Design of advanced post-quantum signature schemes

Corentin Jeudy

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

THÈSE DE DOCTORAT DE

L'UNIVERSITE DE RENNES

ECOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Electronique
Spécialité : *Informatique*

Par
Corentin JEUDY

Design of Advanced Post-Quantum Signature Schemes

Towards efficient privacy

Thèse présentée et soutenue à Rennes, le 18 juin 2024
Unité de recherche : IRISA - UMR 6074

Rapporteurs avant soutenance :

Olivier BLAZY	Professeur, École Polytechnique
Vadim LYUBASHEVSKY	Chercheur, IBM Research Europe Zürich

Composition du Jury :

Président :	Damien VERGNAUD	Professeur, Sorbonne Université, France
Examinatrice :	Alice PELLET--MARY	Chargée de Recherche, Université de Bordeaux, CNRS, Inria, France
Rapporteurs :	Olivier BLAZY	Professeur, École Polytechnique, France
	Vadim LYUBASHEVSKY	Chercheur, IBM Research Europe Zürich, Suisse
Dir. de thèse :	Pierre-Alain FOUQUE	Professeur, Université de Rennes 1, France
Co-dir. de thèse :	Adeline ROUX-LANGLOIS	Directrice de Recherche, Université de Caen, CNRS, GREYC, France
Encadrant :	Olivier SANDERS	Ingénieur de Recherche, Orange Innovation, France

Acknowledgments

My motivation to do research started in the Spring of 2019 with one of my teacher at Carnegie Mellon University noticing a taste for scientific curiosity, creativity and persistence. Since then, I have been fortunate to receive support, guidance, and friendship from a multitude of individuals without whom I would not be where I am today.

I would first like to express my sincere gratitude to Pierre-Alain Fouque, Adeline Roux-Langlois and Olivier Sanders, for their invaluable teaching, counsel and encouragement throughout my doctoral journey. Their expertise and mentorship have been instrumental in shaping my research and professional growth. I particularly thank Adeline for introducing me to this stimulating field of research during my first internship in January 2020. I am immensely grateful for the opportunities she has provided me, this manuscript being the accomplishment of them. She has not only allowed me to expand my academic horizons but has most importantly taught me how to pursue my research with confidence when I did not believe in myself. I also want to show my deepest appreciation to Olivier for his dedication in fostering my work, through riveting discussions and hours spent challenging ideas on the whiteboard, convincing ourselves probabilities are intuitive (until they are not). His help in understanding the industrial research world and in opening doors to pursue my research career at Orange has been priceless. I am truly fortunate to have had all three of them as my advisors.

I am very thankful to Olivier Blazy and Vadim Lyubashevsky for accepting to review this manuscript, even on such short notice, and for their insightful feedback on my work. Their thorough reports have helped me make this manuscript as best as it could be. I also thank Damien Vergnaud and Alice Pellet--Mary for accepting to be part of the jury.

I am happy to have been able to meet so many great people through close collaboration, scientific discussions at conferences, lunches and coffee breaks, or even afterwork drinks (with moderation of course). I would like to particularly thank my first co-authors Katharina Boudgoust and Weiqiang Wen, who I started working with as a research intern in January 2020, and am still glad to exchange with to this day. I also want to thank my other co-authors Sven, Tim and Georg. More broadly, I thank all the members of the EMSEC, SPICY and CAPSULE teams at IRISA with whom I had the pleasure to interact with during my internships and during these last three years of PhD. I am also pleased to have met so many wonderful people at Orange as well, who helped me adjust to this new environment in the nicest of ways. For all the very pleasant lunches, discussions, occasional table tennis tournaments, coffee breaks, climbing sessions, and more, thank you.

I also want to warmly thank all of my friends Maxime, Paul, Élie B, Élie C, Adina, Rémi, André, Anne, Lucas and so many more from Rennes and beyond for helping me adjust to the life in Rennes, for their support, and for just being in my life. I want to thank especially Élie for introducing me to climbing (although my fingers are not so thankful), for having fun and sharing knowledge and visions. Lastly, although there are no words to describe the extent of their unwavering love, encouragement, and understanding, I would like to express my deepest love to my parents and my brother Joran. Their constant comfort and belief in me have been my greatest motivation.

Table of Contents

Résumé en Français

La cryptographie est la discipline visant à concevoir des algorithmes permettant de répondre à certaines exigences de sécurité des systèmes informatiques, comme par exemple l'établissement de communications privées ou l'authentification d'un document. Ces exigences de sécurité se ramènent le plus souvent sont la confidentialité et l'authenticité, bien qu'il en existe beaucoup d'autres en fonction de l'application et du cas d'usage. La confidentialité exprime tout d'abord le fait qu'une information doit rester secrète à tout moment, sauf pour les destinataires prévus. C'est typiquement le cas d'une conversation à distance où l'on souhaite communiquer avec une autre persone sans que quelqu'un d'extérieur puisse écouter. L'authenticité, quant à elle, signifie que les informations présentées sont authentiques, c'est-à-dire qu'elles proviennent effectivement de la bonne personne dans le cas des communications, ou qu'elles sont conformes à ce qui a été déclaré comme étant l'information d'origine. Par nature, elle englobe l'intégrité, qui est généralement définie comme la capacité à garantir que l'information n'a pas subi de modifications (non) intentionnelles par rapport à ce qui est considéré comme authentique.

Afin de répondre à ces besoins de sécurité, on utilise généralement des algorithmes, ou primitives, cryptographiques qui appartiennent à l'une des deux catégories suivantes : la cryptographie *symétrique* et la cryptographie *asymétrique*. Chacune a ses particularités, ses avantages et ses inconvénients, et vise à satisfaire des objectifs de sécurité différents. Historiquement, la cryptographie symétrique, également connue sous le nom de cryptographie à clé secrète, était utilisée pour garantir la confidentialité par le biais d'un processus appelé *chiffrement*. Ce dernier désigne un moyen de brouiller des informations pour les rendre inintelligibles, mais de manière réversible afin qu'elles ne puissent être récupérées que par ses destinataires. Cette capacité repose sur une valeur secrète appelée *clé*, et qui est partagée entre tous les interlocuteurs. Quiconque ne connaissant pas la clé ne peut *déchiffrer* les informations originales une fois qu'elles ont été brouillées.

Au fil du temps, la cryptographie symétrique s'est étendue au-delà du simple processus de chiffrement dans le but de diversifier les garanties de sécurité qu'elle pouvait offrir. Par exemple, les *fonctions de hachage* sont des outils symétriques couramment utilisés pour détecter d'éventuelles modifications par rapport à une valeur certifiée. Elles sont largement utilisées dans les systèmes de gestion des mots de passe. On peut également citer les *codes d'authentification de message* qui ont pour but d'attester l'authenticité d'un message. Les algorithmes symétriques sont bien étudiés et généralement très efficaces. Le problème consiste désormais à trouver des moyens sûrs de transmettre la clé secrète indispensable à ces algorithmes. Par "sûr", nous entendons que des personnes extérieures ne doivent rien apprendre sur cette clé, mais que les utilisateurs légitimes doivent également disposer d'un moyen de s'assurer que la clé qu'ils ont reçue n'a pas été modifiée et qu'elle provient de la bonne source.

Cryptographie à Clé Publique

La cryptographie asymétrique, ou cryptographie à clé publique, permet, entre autres, de résoudre ce problème. Ses fondements remontent aux travaux de Diffie et Hellman en 1976 [DH76]. L'idée sous-jacente est que chaque utilisateur possède désormais une paire de clés, l'une publique et l'autre secrète. Les deux clés sont liées mathématiquement, mais de telle sorte qu'il est supposé impossible de récupérer la clé secrète à partir de la seule clé publique. Pour revenir à l'exemple du chiffrement des communications confidentielles, il est possible de concevoir ce que nous appellons un *chiffrement à clé publique* afin d'échanger des clés symétriques. L'expéditeur utilise en effet la

clé publique du destinataire pour chiffrer la clé symétrique, et le destinataire peut ensuite utiliser sa propre clé secrète pour la déchiffrer. À la fin de cette interaction, les deux parties partagent la connaissance de la clé symétrique. Toutefois, cela ne règle que le problème de la confidentialité. Un attaquant actif pourrait par exemple falsifier la clé chiffrée, même s'il n'a pas connaissance de la clé elle-même (puisque la confidentialité est assurée par le chiffrement sécurisé) ce qui invaliderait son authenticité.

Les signatures numériques constituent la deuxième primitive la plus courante en cryptographie à clé publique. Elles attestent que les données reçues sont effectivement celles transmises et validées par l'expéditeur légitime. Le principe est qu'un signataire certifie un message en produisant une signature à l'aide de sa clé secrète, et que tout le monde puisse ensuite vérifier cette signature à l'aide de la clé publique correspondante. Désormais, deux partis peuvent échanger des clés symétriques de manière sécurisée sans partager de données secrètes au préalable. Bien qu'elles aient été introduites pour certifier les échanges de clés symétriques, les signatures numériques sont devenues intéressantes et même nécessaires dans une pléthore d'autres cas d'usages. De la certification des paiements par carte de crédit aux passeports électroniques ou aux identités numériques en général, les signatures sont devenues de plus en plus importantes avec l'évolution des technologies numériques.

La Menace de l'Ordinateur Quantique

Par définition, la cryptographie asymétrique divulgue des informations supplémentaires (la clé publique), y compris aux attaquants, qui pourraient être utilisées pour retrouver des informations sensibles. On peut naturellement penser à la clé secrète elle-même, ce qui mènerait à une attaque dite par *récupération de clé*. Ces informations privées peuvent néanmoins être d'une autre nature, ce qui nous oblige à envisager d'autres attaques qui sont plus spécifiques à la primitive et aux exigences de sécurité. Par exemple, ce que nous attendons d'un chiffrement est qu'une personne ne disposant pas de la clé de déchiffrement ne puisse rien apprendre sur les données chiffrées. L'une des méthodes d'attaque consiste alors à récupérer la clé et à déchiffrer, mais nous pouvons également imaginer d'autres moyens de récupérer, par exemple, un bit d'information sur les données, ce qui serait considéré comme une attaque réussie. Nous devons donc montrer que même ces attaques sont irréalisables.

Pour cela, nous effectuons une preuve de sécurité ou une réduction de sécurité qui montre que si un adversaire est capable de mener à bien l'attaque, alors il sera capable de résoudre un problème mathématique difficile. Par contraposition, si le problème mathématique est effectivement difficile à résoudre, alors l'attaque du système est infaisable. Cela nous permet de faire reposer la sécurité de systèmes très complexes sur un petit nombre d'hypothèses qui sont beaucoup plus faciles à formuler et à étudier. Par exemple, le lien entre la clé publique et la clé secrète est généralement matérialisé par un problème mathématique que l'on suppose ou que l'on prouve difficile à résoudre. Cela signifie que la récupération de la clé secrète à partir de la clé publique est mathématiquement irréalisable. L'étape suivante consiste donc à trouver des familles de tels problèmes difficiles sur lesquelles construire des primitives cryptographiques. La cryptographie à clé publique déployée aujourd'hui repose en grande partie sur deux familles de problèmes : la *factorisation* (étant donné $N = pq$ pour p, q deux grands nombres premiers, trouver p et q) et le *logarithme discret* (étant donné g et h dans un groupe cyclique, trouver x tel que $h = g^x$). La difficulté de ces problèmes semble bien établie en ce qui concerne les algorithmes classiques, puisque les records actuels [BGG+22] n'attaquent que des paramètres qui sont bien inférieurs à ceux utilisés en pratique. Ce n'est plus le cas pour les algorithmes quantiques.

L'informatique quantique est un domaine de recherche populaire dont l'objectif est la construction d'ordinateurs quantiques de grande échelle. Ces ordinateurs s'appuient sur les principes de la physique quantique pour offrir une nouvelle vision de l'informatique et débloquer de nouvelles méthodes et de nouveaux algorithmes qui peuvent être beaucoup plus efficaces que les méthodes actuelles pour des tâches spécifiques. Tout comme l'unité de mesure d'un ordinateur classique est le bit, la principale unité de mesure des ordinateurs quantiques est appelée qbit ou *bit quantique*. En utilisant les principes de la mécanique quantique tels que la superposition, l'intrication, la quantification, les mesures quantiques, la réduction du paquet d'onde, la dualité, etc, il est possible d'exploiter les propriétés quantiques des particules pour effectuer des calculs. Une caractéristique intéressante est qu'un système de N qbits dans une superposition d'états quantiques a théoriquement la puissance de 2^N bits classiques. On pourrait donc penser qu'un ordinateur quantique avec seulement 50 qbits serait théoriquement plus performant que les meilleurs supercalculateurs actuels. Ce seuil de 50 qbits a souvent été appelé suprématie quantique. Mais au-delà des spécificités des algorithmes quantiques, qui sont très différents des algorithmes classiques, en

pratique, les problèmes de stabilité et le phénomène de décohérence quantique rendent les choses plus compliquées qu'il n'y paraît. En effet, cette décohérence déstabilise le système lors du calcul conduisant alors à son interruption. Actuellement, les industries construisent des ordinateurs quantiques de plus en plus puissants pour atteindre cette suprématie quantique même en présence de décohérence quantique. Par exemple, IBM a construit un ordinateur quantique de 27 qbits en 2019, et l'a amélioré à 65 qbits en 2020, 127 en 2021, 433 en 2022, 1121 en 2023, et vise 1386 d'ici la fin de l'année 2024. Plus généralement, l'informatique quantique reste un domaine de recherche très actif en raison de ses nombreux champs d'application tels que la médecine, la chimie, l'intelligence artificielle ou encore la cryptographie.

En effet, en 1994, Peter Shor [Sho94] proposa un algorithme quantique pour résoudre efficacement les problèmes de factorisation et de logarithme discret que les ordinateurs classiques ne peuvent résoudre qu'en un temps exponentiel (ce qui signifie qu'il faudrait plusieurs (millions de) fois l'âge de l'univers pour les résoudre). L'algorithme de Shor, quant à lui, pourrait les résoudre en un temps polynomial (ce qui signifie qu'il ne prendrait que quelques heures ou quelques jours). Il invaliderait donc la sécurité de tous les algorithmes cryptographiques à clé publique déployés aujourd'hui, mettant ainsi en péril les systèmes de sécurité dans le monde entier. Heureusement, l'algorithme de Shor nécessite une certaine quantité de qbits pour fonctionner avec une telle performance. À titre de référence, il lui faudrait environ un million de qbits pour attaquer un système dont la sécurité repose sur la factorisation d'entiers de 2048 bits. Même avec les récentes avancées dans la construction d'ordinateurs quantiques, ceux-ci sont loin d'être assez puissants pour attaquer la cryptographie. Néanmoins, il est important d'anticiper ces menaces et de se préparer à l'arrivée de ces ordinateurs quantiques. Par exemple, un adversaire pourrait dès aujourd'hui stocker des données chiffrées de manière classique et les déchiffrer dès lors qu'un ordinateur quantique suffisamment puissant sera disponible. Si les données sont encore sensibles à ce moment-là, cela pose un problème de sécurité important. Par ailleurs, il faut tenir compte de l'inertie habituelle des déploiements industriels. Plusieurs années s'écoulent en effet entre la conception d'un système cryptographique, sa standardisation, son déploiement et son utilisation active. C'est pourquoi nous devons commencer dès à présent à trouver des mécanismes cryptographiques alternatifs qui ne seraient pas vulnérables à l'informatique quantique.

Cryptographie Post-Quantique

La *cryptographie post-quantique* vise précisément à répondre à ces préoccupations et correspond à l'ensemble des constructions cryptographiques qui sont résistantes aux algorithmes quantiques. En particulier, elles reposent sur de nouvelles familles d'hypothèses qui ne sont pas seulement immunisées contre l'algorithme de Shor, mais aussi contre tout autre algorithme quantique connu à ce jour. Pour commencer à préparer l'avenir de la cryptographie et remplacer les standards actuels reposant sur la factorisation et le logarithme discret, l'organisme américain NIST (*National Institute of Standards and Technologies*) a lancé en 2016 un processus de standardisation de la cryptographie post-quantique [NISa]. L'objectif était de sélectionner des constructions qui seraient efficaces tout en résistant aux attaques quantiques, c'est-à-dire aux attaques menées par un adversaire ayant accès à un ordinateur quantique. Après avoir reçu 69 algorithmes basés sur les *réseaux euclidiens*, les *codes correcteurs*, les *isogénies*, les *systèmes multivariés* et les *fonctions de hachage*, le NIST a sélectionné 4 algorithmes (1 chiffrement et 3 signatures), dont 3 sont basés sur des réseaux euclidiens : le chiffrement Kyber [BDK+18], et les signatures Falcon [PFH+20] et Dilithium [DKL+18]. Ce sont ces algorithmes qui remplaceront très probablement les standards actuels de cryptographie à clé publique. Mais qu'est-ce que la cryptographie basée sur les réseaux euclidiens ?

Les réseaux euclidiens sont des objets mathématiques étudiés depuis des siècles. La première spécification formelle des réseaux euclidiens et de leurs propriétés remonte à Lagrange (1736 - 1813). Depuis, Gauss (1777 - 1855) a étudié leur utilisation en théorie des nombres, suivi par Minkowski (1864 - 1909) qui en a étudié la géométrie. On peut les définir de manière informelle comme des grilles périodiques de points en d dimensions, c'est-à-dire dans $\mathbb{R}^d$, comme par exemple $\frac{3}{2}\mathbb{Z}^d$. L'une des caractéristiques intéressantes des réseaux euclidiens est qu'ils permettent de formuler plusieurs problèmes mathématiques qui sont à ce jour difficiles à résoudre, même sur le plan quantique. L'exemple le plus courant est le problème du plus court vecteur SVP_γ (*Shortest Vector Problem*), qui consiste à trouver un point du réseau $\mathcal{L}$ dans une boule centrée autour de $\mathbf{0}$ de rayon $\gamma\lambda_1(\mathcal{L})$, où $\gamma \geq 1$ et $\lambda_1(\mathcal{L})$ est la longueur du plus court vecteur non nul de $\mathcal{L}$. Afin d'étudier ce problème de manière algorithmique, nous avons besoin d'une représentation plus compacte d'un réseau euclidien. Il s'avère que tout réseau $\mathcal{L}$ peut être exprimé comme $\mathcal{L} = \mathbf{B}\mathbb{Z}^k$ où $\mathbf{B} \in \mathbb{R}^{d\times k}$ est appelé une base du réseau. Les bases ne sont pas uniques, et la résolution du problème ci-dessus

consiste essentiellement à trouver une base $\mathbf{B}^\star$ qui soit suffisamment plus courte que la base $\mathbf{B}$ donnée en entrée. En 1982, Lenstra, Lenstra et Lovàsz [LLL82] ont introduit l'un des premiers algorithmes de réduction pour les réseaux euclidiens : le célèbre algorithme LLL, qui vise à réduire la base d'un réseau tout en ayant des vecteurs les plus orthogonaux possibles. Il peut ensuite être utilisé pour résoudre SVP_γ, mais ne fonctionne en un temps raisonnable que lorsque γ est exponentiel en d. Le problème est en effet supposé difficile pour γ polynomial en d, même en ayant accès à des ressources quantiques. Malgré cette supposée résistance quantique, SVP_γ n'est pas très adapté à la conception d'algorithmes cryptographiques car il s'agit d'un problème dit *pire-cas*, c'est-à-dire qu'il est facile pour de nombreux réseaux mais difficile pour les pires d'entre eux.

En 1996, Ajtai [Ajt96] publia un article fondamental sur l'utilisation des réseaux euclidiens en cryptographie. Dans cet article, Ajtai introduisit un nouveau problème de réseau appelé *Short Integer Solution* (SIS) et donna la première réduction pire-cas moyen-cas d'une variante de SVP_γ à SIS. Cela a d'énormes conséquences en cryptographie car les constructions basées sur ce problème moyen-cas reposent désormais sur la difficulté des pires instances du problème de réseau sous-jacent, et non sur les instances moyennes. Cela signifie que si la construction est cassée, le problème intermédiaire SIS peut être facilement résolu en moyenne et donc que toutes les instances du problème de réseau sous-jacent peuvent également être résolues facilement, y compris les plus difficiles. En 2005, Regev [Reg05] présenta alors un autre problème intermédiaire qui bénéficie également des hypothèses de difficulté dans le pire des cas. Il décrivit le problème *Learning With Errors* (LWE) et donna une réduction (quantique) des problèmes difficiles sur les réseaux à LWE. Il présenta également un système de chiffrement à clé publique dont la sécurité repose sur la difficulté de LWE.

Depuis, de nombreuses constructions basées sur ces problèmes ont vu le jour, ainsi que de meilleures réductions. Certaines questions ouvertes ont également été résolues grâce aux progrès de la cryptographie sur les réseaux euclidiens, comme le *chiffrement complètement homomorphe* (FHE pour *Fully Homomorphic Encryption*) [BGV12, BV14, DM15], qui a longtemps été considéré comme impossible. En 2009, Gentry [Gen09] présenta en effet le premier schéma de FHE basé sur des réseaux euclidiens, en tant que preuve de concept. Bien qu'attrayants en raison de la base théorique qu'ils fournissent, les problèmes originaux SIS et LWE ont depuis été modifiés sous de nombreux aspects afin d'offrir une meilleure efficacité, par exemple à travers des variantes algébriques [LPR10, LS15, PP19]. Les efforts déployés pour accroître la confiance en ces variantes modifiées, soit par des preuves théoriques, soit par des évaluations cryptanalytiques, ont considérablement contribué au développement de la cryptographie sur les réseaux euclidiens et sont toujours en cours.

Tous ces arguments historiques constituent quelques-unes des raisons pour lesquelles les réseaux euclidiens sont utilisés en cryptographie : ce sont des objets simples, qui s'avèrent particulièrement efficaces si la structure et les paramètres sont bien choisis; ils offrent une sécurité prouvable pour les constructions grâce aux problèmes de réseaux sous-jacents; ils offrent la possibilité de concevoir une grande variété de mécanismes cryptographiques; et enfin, les problèmes de réseaux sont conjecturés comme étant résistants aux attaques quantiques.

Cryptographie pour la Vie Privée

Indépendamment de la résistance aux attaques quantiques, les primitives de base telles que le chiffrement et les signatures ne couvrent malheureusement pas tous les besoins de sécurité que nous attendons dans de nombreuses situations.

Prenons tout d'abord le cas d'une institution devant déléguer un traitement intense calculatoirement sur des données privées à un prestataire de services, par exemple le traitement de données médicales. Si le prestataire de services est parfaitement fiable, il suffit de s'assurer que les données soient protégées pendant les communications, ce qui peut être fait en établissant un canal sécurisé à l'aide de notre boîte à outils cryptographique actuelle. Les calculs peuvent alors être effectués sur les données déchiffrées. Cependant, dans de nombreuses situations, on ne peut pas suffisamment faire confiance au fournisseur de services pour lui confier des informations sensibles. Dans ce cas, il faudrait calculer directement sur les données chiffrées, ce qui n'est pas possible avec les mécanismes de chiffrement habituels. L'utilisation exclusive de chiffrement et de signatures basiques restreint donc la capacité à externaliser les tâches, ce qui est pourtant plus important que jamais avec la montée en puissance de l'informatique dématérialisée (Cloud).

Nous pouvons également considérer le cas d'usage de l'identité numérique. Supposons qu'un client possède un certificat numérique (intégré dans un document d'identification) authentifiant ses informations personnelles (nom, date de naissance, adresse, numéro de sécurité sociale, etc.)

Pour s'identifier à l'aide d'une signature numérique habituelle, ce client n'a pas d'autre choix que de fournir l'ensemble des attributs au contrôleur ou au caissier qui doit exécuter l'algorithme de vérification. Cependant, dans de nombreux cas, on ne s'attendrait pas à devoir donner tous ses attributs personnels pour attester de la validité d'un seul d'entre eux. Par exemple, dans le cas usuel du contrôle d'âge, le client veut seulement révéler qu'il est adulte, mais pas sa date de naissance exacte. Les signatures basiques posent donc de graves problèmes en matière de respect de la vie privée. On pourrait arguer que la situation est similaire dans le monde réel : il est en effet assez courant de présenter une pièce d'identité contenant de nombreuses informations personnelles à un caissier qui a besoin de contrôler son âge. Dans ce cas cependant, il est naturel de supposer que le caissier ne mémorisera pas toutes les informations contenues dans le document en vue d'une exploitation commerciale ou d'une usurpation d'identité. Ce n'est pas le cas dans le monde numérique où les utilisateurs perdent définitivement le contrôle de leurs données dès qu'ils les révèlent. Il est par exemple très probable que le même client sera beaucoup plus réticent à fournir les mêmes informations à un site web qui a besoin de vérifier qu'il est bien un adulte.

Ces exemples, choisis parmi beaucoup d'autres, illustrent certaines des propriétés de sécurité qui ne sont pas nativement garanties par les chiffrements et signatures basiques. Une préoccupation particulière est notamment de pouvoir cacher autant d'informations que possible lorsqu'il n'est pas strictement nécessaire de les divulguer, une propriété généralement désignée sous le nom de *protection de la vie privée* (*privacy* en anglais), que nous désignerons parfois par *anonymat* par abus de langage. Cette propriété recouvre l'ensemble des bonnes pratiques pour collecter, traiter ou communiquer des données privées[1]. Grâce à la prise en compte par les autorités de ces préoccupations en matière de données personnelles, la protection de la vie privée a bénéficié d'une publicité importante et est désormais considérée comme un facteur positif de différenciation. Ces questions se sont en effet étendues à l'échelle mondiale et ne se limitent plus seulement à certaines communautés d'experts, et sont principalement dues à la numérisation des communications, des services et de l'information elle-même.

Comme nous l'avons vu, les mécanismes cryptographiques de base ne permettent pas toujours de résoudre les problèmes de vie privée. Pour combiner les exigences de sécurité habituelles tout en limitant la divulgation d'informations privées, il faut alors concevoir des mécanismes cryptographiques avancés[2] offrant une plus grande souplesse en matière de contrôle des données. La combinaison de la confidentialité et de la protection de la vie privée est apparue principalement dans le cadre du traitement des données sensibles mentionné précédemment. L'une des solutions les plus connues est le FHE. C'est une primitive cryptographique très polyvalente permettant d'effectuer des calculs assez généraux directement sur les données chiffrées, sans avoir à les déchiffrer à aucun moment. Le FHE permettrait donc de répondre au premier cas d'usage présenté ci-dessus. Il existe d'autres types de mécanismes cryptographiques améliorant la confidentialité, par exemple le calcul multipartite sécurisé (*Secure Multiparty Computation*, SMPC). Cette branche de la cryptographie pour la protection de la vie privée a fait l'objet de beaucoup d'attention et de contributions, et nombre d'entre elles, comme le FHE, combinent ces caractéristiques avec une sécurité post-quantique. La situation est très différente pour l'authentification anonyme post-quantique. Seuls des cas d'usages très spécifiques sont abordés, par exemple avec les signatures de groupe [dPLS18, LNPS21] ou les signatures aveugles [dPK22, BLNS23a], et les constructions sont plutôt rares. Cela contraste fortement avec la cryptographie classique basée sur la factorisation ou le logarithme discret, qui a produit d'innombrables mécanismes efficaces pour l'authentification anonyme [Cha82, Bra00, CL01, CL02, CL04, BCC04, BSZ05, ASM06, BB08, CDHK15, PS16, FHS19, San20, BEK+21, San21, BFGP22, CLP22, ST23].

Cet écart entre les constructions classiques et post-quantiques, en termes de diversité des constructions mais aussi d'efficacité, peut s'expliquer en partie par les outils plutôt complexes dont ces constructions ont besoin. Dans le cas du contrôle d'âge présenté ci-dessus, il faut essentiellement prouver l'authenticité des informations divulguées tout en cachant tout le reste. En particulier, pour éviter d'être tracé au fil des authentifications, nous devrions cacher le certificat lui-même. Les éléments clés de ces systèmes sont alors principalement des systèmes de signature polyvalents, qui fourniraient la certification souhaitée tout en permettant la divulgation sélective susmentionnée, et des preuves à divulgation nulle de connaissance, afin de cacher

[1] Dans cette section, nous faisons la distinction entre le terme *secret*, qui désigne les informations dont la divulgation compromettrait la sécurité, et le terme *privé*, qui désigne les informations qui peuvent être divulguées, mais de manière limitée.

[2] Dans cette thèse, nous appelons *avancés* tous les systèmes cryptographiques qui vont au-delà des chiffrement et des signatures numériques basiques. Parmi les exemples, citons le chiffrement complètement homomorphe [Gen09], les méthodes d'authentification pour la vie privée telles que les signatures de groupe [CvH91], les signatures aveugles [Cha82], les accréditations anonymes [Cha85], etc.

les informations sensibles (y compris la signature) au cours de l'authentification. Ces outils ont été étudiés pendant des décennies dans le cadre classique et ont été instanciés de manière très efficace. Dans le cas post-quantique, les avancées sur ces sujets sont plus récentes. Des progrès impressionnants ont été réalisés au cours des dernières années dans la construction de preuves pratiques à divulgation nulle de connaissance à partir de réseaux euclidiens, par exemple [dPLS18, LS18, BBC+18, BLS19, YAZ+19, ALS20, LNS20, LNS21, LNP22], ce qui a permis de développer des mécanismes anonymes plus efficaces. En ce qui concerne les schémas de signature, la plupart des algorithmes efficaces sur les réseaux ne répondent pas aux exigences de construction de ces primitives avancées. Plus précisément, comme nous voulons cacher certains attributs tout en prouvant efficacement qu'ils ont été correctement signés, cela écarte les systèmes de signature qui utilisent des fonctions de hachage. Nous devrions en effet prouver que le hachage des attributs cachés a été correctement évalué, ce qui donnerait un système peu efficace.

Contributions

Les recherches menées dans le cadre de cette thèse de doctorat sont motivées par la poursuite du développement de la cryptographie post-quantique pratique pour la protection de la vie privée, en mettant l'accent sur la branche des mécanismes d'authentification. Nous adoptons une approche globale et étudions la conception générale de ces mécanismes en recherchant des optimisations au niveau des protocoles eux-mêmes, des outils principaux auxquels ils ont recours, ainsi que des hypothèses de sécurité sur lesquels ils reposent.

Les mécanismes classiques ont en effet été grandement améliorés en élargissant le paysage des hypothèses utilisées. Comme la cryptographie basée sur les réseaux euclidiens devient plus mature et approche les limites de ce qui peut être fait avec des hypothèses courantes, une direction prometteuse est d'imiter la démarche fructueuse de la cryptographie classique en relâchant les hypothèses post-quantiques ou en en proposant de nouvelles. Il est alors nécessaire d'évaluer leur difficulté pour éviter de compromettre sur la sécurité. En parallèle, l'un des outils clés des systèmes d'authentification avancés sont les fonctions à trappes, des fonctions calculables publiquement qui ne peuvent être inversées qu'à l'aide d'une information secrète appelée trappe. Plus précisément, les fonctions à trappes introduites par Micciancio et Peikert [MP12] procurent la polyvalence que nous attendons pour concevoir de tels schémas avancés. Pour preuve, la plupart des constructions actuelles pour la protection de la vie privée basées sur les réseaux euclidiens utilisent leur mécanisme. Dans l'espoir d'améliorer l'efficacité de ces constructions, ou d'en concevoir de nouvelles, il est important de réévaluer les outils largement adoptés afin d'identifier les marges d'amélioration possibles. Enfin, la recherche de synergies entre ces différents éléments, ainsi que les optimisations que nous pouvons apporter, permettent de concevoir des primitives plus efficaces pour la protection de la vie privée tout en répondant à certaines des exigences manquantes mises en évidence par les cas d'usages ci-dessus.

Fondations

La formulation de nouvelles hypothèses en vue d'obtenir des mécanismes d'authentification anonymes efficaces est un axe de recherche prometteur, par exemple [AKSY22, BLNS23b]. Néanmoins, ces dernières ne sont pas toujours nécessaires à l'obtention de mécanismes plus efficaces. Les hypothèses fondamentales sont parfois suffisantes, bien qu'elles nécessitent elles aussi une analyse détaillée dans les régimes de paramètres efficaces. Nous nous concentrons donc sur ces hypothèses fondamentales tout en les poussant jusqu'à leurs limites. Parmi celles-ci, le problème de l'apprentissage avec erreurs (LWE) introduit par Regev [Reg05] a été utilisé comme fondement de la sécurité et de l'anonymat dans d'innombrables constructions, y compris les preuves à divulgation nulle de connaissance. Le problème peut être formulé comme suit. Étant donné une matrice de coefficients $\mathbf{A}$ et un vecteur $\mathbf{b}$ correspondant au second membre d'un système d'équations modulaires bruitées en $\mathbf{A}$, c'est-à-dire $\mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e} \bmod q\mathbb{Z}$ où $\mathbf{s}$ est le vecteur inconnu et $\mathbf{e}$ un vecteur de bruit, l'hypothèse LWE stipule qu'il est difficile de trouver la solution $\mathbf{s}$, ou de décider s'il existe même un tel $\mathbf{s}$ ou si $\mathbf{b}$ est purement aléatoire. Le problème est alors paramétré par les distributions choisies pour le *secret* $\mathbf{s}$ et l'*erreur* $\mathbf{e}$. Choisir des distributions produisant des vecteurs $\mathbf{s}$ et $\mathbf{e}$ courts (dans une norme spécifiée) mène généralement à une meilleure efficacité des schémas basés sur LWE. Cela rend néanmoins le problème légèrement plus facile puisque nous réduisons l'espace des solutions possibles. Nous avons donc besoin d'étudier la difficulté de ce problème lorsque l'on réduit les paramètres pour avoir des vecteurs de plus en plus courts. Plusieurs travaux [GKPV10, BLP+13, MP13, Mic18, BD20] ont abordé la question en montrant que LWE peut être raisonnablement sûr même pour des distribu-

tions courtes, par exemple des vecteurs $\mathbf{s}$ ou $\mathbf{e}$ uniformes binaires.

Malheureusement, les schémas résultants restent relativement inefficaces. C'est pourquoi une ligne de recherche parallèle visait à ajouter une structure algébrique supplémentaire au problème afin de permettre des calculs plus rapides et un stockage plus efficace. Parmi ces variantes structurées [SSTX09, LPR10, LS15, PP19], l'hypothèse *Module*-LWE (M-LWE) offre une polyvalence intéressante permettant de faire un compromis entre l'efficacité et la sécurité et vice-versa. Au lieu d'utiliser des coefficients entiers, elle considère des *entiers algébriques* qui peuvent être vus comme des polynômes à coefficients entiers. Les opérations sur les polynômes peuvent être traitées plus efficacement, pour le même volume de données, et cela permet également de compacter la dimension et le stockage des matrices. Bien que Langlois et Stehlé [LS15] aient étudié en profondeur sa difficulté dans le cas général, aucun résultat n'était connu pour les régimes relâchés par exemple celui où $\mathbf{s}$ ou $\mathbf{e}$ sont composés de polynômes à coefficients binaires.

Dans cette thèse, nous obtenons des conclusions similaires à celles de [GKPV10, BLP+13, MP13, Mic18, BD20] mais pour l'hypothèse M-LWE. Dans quatre articles [BJRW20, BJRW21, BJRW22, BJRW23], nous montrons que le problème M-LWE reste aussi difficile que sa définition originale (avec $\mathbf{s}$ uniforme modulo q, et $\mathbf{e}$ gaussien) lorsque l'on modifie les distributions du secret et de l'erreur. Bien qu'avec des paramètres légèrement différents, nous utilisons plus tard ces variantes comme fondements de sécurité de nos constructions pour la vie privée. Nous notons que nos contributions ont une portée plus large car ces hypothèses sont utilisées dans la plupart des conceptions cryptographiques efficaces basées sur les réseaux euclidiens. Notre étude contribue donc à renforcer notre confiance dans la sécurité de ces schémas.

Ces contributions sont abordées dans la partie I. Le chapitre 2 étudie tout d'abord la difficulté de M-LWE où $\mathbf{s}$ est uniforme avec de petits coefficients et $\mathbf{e}$ reste Gaussien. Ensuite, dans le chapitre 3, nous étudions M-LWE où $\mathbf{e}$ est uniforme avec de petits coefficients, d'abord avec $\mathbf{s}$ uniforme modulo q puis avec $\mathbf{s}$ distribué de la même façon que $\mathbf{e}$.

Échantillonneurs et Signatures

Les mécanismes d'authentification avancés reposent principalement sur les trappes polyvalentes de Micciancio et Peikert [MP12], ainsi que sur l'échantillonneur d'antécédents qui leur est associé. Ils définissent des fonctions à trappes, sous la forme de matrices $\mathbf{A}$, où l'image d'un vecteur d'entrée peut être calculée publiquement et efficacement, tandis que la recherche d'un antécédent court $\mathbf{x}$ telle que $\mathbf{Ax} = \mathbf{u} \bmod q$ est difficile sans connaître la trappe. En particulier, la connaissance de cette dernière permet de "randomiser" le processus de recherche d'antécédent et d'obtenir ainsi un *échantillonneur d'antécédent*. À partir de ces fonctions à trappes, on peut directement imaginer un système de signature dans lequel $\mathbf{A}$ serait la clé publique, la trappe serait la clé secrète et les antécédents $\mathbf{x}$ seraient les signatures. Pour garantir la sécurité d'un tel échantillonneur, c'est-à-dire pour s'assurer que les antécédents $\mathbf{x}$ ne divulguent pas d'informations sur la trappe, les auteurs proposent des contre-mesures qui imposent des restrictions sur les paramètres, et donc sur la compacité des schémas résultants. La plupart des utilisations des outils développés dans [MP12] reposent sur l'échantillonnage d'antécédents $\mathbf{x}$ suivants une distribution gaussienne *sphérique* dont la qualité, c'est-à-dire la variance, est dictée par la taille de la trappe secrète.

La deuxième étape de cette thèse consiste alors à améliorer la qualité des échantillonneurs qui affecte directement celle des mécanismes les utilisant, y compris les signatures pour respect de la vie privée. Ce faisant, nous montrons que nous pouvons atteindre une meilleure efficacité tout en conservant les caractéristiques intéressantes de ces trappes et échantillonneurs. Au-delà de l'application des signatures anonymes, nous concevons une famille de schémas de signature appelée Phoenix, qui est basée sur une version optimisée de l'échantillonneur proposé par Lyubashevsky et Wichs [LW15]. Notre système est compétitif par rapport à certaines signatures à base de réseaux sélectionnées pour la standardisation, telles que Dilithium [DKL+18], tout en bénéficiant de certaines des caractéristiques intéressantes des conceptions basées sur les trappes, telles qu'une meilleure sécurité, une évaluation plus simple des paramètres, etc.

Ces contributions sont abordées dans la partie II. Nous proposons tout d'abord plusieurs optimisations pour l'échantillonneur Micciancio-Peikert [MP12, LW15] dans le chapitre 4, lesquelles sont utilisées par la suite dans nos constructions. En particulier, le chapitre 5 présente une nouvelle famille de signatures appelée Phoenix reposant sur notre échantillonneur approché par rejet, qui combine les idées de [LW15] et [CGM19].

Signatures Avancées et Vie Privée

Nous concluons cette thèse en fournissant des schémas de signature polyvalents appelés *signatures avec protocoles efficaces* (SEP), tels qu'introduits par Camenisch et Lysyanskaya [CL02], qui représentent un outil très utile pour une grande variété d'applications de protection de la vie privée. Les SEP ont conduit à certains des mécanismes d'authentification avancés les plus efficaces dans le cadre classique [CL04, BB08, ASM06, PS16]. Outre la preuve de concept de [LLM+16] qui a abouti à des tailles totalement irréalisables en pratique, il n'y avait pas d'équivalent post-quantique avant notre travail. Nous proposons deux constructions sur les réseaux euclidiens, représentant deux vagues d'améliorations par rapport à [LLM+16]. En particulier, la seconde embarque les contributions discutées précédemment pour produire un schéma beaucoup plus compact sans sacrifier la sécurité.

Afin de démontrer l'utilité de notre construction, nous utilisons notre SEP pour concevoir un système d'accréditations anonymes. À haut-niveau, les accréditations anonymes impliquent un utilisateur possédant certains attributs, un signataire chargé de produire un certificat sur les attributs (éventuellement cachés) de l'utilisateur, et un vérificateur contrôlant la validité de ce certificat sur les attributs de manière anonyme (pour l'utilisateur). Ces systèmes répondent aux contraintes de nombreux cas d'usages, tels que la certification d'attributs nombreux et éventuellement secrets, l'authentification anonyme tout en permettant la divulgation sélective de certains attributs, etc. Par exemple, il répond parfaitement au problème du contrôle d'âge à l'aide d'un passeport électronique évoqué plus haut. Le signataire, incarné par une autorité nationale, produirait un certificat (le passeport) sur les attributs d'une personne. Cette personne (l'utilisateur) pourrait alors montrer à un contrôleur (le vérificateur) qu'elle possède un passeport légitimement délivré certifiant ses attributs personnels cachés, y compris son âge prouvant qu'elle est adulte. Notre système est directement basé sur notre SEP et sur le système de preuves à divulgation nulle de connaissance de Lyubashevsky et al. [LNP22], et repose ainsi sur les hypothèses habituelles, bien qu'avec des paramètres relâchés comme indiqué ci-dessus. Néanmoins, il est compétitif (et même plus performant) que la plupart des constructions existantes de mécanismes d'accréditations anonymes sur les réseaux euclidiens [BLNS23b, LLLW23, BCR+23]. Nos travaux, initialement publiés dans [JRS23] puis améliorés dans [AGJ+24], ont fourni les premiers systèmes d'accréditations anonymes post-quantiques. La version améliorée est la première à réaliser des preuves de certification de moins de 100 Ko tout en s'appuyant sur des hypothèses courantes en réseaux euclidiens.

Au-delà du seul critère de taille de la preuve de certification, nous démontrons l'aspect pratique de notre système en l'implémentant en C. Notre implémentation est plus performante que celle de [BCR+23] qui était jusqu'à présent la seule implémentation existante d'accréditations anonymes post-quantiques. En particulier, le protocole complet pour l'émission (aveugle) d'un certificat prend 400 millisecondes en moyenne, tandis que la présentation de ces certificats pour permettre l'authentification anonyme prend 500 millisecondes en moyenne. Ces délais devraient être imperceptibles pour l'utilisateur dans la plupart des cas d'usages.

Ces contributions sont abordées dans la partie III. Nous proposons deux constructions de SEP à base de réseaux euclidiens dans le chapitre 6, que nous utilisons ensuite pour obtenir un système d'accréditations anonymes dans le chapitre 7. Enfin, le chapitre 8 est dédié à l'implémentation de ce dernier.

Publications

We list here the different research and dissemination papers that have been published in international venues during this thesis. Research papers have gone through a peer-review process whether it is for a journal or conference publication.

- ∘ [BJRW20] **Towards Classical Hardness of Module-LWE: The Linear Rank Case**. Co-authored with Katharina Boudgoust, Adeline Roux-Langlois and Weiqiang Wen. Published in the proceedings of *Asiacrypt 2020*.

- ∘ [BJRW21] **On the Hardness of Module-LWE with Binary Secret**. Co-authored with Katharina Boudgoust, Adeline Roux-Langlois and Weiqiang Wen. Published in the proceedings of *CT-RSA 2021*.

  [BJRW22] **Entropic Hardness of Module-LWE from Module-NTRU**. Co-authored with Katharina Boudgoust, Adeline Roux-Langlois and Weiqiang Wen. Published in the proceedings of *Indocrypt 2022*.

- ⋆ [BJRW23] **On the Hardness of Module Learning With Errors with Short Distributions**. Co-authored with Katharina Boudgoust, Adeline Roux-Langlois and Weiqiang Wen. Published at the *IACR Journal of Cryptology (2023, vol. 36)*.

  [JR23] **Cryptographie Reposant sur les Réseaux Euclidiens**. Co-authored with Adeline Roux-Langlois. Dissemination paper published in the French journal *Techniques de l'Ingénieur*.

- ⋆ [JRS23] **Lattice Signature with Efficient Protocols, Application to Anonymous Credentials**. Co-authored with Adeline Roux-Langlois and Olivier Sanders. Published in the proceedings of *Crypto 2023*.

- ⋆ [JRS24] **Phoenix: Hash-and-Sign with Aborts from Lattice Gadgets**. Co-authored with Adeline Roux-Langlois and Olivier Sanders. Published in the proceedings of *PQCrypto 2024*.

- ⋆ [AGJ+24] **Practical Post-Quantum Signatures for Privacy**. Co-authored with Sven Argo, Tim Güneysu, Georg Land, Adeline Roux-Langlois and Olivier Sanders. Published in the proceedings of *ACM CCS 2024*.

Only the papers marked with ⋆ are covered in this thesis. The paper [BJRW23] represents a long version of the papers marked with ∘. As such some aspects from the papers marked with ∘ appear in Chapter 2.

Introduction

RYPTOGRAPHY is the discipline of designing algorithms to satisfy certain security requirements we expect from a real-life system, such as establishing private communications, or authenticating a document. The main security objectives are confidentiality and authenticity, although there are many more depending on the application and use case. Confidentiality captures the fact that a piece of information should remain secret at all times unless to the intended recipients. In communications, it is usual for a person to want to interact with another without having anyone else listening in on their conversation. Authenticity on the other hand argues that the presented information is authentic, meaning it indeed originates from the intended party for communications, or is consistent with what has been declared as being the proper information. By nature, it encompasses integrity, which is generally defined as being able to ensure that information has not suffered from (un)intentional modifications compared to what is considered authentic.

Achieving these security goals is generally done with cryptographic algorithms, or primitives, which fall in one of two categories: *symmetric* and *asymmetric* cryptography. Each has its peculiarities, advantages and drawbacks, and aims at different security objectives. Historically, symmetric cryptography, also known as secret key cryptography, was used to ensure confidentiality through a process called *encryption*. The latter designates a way to scramble information to make it unintelligible, but in a reversible way so that it can be recovered by the intended party only. It is done with the help of a secret value known as *key* which allows one to reversibly scramble and unscramble data, and that is shared among all the parties that must have this ability. Anyone oblivious to the key would not be able to *decrypt* the original information once scrambled.

Over the years, symmetric cryptography extended further than just encryption with the goal of diversifying the security guarantees it could provide. For example, *hash functions* are a common symmetric tool to detect modifications compared to a certified ground truth. They are widely used in password management systems. We can also mention *message authentication codes* which serve the purpose of attesting the authenticity of a message. Symmetric algorithms are well studied and usually very efficient. The problem now lies in findind secure ways for the intended parties to agree upon a shared key. By secure, we mean that outsiders should not learn anything about it, but the parties should also have a way to ensure the key they received has not been modified and originates from the correct source.

Public Key Cryptography

Asymmetric cryptography, or public key cryptography, allows for solving this problem, among other things. Its foundations can be traced back to the seminal work of DIFFIE and HELLMAN [DH76]. The idea is that each user now possesses a pair of keys, one of which is public while the other is secret. The two keys are linked mathematically but in a way that it should be infeasible to recover the secret key from the public key only. Going back to the example of encryption for confidential communication, one can devise what we would call public key encryption in order to exchange symmetric keys. For that the sender can use the public key of the recipient to encrypt the symmetric key, and the recipient can then use their own secret key to decrypt it. At the end of this interaction, the two parties share the knowledge of the symmetric key. This however only addresses the confidentiality issue. An active eavesdropper could for example tamper with the

encrypted key, even being oblivious to the key itself – as confidentiality is ensured by the secure encryption – which would then hinder authenticity.

Digital signatures are the second most common primitive in public key cryptography. They represent a certification from the legitimate sender that the data has not been tampered with, and that guarantees the sender's identity. More precisely, the idea is for a user to sign a message using their secret key, and everyone can then verify the signature using the signer's public key. If the verification passes, it means that the signature indeed certifies the data and was issued by the signer holding the corresponding secret key, thus ensuring authenticity. Now, two parties can exchange symmetric keys in a secure way without sharing a secret data a priori. Although they were originally introduced for the purpose of certyfing symmetric key exchanges, digital signatures have become relevant and even necessary in a plethora of other use cases. From certifying credit card payments to electronic passports or more general digital identities, signatures have become more and more important with the evolution of digital technologies.

The Threat of Quantum Computing

By nature, asymmetric cryptography divulges some extra information (the public key) to the public, including attackers, that could be used to learn undesired information. The latter could for instance be the secret key itself, which we would call a *key recovery attack*, but it is actually more general, compelling us to consider other attacks that are more specific to the primitive and security requirements. Taking the example of encryption, what we expect from it is that an eavesdropper should not be able to learn anything about the data that was encrypted. One way to attack is to recover the key and decrypt, but we can also imagine other ways of recovering say one bit of information on the data. This could be considered as a successful attack because it means that one bit has leaked. We thus need to show that even these attacks are infeasible.

For that, we perform a security proof or security reduction which shows that if an adversary is able to successfully carry out the attack, then they would be able to solve a hard mathematical problem. By contraposition, if the mathematical problem is indeed hard to solve, then attacking the system is infeasible. It allows us to make the security of very complex systems depend on a small number of assumptions that are much easier to formulate and study. For example, the link between the public key and the secret key is usually materialized by such a mathematical problem which is assumed or proven hard to solve. This means that recovering the secret key from the public key is mathematically infeasible. The next step is therefore to find families of such hard problems, upon which to build cryptographic primitives. Currently deployed public key cryptography relies for the most part on two families of problems: *factorization* (given $N = pq$ for p, q large primes, find p and q) and *discrete logarithm* (given g and h in a cyclic group, find x such that $h = g^x$). The hardness of these problems seems well established with respect to classical algorithms, as the current records [BGG+22] target a security level which is much below what is used in practice. This is no longer true when considering quantum algorithms.

Quantum computing is a popular area of research whose purpose is driven by the construction of large-scale quantum computers. Such computers rely on the principles of quantum physics to provide a new vision of computation, and unlock new ways and new algorithms that can be much more efficient than the current ones for specific tasks. Just like the unit of measure in a classical computer is the bit, the main metric in quantum computers is called qbit or *quantum bit*. Using the principles of quantum mechanics such as superposition, entanglement, quantification, quantum measurements, wave function collapse, duality, and so on, one can leverage the quantum properties of particles to perform computations. An attractive feature is that a system of N qbit in a superposition of quantum states theoretically has the power of 2^N classical bits. One may thus think that a quantum computer with only 50 qbits would theoretically outperform the best supercomputers today. This threshold of 50 qbits was often referred to as quantum supremacy. But beyond the specificities of quantum algorithms which are very different from classical ones, in practice, stability issues and the phenomenon of quantum decoherence makes it more complicated than it seems. Essentially, by the time the computation is done, decoherence would have already destabilized the quantum system thus interrupting the computation. Currently, industries are building larger and larger quantum computers to achieve this quantum supremacy even in the presence of quantum decoherence. For example, IBM has built a 27-qbit quantum computer in 2019, and upgraded it to 65 qbits in 2020, 127 in 2021, 433 in 2022, 1121 in 2023, and aims for 1386 by the end of 2024. Nevertheless, it remains a very active area of research due to its many fields of applications such as medicine, chemistry, artificial intelligence and also cryptography.

Indeed, in 1994, Peter Shor [Sho94] proposed a quantum algorithm for efficiently solving the factorization and discrete logarithm problems. More precisely, classical computers are only able

to solve them in time exponential in the main parameter (meaning it would take several (millions of) times the age of the universe to solve). Shor's algorithm however could solve them in time polynomial in the main parameter (meaning it would only take a few hours or days). This means that it would wreak havoc on all the currently deployed public key cryptographic algorithms, thus endangering security systems worldwide. Fortunately, Shor's algorithm requires a certain amount of qbits to run with such performance. As a frame of reference, it would take around one million qbits for it to attack a scheme whose security relies on factoring 2048-bits integers. Even with the recent advances in building quantum computers, they are far from being powerful enough to attack cryptography. Nevertheless, it is important to anticipate these threats and prepare for the arrival of such quantum computers. For example, an adversary could store classically encrypted data and decrypt it whenever a large-scale quantum computer is available. If the data is still sensitive by that time, this would cause an important security issue. Just like in many areas, it takes several years between the design of a cryptographic scheme and its standardization, deployment and active use. That is why we need to start finding alternative cryptographic mechanisms that would not be vulnerable to quantum computing.

Post-Quantum Cryptography

Post-Quantum Cryptography aims at addressing those exact concerns and refers to cryptographic constructions that are resistant to quantum algorithms. In particular, they rely on new families of assumptions which are not only immune to Shor's algorithm but also to any other existing quantum algorithm. To start preparing the future of cryptography and replace the current standards relying on factoring and discrete logarithms, the US National Institute of Standards and Technology (NIST) launched a post-quantum cryptography standardization effort in 2016 [NISa]. The goal was to select constructions that would be efficient while withstanding quantum attacks, i.e., attacks carried out by an adversary who has access to a quantum computer on hand. After 69 submitted algorithms based on *lattices*, *codes*, *isogenies*, *multivariate*, and *hashes*, NIST selected 4 algorithms (1 encryption and 3 signatures), 3 of which are lattice-based: the encryption Kyber [BDK+18], and the signatures Falcon [PFH+20] and Dilithium [DKL+18]. They are the ones that are most likely going to replace the current public key cryptographic standards. But what is lattice-based cryptography exactly?

Lattices have been studied as a mathematical object for centuries. The first formal specification of lattices and their properties goes back to Lagrange (1736 - 1813). Since then, Gauss (1777 - 1855) studied the use of lattices in Number Theory, followed by Minkowski (1864 - 1909) who studied the geometry of lattices. They can informally be seen as periodic grids of points in d dimensions, i.e., in $\mathbb{R}^d$. One of the nice features of lattices is that they allow for formulating several mathematical problems that are to this day conjectured hard to solve, even quantumly. A popular example is the *Shortest Vector Problem* (SVP_γ), which consists in finding a point of the given lattice $\mathcal{L}$ in a ball centered around $\mathbf{0}$ of radius $\gamma\lambda_1(\mathcal{L})$, where $\gamma \geq 1$ and $\lambda_1(\mathcal{L})$ is the length of a shortest non-zero vector of $\mathcal{L}$. To study this problem algorithmically, we need a more compact representation of a lattice. It turns out that every lattice $\mathcal{L}$ can be expressed as $\mathcal{L} = \mathbf{B}\mathbb{Z}^k$ where $\mathbf{B} \in \mathbb{R}^{d\times k}$ is called a basis. Bases are not unique, and solving the above problem essentially consists in finding a basis $\mathbf{B}^\star$ that is sufficiently shorter than the given one $\mathbf{B}$. In 1982, Lenstra, Lenstra and Lovàsz [LLL82] introduced one of the first such reduction algorithm for lattices: the well-known LLL algorithm, that aims at reducing the basis of a lattice while having highly orthogonal vectors. It can then be used to solve SVP_γ, but only runs in a reasonable time when γ is exponential in d. The problem is indeed assumed hard for γ polynomial in d, even having access to quantum capabilities. Despite this conjectured *quantum resistance*, SVP_γ is not very suitable to design cryptography upon as it is a *worst-case* problem, i.e., it is easy for many lattices but hard for the worst ones.

Then, in 1996, Ajtai [Ajt96] published a groundbreaking paper on the use of lattices in cryptography. In this paper, Ajtai introduced a new lattice problem called *Short Integer Solution* (SIS) and give the first worst-case to average-case reduction from (a variant of) SVP_γ to SIS. This has tremendous consequences in cryptography because constructions based on this average-case problem now rely on the hardness of the worst instances of lattice problems, and not average ones. It means that if the construction is broken, then the intermediate problem can be easily solved on average and thus that all the instances of the underlying lattice problem can be solved easily as well, even the hardest ones. In 2005, Regev [Reg05] introduced another intermediate problem that benefits from worst-case hardness assumptions as well. He described the *Learning With Errors* (LWE) problem and gave a (quantum) reduction from hard lattice problems to LWE, and also introduced a public-key encryption scheme whose security rely on the hardness of LWE.

Since then, there has been numerous constructions based on these problems, as well as better reductions. Some open questions were also solved thanks to the progress of lattice-based cryptography such as *Fully Homomorphic Encryption* (FHE) [BGV12, BV14, DM15] which was debated to be impossible for a long time. In 2009, Gentry [Gen09] introduced the first FHE scheme based on lattices, as a proof of concept. Albeit appealing due to the theoretical foundation they provide, the original SIS and LWE problems have since then been tweaked in many aspects to offer a better efficiency, e.g., through algebraically structured variants [LPR10, LS15, PP19]. The effort made towards gaining confidence in these tweaked variants, either through theoretical proofs or cryptanalytic assessments, has been extremely important in the development of lattice-based cryptography and is still ongoing.

All these historical arguments constitute some of the reasons why lattices are used for cryptography: they are simple objects, and can be really efficient under a good choice of structure and parameters; they offer provable security for constructions thanks to the underlying lattice problems; they yield the possibility of designing a variety of cryptographic mechanisms; and finally the lattice problems are conjectured to be resistant to quantum attacks.

Cryptography for Privacy

Independently of quantum resistance, basic primitives such as encryption and signatures unfortunately do not cover all the security needs we expect in many situations.

We first consider the situation where an institution needs to delegate some computationally intensive processing of private data to some outside service provider, e.g., the processing of medical data. If the service provider is fully trusted, one only needs to ensure the data is protected during the communications, which can be done by establishing a secure channel using our current cryptographic toolbox. Performing computations could then be done on the decrypted data. However, in many situations, the service provider cannot be trusted with sensitive information. In that case, one would need to compute directly on encrypted data which is not possible for regular encryption mechanisms. The sole use of simple encryption and signatures thus precludes the possibility of trustless external processing, which is now more important than ever through cloud technologies.

We can also consider the use case of digital identity. Assume that a customer owns a digital certificate (embedded in some identification document) authenticating their personal information (name, birthdate, address, social security number, etc). To identify themselves with a standard digital signature, this customer has no other choice than to provide the full set of attributes to the controller or cashier as they are required to run the verification algorithm. However, in many cases, one would not expect having to give out all its personal attributes to attest the validity of only one of them. For example, in the classical situation of age control, the customer only wants to reveal that they are an adult but not even their exact birthdate. Standard signatures thus lead to severe privacy issues. Here one could argue that the situation already occurs in the real world: it is indeed quite common to present an ID document displaying many personal information to a cashier that needs to control your age. In the former, it is natural to assume that the cashier will not memorize all the information contained in the document for further commercial exploitation or identity theft. This does not hold true in the digital world where the users definitely lose control of their data as soon as they reveal them and it is very likely that the same customer will be much more reluctant to provide the same information to a website that needs to verify that they are an adult.

These examples, among many others, portray some of the missing security properties we expect from digital systems. A particular concern is precisely to be able to hide as much information as possible when it is not strictly necessary to disclose it, a property usually referred to as *privacy*. The latter has mostly been considered as a guideline for how private[3] or personal data should be collected, processed, or communicated. Through law enforcement addressing these concerns on personal information[4], privacy has received public advertisement and is now seen as a positive differentiator. These concerns indeed spreaded to a global scale and are no longer limited to specific communities of experts. It is mostly due to the digitalization of communications, services and information itself.

As we have seen, the basic cryptographic mechanisms do not always solve privacy issues. Combining the usual security requirements while limiting the disclosure of private information then calls

[3] In this section, we distinguish between the term *secret*, which denotes information whose disclosure would jeopardize the standard security, and the term *private*, which refers to information which can be disclosed but in a limited fashion.

[4] European General Data Protection Regulation (GDPR)

for the design of advanced[5] cryptographic mechanisms that offers more flexibility on data control. Combining confidentiality and privacy has emerged mostly through processing on sensitive data. One of the most advertised example is FHE which is very versatile as it allows for performing rather general computations directly on the encrypted data, without having to decrypt at any point. The latter would then perfectly address the first use case presented above. There are other types of privacy-enhancing cryptographic mechanisms for confidentiality, e.g., Secure Multiparty Computation (SMPC). This branch of cryptography for privacy has received a lot of attention and contributions and many of them, e.g., FHE, combine these features with a post-quantum security. The state of affairs is vastly different for post-quantum and privacy-enhanced authentication. Only very specific use cases are tackled, e.g., with group signatures [dPLS18, LNPS21] or blind signatures [dPK22, BLNS23a], and constructions are rather scarce. This is in sharp contrast with classical cryptography based on factoring or discrete logarithms which has produced countless efficient constructions of such authentication mechanisms [Cha82, Bra00, CL01, CL02, CL04, BCC04, BSZ05, ASM06, BB08, CDHK15, PS16, FHS19, San20, BEK+21, San21, BFGP22, CLP22, ST23].

This discrepancy between classical and post-quantum constructions, in terms of diversity of constructions but also efficiency of the prior, can be somewhat explained by the rather complex tools that are needed by such constructions. In the case of age control presented above, one essentially needs to prove the authenticity of the disclosed information while hiding everything else. In particular, to avoid being traced across several authentications, we would need to hide the certificate itself. The key components of such systems are then mainly multi-purpose signature schemes, which would provide the desired certification while allowing for the aforementioned selective disclosure, and zero-knowledge proofs, so as to hide sensitive information (including the signature) during authentication. These tools have been studied for decades in the classical setting and been instantiated very efficiently. In the post-quantum case, advances on these subjects are more recent. Impressive progress was made in the last few years on building practical zero-knowledge proofs from lattices, e.g., [dPLS18, LS18, BBC+18, BLS19, YAZ+19, ALS20, LNS20, LNS21, LNP22], which helped developping some more efficient primitives. Regarding signature schemes, most of the efficient designs on lattices do not fit the requirements for building these advanced primitives. More precisely, as we want to hide certain attributes while efficiently proving they have been properly signed, this discards signature schemes that use hash functions. We would indeed need to prove that the hash of the hidden attributes has been properly evaluated which would yield a rather impractical system.

Contributions

The research carried throughout this thesis is motivated by pursuing the development of practical post-quantum cryptography for privacy with a focus on the branch of authentication mechanisms. We adopt a global approach and investigate the overall design of such mechanisms by searching for optimizations at the levels of the protocols themselves, the building blocks they resort to, as well as the security foundations they rely on.

Classical mechanisms have indeed been greatly improved by stretching or formulating new assumptions. As lattice-based cryptography is becoming more mature and is approaching the limits of what can be done with standard lattice assumptions, a promising direction is to mimic the success of classical cryptography by stretching or proposing new post-quantum assumptions. It is then necessary to evaluate their hardness to avoid compromising on security. In parallel, one of the key tools in privacy-driven schemes are trapdoor functions, a publicly computable function that can be inverted only using a secret information called trapdoor. More precisely, the trapdoor functions introduced by Micciancio and Peikert [MP12] procure the versatility we expect to design such advanced schemes. As evidence, most of the current lattice-based designs for privacy are using their mechanism. In the hope of improving the efficiency of these constructions, or design new ones, it is important to reassess the widely adopted tools to identify possible margins for improvement. Finally, finding the synergies between the different building blocks, along with the optimizations we can bring, enables designs of privacy-enhanced primitives to be more efficient while addressing some of the missing requirements pointed out by the use cases above.

[5] In this thesis, we call *advanced* any cryptographic design that goes beyond regular encryption and digital signatures. Examples include fully homomorphic encryption [Gen09], privacy-enhanced authentications such as group signatures [CvH91], blind signatures [Cha82], anonymous credentials [Cha85], etc.

Foundations

Although devising new ad-hoc assumptions to obtain efficient cryptography for privacy is a promising research direction, e.g., [BLNS23b], we focus on keeping the fundamental yet versatile assumptions while stretching them to their limit. Among them, the widely adopted Learning With Errors (LWE) problem introduced by Regev [Reg05] has been used as the security and privacy foundation of countless constructions, including zero-knowledge proofs. It can be formulated as follows. Given a matrix of coefficients $\mathbf{A}$ and a vector $\mathbf{b}$ corresponding to the right-hand side of a noisy modular system of equations in $\mathbf{A}$, i.e., $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q\mathbb{Z}$ where $\mathbf{s}$ is the unknown vector and $\mathbf{e}$ a noise vector, the LWE assumption argues it is hard to solve for $\mathbf{s}$, or to decide if there is even such an $\mathbf{s}$ or if $\mathbf{b}$ is fully random. The problem is then parameterized by the distributions chosen for the *secret* $\mathbf{s}$ and the *error* $\mathbf{e}$. Choosing distributions so that $\mathbf{s}$ and $\mathbf{e}$ are short vectors (in a specified norm) generally translates to a better efficiency of the schemes based on LWE. This comes with the downside of making the problem slightly easier as we essentially reduce the space of possible solutions. We thus need to study the hardness of this problem when stretching the parameters that yield shorter and shorter vectors. Several works [GKPV10, BLP+13, MP13, Mic18, BD20] have taken the matter at hand showing that LWE can be proven reasonably safe even for short distributions, e.g., $\mathbf{s}$ or $\mathbf{e}$ uniform binary vectors.

Unfortunately, the resulting schemes remain quite inefficient, which is why a parallel line of research aimed at adding an additional algebraic structure to the problem in order to enable faster computations and more efficient storage. Among these structured variants [SSTX09, LPR10, LS15, PP19], the *Module*-LWE (M-LWE) assumption proposed an interesting versatility allowing to trade-off efficiency for security and vice-versa. Instead of considering integer coefficients, it considers *algebraic integers* which can essentially be seen as polynomials with integer coefficients. Performing operations on the polynomials can be dealt more efficiently, for the same volume of data, and this also allows one to compact the dimension and storage of matrices. Although Langlois and Stehlé [LS15] thoroughly studied its hardness in the general case, no result was known for stretched parameter regimes, e.g., $\mathbf{s}$ or $\mathbf{e}$ composed of polynomials with binary coefficients.

In this thesis, we provide similar conclusions to those drawn by [GKPV10, BLP+13, MP13, Mic18, BD20] but for the M-LWE assumption. Through four papers [BJRW20, BJRW21, BJRW22, BJRW23], we show that the M-LWE problem remains as hard as its original definition (with $\mathbf{s}$ uniform modulo q, and $\mathbf{e}$ Gaussian) when stretching the secret and error distributions. Albeit with slightly updated parameters, we later use these variants as the security foundations of our privacy-enhanced constructions. We note that our contributions has a larger scope as these assumptions are used in most of the efficient lattice-based cryptographic designs. Our study helps building our confidence in the security of these schemes as well.

These contributions are described in Part I. Chapter 2 first studies the hardness of M-LWE where $\mathbf{s}$ is uniform with small coefficients while $\mathbf{e}$ remains Gaussian. Then, in Chapter 3, we look at M-LWE with $\mathbf{e}$ uniform with small coefficients, first while $\mathbf{s}$ is uniform modulo q and then with $\mathbf{s}$ distributed as $\mathbf{e}$.

Samplers and Signatures

Privacy-oriented authentication mechanisms mostly rely on the versatile trapdoors of Micciancio and Peikert [MP12], and the associated preimage sampler. They define trapdoor functions, in the form of a matrix $\mathbf{A}$, where the image of an input vector can be computed publicly and efficiently, while finding a short preimage $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{u} \bmod q$ is hard without knowledge of the trapdoor information. In particular, the knowledge of the latter allows one to randomize the preimage finding process thus yielding a *preimage sampler*. From these trapdoor functions, one can straightforwardly imagine a signature scheme where $\mathbf{A}$ would be the public key, the trapdoor would be the secret key, and the preimages $\mathbf{x}$ would be signatures. To ensure security of such a sampler, i.e., to make sure preimages $\mathbf{x}$ do not leak information on the trapdoor, the authors need to provide countermeasures which place restrictions on the parameters, and thus on the compactness of the resulting schemes. Most uses of the framework of [MP12] rely on sampling preimages $\mathbf{x}$ following a *spherical* Gaussian distribution whose quality, i.e., variance, is dictated by the size of the secret trapdoor.

The second step of this thesis consists in optimizing the quality of the sampling procedure which directly affects the mechanisms based on it, including privacy-enhanced signatures. By doing so, we show that we can reach promising efficiency while keeping the interesting features of these trapdoors and samplers. To give evidence beyond the application of signatures for privacy,

we design a family of standard signature schemes called Phoenix, which is based on an optimized version of the sampler proposed by Lyubashevsky and Wichs [LW15]. Our scheme is competitive with some lattice signatures selected for standardization such as Dilithium [DKL+18], while enjoying some of the nice features of trapdoor-based designs like tight security proofs, easy security and parameter assessments, etc.

These contributions are described in Part II. We first look at several optimizations of the Micciancio-Peikert sampler [MP12, LW15] in Chapter 4, optimizations that are used in our subsequent constructions. In particular, Chapter 5 presents our new signature schemes Phoenix relying on our *approximate rejection sampler* which combines ideas of [LW15] and [CGM19].

Advanced Signatures and Privacy

We conclude this thesis by providing general-purpose signature schemes called *signature with efficient protocols* (SEP), as coined by Camenisch and Lysyanskaya [CL02], which represent an interesting building block for a large variety of privacy-preserving applications. SEP have led to some of the most efficient advanced authentication mechanisms in the classical setting [CL04, BB08, ASM06, PS16]. Besides the proof-of-concept of [LLM+16] which resulted in totally intractable sizes, there was no post-quantum equivalent prior to our work. We propose two constructions on lattices, representing two waves of improvements over [LLM+16]. In particular, the second one embarks the previously discussed contributions to yield a much more compact scheme without compromising on security.

To demonstrate the utility of our construction, we use our SEP to devise an anonymous credentials system. At a high-level, anonymous credentials involve a user owning some attributes, an issuer in charge of producing a certificate on the user's (possibly hidden) attributes, and a verifier checking the validity of said certificate on the attributes in an (user-)anonymous manner. Such systems encompass the constraints of many use cases such as certifying numerous and possibly secret attributes, authentifying in an anonymous way while enabling selective disclosure of some attributes, etc. For instance, it perfectly addresses the case of age control using an electronic passport discussed above. The issuer embodied by a national authority would produce a credential (the passport) on a person's attributes. Said person (the user) could then show to a controller (the verifier) that they own a legitimately issued passport certifying their hidden personal attributes, including their age proving they are an adult. Our system is directly based on our SEP and the zero-knowledge framework of Lyubashevsky et al. [LNP22], and thus relies on standard assumptions, albeit with stretched parameters as discussed above. Nevertheless, it is competitive with (and even outperforms most of) the exisiting constructions of anonymous credentials on lattices [BLNS23b, LLLW23, BCR+23]. Our work, initially published in [JRS23] and later improved in [AGJ+24], provided the first post-quantum anonymous credentials. The improved version is the first to achieve credential proofs of less than 100 KB while still relying on common lattice assumptions.

Beyond the sole metric of the credential proof size, we showcase the practicality of our design by implementing it in C. Our implementation outperforms that of [BCR+23] which was so far the only existing implementation of post-quantum anonymous credentials. In particular, the full protocol for the (blind) issuance of a credential takes 400 milliseconds on average, while the showing of said credentials to allow for anonymous authentication takes 500 milliseconds on average. These timings should be imperceptible on the user experience in most use cases.

These contributions are described in Part III. We propose our two constructions of SEPs on lattices in Chapter 6, which we then use to derive our anonymous credentials in Chapter 7. Finally, Chapter 8 is dedicated to the implementation of the latter.

1

Preliminaries

In this chapter, we introduce the different mathematical notions that we need throughout the following chapters of this thesis. We start by detailing the notations, before giving the necessary background in algebraic number theory, lattices and discrete probability. We also present the different mathematical assumptions on which relies the security of our constructions. Finally, we give the security models of the cryptographic primitives and protocols that are covered in this thesis.

Contents

General Notations

We start by presenting some of the notations that are used all throughout this thesis. We use $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ to respectively denote the set of natural integers, the ring of integers, the field of rationals, the field of reals and the field of complex numbers. The complex conjugation is denoted by a bar, i.e., $\overline{z}$.

For two integers $a \leq b$, we define the (closed) integer segment between a and b by $[\![a, b]\!] = \{k \in \mathbb{Z} : a \leq k \leq b\}$. From there, we can define $[\![a, b[\![= [\![a, b-1]\!]$, $]\!]a, b]\!] = [\![a+1, b]\!]$ and $]\!]a, b[\![= [\![a+1, b-1]\!]$. When $a = 1$, we simplify the notation $[\![1, b]\!]$ by $[\![b]\!]$.

For a positive integer q, we also define $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ the quotient ring of integers modulo q, which we sometimes identify with the set of representatives $(-q/2, q/2] \cap \mathbb{Z} = [\![-\lfloor (q-1)/2 \rfloor, \lceil (q-1)/2 \rceil]\!]$. More generally, for a ring R, we write $\langle p \rangle$ the principal ideal generated by $p \in R$, and R_p the quotient ring $R/\langle p \rangle = R/pR$.

Vectors are written in bold lowercase letters $\mathbf{x}$, and by convention they are column vectors. To specify the entries of a vector $\mathbf{x} \in R^k$ in the canonical basis, we may write $\mathbf{x} = [x_i]_{i \in [\![k]\!]}$. The k-dimensional vector with only zero entries (resp. 1 entries) is denoted by $\mathbf{0}_k$ (resp. $\mathbf{1}_k$) or simply $\mathbf{0}$ if k is clear from the context. We use $\|\cdot\|_p$ to denote the ℓ_p norm of $\mathbb{R}^k$, i.e., $\|\mathbf{x}\|_p = (\sum_{i \in [\![k]\!]} |x_i|^p)^{1/p}$ for any positive integer p, and $\|\mathbf{x}\|_\infty = \max_{i \in [\![k]\!]} |a_i|$.

Matrices are written in bold capital letters $\mathbf{A}$. For a ring R and two positive integers m and d, we write $R^{m \times d}$ to be the set of matrices with entries in R having m rows and d columns. We may specify a matrix $\mathbf{A} \in R^{m \times d}$ by its columns as $\mathbf{A} = [\mathbf{a}_i]_{i \in [\![d]\!]}$, which means that the columns of $\mathbf{A}$ are the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_d \in R^m$. We sometimes specify a matrix by its entries as $\mathbf{A} = [a_{i,j}]_{i \in [\![m]\!], j \in [\![d]\!]}$. Finally, we can also define a matrix by its blocks. The transpose of a matrix is denoted by a superscript T, i.e., $\mathbf{A}^T$. If the matrix is invertible, we use $\mathbf{A}^{-1}$ to denote its inverse and $\mathbf{A}^{-T} = (\mathbf{A}^T)^{-1}$ to denote the inverse of its transpose. Further, for matrices over $\mathbb{C}$, we use the superscript H to designate the Hermitian operator, i.e., conjugate transpose. The identity matrix of dimension k is denoted by $\mathbf{I}_k$, and, more generally, for a vector $\mathbf{x} \in R^k$, we define $\mathrm{diag}(\mathbf{x})$ to be the matrix of $R^{k \times k}$ whose diagonal entries are the entries of $\mathbf{x}$. We define the spectral norm of a matrix over $\mathbb{C}$ by $\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \|\mathbf{A}\mathbf{x}\|_2 / \|\mathbf{x}\|_2$, and the max-norm by $\|\mathbf{A}\|_{\max} = \max_{i \in [\![m]\!], j \in [\![d]\!]} |a_{i,j}|$.

We use the standard Landau notations, i.e., $O(\cdot), o(\cdot), \omega(\cdot), \Omega(\cdot), \Theta(\cdot)$. In rare occasions, we use $\widetilde{O}$ which further ignores poly-logarithmic factors. We say that a function ε is negligible in λ if $\varepsilon = \lambda^{-\omega(1)}$, e.g., $\varepsilon = 2^{-\lambda}$. We also say that a probability p is overwhelming in λ if $1 - p$ is negligible in λ. When used with subscript, λ is used to denote a security parameter.

1.1 Algebraic Number Theory

We start by presenting the necessary objects and results within algebraic number theory that are relevant in lattice-based cryptography.

1.1.1 Number Fields

An complex number $\zeta \in \mathbb{C}$ is called *algebraic number* if it is a root of a rational polynomial of $\mathbb{Q}[x]$. The unique irreducible monic polynomial $f \in \mathbb{Q}[x]$ of smallest degree that vanishes at ζ is called the *minimal polynomial* of ζ. When f only has integer coefficients, i.e., $f \in \mathbb{Z}[x]$, then ζ is called *algebraic integer*. A *number field* $K = \mathbb{Q}(\zeta)$ is an extension field of $\mathbb{Q}$ of finite degree $n = [K : \mathbb{Q}]$ obtained by adjoining an algebraic number ζ. The degree of the field corresponds to the degree of the minimal polynomial f of ζ. We sometimes call f the *defining polynomial* of K. The set of algebraic integers in K forms a ring called the *ring of integers of* K. The latter is sometimes denoted by $\mathcal{O}_K$ but it will be called R throughout this work, the fraction field K being implicit. We also define the field tensor product $K_{\mathbb{R}} = K \otimes_{\mathbb{Q}} \mathbb{R}$ which can be seen as replacing $\mathbb{Q}$ by $\mathbb{R}$, although $K_{\mathbb{R}}$ is not a field. Finally, for an integer q, we define $R_q = R/qR$ and $\mathbb{T}_q = K_{\mathbb{R}}/qR$.

The inclusion $\mathbb{Z}[\zeta] \subseteq R$ is always true, that is integer polynomials evaluated at ζ are algebraic integers in K. The converse inclusion however only holds for a specific class of number fields. This is the case for some quadratic extensions (i.e., when $\zeta = \sqrt{D}$ with D being a square-free integer such that $D \neq 1 \bmod 4$), cyclotomic fields, or number fields where the defining polynomial f is of square-free discriminant Δ_f. In that context, we often identify by isomorphism the number field K with the quotient $\mathbb{Q}[x]/\langle f \rangle$ and its ring of integers R with $\mathbb{Z}[x]/\langle f \rangle$. In this thesis, we call such fields *monogenic*, although, rigorously speaking, a monogenic number field is some $K = \mathbb{Q}(\zeta)$ for which $R = \mathbb{Z}[\zeta']$ for a possibly different ζ'.

In lattice-based cryptography, we are mostly interested in number fields having good algorithmic properties and representations. In particular, we need the ring of integers to be efficiently computable and have a good basis representation. All these expected properties are met by *cyclotomic fields*.

Definition 1.1 (Cyclotomic Polynomial and Cyclotomic Field)

Let ν be a positive integer. The ν-th cyclotomic polynomial is defined by

$$\Phi_\nu = \prod_{\substack{k \in [\![0,\nu[\![\\ \gcd(k,\nu)=1}} \left(x - e^{i\frac{2k\pi}{\nu}}\right).$$

It is monic, has integer coefficients, is irreducible in $\mathbb{Q}[x]$ and has degree $n = \varphi(\nu) = \left|\mathbb{Z}_\nu^\times\right|$. If ζ_ν to be a root of Φ_ν, then ζ_ν is an algebraic integer and $K_\nu = \mathbb{Q}(\zeta_\nu)$ is called the ν-th cyclotomic field. When $\nu \neq 2 \bmod 4$, ν is also corresponds to the *conductor* of K_ν.

The most popular choice for ν that leads to the most efficient constructions is $\nu = 2^{\mu+1}$ for some non-negative integer μ. In this case, $\Phi_\nu = x^n + 1$ with $n = \varphi(2^{\mu+1}) = 2^\mu$. We later refer to them as power-of-two cyclotomic fields. Such fields have good geometric and algorithmic properties as we will see below when introducing embeddings of the field. Another nice feature is the simple tower structure. Indeed, when $\nu = 2^{\mu+1}$, it holds that $K_{\nu/2} \subset K_\nu$. Unless stated otherwise, $K = \mathbb{Q}(\zeta)$ denotes a general number field of degree n.

1.1.2 Coefficient, Canonical and Minkowski Embeddings

Given a number field $K = \mathbb{Q}(\zeta)$ of degree n, there are several ways of mapping it into more usual objects. We call such maps embeddings. In this work, we consider several embeddings which naturally arise in lattice-based cryptography, namely the *coefficient embedding*, the *canonical embedding* and the *Minkowski embedding*.

Coefficient Embedding

A number field can be seen as a vector space of finite dimension n over the rationals with basis $\{1, \zeta, \ldots, \zeta^{n-1}\}$, meaning that each element $a \in K$ can be expressed as $a = \sum_{j \in [\![0,n[\![} a_j \zeta^j$ with $a_j \in \mathbb{Q}$ for all j. As such, a is naturally mapped to a vector of $\mathbb{Q}^n$. The *coefficient embedding* is then the isomorphism, which we denote by τ, between K and $\mathbb{Q}^n$ that maps the element $a \in K$ to its coefficient vector $\tau(a) = [a_0 | \ldots | a_{n-1}]^T$. For simplicity, for $k \in [\![0, n[\![$ we use τ_k to denote the k-th projection, that is $\tau_k(a) = a_k \in \mathbb{Q}$. The coefficient embedding can also be extended to $K_\mathbb{R}$ and thus maps to $\mathbb{R}^n$. We also extend the notation to vectors by applying the embedding entrywise and concatenating the resulting vectors.

We can then consider the usual ℓ_p norms over K by $\|a\|_p := \|\tau(a)\|_p$. Unless specified otherwise, the norm of an element in $K, K_\mathbb{R}, R$ or vectors of such spaces is with respect to the coefficient embedding. For a positive integer η, we define $S_\eta = \tau^{-1}([\![-\eta, \eta]\!]^n)$ corresponding to elements with integer coefficients and ℓ_∞ norm at most η. We also define $T_\eta = \tau^{-1}([\![0, \eta]\!]^n)$.

Canonical Embedding

Another way to embed K is via the *canonical embedding*. A number field K has exactly n field homomorphisms $\sigma_1, \ldots, \sigma_n$ which are characterized by the fact they map ζ to one of the distinct roots of the defining polynomial f. We use t_1 to denote the number of real roots of f and t_2 the number of pairs of complex roots. Since f has rational coefficients, its roots come in conjugate pairs. We order the field embeddings so that $\sigma_1, \ldots, \sigma_{t_1}$ map ζ to one of the real roots, and $\sigma_{t_1+1}, \ldots, \sigma_{t_1+2t_2}$ map it to one of the complex roots and such that $\sigma_{t_1+t_2+j}(\zeta) = \overline{\sigma_{t_1+j}(\zeta)}$ for $j \in [\![t_2]\!]$. The canonical embedding is then denoted by σ and defines the ring homomorphism from K to $\mathbb{C}^n$ by $\sigma(a) = [\sigma_1(a) | \ldots | \sigma_n(a)]^T$ where addition and multiplication are performed entrywise. It is once again extended to vectors in the natural way by concatenation. We then define two different norms over K^d for some positive integer d. For $\mathbf{a} \in K^d$, we define $\|\mathbf{a}\|_{\infty,\infty} = \|\sigma(\mathbf{a})\|_\infty = \max_{i \in [\![d]\!], j \in [\![n]\!]} |\sigma_j(a_i)|$ and $\|\mathbf{a}\|_{2,\infty} = \max_{j \in [\![n]\!]} \sqrt{\sum_{i \in [\![d]\!]} |\sigma_j(a_i)|^2}$. We also define the algebraic norm, or field norm, of an element $a \in K$ as $N(a) = \prod_{j \in [\![n]\!]} \sigma_j(a) \in \mathbb{Q}$. Note that the algebraic norm is multiplicative, namely for all $a, b \in K$, it holds that $N(ab) = N(a)N(b)$.

We additionally define the conjugate of an element via the canonical embedding. More precisely, for all $a \in K_\mathbb{R}$, we define $a^* = \sigma^{-1}(\overline{\sigma(a)})$. In the case of cyclotomic fields where ζ is a root of unity, conjugation comes down to evaluating at ζ^{-1}. That is that for all $a \in K_\mathbb{R}$, $a^* = \sum_{j \in [\![0,n[\![} \tau_j(a) \zeta^{-j}$. When the conjugate is applied to a matrix of $K_\mathbb{R}^{m \times d}$, it actually corresponds to the conjugate

transpose, that is $\mathbf{A}^* = [a^*_{j,i}]_{(i,j)} \in K_{\mathbb{R}}^{d\times m}$. We denote by $K_{\mathbb{R}}^{+}$ the subspace of $K_{\mathbb{R}}$ of self-adjoint elements, i.e., that verifies $a^* = a$. We also define $K_{\mathbb{R}}^{++}$ to be the subset $\{a \in K_{\mathbb{R}}^{+} : \sigma(a) \in (\mathbb{R}^{+*})^n\}$.

Minkowski Embedding

Due to the conjugation symmetry of the canonical embedding, its range is a subset of the following space.

$$H = \{\mathbf{a} \in \mathbb{R}^{t_1} \times \mathbb{C}^{2t_2} : \forall j \in [\![t_2]\!], a_{t_1+t_2+j} = \overline{a_{t_1+j}}\}.$$

The latter is often called the space H or Minkowski space. We can easily verify that H is an $\mathbb{R}$-vector space of dimension $n = t_1 + 2t_2$ where an orthonormal basis is given by the columns of the following matrix $\mathbf{U}$.

$$\mathbf{U} = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{2}\mathbf{I}_{t_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{t_2} & i\mathbf{I}_{t_2} \\ \mathbf{0} & \mathbf{I}_{t_2} & -i\mathbf{I}_{t_2} \end{bmatrix}.$$

In particular, it means that K can be mapped to $\mathbb{R}^n$ by what is called the *Minkowski embedding*, which is defined by $\sigma_H = \mathbf{U}^H\sigma$.

Distortion Between Embeddings

These embeddings play an important role in lattice-based cryptography. Grasping the relation between the three embeddings helps understanding some more fundamental properties of the underlying field, including geometrical aspects. We have seen that σ_H and σ are linked linearly by the orthonormal transformation $\mathbf{U}^H$. However, this is not the case between τ and σ. More precisely, we have the following relation

$$\sigma(a) = \mathbf{V}\tau(a) \text{ for all } a \in K, \text{ where } \mathbf{V} = \begin{bmatrix} 1 & \alpha_1 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \dots & \alpha_2^{n-1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & \alpha_n & \dots & \alpha_n^{n-1} \end{bmatrix},$$

is the Vandermonde matrix defined by the roots $(\alpha_j)_{j\in[\![n]\!]} = (\sigma_j(\zeta))_{j\in[\![n]\!]}$ of the defining polynomial f. The transformation does not necessarily carry the structure from one embedding to the other, e.g., a vector that is binary with respect to τ need not to be binary with respect to σ. Changing the embedding naturally impacts the norm which can be captured by the inequalities

$$\frac{1}{\|\mathbf{V}^{-1}\|_2}\big\|\tau(a)\big\|_2 \le \big\|\sigma(a)\big\|_2 \le \|\mathbf{V}\|_2\big\|\tau(a)\big\|_2$$

More specifically, the singular values of $\mathbf{V}$ help assessing the distortion between both embeddings. Roşca et al. [RSW18] and Blanco-Chacón [Bla22] give additional insight on this distortion for specific number fields. For example, the case of power-of-two cyclotomic fields is very favorable as $\mathbf{V} = \sqrt{n}\mathbf{P}$ where $\mathbf{P}$ is a unitary matrix. In that case, τ and σ are isometric up to a factor of $\sqrt{n}$. In Chapter 2, we are interested in the parameter $B_\eta = \max_{a\in S_\eta}\big\|\sigma(a)\big\|_\infty$ for a positive integer η. It is inherent to the number field and intervenes in the proof of Lemma 2.2, 2.5 and 3.7. Here, we provide an upper-bound on B_η, that is further simplified for cyclotomic fields.

Lemma 1.1 (Bound on B_η)

Let K be a number field of degree n, and $\mathbf{V}$ the associated Vandermonde matrix. Let η be a positive integer. Then, it holds that $1 \le B_\eta = \max_{a\in S_\eta}\big\|\sigma(a)\big\|_\infty \le n\eta\|\mathbf{V}\|_{\max}$. In particular, for cyclotomic fields, it yields $1 \le B_\eta \le n\eta$.

Proof (Lemma 1.1). The lower bound is due to the fact that every non-zero element a of $R = \mathcal{O}_K$ has algebraic norm $N(a) \ge 1$, which implies that $\big\|\sigma(a)\big\|_\infty \ge 1$. Let a be in S_η and

$i \in [\![n]\!]$. Then it holds that

$$\begin{aligned}\left|\sigma_i(a)\right| \leq \sum_{j\in[\![0,n[\![} \left|\tau_j(a)\sigma_i(\zeta^j)\right| &= \sum_{j\in[\![0,n[\![} \left|\tau_j(a)\right|\left|\alpha_i^j\right| \\ &\leq \left\|\tau(a)\right\|_1 \|\mathbf{V}\|_{\max} \leq n\eta\|\mathbf{V}\|_{\max}.\end{aligned}$$

Taking the maximum over all $i \in [\![n]\!]$ and $a \in S_\eta$ gives $B_\eta \leq n\eta\|\mathbf{V}\|_{\max}$. In the case of cyclotomic fields, the α_i are roots of unity and therefore of magnitude 1. Hence $\|\mathbf{V}\|_{\max} = 1$ which yields the bound $B_\eta \leq n\eta$.

1.1.3 Multiplication Matrices

The multiplication in K can be interpreted in the embedded spaces with respect to each embedding. It translates into a matrix-vector multiplication once embedded. In the canonical embedding, the multiplication matrix can be easily expressed as we have that for all a, b in K, $\sigma(a \cdot b) = \sigma(a) \odot \sigma(b) = \mathrm{diag}(\sigma(a))\sigma(b)$, where $\odot$ denotes the Hadamard product (entrywise product). Therefore, we can consider the ring homomorphism M_σ from K to $\mathbb{C}^{n\times n}$ defined by $M_\sigma(a) = \mathrm{diag}(\sigma(a))$. It then verifies

$$\forall (a, b) \in K^2, \sigma(ab) = M_\sigma(a)\sigma(b).$$

This is not specific to the canonical embedding and we can define similar homomorphism M_τ and M_{σ_H} with respect to the coefficient and Minkowski embeddings. Due to the linear relation between embeddings, we have that for all a in K:

$$M_\tau(a) = \mathbf{V}^{-1}M_\sigma(a)\mathbf{V}, \text{ and } M_{\sigma_H}(a) = \mathbf{U}^H M_\sigma(a)\mathbf{U}.$$

Because $\mathbf{V}$ is not orthonormal as opposed to $\mathbf{U}$, the expression of M_τ seems quite involved. We give in Lemma 1.2 another expression of M_τ for general number fields which is greatly simplified for power-of-two cyclotomic fields.

Lemma 1.2 (Expression of M_τ)

Let K be a number field of degree n, and let us write its defining polynomial as $f = x^n + \sum_{j\in[\![0,n[\![} f_j x^j$. Then for all a in K, it holds that

$$M_\tau(a) = \sum_{j\in[\![0,n[\![} \tau_j(a)\mathbf{C}^j, \text{ with } \mathbf{C} = \begin{bmatrix} 0 & & 0 & -f_0 \\ & \mathbf{I}_{n-1} & & -f_1 \\ & & & \vdots \\ & & & -f_{n-1} \end{bmatrix}$$

the companion matrix of f.

Proof (Lemma 1.2). Let $f = x^n + \sum_{k=0}^{n-1} f_k x^k$ denote the defining polynomial of $K = \mathbb{Q}(\zeta)$. Let $\mathbf{C}$ denote the companion matrix of f, as in the lemma statement. It is well known that the characteristic (and minimal) polynomial of the companion matrix of f is f itself. This entails that $\mathbf{C}$ has the roots of f for eigenvalues, which we denote by $\alpha_1, \ldots, \alpha_n$. Recall that the field embeddings are such that $\sigma_i(\zeta) = \alpha_i$ for all $i \in [\![n]\!]$. Since the roots of f are distinct, it means that $\mathbf{C}$ is diagonalizable. More precisely, the diagonalization of companion matrices is well-known and gives that $\mathbf{C} = \mathbf{V}^{-1}\mathrm{diag}(\alpha_1, \ldots, \alpha_n)\mathbf{V} = \mathbf{V}^{-1}\mathrm{diag}(\sigma(\zeta))\mathbf{V}$. Now let a be in K. We have that $M_\tau(a) = \mathbf{V}^{-1}\mathrm{diag}(\sigma(a))\mathbf{V}$. We can then rewrite this expression in terms

of the τ_k and $\mathbf{C}$ as follows.

$$\begin{aligned}\mathbf{V}^{-1}\mathrm{diag}(\sigma(a))\mathbf{V} &= \mathbf{V}^{-1}\mathrm{diag}\left(\sigma_1\left(\sum_{j\in[\![0,n[\![}\tau_j(a)\zeta^j\right),\dots,\sigma_n\left(\sum_{j\in[\![0,n[\![}\tau_j(a)\zeta^j\right)\right)\mathbf{V}\\ &= \sum_{j\in[\![0,n[\![}\tau_j(a)\mathbf{V}^{-1}\mathrm{diag}(\sigma_1(\zeta)^j,\dots,\sigma_n(\zeta)^j)\mathbf{V}\\ &= \sum_{j\in[\![0,n[\![}\tau_j(a)\mathbf{V}^{-1}\mathrm{diag}(\sigma(\zeta))^j\mathbf{V}\\ &= \sum_{j\in[\![0,n[\![}\tau_j(a)\mathbf{C}^j,\end{aligned}$$

concluding the proof.

In power-of-two cyclotomic fields, we have $f = x^n + 1$ which gives that $\mathbf{C}$ is the generating nega-circulant matrix. The expression of $M_\tau(a)$ can thus be simplified to

$$M_\tau(a) = \begin{bmatrix} a_0 & -a_{n-1} & \dots & -a_1 \\ a_1 & a_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & -a_{n-1} \\ a_{n-1} & a_{n-2} & \dots & a_0 \end{bmatrix} \in \mathbb{Q}^{n\times n},$$

which is itself nega-circulant, with $a_j = \tau_j(a)$. It also holds for general number fields that $M_\tau(a^*) = M_\tau(a)^T$.

Using the multiplication matrix maps, we can translate matrix-vector operations over K^d into matrix-vector operations over $\mathbb{R}^{nd}$ or $\mathbb{C}^{nd}$ by extending the maps to a matrix in $K^{m\times d}$. More precisely, for a matrix $\mathbf{A} = [a_{i,j}]_{(i,j)} \in K^{m\times d}$, we define the block matrix $M_\sigma(\mathbf{A}) = [M_\sigma(a_{i,j})]_{(i,j)} \in \mathbb{C}^{nm\times nd}$. We define $M_\tau(\mathbf{A})$ and $M_{\sigma_H}(\mathbf{A})$ similarly. The multiplication matrix maps allow us to determine spectral properties of field elements or matrices over K. We show in the following lemma that although the embeddings are fairly different in nature, they preserve the spectral behavior, i.e. the singular values of $M_\tau(\mathbf{A})$, $M_\sigma(\mathbf{A})$, and $M_{\sigma_H}(\mathbf{A})$ are equal. As a result, in the remainder of this thesis, we write $\|\mathbf{A}\|_2$ to denote $\left\|M_\tau(\mathbf{A})\right\|_2$. This relies on a unified analysis by Rjasanow [Rja94] which gives conditions to obtain the eigenvalues of a matrix when described by blocks. In our setting, we end up showing that the spectral analysis of the entire block matrix $M_\tau(\mathbf{A})$ comes down to finding the singular values of the n embedded matrices $\sigma_k(\mathbf{A})$. For convenience, we write $S(\mathbf{A})$ the set of all singular values of a complex matrix $\mathbf{A}$.

Lemma 1.3 (Spectral Analysis)

Let K be a number field of degree n, and m, d positive integers. Let $\mathbf{A}$ be a matrix in $K^{m\times d}$. It holds that

$$S(M_\tau(\mathbf{A})) = \bigcup_{k\in[\![n]\!]} S(\sigma_k(\mathbf{A})) = S(M_\sigma(\mathbf{A})) = S(M_{\sigma_H}(\mathbf{A})),$$

where $\sigma_k(\mathbf{A}) = [\sigma_k(a_{i,j})]_{(i,j)\in[\![m]\!]\times[\![d]\!]}$. In particular, we have $\left\|M_\tau(\mathbf{A})\right\|_2 = \max_{k\in[\![n]\!]} \left\|\sigma_k(\mathbf{A})\right\|_2$.

Proof (Lemma 1.3). For (i,j) in $[\![m]\!]\times[\![d]\!]$, we define the polynomial evaluation function $a_{i,j}(\cdot) : t \mapsto \sum_{k\in[\![0,n[\![}\tau_k(a_{i,j})t^k$. The way $a_{i,j} \in K$ is defined, we have $a_{i,j} = a_{i,j}(\zeta)$. Lemma 1.2 gives $M_\tau(a_{i,j}) = \sum_{k\in[\![0,n[\![}\tau_k(a_{i,j})\mathbf{C}^k = a_{i,j}(\mathbf{C})$. Finally, for $k \in [\![n]\!]$, if α_k denotes $\sigma_k(\zeta)$, it holds that $a_{i,j}(\alpha_k) = \sigma_k(a_{i,j})$. We then define the function over complex matrices by $\mathbf{A}(t) = [a_{i,j}(t)]_{(i,j)}$ for all t. By the prior observations, we get that $\mathbf{A} = \mathbf{A}(\zeta)$, $M_\tau(\mathbf{A}) = \mathbf{A}(\mathbf{C})$, and $\mathbf{A}(\alpha_k) = \sigma_k(\mathbf{A})$.

Consider $\mathbf{B}(t) = \mathbf{A}(t)^H\mathbf{A}(t)$. The same reasoning holds for $\mathbf{A}(t)\mathbf{A}(t)^H$. First, notice that $\mathbf{C}$ is diagonalizable with eigenvalues $\alpha_1,\dots,\alpha_n$, as its minimal polynomial is the minimal

polynomial of ζ. [Rja94] then states that $\mathbf{B}(\mathbf{C})$ is diagonalizable if and only if the n matrices $\mathbf{B}(\alpha_k)$ are diagonalizable, in which case the spectrum (set of eigenvalues) of $\mathbf{B}(\mathbf{C})$ is the union of the spectra of the $\mathbf{B}(\alpha_k)$. By construction, for every k in $[\![n]\!]$, $\mathbf{B}(\alpha_k)$ is Hermitian and therefore diagonalizable. Since the eigenvalues of $\mathbf{B}(\alpha_k)$ (resp. $\mathbf{B}(\mathbf{C})$) are the square singular values of $\mathbf{A}(\alpha_k)$ (resp. $\mathbf{A}(\mathbf{C})$), we directly get that

$$S(\mathbf{A}(\mathbf{C})) = \bigcup_{k\in[\![n]\!]} S(\mathbf{A}(\alpha_k)),$$

which proves the first equality.

For the third equality, recall that $M_{\sigma_H} = \mathbf{U}^H M_\sigma \mathbf{U}$. It implies that $M_{\sigma_H}(\mathbf{A}) = (\mathbf{I}_m \otimes \mathbf{U}^H) M_\sigma(\mathbf{A})(\mathbf{I}_d \otimes \mathbf{U})$. Since $\mathbf{U}$ is unitary, we have $S(M_{\sigma_H}(\mathbf{A})) = S(M_\sigma(\mathbf{A}))$.

We now prove the second equality. Recall that $M_\sigma(\mathbf{A})$ is the block matrix of size $nm \times nd$ whose block $(i,j) \in [\![m]\!] \times [\![d]\!]$ is $\mathrm{diag}(\sigma(a_{i,j}))$. The matrix can therefore be seen as a $m \times d$ matrix with blocks of size $n \times n$. The idea is now to permute the rows and columns of $M_\sigma(\mathbf{A})$ to end up with a matrix of size $n \times n$ with blocks of size $m \times d$ only on the diagonal. For that, we define the following permutation π_k of $[\![nk]\!]$ for any positive integer k. For all $i \in [\![nk]\!]$, write $i - 1 = k_1^{(i)} + nk_2^{(i)}$, with $k_1^{(i)} \in [\![0, n[\![$ and $k_2^{(i)} \in [\![0, k[\![$. Then, define $\pi_k(i) = 1 + k_2^{(i)} + k \cdot k_1^{(i)}$. This is a well-defined permutation based on the uniqueness of the Euclidean division. We can then define the associated permutation matrix $\mathbf{P}_{\pi_k} = [\delta_{i,\pi_k(j)}]_{(i,j)\in[\![nk]\!]^2} \in \mathbb{R}^{nk\times nk}$. Then, by defining $\mathbf{P}_{\pi_m}$ and $\mathbf{P}_{\pi_d}$ as described, it holds that

$$\mathbf{P}_{\pi_m} M_\sigma(\mathbf{A}) \mathbf{P}_{\pi_d}^T = \begin{bmatrix} \sigma_1(\mathbf{A}) & & \\ & \ddots & \\ & & \sigma_n(\mathbf{A}) \end{bmatrix}.$$

Since $\mathbf{P}_{\pi_m}, \mathbf{P}_{\pi_d}$ are permutation matrices, they are also unitary and therefore $S(M_\sigma(\mathbf{A})) = S(\mathbf{P}_{\pi_m} M_\sigma(\mathbf{A}) \mathbf{P}_{\pi_d}^T)$. As $\mathbf{P}_{\pi_m} M_\sigma(\mathbf{A}) \mathbf{P}_{\pi_d}^T$ is block-diagonal, it directly holds that $S(\mathbf{P}_{\pi_m} M_\sigma(\mathbf{A}) \mathbf{P}_{\pi_d}^T) = \cup_{k\in[\![n]\!]} S(\sigma_k(\mathbf{A}))$, thus proving the second equality.

Finally, by taking the maximum of the sets involved in the first equality, we obtain $\left\|M_\tau(\mathbf{A})\right\|_2 = \max_{k\in[\![n]\!]} \left\|\sigma_k(\mathbf{A})\right\|_2$ as claimed.

1.1.4 Subring Embedding in Power-of-Two Cyclotomics

In our construction of signature with efficient protocols of Section 6.4, we leverage the tower structure of power-of-two cyclotomic rings. In this section we only consider the cyclotomic field of power-of-two conductors ν. As noted in Section 1.1.1, we have $K_{\nu/2} \subset K_\nu$. But more formally, we can embed K_ν into $K_{\nu/2}^2$. Let $a \in K_\nu$ be written as $a = \sum_{j\in[\![0,n[\![} a_j \zeta_\nu^j$, with $\zeta_\nu = \exp(i \cdot 2\pi/\nu)$. Then, we can write a as $a^{(e)}(\zeta_\nu^2) + \zeta_\nu a^{(o)}(\zeta_\nu^2)$ where

$$a^{(e)}(t) = \sum_{j\in[\![0,n/2[\![} a_{2j} t^j, \text{ and } a^{(o)}(t) = \sum_{j\in[\![0,n/2[\![} a_{2j+1} t^j.$$

Because $\zeta_\nu^2 = \zeta_{\nu/2}$, we can thus see $a^{(e)}(\zeta_\nu^2)$ and $a^{(o)}(\zeta_\nu^2)$ as elements of $K_{\nu/2}$, thus mapping K_ν to $K_{\nu/2}^2$. This represents one step in the tower. We can generalize it as follows.

Subring Embedding

This generalization and the use of subrings lead to interesting performance improvements in our systems of Chapter 6 when using zero-knowledge arguments. In [LNPS21], the authors explain that using a ring of smaller degree allows for reducing the proof size. This is however at the expense of a lower compression of the keys for the signature scheme. A solution to obtain the best of both worlds is to use a ring R of degree n for the signature, and a subring $\widehat{R}$ of degree $\widehat{n}|n$ for the zero-knowledge proof. This requires embedding the relations over R into relations over $\widehat{R}$ (in turn increasing the dimension by $\widehat{k} = n/\widehat{n}$). This subring embedding strategy is already used implicitly in [LNPS21, LNP22], and we give for completeness all the algebraic details needed to map R to $\widehat{R}$ or more generally K_ν to $K_{\nu/k}^k$.

We let $\widehat{\nu}|\nu$ be a power of two, and $\widehat{n}|n$ be the corresponding degree. We also call $\widehat{k}$ the ratio $\widehat{k} = \nu/\widehat{\nu} = n/\widehat{n}$. For clarity, we define $K = K_\nu$ and $\widehat{K} = K_{\widehat{\nu}}$. To avoid confusion in this section, when relevant and not clear from the context, we use $\otimes_K$ to denote the product in K, and $\otimes_{\widehat{K}}$ for the product in $\widehat{K}$. Also, for clarity we reason over ring elements as if they were polynomials in some x and not ζ_ν or $\zeta_{\widehat{\nu}}$.

Even though there are many ways to embed K into $\widehat{K}^{\widehat{k}}$, we define the embedding $\theta : K \to \widehat{K}^{\widehat{k}}$ as follows. For $a = \sum_{\ell \in [\![0,n[\![} a_\ell x^\ell \in K$ with $(a_\ell)_\ell \in \mathbb{Q}^n$, and for all $i \in [\![0, \widehat{k}[\![$, define $\widehat{a}_i = \sum_{j \in [\![0,\widehat{n}[\![} a_{\widehat{k}j+i} x^j \in \widehat{K}$. Then, the embedding of a is defined by $\theta(a) = [\widehat{a}_0 | \dots | \widehat{a}_{\widehat{k}-1}]^T \in \widehat{K}^{\widehat{k}}$. This embedding relies on the fact that a can be uniquely written as $a = \sum_{i \in [\![0,\widehat{k}[\![} \sum_{j \in [\![0,\widehat{n}[\![} a_{\widehat{k}j+i} x^{\widehat{k}j+i}$, which itself equals $\sum_{i \in [\![0,\widehat{k}[\![} \widehat{a}_i(x^{\widehat{k}}) \otimes_K x^i$. This in particular defines the inverse embedding θ^{-1}.

Operations and Multiplication Matrix

The embedding θ (and its inverse) is clearly linear, which means that addition in K can be performed over $\widehat{K}^{\widehat{k}}$ coefficient-wise and vice-versa. In [LNPS21, Lem. 2.11], Lyubashevsky et al. recall that the multiplication $a \otimes_K b$ can also be performed on the embeddings $\theta(a), \theta(b)$ using a carefully defined multiplication $\otimes_{\widehat{K}^{\widehat{k}}} : \widehat{K}^{\widehat{k}} \times \widehat{K}^{\widehat{k}} \to \widehat{K}^{\widehat{k}}$, that can be carried using only additions and $\otimes_{\widehat{K}}$. For two elements $a, b \in K$ such that $\theta(a) = [\widehat{a}_0 | \dots | \widehat{a}_{\widehat{k}-1}]^T$ and $\theta(b) = [\widehat{b}_0 | \dots | \widehat{b}_{\widehat{k}-1}]^T$, we have $\theta(a) \otimes_{\widehat{K}^{\widehat{k}}} \theta(b) = [\widehat{c}_0 | \dots | \widehat{c}_{\widehat{k}-1}]^T$, where

$$\widehat{c}_\ell = \sum_{\substack{i,j \in [\![0,k[\![\\ i+j = \ell \bmod \widehat{k}}} \widehat{a}_i \otimes_{\widehat{K}} \widehat{b}_j \otimes_{\widehat{K}} x^{\lfloor \frac{i+j}{k} \rfloor},$$

for all $\ell \in [\![0, \widehat{k}[\![$. We can simplify this expression by observing that for a fixed $j \in [\![0, \widehat{k}[\![$, there is only one $i \in [\![0, \widehat{k}[\![$ verifying $i + j = \ell \bmod \widehat{k}$, namely $i = \ell - j$ if $\ell \geq j$, and $i = \ell - j + \widehat{k}$ otherwise. We thus get

$$\begin{aligned} \widehat{c}_\ell &= \sum_{j \in [\![0,\ell]\!]} \widehat{a}_{\ell-j} \otimes_{\widehat{K}} \widehat{b}_j + \sum_{j \in [\![\ell+1,\widehat{k}[\![} \widehat{a}_{\ell-j+\widehat{k}} \otimes_{\widehat{K}} x \otimes_{\widehat{K}} \widehat{b}_j \\ &= [\widehat{a}_\ell | \dots | \widehat{a}_0 | \widehat{a}_{\widehat{k}-1} x | \dots | \widehat{a}_{\ell+1} x] \cdot \theta(b). \end{aligned}$$

This rewriting highlights the expression of a multiplication matrix $M_\theta(a)$ so that $\theta(a \otimes_K b) = \theta(a) \otimes_{\widehat{K}^{\widehat{k}}} \theta(b) = M_\theta(a)\theta(b)$ where the latter matrix-vector product is performed in $\widehat{K}$. Formally, we have

$$M_\theta(a) = \begin{bmatrix} \widehat{a}_0 & \widehat{a}_{\widehat{k}-1} x & \dots & \widehat{a}_1 x \\ \widehat{a}_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \widehat{a}_{\widehat{k}-1} x \\ \widehat{a}_{\widehat{k}-1} & \dots & \widehat{a}_1 & \widehat{a}_0 \end{bmatrix},$$

Another useful way to express $M_\theta(a)$ is by observing that for $i \in [\![0, \widehat{k}[\![$, the i-th column of $M_\theta(a)$ corresponds to $\theta(a \otimes_K x^i)$. Hence $M_\theta(a) = [\theta(a) | \theta(a \otimes_K x) | \dots | \theta(a \otimes_K x^{\widehat{k}-1})]$. We naturally extend the embedding θ to vectors and the multiplication map M_θ blockwise to vectors and matrices over K, i.e., for $\mathbf{A} = [a_{i,j}]_{i,j} \in K^{m \times d}$ by $M_\theta(\mathbf{A}) = [M_\theta(a_{i,j})]_{i,j} \in \widehat{K}^{\widehat{k}m \times \widehat{k}d}$.

Remark 1.1 (Subring and Coefficient Embeddings)

The coefficient embedding now appears as a specific subring embedding. Indeed, when $\widehat{K}$ is of degree 1, we get $\widehat{n} = 1$, $\widehat{k} = n$ and $\widehat{K} = \mathbb{Q}$. The multiplication matrices $M_\tau(a)$ and $M_\theta(a)$ then perfectly match because in this ring of degree 1, x is equal to -1.

The coefficient embedding can be defined with respect to K but also with respect to a subring $\widehat{K}$ of K. If needed, we differentiate them by τ_K and $\tau_{\widehat{K}}$. When both are present, we use $\widehat{S}_\eta$ and $\widehat{T}_\eta$ for the corresponding sets S_η and T_η but with respect to the subring $\widehat{K}$.

1.1.5 Ideals, Units and Modules

In lattice-based cryptography, we usually work over the quotient ring $R_q = R/qR$. It then becomes necessary to talk about ideals, ideal factorization, and modules.

Ideals and Modules

An ideal $\mathcal{I} \subseteq R$ is a non-zero additive subgroup of R that is closed under multiplication by R. Just like we consider prime numbers in the ring $\mathbb{Z}$, we can define prime ideals of R. An ideal $\mathfrak{p} \neq R$ is *prime* if for all a, b in R, $ab \in \mathfrak{p}$ implies that either a or b lies in $\mathfrak{p}$. In R, an ideal $\mathfrak{p}$ is prime if and only if it is maximal, implying that $R/\mathfrak{p}$ is a field. For two ideals $\mathcal{I}, \mathcal{J}$, the sum $\mathcal{I} + \mathcal{J}$ is the set of all $a + b$, where $(a, b) \in \mathcal{I} \times \mathcal{J}$, while the product $\mathcal{I}\mathcal{J}$ is the set of all finite sums of ab for $(a, b) \in \mathcal{I} \times \mathcal{J}$.

It turns out that every ideal of R can be factored in a product of prime ideals. We now focus on the principal ideal $\langle q \rangle = qR$ for a positive integer q. The integer q is said to be *unramified* in R if the ideal $\langle q \rangle$ can be factored into a product of *distinct* prime ideals $\prod_{i \in [\![\kappa]\!]} \mathfrak{p}_i$. We say that q is *fully split* in R if $\kappa = n$ in the above factorization, where n is the degree of the number field. On the contrary, we say that q is *inert* when $\kappa = 1$, that is $\langle q \rangle$ is prime.

We extend the field norm and define the norm of an ideal $N(\mathcal{I})$ as the index of $\mathcal{I}$ as an additive subgroup of R, which corresponds to $N(\mathcal{I}) = \big|R/\mathcal{I}\big|$. The norm is still multiplicative and verifies $N(\langle a \rangle) = \big|N(a)\big|$ for any $a \in R$.

Modules generalize the notion of vector spaces where the underlying field is only a ring. In particular, modules are abelian groups endowed with a *scalar multiplication* by elements of the underlying ring. In this work, we only consider free modules, i.e., which have a basis. For example, for a positive integer d, R^d is a free R-module of rank d. Considering matrices and vectors over R (or $K_{\mathbb{R}}$ or R_q) resorts to module theory and not exactly linear algebra. We insist that certain intuitions from linear algebra do not carry or have equivalent in module theory.

Units

At many occasions in this thesis, we need certain elements to be invertible in R_q, which thus depends on the factorization of $\langle q \rangle$. For that we use the following lemma proven in, e.g., [LS18, Thm. 1.1] for cyclotomic rings.

Lemma 1.4 ([LS18, Thm. 1.1])

Let K be the ν-th cyclotomic field, with $\nu = \prod_i p_i^{e_i}$ be its factorization into primes with $e_i \geq 1$. We let R be the ring of integers of K. Also, let $\mu = \prod_i p_i^{f_i}$ for any $f_i \in [\![e_i]\!]$. Let q be a prime such that $q = 1 \bmod \mu$ and $ord_\nu(q) = \nu/\mu$, where ord_ν is the multiplicative order modulo ν. Then, for any element a of R satisfying $0 < \big\|\tau(a)\big\|_\infty < q^{1/\varphi(\mu)}/\mathfrak{s}_1(\mu)$, it holds that $a \bmod qR \in R_q^\times$. Here $\mathfrak{s}_1(\mu)$ denotes the spectral norm of the Vandermonde matrix of the μ-th cyclotomic field.

The number theoretic conditions on q essentially state that $\langle q \rangle$ splits into $\varphi(\mu)$ distinct prime ideal factors, each of algebraic norm $q^{\varphi(\nu)/\varphi(\mu)} = q^{\nu/\mu}$. In the case where ν is a power of an odd prime, then so is μ and then [LPR10] states that $\mathfrak{s}_1(\mu) = \sqrt{\mu}$. For more general cases, we refer to the discussions from Lyubashevsky and Seiler [LS18, Conj. 2.6]. We also refer the reader to [LS18, Thm. 2.5] which discusses the existence of such primes q for specific values of ν and μ.

Remark 1.2 (The Case of Power-of-Two Cyclotomics)

Lemma 1.4 is simplified in the power-of-two case [LS18, Cor. 1.2] where it is conditioned on the number $\kappa > 1$ of factors of $x^n + 1$ in $\mathbb{Z}_q[x]$. Choosing κ as a power of two less than $n = 2^\ell$ gives that the only conditions on q are that q has to be a prime congruent to $2\kappa + 1$ modulo 4κ. The invertibility condition then becomes $0 < \big\|\tau(a)\big\|_\infty < q^{1/\kappa}/\sqrt{\kappa}$ for any $a \in R_q$. The upper bound is decreasing with κ so the smaller κ, the more invertible elements. The smallest choice for κ is 2, which leads to choosing a prime $q = 5 \bmod 8$, meaning that $\langle q \rangle$ splits into two prime ideal factors of norm $q^{n/2}$. However, it is better to have *low splitting* for computational efficiency so as to rely on the *Number Theoretic Transform* (NTT) which is some kind of discrete Fast Fourier Transform. We thus fix the bound on the elements a that need to be invertible, and then choose the largest κ such that $q^{1/\kappa}/\sqrt{\kappa}$ exceeds this bound.

1.1.6 Module Theory over R_q: Singularity of Uniform Matrices

Although modules over rings share similarities with vector spaces over fields, certain properties have no equivalent in module theory. For example, as a ring may contain zero divisor, a non-zero vector over such ring may not form a linearly independent family. In this thesis, we consider the ring R_q and in particular matrices over R_q that need to be invertible, or have full column-rank, etc. We thus remind a few preliminary results on the singularity of uniform matrices over R_q. In this section, K denotes an arbitrary number field and R its ring of integers. The integer q is a prime that does not ramify in R and that splits as $\langle q \rangle = \prod_{i \in [\![\kappa]\!]} \mathfrak{p}_i$, where $\kappa \leq n = [K : \mathbb{Q}]$. We still use R_q to define $R/\langle q \rangle$ and we also define $\mathbb{F}_i = R/\mathfrak{p}_i$ for each $i \in [\![\kappa]\!]$. We recall that for each $i \in [\![\kappa]\!]$, $\mathbb{F}_i$ is a finite field of size $N(\mathfrak{p}_i)$, see e.g. [LPR13a, Sec. 2.5.3].

We give two useful results on the probability that a uniformly random matrix $\mathbf{A} \in R_q^{d \times d}$ is invertible in R_q. We note that such results were provided in [WW19]. However, the proofs were based on a flawed argument which was that a vector of R_q^d which is linearly independent (with itself) must contain a coefficient in $R_q^\times$. This is not the case as a vector of R_q^d consisting only of zero divisors can still be linearly independent. The details and proofs can be found in our original paper [BJRW23, App. A].

Lemma 1.5 (Linear Independence in Uniform Matrices [BJRW23, Lem. 2.5])

Let K be a number field, and R its ring of integers. Let d, q be positive integers such that q is an unramified prime which factors as $\langle q \rangle = \prod_{i \in [\![\kappa]\!]} \mathfrak{p}_i$. Let ℓ be in $[\![0, d[\![$, and $\mathbf{a}_1, \ldots, \mathbf{a}_\ell \in R_q^d$ be R_q-linearly independent vectors of R_q^d. Then

$$\mathbb{P}_{\mathbf{b} \hookleftarrow U(R_q^d)}[\mathbf{a}_1, \ldots, \mathbf{a}_\ell, \mathbf{b} \text{ are } R_q\text{-linearly independent}] = \prod_{i \in [\![\kappa]\!]} \left(1 - \frac{1}{N(\mathfrak{p}_i)^{d-\ell}}\right).$$

As a result, for any $1 \leq k \leq d$, it holds that

$$\mathbb{P}_{(\mathbf{a}_i)_i \sim U(R_q^d)^k}[(\mathbf{a}_i)_{i \in [\![k]\!]} \text{ are } R_q\text{-linearly independent}] = \prod_{\ell \in [\![0,k[\![} \prod_{i \in [\![\kappa]\!]} \left(1 - \frac{1}{N(\mathfrak{p}_i)^{d-\ell}}\right).$$

We note that the number of columns is $k \leq d$. Additionally, in R_q we still have the fact that a linearly independent family of vectors of R_q^d cannot contain more that d vectors. As a result, when $k > d$, we have to analyze the following probability

$$\mathbb{P}_{(\mathbf{a}_i)_{i \in [\![k]\!]} \sim U(R_q^d)^k}[\exists S \subseteq [\![k]\!], |S| = d \wedge (\mathbf{a}_i)_{i \in S} \text{ are } R_q\text{-l. i.}]$$

However, even if there exists subsets $S_i \subseteq [\![k]\!]$ with $|S_i| = d$ and $(\mathbf{a}_j \bmod \mathfrak{p}_i)_{j \in S_i}$ are $\mathbb{F}_i$-l. i., there is no guarantee that all the S_i are equal. The following lemma argues that the equality of the S_i is not necessary to guarantee that the columns form a spanning set of R_q^d.

Lemma 1.6 (Singularity of Uniform Matrices [BJRW23, Lem. 2.6])

Let K be a number field, and R its ring of integers. Let q be a prime integer that is unramified in R which splits as $\langle q \rangle = \prod_{i \in [\![\kappa]\!]} \mathfrak{p}_i$. Let $m \geq d$ be two integers. It holds

$$\mathbb{P}_{\mathbf{A} \sim U(R_q^{d \times m})}[\mathbf{A} \cdot R_q^m = R_q^d] \geq \prod_{\ell \in [\![0,d[\![} \prod_{i \in [\![\kappa]\!]} \left(1 - \frac{1}{N(\mathfrak{p}_i)^{m-\ell}}\right).$$

When R and q are clear from the context, for $m \geq d$, we define $\delta(m, d) = 1 - \mathbb{P}_{\mathbf{A} \sim U(R_q^{d \times m})}[\mathbf{A} \cdot R_q^m = R_q^d]$, which we use extensively throughout Chapter 3. If q is not clear, we write $\delta_q(m, d)$ instead. We note that $\delta(m, d)$ can be upper-bounded by $\frac{d \cdot \kappa}{(\min_{i \in [\![\kappa]\!]} N(\mathfrak{p}_i))^{m-d+1}}$. Hence, if q splits into only high-norm ideal factors so that $\min_{i \in [\![\kappa]\!]} N(\mathfrak{p}_i) \geq \lambda^{\omega(1/(m-d+1))}$, the probability $\delta(m, d)$ becomes negligible in λ.

We also define $\delta'(k, d)$ to be the probability that among $k \geq d$ independent uniform columns of R_q^d, there is no subset of d of those columns that are R_q-linearly independent. Formally, we define

$$\delta'(k, d) = 1 - \mathbb{P}_{(\mathbf{a}_i)_{i \in [\![k]\!]} \sim U(R_q^d)^k}[\exists S \subseteq [\![k]\!], |S| = d \wedge (\mathbf{a}_i)_{i \in S} \text{ are } R_q\text{-l. i.}]$$

We note that if R_q was a field, we would have $\delta(k,d) = \delta'(k,d)$. However, in the general case, $\delta(k,d) \neq \delta'(k,d)$ as a minimal spanning set of an R_q-submodule of R_q^d is not necessarily a basis of said submodule. Additionally, note that $\delta'(d,d)$ is given by Lemma 1.5 as

$$\delta'(d,d) = 1 - \prod_{\ell \in [\![0,d[\![} \prod_{i \in [\![\kappa]\!]} \left(1 - \frac{1}{N(\mathfrak{p}_i)^{d-\ell}}\right).$$

The probability $\delta'(k,d)$ is discussed in Section 3.4 as it only appears in the latter.

1.2 Lattices

We now the fundamental notions related to lattices that are going to be used in this thesis. We start by giving the standard definitions and then combine them with the notions of Section 1.1 to introduce *structured lattices*.

1.2.1 Standard Lattices

We recall that $\mathbb{R}^k$ is equipped with the Euclidean norm $\|\cdot\|_2$ and inner product $\langle \cdot, \cdot \rangle$. We define the closed ℓ_p hyperball of radius r and center $\mathbf{c} \in \mathbb{R}^k$ by $\mathcal{B}_p(\mathbf{c}, r) = \{\mathbf{x} \in \mathbb{R}^k : \|\mathbf{x} - \mathbf{c}\|_p \leq r\}$ for any non-negative integer p. For the sake of completeness, we also consider the hypercube of center $\mathbf{c} \in \mathbb{R}^k$ and half-side r by $\mathcal{B}_\infty(\mathbf{c}, r) = \{\mathbf{x} \in \mathbb{R}^k : \|\mathbf{x} - \mathbf{c}\|_\infty \leq r\}$. We use the superscript o on a hyperball to specify that it is open. As a result, we have

$$\forall p \in \mathbb{N} \cup \{\infty\}, \forall \mathbf{c} \in \mathbb{R}^k, \forall r \geq 0, \mathcal{B}_p^o(\mathbf{c}, r) = \{\mathbf{x} \in \mathbb{R}^k : \|\mathbf{x} - \mathbf{c}\|_p < r\}.$$

We recall the definition of a discrete set. It can be more generally defined in topological spaces but we only give the simpler definition with respect to normed spaces.

Definition 1.2 (Discrete Set)

Let $(V, \|\cdot\|)$ be a normed space. Let S be an arbitrary subset of V. We say that S is a *discrete set* if and only if: $\forall \mathbf{x} \in S$, $\exists r > 0$, $\mathcal{B}^o_{\|\cdot\|}(\mathbf{x}, r) \cap S = \{\mathbf{x}\}$.

We are now able to define a lattice of $\mathbb{R}^k$. There are other ways to define lattices that encompass our definitions, but they are not necessary for this thesis.

Definition 1.3 (Euclidean Lattice)

Let k be a positive integer, and $\mathcal{L} \subset \mathbb{R}^k$. The set $\mathcal{L}$ is called a (Euclidean) lattice of $\mathbb{R}^k$ if and only if $\mathcal{L}$ is a discrete subgroup of $(\mathbb{R}^k, +)$. A sublattice $\mathcal{L}' \subseteq \mathcal{L}$ is a discrete subgroup of the lattice $\mathcal{L}$.

As is, lattices seem difficult to handle from an algorithmic perspectives. Fortunately, every lattice $\mathcal{L}$ can be expressed as $\mathcal{L} = \mathbb{Z}\mathbf{b}_1 \oplus \ldots \oplus \mathbb{Z}\mathbf{b}_d$, where $(\mathbf{b}_i)_{i \in [\![d]\!]}$ is a family of linearly independent vectors of $\mathbb{R}^k$. It is called a *basis* of the lattice $\mathcal{L}$ and can be represented by the matrix $\mathbf{B} = [\mathbf{b}_1 | \ldots | \mathbf{b}_d] \in \mathbb{R}^{k \times d}$. We sometimes use the notation $\mathcal{L}(\mathbf{B})$ to specify that the lattice is spanned by the columns of $\mathbf{B}$ through integer linear combinations. The integer k is called the dimension of the lattice, refering to the dimension of the ambient space $\mathbb{R}^k$, while d is called the *rank* of the lattice. When $k = d$, we say that the lattice is full-rank. Just like for vector spaces, each lattice has an infinite number of bases but they all differ by a unitary transformation. More precisely, given $\mathbf{B}, \mathbf{B}'$, it holds that $\mathcal{L}(\mathbf{B}) = \mathcal{L}(\mathbf{B}')$ if and only if there exists $\mathbf{U} \in GL_d(\mathbb{Z})$ such that $\mathbf{B}' = \mathbf{B}\mathbf{U}$. This allows us to define the following basis-invariant quantity.

Definition 1.4 (Volume of a Lattice)

Let k, d be positive integers, and $\mathcal{L} \subset \mathbb{R}^k$ a lattice represented by a basis $\mathbf{B} \in \mathbb{R}^{k \times d}$. The *determinant* or *volume* of $\mathcal{L}$ is defined by $\mathrm{Vol}(\mathcal{L}) = \sqrt{\det \mathbf{B}^T \mathbf{B}}$. When $\mathcal{L}$ is full-rank, we have $\mathrm{Vol}(\mathcal{L}) = |\det \mathbf{B}|$.

We also define the dual of a lattice $\mathcal{L}$ as $\mathcal{L}^\star = \{\mathbf{y} \in \mathrm{Span}_{\mathbb{R}}(\mathcal{L}) : \forall \mathbf{x} \in \mathcal{L}, \langle \mathbf{x}, \mathbf{y} \rangle \in \mathcal{L}\}$. Finally, we define the first minimum of a lattice.

Definition 1.5 (First Minimum of a Lattice)

Let k be positive integers, and $\mathcal{L} \subset \mathbb{R}^k$ a lattice. Let p be in $\mathbb{N} \cap \{\infty\}$. The first minimum of $\mathcal{L}$ with respect to $\|\cdot\|_p$ is defined by

$$\lambda_1^p(\mathcal{L}) = \min\{r > 0 : \left|\mathcal{B}_p(\mathbf{0}, r) \cap \mathcal{L}\right| > 1\} = \min_{\mathbf{x} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{x}\|_p .$$

When $p = 2$, we omit the superscript.

1.2.2 Structured Lattices

So far we defined lattices as objects lying in the reals. In lattice-based cryptography, we extend this notion a little through objects that *embed into* the reals. More concretely, we now look at some classes of *algebraically structured lattices* originating from ideals and modules over rings of algebraic integers introduced in Section 1.1. We consider a number field K and its ring of integers R.

Ideal Lattices

Any ideal $\mathcal{I}$ of R embeds into a lattice of $\mathbb{R}^n$ through the coefficient or Minkowski embeddings denoted by τ and σ_H respectively. We can also consider the notion of lattices over the space H in which case $\sigma(\mathcal{I})$ is also a lattice. These are called *ideal lattices*. They represent a subclass of the set of all lattices which admit an additional structure resulting from the underlying ideal properties.

Ideal are additive subgroups which do not add to the subgroup structure of a standard lattice. However, ideals are also absorbant, meaning they are closed under multiplication by R, which this time represents a non-trivial property. If we call $\mathcal{L} = \iota(\mathcal{I})$ the ideal lattice with respect to the embedding $\iota \in \{\tau, \sigma, \sigma_H\}$, the multiplication matrix map introduced in Section 1.1.3 gives that $\mathcal{L}$ is stable by multiplication by any element of $M_\iota(R)$. This additional structure imposes a certain geometry of the lattice and is specific to each embedding. For example, because τ and σ_H are not isometric, the same ideal leads to potentially very different ideal lattices which are distorted one with respect to the other.

Module Lattices

Following the same logic, we can also consider *module lattices* by embedding modules. Since we mostly consider modules like $\mathcal{I}^d$, and because the embedding of $\mathcal{I}^d$ is the concatenation of the embeddings of $\mathcal{I}$, module lattices sort of interpolate ideal lattices and standard lattices. For example, if $d = 1$, the module lattice $\iota(\mathcal{I}^d)$ is actually an ideal lattice, and if $n = [K : \mathbb{Q}] = 1$, the module lattice $\iota(\mathcal{I}^d)$ is actually a standard lattice with no algebraic structure.

Again, module lattices inherit from an additional property compared to standard ones which stems from the fact that the module is completed with a scalar product. Multiplying $\iota(\mathcal{I}^d)$ by $M_\iota(R^{d\times d})$ preserves the lattice.

Specific Lattices: q-ary Lattices

In the remainder of this thesis, we are mostly interested in a specific kind of lattices called q-ary or q-periodic lattices. They satisfy the property that $q\mathbb{Z}^d \subseteq \mathcal{L} \subseteq \mathbb{Z}^d$ for an integer q. More specifically, we are interested in the following q-ary lattices.

Example 1.1 (q-ary Lattices)

Let m, d be positive integers, and $q \geq 2$ an integer. Let $\mathbf{A}$ be in $\mathbb{Z}^{d\times m}$. We define the following q-ary lattices of rank m.

$$\mathcal{L}_q(\mathbf{A}) = \mathbf{A}^T\mathbb{Z}^d + q\mathbb{Z}^m$$
$$\mathcal{L}_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod q\mathbb{Z}\}$$

We usually take $\mathbf{A}$ to be in $\mathbb{Z}_q^{d\times m}$ directly, and sometimes identifying it with one of its representatives. We also consider the lattice cosets for $\mathbf{u} \in \mathbb{Z}_q^d$ defined by

$$\mathcal{L}_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A}\mathbf{x} = \mathbf{u} \bmod q\mathbb{Z}\}$$

We can define similar lattices in the structured case by considering matrices over $R_q^{d\times m}$. More precisely, for $\mathbf{A} \in R_q^{d\times m}$, $\mathbf{u} \in R_q^d$, we use the same notation to denote $\mathcal{L}_q(\mathbf{A}) = \mathbf{A}^T R^d + qR^m$ and $\mathcal{L}_q^\perp(\mathbf{A}) = \{\mathbf{x} \in R^m : \mathbf{A}\mathbf{x} = \mathbf{0} \bmod qR\}$, as well as $\mathcal{L}_q^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in R^m : \mathbf{A}\mathbf{x} = \mathbf{u} \bmod qR\}$. Notice that these are subsets of R^m and are thus not lattices per se until embedded with τ, σ, σ_H.

1.2.3 Computational Problems over Lattices

The reason why lattices are so attractive from an algorithmic standpoint is because they offer a variety of mathematical problems that are computationally hard to solve. This makes it a very interesting research area in cryptology on three levels: (1) cryptanalysis which aims at finding better algorithms to solve such problems efficiently, (2) cryptography which aims at harnessing the hardness of such problems to construct secure primitives, and (3) mathematics and complexity theory which aims at theoretically establishing a hierarchy among these problems and prove their hardness mathematically. We now present a small selection of lattice problems which we deem important to have a good grasp and comprehension of the stakes in those three directions.

We first introduce the most common problem consisting in finding a vector of shortest (non-zero) norm in the lattice. Building cryptography on such a strict variant is however delicate, which is why we consider an approximate version of the problem. From now on, we only consider full-rank lattices, but it is possible to generalize to arbitrary lattices.

Definition 1.6 ((Approximate) Shortest Vector Problem)

Let d be a positive integer, p in $\mathbb{N} \cup \{\infty\}$, and $\gamma \geq 1$ a real. The *Approximate Shortest Vector Problem* $\mathrm{SVP}_\gamma^{(d,p)}$ asks to find $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\|_p \leq \gamma\lambda_1^p(\mathcal{L})$ given a (full-rank) lattice $\mathcal{L}$ of rank d. When $p = 2$ and d is clear from the context, we omit the superscript.

Another popular problem over lattices consists in finding the closest lattice point (up to an approximation factor) to a given target $\mathbf{t}$ in the ambient space. Note that SVP_γ is not exactly the specific instance $\mathbf{t} = \mathbf{0}$ of this new problem because when $\mathbf{t}$ is in the lattice, the problem becomes trivial (since $\mathbf{t}$ itself is solution). The quantity $\mathrm{dist}_p(\mathbf{t}, \mathcal{L})$ is defined as the minimal ℓ_p distance between $\mathbf{t}$ and a vector of $\mathcal{L}$.

Definition 1.7 ((Approximate) Closest Vector Problem)

Let d be a positive integer, p in $\mathbb{N} \cup \{\infty\}$, and $\gamma \geq 1$ a real. The *Approximate Closest Vector Problem* $\mathrm{CVP}_\gamma^{(d,p)}$ asks to find $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v} - \mathbf{t}\|_p \leq \gamma\mathrm{dist}_p(\mathbf{t}, \mathcal{L})$ given a (full-rank) lattice $\mathcal{L}$ of rank d and a target $\mathbf{t} \in \mathbb{R}^d$. When $p = 2$ and d is clear from the context, we omit the superscript.

These problems can be restricted to a specific class of lattices, e.g., ideal or module lattices, in which case we usually prepend the class to the name of the problem, e.g., Ideal-SVP_γ.

1.3 Probabilities

This section is dedicated to providing the necessary notions and results from probability theory that are going to be used extensively in the rest of this thesis. We first fix some notations before diving into the definitions and results related to Gaussian distributions, regularity, concentration, etc.

For a finite set S, we denote by $|S|$ its cardinality and we use $U(S)$ to denote the discrete uniform probability distribution of support S. If S is not finite but with bounded volume in a metric space, $U(S)$ can still be defined as the continuous uniform distribution over S, with probability density function $\mathrm{Vol}(S)^{-1}\mathbf{1}_S(\cdot)$. We also let ψ_η be the centered binomial distribution of parameter $\eta \in \mathbb{N} \setminus \{0\}$ defined by the distribution of $\sum_{i\in[\![\eta]\!]} a_i - b_i$ for $a_1, b_1, \ldots, a_\eta, b_\eta$ independently drawn from $U(\{0,1\})$. We then use $\mathcal{B}_\eta$ to denote the distribution over R whose coefficients follow ψ_η, that is $\mathcal{B}_\eta = \tau^{-1}(\psi_\eta^n)$ where n is the ring degree.

The action of sampling $a \in S$ from a distribution $\mathcal{P}$ is denoted by $a \hookleftarrow \mathcal{P}$, whereas the notation $a \sim \mathcal{P}$ means that the random variable a is distributed according to $\mathcal{P}$. When not clear from the context, $\mathrm{Supp}(\mathcal{P})$ denotes the support of the distribution $\mathcal{P}$. When using random variable, we use the notation $\mathbb{P}_{a\sim\mathcal{P}}[E]$ or $\mathcal{P}(E)$ to denote the probability of event E happening where the random

variable a follows the distribution $\mathcal{P}$. Finally, for a discrete distribution $\mathcal{P}$ (or a random variable $a \sim \mathcal{P}$), we define its min-entropy by $H_\infty(\mathcal{P}) = -\log_2(\max_{s\in\mathrm{Supp}(\mathcal{P})} \mathbb{P}_{a\sim\mathcal{P}}[a = s])$. Finally, we sometimes use the notation $\mathcal{P} \otimes \mathcal{Q}$ to denote the joint distribution of $\mathcal{P}$ and $\mathcal{Q}$.

1.3.1 Divergences

At many occasions in cryptography we need to measure the closeness of some probability distributions, for example to ensure that the actual distribution of the provided random elements does not leak secret information by being close to an idealized public distribution. There are several ways of measuring how close two probability distributions are. The most natural way is to compare the marginal distributions entrywise, i.e., evaluating the discrepancy between the probability of each event occuring for each distribution. This is usually possible but certain situations in cryptography call for the worst-case and we thus need a more global comparison. Such tools are generally called *divergences* or sometimes *distances* if they meet all the expected properties. In this thesis, we focus on three ways of comparing distributions: the statistical distance, Rényi divergences [Ré61, vEH14], and sometimes directly on the marginal distributions which is usually related to the infinite-order Rényi divergence.

Definition 1.8 (Comparing Marginal Distributions)

Let $\mathcal{P}$ and $\mathcal{Q}$ be two discrete probability distributions. For $0 \leq \delta_1 \leq 1 \leq \delta_2$, we write $\mathcal{P} \approx_{\delta_1,\delta_2} \mathcal{Q}$ if and only if for all $s \in \mathrm{Supp}(\mathcal{P}) \cup \mathrm{Supp}(\mathcal{Q}), \mathcal{P}(s) \in [\delta_1, \delta_2] \cdot \mathcal{Q}(s)$.

Statistical Distance

One of the divergences extensively used in lattice-based cryptography is the *statistical distance*, sometimes referred to as the *total variation distance*. It gives an *additive* way of comparing probability distributions. We recall its definition in Definition 1.9 and some properties that are going to be useful in subsequent proofs.

Definition 1.9 (Statistical Distance)

Let S be a countable set, and $\mathcal{P}, \mathcal{Q}$ two discrete probability distributions over S. The statistical distance between $\mathcal{P}$ and $\mathcal{Q}$ is defined by

$$\Delta(\mathcal{P}, \mathcal{Q}) = \frac{1}{2} \sum_{s\in S} \left|\mathcal{P}(s) - \mathcal{Q}(s)\right|.$$

We define the statistical distance between random variables in a similar way by the distributions they represent. We say that $\mathcal{P}$ and $\mathcal{Q}$ are ε-close or ε-indistinguishable if $\Delta(\mathcal{P}, \mathcal{Q}) \leq \varepsilon$.

The statistical distance follow all the expected properties of a distance. It is symmetric, positive-definite, and verifies the triangle inequality. Formally, we have that $\Delta(\mathcal{P}, \mathcal{Q}) \geq 0$, with equality if and only if $\mathcal{P}$ and $\mathcal{Q}$ are identical. It also holds that $\Delta(\mathcal{P}, \mathcal{Q}) = \Delta(\mathcal{Q}, \mathcal{P})$, and $\Delta(\mathcal{P}, \mathcal{R}) \leq \Delta(\mathcal{P}, \mathcal{Q}) + \Delta(\mathcal{Q}, \mathcal{R})$. What is most interesting and implicitly used at many occasions in cryptography are the probability preservation property and the data processing inequality which we recall in Lemma 1.7.

Lemma 1.7 (Probability Preservation and Data Processing - SD)

Let S be a countable set, and $\mathcal{P}, \mathcal{Q}$ two discrete probability distributions over S. Then, for any event $E \subseteq S$, the probability preservation property gives that

$$\mathcal{P}(E) \leq \Delta(\mathcal{P}, \mathcal{Q}) + \mathcal{Q}(E).$$

Then, for any possibly randomized function f, the data processing inequality is

$$\Delta(f(\mathcal{P}), f(\mathcal{Q})) \leq \Delta(\mathcal{P}, \mathcal{Q}),$$

where $f(\mathcal{P})$ denotes the distribution obtained by sampling $a \hookleftarrow \mathcal{P}$ and outputting $f(a)$.

In a cryptographic context, the distributions we consider depend on a multitude of parameters including the security parameter λ or another asymptotic parameter (e.g., the ring degree n). We

then say that two distributions are statistically close or statistically indistinguishable if they are ε-close for some function $\varepsilon(\lambda)$ that is negligible in λ, e.g., $\varepsilon(\lambda) = 2^{-\lambda}$.

Rényi Divergences

The Rényi divergences [Rén61, vEH14] represent another way of measuring the closeness of two distributions which has a rather *multiplicative* nature. It was thoroughly studied for its use in cryptography as a powerful alternative to the statistical distance by Bai et al. [BLL+15, BLR+18] and later by Prest [Pre17]. It is characterized by an order a which offers a trade-off between parameter selection and security loss. We recall it in Definition 1.10 and give the properties that we will need in the rest of this thesis.

Definition 1.10 (Rényi Divergences)

Let $\mathcal{P}, \mathcal{Q}$ be two discrete probability distributions such that $\mathrm{Supp}(\mathcal{P}) \subseteq \mathrm{Supp}(\mathcal{Q})$. Let α be a real in $(1, \infty]$. The Rényi divergence of order α from $\mathcal{P}$ to $\mathcal{Q}$ is defined by

$$\mathrm{RD}_\alpha(\mathcal{P}\|\mathcal{Q}) = \left(\sum_{s \in \mathrm{Supp}(\mathcal{P})} \frac{\mathcal{P}(s)^\alpha}{\mathcal{Q}(s)^{\alpha-1}} \right)^{\frac{1}{\alpha-1}}.$$

Note that RD_∞ simplifies to

$$\mathrm{RD}_\infty(\mathcal{P}\|\mathcal{Q}) = \max_{s \in \mathrm{Supp}(\mathcal{P})} \frac{\mathcal{P}(s)}{\mathcal{Q}(s)}.$$

We define the Rényi divergence between random variables in a similar way by the distributions they represent.

Even though this divergence are very different from the statistical distance, they retain similar properties. In particular it holds that $\mathrm{RD}_a(\mathcal{P}\|\mathcal{Q}) \geq 1$ with equality only if the distributions are identical. We give additional relevant properties in Lemma 1.8. For example, the Rényi divergence also enjoys probability preservation and data processing inequalities, multiplicativity, and a weak form of the triangle inequality. The latter will not be needed in this work.

Lemma 1.8 (Rényi Divergence Properties [LSS14, Lem. 4.1])

Let $\mathcal{P}, \mathcal{Q}$ be two discrete probability distributions such that $\mathrm{Supp}(\mathcal{P}) \subseteq \mathcal{Q}$. Let $\alpha \in (1, \infty]$. Then, for any event $E \subseteq \mathrm{Supp}(\mathcal{Q})$, the probability preservation property gives that

$$\mathcal{P}(E)^{\frac{\alpha}{\alpha-1}} \leq \mathrm{RD}_\alpha(\mathcal{P}\|\mathcal{Q}) \cdot \mathcal{Q}(E).$$

Then, for any possibly randomized function f, the data processing inequality is

$$\mathrm{RD}_\alpha(f(\mathcal{P})\|f(\mathcal{Q})) \leq \mathrm{RD}_\alpha(\mathcal{P}\|\mathcal{Q}),$$

where $f(\mathcal{P})$ denotes the distribution obtained by sampling $a \hookleftarrow \mathcal{P}$ and outputting $f(a)$. Then, let $(\mathcal{P}_i)_{i \in [\![n]\!]}, (\mathcal{Q}_i)_{i \in [\![n]\!]}$ be two families of independent discrete probability distributions such that for all i in $[\![n]\!]$, $\mathrm{Supp}(\mathcal{P}_i) \subseteq \mathrm{Supp}(\mathcal{Q}_i)$. It holds that

$$\mathrm{RD}_\alpha\left(\bigotimes_{i \in [\![n]\!]} \mathcal{P}_i \middle\| \bigotimes_{i \in [\![n]\!]} \mathcal{Q}_i \right) = \prod_{i \in [\![n]\!]} \mathrm{RD}_\alpha(\mathcal{P}_i\|\mathcal{Q}_i),$$

where $\bigotimes_{i \in [\![n]\!]} \mathcal{P}_i$ denotes the joint probability distribution of the family $(\mathcal{P}_i)_{i \in [\![n]\!]}$.

In Chapter 6, we leverage the relative error lemma from [Pre17, Lem. 3] which enables tighter security arguments. We thus formulate the following lemma which combines the probability preservation property of Lemma 1.8 and the relative error lemma.

Lemma 1.9 (Relative Error Lemma [Pre17, Lem. 3])

Let $\mathcal{P}, \mathcal{Q}$ be two discrete probability distributions having the same support S. Let $\delta > 0$ be such that $\mathcal{P} \approx_{1-\delta,1+\delta} \mathcal{Q}$. Then, for all $\alpha \in (1, \infty)$ and event $E \subseteq S$, it holds that

$$\mathcal{P}(E) \leq \left(1 + \frac{\alpha(\alpha-1)\delta^2}{2(1-\delta)^{\alpha+1}}\right)^{\frac{1}{\alpha}} \cdot \mathcal{Q}(E)^{\frac{\alpha-1}{\alpha}} \underset{\delta \to 0}{\sim} \left(1 + \frac{\alpha-1}{2}\delta^2\right) \cdot \mathcal{Q}(E)^{\frac{\alpha-1}{\alpha}}.$$

Smooth Rényi Divergence

The Rényi divergence is a powerful tool to compare distributions. It presents however certain limitations. For example, the infinite order Rényi divergence between shifted discrete Gaussians (which are defined in Section 1.3.2) is infinite and thus cannot be exploited to draw any conclusions. This lead Devevey et al. [DFPS22] to introduce a relaxed version called *smooth Rényi divergence* which allows one to discard (in a quantified and controlled way) a small portion of the distributions support which may cause problems in the exact Rényi divergence computation. It is parameterized by some $\varepsilon \geq 0$ that quantifies the probability mass of the discarded points.

Definition 1.11 (Smooth Rényi Divergence [DFPS22, Def. 2.1])

Let $\varepsilon \geq 0$ and $\mathcal{P}, \mathcal{Q}$ be two discrete probability distributions such that $\mathcal{P}(\mathrm{Supp}(\mathcal{Q})) \geq 1 - \varepsilon$. The ε-smooth Rényi divergence of infinite order from $\mathcal{P}$ to $\mathcal{Q}$ is defined by

$$\mathrm{RD}_\infty^\varepsilon(\mathcal{P}\|\mathcal{Q}) = \inf\left\{M > 0 : \mathbb{P}_{a\sim\mathcal{P}}[\mathcal{P}(a) \leq M \cdot \mathcal{Q}(a)] \geq 1 - \varepsilon\right\}.$$

It essentially allows one to use the infinite-order Rényi divergence and its properties, while discarding a fraction ε of the problematic points. In particular, we can now define the divergence even when $\mathrm{Supp}(\mathcal{P}) \not\subseteq \mathrm{Supp}(\mathcal{Q})$ as it allows for discarding the points in $\mathrm{Supp}(\mathcal{P}) \setminus \mathrm{Supp}(\mathcal{Q})$ that would lead to an undefined quantity in Definition 1.10. We insist that it is a different notion and thus does not have the exact same properties. In particular, it is not log-positive in the sense that $\mathrm{RD}_\infty^\varepsilon(\mathcal{P}\|\mathcal{Q})$ is not always above 1. It however still verifies the probability preservation property [DFPS22, Lem. A.5], with an extra additive loss of ε: $\mathcal{P}(E) \leq \mathrm{RD}_\infty^\varepsilon(\mathcal{P}\|\mathcal{Q}) \cdot \mathcal{Q}(E) + \varepsilon$. We also note that this notion does not relate directly to the usual concept of ε-smooth entropy of a distribution $\mathcal{P}$ which considers it to be the maximal entropy over all *other distributions* that are ε-close to $\mathcal{P}$.

1.3.2 Gaussian Measures

Gaussian probability distributions have very interesting probabilistic and geometric properties, and they appear everywhere in mathematics. They offer many possibilities in order to randomize certain processes, typically instances of problems. Unfortunately, Gaussian distributions are by nature continuous over $\mathbb{R}^d$ which seems incompatible with the discrete aspect of lattices. There are however ways to discretize Gaussians onto discrete sets such as lattices. The idea is essentially to condition a continuous one to be supported in the lattice, thus requiring a normalization by the Gaussian mass of the lattice itself.

Albeit perfectly valid in theory, for it to be relevant, one needs a way to efficiently sample from such distributions. Gaussian distributions are known to be concentrated around their center, which means that if one is able to sample very narrow discrete Gaussians centered around $\mathbf{0}$ (or around a target $\mathbf{t}$ in the case of CVP_γ), they would be able to solve SVP_γ, CVP_γ and possibly other variants. As it turns out, sampling a Gaussian on a lattice and the quality of the samples highly depends on the size of the basis of the lattice that is used. As a result, only a good basis with short and almost orthogonal vectors allows for efficient and qualitative sampling.

Continuous Gaussian Distributions

We first introduce continuous Gaussian Distributions which are also going to be relevant for Chapter 2. For clarity, we call $\mathcal{S}_d^+$ the set of symmetric positive semi-definite matrices of $\mathbb{R}^{d\times d}$, and $\mathcal{S}_d^{++} = \mathcal{S}_d^+ \cap GL_d(\mathbb{R})$ the set of symmetric positive-definite matrices of $\mathbb{R}^{d\times d}$.

Definition 1.12 (Gaussian Function and Continuous Gaussian)

Let d be a positive integer. For a matrix $\mathbf{S} \in \mathcal{S}_d^{++}$ and a vector $\mathbf{c} \in \mathbb{R}^d$, we define the *Gaussian function of center* $\mathbf{c}$ *and width* $\sqrt{\mathbf{S}}$ by

$$\forall \mathbf{x} \in \mathbb{R}^d, \rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x}) = \exp(-\pi(\mathbf{x}-\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{c})).$$

By renormalizing, we define the continuous Gaussian distribution $D_{\sqrt{\mathbf{S}},\mathbf{c}}$ by its probability density function

$$\forall \mathbf{x} \in \mathbb{R}^d, D_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x}) = \frac{1}{\sqrt{\det \mathbf{S}}}\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x}) = \frac{1}{\sqrt{\det \mathbf{S}}}\exp(-\pi(\mathbf{x}-\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{c})).$$

In Chapter 2, we also need to consider cases where the covariance matrix[1] is not invertible. This requires extending the definition to use positive *semi*-definite matrices, which results in probability distributions that are called *degenerate* or *singular*. In the degenerate case, the probability density function cannot be defined with respect to the standard Lebesgue measure as $\mathbf{S}$ is not necessarily invertible. Standard results on non-singular Gaussian distributions can however be extended to the singular case by using the characteristic function which always exists and is defined by

$$\forall \mathbf{t} \in \mathbb{R}^d, \varphi_{D_{\sqrt{\mathbf{S}},\mathbf{c}}}(\mathbf{t}) = \mathbb{E}_{\mathbf{x}\sim D_{\sqrt{\mathbf{S}},\mathbf{c}}}[\exp(i\mathbf{x}^T\mathbf{t})] = \exp(i\mathbf{c}^T\mathbf{t} - \pi\mathbf{t}^T\mathbf{S}\mathbf{t}).$$

We also note that one can still define a density for degenerate Gaussian distributions as $D_{\sqrt{\mathbf{S}},\mathbf{c}} : \mathbf{x} \in \mathbb{R}^d \mapsto (\det{}^+ \mathbf{S})^{-1/2}\exp(-\pi(\mathbf{x}-\mathbf{c})^T\mathbf{S}^+(\mathbf{x}-\mathbf{c}))$, where $\mathbf{S}^+$ is the Moore-Penrose pseudo-inverse of $\mathbf{S}$ and $\det^+$ is the pseudo-determinant. This density is however defined with respect to a skewed measure.

For clarity, when the center is $\mathbf{c} = \mathbf{0}$, we omit it from the subscripts. Additionally, if $\mathbf{S} = \mathrm{diag}(\mathbf{s}^2) = \mathrm{diag}(s_1^2, \ldots, s_d^2)$, we use $\mathbf{s}$ instead of $\sqrt{\mathbf{S}}$ in the notation. Finally, when $\mathbf{S} = s^2\mathbf{I}_d$, we simply use s and call this distribution *spherical*. Otherwise, it is called *elliptical*. We also define $\Psi_{\leq s} = \{D_{\mathbf{s}}; \mathbf{s} \in (\mathbb{R}^+)^d \cap \mathcal{B}_\infty(\mathbf{0}, s)\}$, that is the set of Gaussian distributions $D_{\mathbf{s}}$ for which $\|\mathbf{s}\|_\infty \leq s$.

Discrete Gaussian Distributions

We can now define discrete Gaussian distributions. Although we use them over lattices or lattice cosets, it is possible to define them over arbitrary countable sets.

Definition 1.13 (Discrete Gaussian)

Let d be a positive integer and $S \subset \mathbb{R}^d$ a countable set. For a matrix $\mathbf{S} \in \mathcal{S}_d^{++}$ and a vector $\mathbf{c} \in \mathbb{R}^d$, we define the *discrete Gaussian distribution over* S *of center* $\mathbf{c}$ *and width* $\sqrt{\mathbf{S}}$ by its probability density function

$$\forall \mathbf{x} \in \mathbb{R}^d, \mathcal{D}_{S,\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x})}{\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(S)} = \frac{\exp(-\pi(\mathbf{x}-\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{c}))}{\sum_{\mathbf{y}\in S}\exp(-\pi(\mathbf{y}-\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{y}-\mathbf{c}))}.$$

Although the density of discrete Gaussian distributions looks rather similar to that of a continuous one, they should not be treated as continuous ones without care. A key quantity related to discrete Gaussians over lattices is called the *smoothing parameter* of a lattice coined by Micciancio and Regev [MR07]. Given some $\varepsilon > 0$ and a lattice $\mathcal{L}$, we define the smoothing parameter of $\mathcal{L}$ with smoothing loss ε by $\eta_\varepsilon(\mathcal{L}) = \inf\{s > 0 : \rho_{1/s}(\mathcal{L}^\star) = 1 + \varepsilon\}$. For a matrix $\mathbf{S} \in \mathcal{S}_d^{++}$, we say that $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$ if $\rho_{\sqrt{\mathbf{S}^{-1}}}(\mathcal{L}^\star) \leq 1 + \varepsilon$. Note that if $\mathbf{S} - \eta_\varepsilon(\mathcal{L})^2\mathbf{I}_d \in \mathcal{S}_d^+$, then $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$. It essentially captures the standard deviation threshold above which a discrete Gaussian behaves almost like a continuous one. Regardless, $\mathcal{D}_{\mathcal{L},s}$ is always a sub-Gaussian distribution with sub-Gaussian moment $s/\sqrt{2\pi}$ [MP12, Lem. 2.8]. We recall that a (discrete or continuous) distribution $\mathcal{P}$ over $\mathbb{R}^d$ is sub-Gaussian with moment α, if for all unit vector $\mathbf{u} \in \mathbb{R}^d$ and $t \in \mathbb{R}$, it holds that $\mathbb{E}_{\mathbf{x}\sim\mathcal{P}}[\exp(t\mathbf{x}^T\mathbf{u})] \leq \exp(\alpha^2t^2/2)$.

It also holds that the discrete Gaussian carries almost the same entropy as the continuous one, to which it is withdrawn the log-volume of the lattice. We formalize it in the following lemma. A

[1] We call $\mathbf{S}$ the covariance of $D_{\sqrt{\mathbf{S}},\mathbf{c}}$ by abuse of language. The actual covariance is $(2\pi)^{-1}\mathbf{S}$.

similar result on the min-entropy of discrete Gaussian is given in [PR06, Lem. 2.10] but we give a tighter bound directly resulting from Poisson's summation formula.

Lemma 1.10 (Min-entropy of Discrete Gaussians)

Let d be a positive integer, and $\mathcal{L} \subset \mathbb{R}^d$ a lattice of rank d. Let $\varepsilon > 0$ and $\mathbf{c} \in \mathbb{R}^d$. We also take $\mathbf{S} \in \mathcal{S}_d^{++}$ be such that $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$. Then, it holds that

$$H_\infty(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}) \geq \log_2 \sqrt{\det \mathbf{S}} - \log_2 \mathrm{Vol}(\mathcal{L}) + \log_2(1-\varepsilon).$$

When, $\mathbf{c} = \mathbf{0}$, the condition $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$ is not needed and the min-entropy is lower-bounded by $\log_2 \sqrt{\det \mathbf{S}} - \log_2 \mathrm{Vol}(\mathcal{L})$.

Proof (Lemma 1.10). Let $\mathcal{L} \subset \mathbb{R}^d$ be a lattice of rank d, $\varepsilon > 0$, $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$ and $\mathbf{c} \in \mathbb{R}^d$. We look at $\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathcal{L})$. By the Poisson summation formula, it holds that

$$\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathcal{L}) = \sqrt{\det \mathbf{S}} \cdot \mathrm{Vol}(\mathcal{L})^{-1} \sum_{\mathbf{x} \in \mathcal{L}^\star} e^{-i \cdot 2\pi \mathbf{x}^T \mathbf{c}} \rho_{\sqrt{\mathbf{S}^{-1}}}(\mathbf{x}).$$

Yet, it holds that $\left| \sum_{\mathbf{x} \in \mathcal{L}^\star} e^{-i \cdot 2\pi \mathbf{x}^T \mathbf{c}} \rho_{\sqrt{\mathbf{S}^{-1}}}(\mathbf{x}) - 1 \right| \leq \rho_{\sqrt{\mathbf{S}^{-1}}}(\mathcal{L}^\star \setminus \{\mathbf{0}\}) \leq \varepsilon$, as $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$. Since the sum is a positive real, it yields that the latter is bounded below by $1 - \varepsilon$. Thence,

$$\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathcal{L}) \geq \sqrt{\det \mathbf{S}} \cdot \mathrm{Vol}(\mathcal{L})^{-1} (1-\varepsilon).$$

Since $\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{x}) \leq 1$ for all $\mathbf{x} \in \mathcal{L}$, we have that $H_\infty(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}) \geq \log_2 \rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathcal{L})$, which gives the desired inequality.

When $\mathbf{c} = \mathbf{0}$, the previous calculations yield $\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}) = \sqrt{\det \mathbf{S}} \cdot \mathrm{Vol}(\mathcal{L})^{-1} \rho_{\sqrt{\mathbf{S}^{-1}}}(\mathcal{L}^\star) \geq \sqrt{\det \mathbf{S}} \cdot \mathrm{Vol}(\mathcal{L})^{-1}$. Additionally, we get that $H_\infty(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}}) = \log_2 \rho_{\sqrt{\mathbf{S}}}(\mathcal{L})$ which concludes the proof.

Finally, we also provide the necessary results on the (smooth) Rényi divergence between shifted discrete Gaussians which generalize that of [LSS14, Lem. 4.2] and [DFPS22, Lem. C.2].

Lemma 1.11 (Adapted from [LSS14, Lem. 4.2][DFPS22, Lem. C.2])

Let d be a positive integer, and $\mathcal{L} \subset \mathbb{R}^d$ a lattice of rank d. Let $\varepsilon > 0, \varepsilon' \in (0,1)$ and $\mathbf{c} \in \mathbb{R}^d$. We also take $\mathbf{S} \in \mathcal{S}_d^{++}$ and $\alpha \in (1, \infty)$. Then, it holds that

$$\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}} \| \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}) \leq \exp\left(\alpha \pi \mathbf{c}^T \mathbf{S}^{-1} \mathbf{c}\right),$$

$$\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}} \| \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}}) \leq \exp\left(\alpha \pi \mathbf{c}^T \mathbf{S}^{-1} \mathbf{c}\right) \cdot \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\frac{\alpha}{\alpha-1}}, \text{ if } \sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L})$$

$$\mathrm{RD}_\alpha^{\varepsilon'}(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}} \| \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}) \leq \exp\left(\pi \mathbf{c}^T \mathbf{S}^{-1} \mathbf{c} + 2\sqrt{\pi \mathbf{c}^T \mathbf{S}^{-1} \mathbf{c} \cdot \ln \varepsilon'^{-1}}\right)$$

When $\mathbf{c} \in \mathcal{L}$ and $\alpha \in \mathbb{N} \setminus \{0,1\}$, the first two inequalites become equalities, and the smoothing requirement is no longer needed. Also, when $\mathbf{S} = s^2 \mathbf{I}_d$, the last upper-bound is itself bounded by $M > 1$ if $s \geq \|\mathbf{c}\|_2 \cdot \frac{\sqrt{\pi}}{\ln M}(\sqrt{\ln \varepsilon'^{-1} + \ln M} + \sqrt{\ln \varepsilon'^{-1}})$.

Proof (Lemma 1.11). We have the following.

$$\begin{aligned}
&\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}}\|\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}})^{\alpha-1} \\
&= \sum_{\mathbf{x}\in\mathcal{L}} \frac{\exp(-\pi\mathbf{x}\mathbf{S}^{-1}\mathbf{x})^\alpha/\rho_{\sqrt{\mathbf{S}}}(\mathcal{L})^\alpha}{\exp(-\pi(\mathbf{x}-\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{c}))^{\alpha-1}/\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}-\mathbf{c})^{\alpha-1}} \\
&= \frac{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}-\mathbf{c})^{\alpha-1}}{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L})^\alpha} \sum_{\mathbf{x}\in\mathcal{L}} \exp\left(-\pi(\alpha\mathbf{x}^T\mathbf{S}^{-1}\mathbf{x} - (\alpha-1)(\mathbf{x}-\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{x}-\mathbf{c}))\right) \\
&= \frac{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}-\mathbf{c})^{\alpha-1}}{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L})^\alpha} \sum_{\mathbf{x}\in\mathcal{L}} \exp\left(-\pi((\mathbf{x}+(\alpha-1)\mathbf{c})^T\mathbf{S}^{-1}(\mathbf{x}+(\alpha-1)\mathbf{c}) - \alpha(\alpha-1)\mathbf{c}^T\mathbf{S}^{-1}\mathbf{c})\right) \\
&= \frac{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}-\mathbf{c})^{\alpha-1}\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}+(\alpha-1)\mathbf{c})}{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L})^\alpha} \cdot \exp\left(\pi\alpha(\alpha-1)\mathbf{c}^T\mathbf{S}^{-1}\mathbf{c}\right).
\end{aligned}$$

Yet, it holds by Poisson's summation formula that $\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}+\mathbf{c}') \le \rho_{\sqrt{\mathbf{S}}}(\mathcal{L})$ for all $\mathbf{c}' \in \mathbb{R}^n$. Therefore, we get

$$\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}}\|\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}})^{\alpha-1} \le \exp\left(\pi\alpha(\alpha-1)\mathbf{c}^T\mathbf{S}^{-1}\mathbf{c}\right),$$

leading to

$$\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}}\|\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}) \le \exp\left(\pi\alpha\mathbf{c}^T\mathbf{S}^{-1}\mathbf{c}\right),$$

as desired. Note that when $\mathbf{c} \in \mathcal{L}$ and α is an integer, the ratio of Gaussian masses above is directly equal to 1, which proves the equality case. We now look at the other direction. Using the same methodology, it holds that

$$\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}\|\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}})^{\alpha-1} = \frac{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L})^{\alpha-1}\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}-\alpha\mathbf{c})}{\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}-\mathbf{c})^\alpha} \cdot \exp\left(\pi\alpha(\alpha-1)\mathbf{c}^T\mathbf{S}^{-1}\mathbf{c}\right).$$

Yet, if $\sqrt{\mathbf{S}} \ge \eta_\varepsilon(\mathcal{L})$, we obtain that $\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}) \le \mathrm{Vol}(\mathcal{L})^{-1}\sqrt{\det\mathbf{S}}(1+\varepsilon)$ and that $\rho_{\sqrt{\mathbf{S}}}(\mathcal{L}-\mathbf{v}) \in \mathrm{Vol}(\mathcal{L})^{-1}\sqrt{\det\mathbf{S}}\cdot[1-\varepsilon, 1+\varepsilon]$, for any center $\mathbf{v}$. Hence,

$$\mathrm{RD}_\alpha(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}\|\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}}}) \le \exp\left(\pi\alpha(\alpha-1)\mathbf{c}^T\mathbf{S}^{-1}\mathbf{c}\right) \cdot \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{\alpha/(\alpha-1)},$$

The last inequality follows from the first one combined with [DFPS22, Lem. A.7] which states that $\mathrm{RD}^\varepsilon_\infty(\mathcal{P}\|\mathcal{Q}) \le \mathrm{RD}_\alpha(\mathcal{P}\|\mathcal{Q})/\varepsilon'^{1/(\alpha-1)}$ for any $\alpha > 1$. Optimizing over the latter gives the bound. Finally the bound in the spherical case is directly taken from [DFPS22, Lem. C.2].

Structured Gaussians

Our work relies almost exclusively on number fields and rings of integers as introduced in Section 1.1. In that regard, it is natural to consider Gaussian distributions directly over the fields and rings. They are naturally defined through a chosen embedding to the reals, i.e., either τ or σ_H. In Chapter 2, we consider Gaussians with respect to the Minkowski embedding σ_H. More precisely, we use continuous Gaussian distributions over $K_\mathbb{R}$ defined by $D_{\sqrt{\mathbf{S}},\mathbf{c}} = \sigma_H^{-1}(D_{\sqrt{\mathbf{S}},\sigma_H(\mathbf{c})})$ where $\mathbf{S} \in \mathcal{S}^{++}_{nd}$ and $\mathbf{c} \in K^d_\mathbb{R}$. Note that because σ is an isomorphism between $K_\mathbb{R}$ and H (and not K and H), the resulting samples are not necessarily in K but in $K_\mathbb{R}$. Gaussians over $K_\mathbb{R}$ have been introduced alongside the R-LWE problem in [LPR10] with respect to the canonical embedding as they provide tighter reductions this way. Since then, their use in various reductions has spread when dealing with structured variants of LWE.

However, when designing lattice-based cryptosystems, the coefficient embedding is much more convenient as it usually avoids storing elements with floating points. As a result, with the exception of Chapter 2, we consider Gaussians with respect to the coefficient embedding τ in this thesis. That is that we define $\mathcal{D}_{M,\sqrt{\mathbf{S}},\mathbf{c}} = \tau^{-1}(\mathcal{D}_{\tau(M),\sqrt{\mathbf{S}},\tau(\mathbf{c})})$ for some R-module $M \subset K^d$, $\mathbf{S} \in \mathcal{S}^{++}_{nd}$ and $\mathbf{c} \in K^d_\mathbb{R}$.

Convolution of Discrete Gaussians

Our work necessitates a few results on the convolution of discrete Gaussian distributions, or with continuous ones. As opposed to that of solely continuous Gaussians, involving their discrete counterparts require certain smoothing conditions. First, the sum of indepentent discrete Gaussians follows the standard intuition when above the smoothing parameter of the lattice. Lemma 1.12 is an adaption from [Reg05, Claim 3.9] and [MP13, Thm. 3.3]. Also we note that the results of this section have been subsumed by a more general theorem due to Genise et al. [GMPW20].

Lemma 1.12 (Summing Independent Discrete Gaussians)

Let d be a positive integer, and $\mathcal{L} \subset \mathbb{R}^d$ a lattice of rank d. Let $\mathbf{S}, \mathbf{R}$ be in $\mathcal{S}_d^{++}$ and define $\mathbf{T} = \mathbf{R} + \mathbf{S}$ and $\mathbf{U} = (\mathbf{R}^{-1} + \mathbf{S}^{-1})^{-1}$. For some $\varepsilon \in (0,1)$, we assume $\sqrt{\mathbf{U}} \geq \eta_\varepsilon(\mathcal{L})$. Also, let $\mathbf{c}_1, \mathbf{c}_2$ be in $\mathbb{R}^d$ and $\mathbf{c} = \mathbf{c}_1 + \mathbf{c}_2$. It then holds that

$$\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{R}},\mathbf{c}_1} + \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}_2} \approx_{\delta_1,\delta_2} \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{T}},\mathbf{c}},$$

where $\delta_1 = (1-\varepsilon)^2/(1+\varepsilon)^2$ and $\delta_2 = \delta_1^{-1}$. In particular, it yields

$$\Delta\left(\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{R}},\mathbf{c}_1} + \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}_2}, \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{T}},\mathbf{c}}\right) \leq \frac{2\varepsilon}{(1-\varepsilon)^2} \underset{\varepsilon\to 0}{\sim} 2\varepsilon,$$

When $\mathbf{c}_1 = \mathbf{c}_2 = 0$, the statistical distance can be bounded by $\varepsilon(3+\varepsilon)/2(1+\varepsilon)^2 \leq 3\varepsilon/2$.

The previous lemma dealt with independent discrete Gaussians. In some occasions, the convoluted distributions may not be independent of one another. A result by Peikert [Pei10, Thm. 3.1] looks at this specific case in depth and also tackles the convolution with continuous Gaussians. We only mention the specific case that is going to be used in this thesis.

Lemma 1.13 (Convolved Gaussians [Pei10, Thm. 3.1])

Let d be a positive integer, and $\mathcal{L} \subset \mathbb{R}^d$ a lattice of rank d. Let $\varepsilon \in (0,1)$ and $r, s > 0$ be such that $s \geq \eta_\varepsilon(\mathcal{L})$. By calling $\mathcal{P}$ the distribution of $\mathbf{x} + \mathbf{y}$ obtained by first sampling $\mathbf{x}$ from D_r and then $\mathbf{y}$ sampled from $\mathcal{D}_{\mathcal{L}-\mathbf{x},s}$, we have

$$\mathcal{P} \approx_{\delta_1,\delta_2} \mathcal{D}_{\mathcal{L},\sqrt{r^2+s^2}},$$

where $\delta_1 = (1-\varepsilon)^2/(1+\varepsilon)^2$ and $\delta_2 = \delta_1^{-1}$.

Finally, in Chapter 2, we also need another lemma related to the inner product of $K_{\mathbb{R}}^d$ (which results in an element of $K_{\mathbb{R}}$) between a discrete Gaussian vector and an arbitrary one. In particular, we use Lemma 1.14 in the proof of Lemma 2.5 in order to decompose a Gaussian noise into an inner product. It generalizes [Reg09, Cor. 3.10] to the module case. A specific instance is proven in the proof of [LS15, Lem. 4.15], which is later mentioned (without proof) in [RSW18, Lem. 5.5]. Note here that the Gaussian distribution is with respect to the Minkowski embedding σ_H.

Lemma 1.14 (Module Gaussians Inner Product)

Let K be a number field and R its ring of integers. Let d be a positive integer and $M \subseteq K^d$ be an R-module (yielding a module lattice). Let $\mathbf{u}, \mathbf{z} \in K^d$ be fixed, and let $r, s > 0$ be such that $(1/r^2 + \|\mathbf{z}\|_{2,\infty}^2/s^2)^{-1/2} \geq \eta_\varepsilon(\sigma_H(M))$ for some $\varepsilon \in (0, 1/2)$. Then, the distribution of $\mathbf{z}^T\mathbf{v} + e$ where $\mathbf{v} \sim \mathcal{D}_{M+\mathbf{u},r}$ and $e \in K_{\mathbb{R}}$ is sampled from D_s, is withing statistical distance at most 2ε from the elliptical Gaussian $D_{\mathbf{r}}$ over $K_{\mathbb{R}}$, where $r_j = \sqrt{r^2 \sum_{i \in [\![d]\!]} \left|\sigma_j(z_i)\right|^2 + s^2}$ for $j \in [\![n]\!]$.

Proof (Lemma 1.14). Consider $\mathbf{h} \in K_{\mathbb{R}}^d$ distributed according to $D_{\mathbf{r}',\ldots,\mathbf{r}'}$, where $\mathbf{r}'$ is given by $r_j' = s/\sqrt{\sum_{i \in [\![d]\!]} \left|\sigma_j(z_i)\right|^2}$ for $j \in [\![n]\!]$. We now argue by Lemma 1.15 that $\langle \mathbf{z}, \mathbf{h} \rangle$ is distributed according D_s.

Lemma 1.15 ([LS15, Lem. 2.13])

Let $\mathbf{r} \in (\mathbb{R}^+)^n \cap H$, $\mathbf{z} \in K^d$ fixed and $\mathbf{e} \in K_{\mathbb{R}}^d$ sampled from $D_{\sqrt{\mathbf{S}}}$, where $\sqrt{\mathbf{S}} = [\delta_{i,j}\mathrm{diag}(\mathbf{r})]_{i,j\in[\![d]\!]} \in \mathbb{R}^{nd\times nd}$. Then $\langle \mathbf{z}, \mathbf{e}\rangle = \sum_{i\in[\![d]\!]} z_i e_i$ is distributed according to $D_{\mathbf{r}'}$ with $r'_j = r_j\sqrt{\sum_{i\in[\![d]\!]}|\sigma_j(z_i)|^2}$.

It then holds that $\Delta(\langle \mathbf{z}, \mathbf{v}\rangle + e, D_{\mathbf{r}}) = \Delta(\langle \mathbf{z}, \mathbf{v} + \mathbf{h}\rangle, D_{\mathbf{r}})$. Now, we denote $\mathbf{t}$ such that $t_j = \sqrt{r^2 + (r'_j)^2}$ for $j \in [\![n]\!]$. Note that by assumption

$$\begin{aligned}\min_{j\in[\![n]\!]} r \cdot r'_j/t_j &= (1/r^2 + \max_{j\in[\![n]\!]} \sum_{i\in[\![d]\!]} |\sigma_j(z_i)|^2 / s^2)^{-1/2} \\ &= (1/r^2 + \|\mathbf{z}\|_{2,\infty}^2 / s^2)^{-1/2} \geq \eta_\varepsilon(\sigma_H(M)).\end{aligned}$$

Following Lemma 1.12, or rather a variant where we sum a discrete and a continuous Gaussian, we get that $\mathbf{v} + \mathbf{h}$ is distributed as $D_{\mathbf{t},\dots,\mathbf{t}}$, within statistical distance at most 2ε. By applying once more Lemma 1.15 and the data processing inequality for the statistical distance recalled in Lemma 1.7, then we get that $\langle \mathbf{z}, \mathbf{v} + \mathbf{h}\rangle$ is distributed as $D_{\mathbf{r}}$ within statistical distance at most 2ε, where $r_j = t_j\sqrt{\sum_{i\in[\![d]\!]}|\sigma_j(z_i)|^2} = \sqrt{r^2\sum_{i\in[\![d]\!]}|\sigma_j(z_i)|^2 + s^2}$ for $j \in [\![n]\!]$.

1.3.3 Regularity

In this thesis, we need a few regularity results that have become common in lattice-based cryptography. We start by the celebrated lefotver hash lemma that has been formulated and generalized in many different flavors, e.g., [HILL99, Mic07, DORS08, LW20]. In our case, we need an adaption of the one by Micciancio [Mic07], which, instead of working with vectors over the finite field $\mathbb{Z}_q$, operates over the principal ideal domain $\mathbb{Z}_q[x]$ for q prime. Given a monogenic number field, as defined in Section 1.1.1, and a prime q, then the ideals of $R_q = R/qR$ can be characterized via the ideals of $\mathbb{Z}_q[x]$, which is needed in the proof. Further, we provide not only a bound on the statistical distance but also on the Rényi divergence of order 2.

Lemma 1.16 (Leftover Hash Lemma)

Let n, k, d, q, η be positive integers with q prime. Let K be a monogenic field of degree n and R be its ring of integers. Then it holds that

$$\mathrm{RD}_2((\mathbf{A}, \mathbf{Az})\|(\mathbf{A}, \mathbf{u})) \leq \left(1 + \frac{q^k}{(2\eta+1)^d}\right)^n,$$

$$\Delta((\mathbf{A}, \mathbf{Az}), (\mathbf{A}, \mathbf{u})) \leq \frac{1}{2}\sqrt{\left(1 + \frac{q^k}{(2\eta+1)^d}\right)^n - 1},$$

where $\mathbf{A} \sim U(R_q^{k\times d})$, $\mathbf{z} \sim U(S_\eta^d)$, and $\mathbf{u} \sim U(R_q^k)$.

Proof (Lemma 1.16). Let $\mathcal{P}$ be the distribution that samples $\mathbf{A} \hookleftarrow U(R_q^{k\times d})$ and $\mathbf{z} \hookleftarrow U(S_\eta^d)$ and outputs $(\mathbf{A}, \mathbf{Az}) \in R_q^{k\times d} \times R_q^k$. Let $\mathcal{Q} = U(\mathrm{Supp}(\mathcal{P}))$, i.e., it samples $\mathbf{A} \hookleftarrow U(R_q^{k\times d})$ and $\mathbf{u} \hookleftarrow U(R_q^k)$, and outputs $(\mathbf{A}, \mathbf{u}) \in R_q^{k\times d} \times R_q^k$. Note that $|\mathrm{Supp}\mathcal{P}| = q^{nk(d+1)}$.

We start by bounding the collision probability of $\mathcal{P}$. We conclude by linking the collision probability with the Rényi divergence of order 2 and get the statistical distance using Pinsker's inequality [FHT03].

For $\mathbf{A}, \mathbf{A}' \sim U(R_q^{k\times d})$ and $\mathbf{z}, \mathbf{z}' \sim U(S_\eta^d)$ it yields

$$\begin{aligned}\mathbb{P}[\mathbf{A} = \mathbf{A}' \wedge \mathbf{Az} = \mathbf{A}'\mathbf{z}'] &= \mathbb{P}[\mathbf{A} = \mathbf{A}'] \cdot \mathbb{P}[\mathbf{Az} = \mathbf{A}'\mathbf{z}'|\mathbf{A} = \mathbf{A}'] \\ &= \frac{1}{|R_q|^{k\cdot d}} \cdot \mathbb{P}[\mathbf{A}(\mathbf{z} - \mathbf{z}') = \mathbf{0}].\end{aligned}$$

We now need the following lemma to further transform this equation.

Lemma 1.17 (Adapted from [Mic07, Lem. 4.4])

Let A be a finite ring and k, d be positive integers. Further, take an arbitrary vector $\mathbf{z} = (z_j)_{j \in [\![d]\!]} \in A^d$. If $\mathbf{A} \sim U(A^{k \times d})$, then $\mathbf{Az}$ is uniformly distributed over the module $\langle z_1, \dots, z_d \rangle^k$. In particular, the probability that $\mathbf{Az} = \mathbf{0}$ is exactly $\frac{1}{\left|\langle z_1, \dots, z_d \rangle\right|^k}$.

Proof (Lemma 1.17). Let $\mathbf{z} \in A^d$. For $\mathbf{b} \in A^k$ we define $T_{\mathbf{b}} = \left\{\mathbf{A} \in A^{k \times d} \colon \mathbf{Az} = \mathbf{b}\right\}$. Notice that the probability that $\mathbf{Az} = \mathbf{b}$ over the uniform random choice of $\mathbf{A}$ is exactly $\frac{|T_{\mathbf{b}}|}{|A|^{k \cdot d}}$. If $\mathbf{b} \notin \langle z_1, \dots, z_d \rangle^k$, then $T_{\mathbf{b}} = \emptyset$ and hence $\mathbb{P}_{\mathbf{A} \sim U(A^{k \times d})}[\mathbf{Az} = \mathbf{b}] = 0$. We now show that all $\mathbf{b} \in \langle z_1, \dots, z_d \rangle^k$ have the same probability. Let $\mathbf{b}$ be an arbitrary element of $\langle z_1, \dots, z_d \rangle^k$, i.e., it can be represented as $\mathbf{Az} = \mathbf{b}$ for some fixed $\mathbf{A} \in A^{k \times d}$. It follows that $\mathbf{A}' \in T_{\mathbf{b}}$ if and only if $\mathbf{A}' - \mathbf{A} \in T_{\mathbf{0}}$. Further, the mapping $\mathbf{A}' \mapsto \mathbf{A}' - \mathbf{A}$ is a bijection between $T_{\mathbf{b}}$ and $T_{\mathbf{0}}$, which implies that $|T_{\mathbf{b}}| = |T_{\mathbf{0}}|$. This shows that all $\mathbf{b} \in \langle z_1, \dots, z_d \rangle^k$ have the same probability, completing the proof.

By Lemma 1.17 over the random choice of $\mathbf{A}$ and the size of the finite ring R_q, the previous equation can be transformed into

$$\begin{aligned}
\frac{1}{q^{n \cdot k \cdot d}} \cdot \mathbb{P}[\mathbf{A}(\mathbf{z} - \mathbf{z}') = \mathbf{0}] &= \frac{1}{q^{nkd}} \cdot \sum_{\mathcal{I} \in \mathscr{I}} \frac{\mathbb{P}\left[\langle z_1 - z_1', \dots, z_d - z_d' \rangle^k = \mathcal{I}^k\right]}{|\mathcal{I}|^k} \\
&\leq \frac{1}{q^{nkd}} \cdot \sum_{\mathcal{I} \in \mathscr{I}} \frac{\mathbb{P}\left[\langle z_1 - z_1', \dots, z_d - z_d' \rangle^k \subseteq \mathcal{I}^k\right]}{|\mathcal{I}|^k} \\
&= \frac{1}{q^{(nk) \cdot (d+1)}} \cdot \sum_{\mathcal{I} \in \mathscr{I}} \frac{q^{nk}}{|\mathcal{I}|^k} \cdot \prod_{j \in [\![d]\!]} \mathbb{P}\left[(z_j - z_j') \in \mathcal{I}\right],
\end{aligned}$$

where $\mathscr{I}$ denotes the set of all ideals in R_q and we conditioned on the ideal $\langle z_1 - z_1', \dots, z_k - z_k' \rangle$.

We now specify $\mathscr{I}$. For $K = \mathbb{Q}(\zeta)$, let f be the minimal polynomial of ζ and let $f = \prod_{i \in [\![\kappa]\!]} f_i$ be its factorization in irreducible polynomials in $\mathbb{Z}_q[x]$. As $\mathbb{Z}_q$ is a field, $\mathbb{Z}_q[x]$ is a principal ideal domain. The ideal correspondence theorem in commutative algebra states that every ideal in R_q corresponds to an ideal in $\mathbb{Z}_q[x]$ containing $\langle f \rangle$. As each ideal in $\mathbb{Z}_q[x]$ itself is principal, thus of the form $\langle g \rangle$ for a polynomial $g \in \mathbb{Z}[x]$, this is equivalent to g dividing f. Hence, we know that the ideals of R_q are given by $\mathscr{I} = \left\{\langle f_G \rangle \colon G \subseteq \{1, \dots, \kappa\}\right\}$, where we define $f_G = \prod_{i \in G} f_i$. By convention, we say that the empty set $\emptyset$ defines the constant polynomial $f_\emptyset = 1$. For any f_G, it holds that

$$\mathbb{P}\left[(z_j - z_j') \in \langle f_G \rangle\right] = \mathbb{P}\left[z_j = z_j' \bmod f_G\right] \leq \max_{\widetilde{z}} \mathbb{P}\left[z_j \bmod f_G = \widetilde{z}\right] \leq \frac{1}{(2\eta + 1)^{\deg(f_G)}},$$

where the maximum is taken over all $\widetilde{z} \in R$ with $\deg(\widetilde{z}) < \deg(f_G)$. As explained in [Mic07], the last inequality follows from the fact that for any fixed value of the $n - \deg(f_G)$ highest degree coefficients of z, the map $z \mapsto z \bmod f_G$ is a bijection between sets of size $(2\eta+1)^{\deg(f_G)}$. We then get

$$\frac{q^{nk}}{\left|\langle f_G \rangle\right|^k} \prod_{j \in [\![d]\!]} \mathbb{P}\left[(z_j - z_j') \in \langle f_G \rangle\right] \leq \frac{q^{nk}}{\left(q^{n - \deg(f_G)}\right)^k} \left(\frac{1}{(2\eta + 1)^{\deg(f_G)}}\right)^d = \left(\frac{q^k}{(2\eta + 1)^d}\right)^{\deg(f_G)}.$$

Adding up over all ideals we can deduce

$$\begin{aligned}\sum_{\langle f_G\rangle\in\mathscr{I}} \frac{q^{nk}}{\left|\langle f_G\rangle\right|^k}\cdot\prod_{j\in[\![d]\!]}\mathbb{P}\left[(z_j-z'_j)\in\langle f_G\rangle\right] &\leq \sum_{G\subseteq\{1,\dots,\kappa\}}\left(\frac{q^k}{(2\eta+1)^d}\right)^{\deg(f_G)}\\ &=\prod_{i\in[\![\kappa]\!]}\left(1+\left(\frac{q^k}{(2\eta+1)^d}\right)^{\deg(f_i)}\right)\\ &\leq\prod_{i\in[\![\kappa]\!]}\left(1+\frac{q^k}{(2\eta+1)^d}\right)^{\deg(f_i)}\\ &=\left(1+\frac{q^k}{(2\eta+1)^d}\right)^n.\end{aligned}$$

Putting everything together, it holds

$$\mathbb{P}_{X,X'\sim\mathcal{P}}[X=X']\leq\left(1+\frac{q^k}{(2\eta+1)^d}\right)^n q^{-nk(d+1)},$$

where X and X' are independent and identically distributed according to $\mathcal{P}$. Finally, we observe that $\mathrm{RD}_2(\mathcal{P}\|\mathcal{Q})=|\mathrm{Supp}\mathcal{P}|\cdot\mathbb{P}_{X,X'\sim\mathcal{P}}[X=X']$. We indeed have

$$\begin{aligned}\mathrm{RD}_2(\mathcal{P}\|\mathcal{Q})=\sum_{x\in\mathrm{Supp}\mathcal{P}}\frac{\mathcal{P}(x)^2}{\mathcal{Q}(x)}&=|\mathrm{Supp}\mathcal{P}|\cdot\sum_{x\in\mathrm{Supp}\mathcal{P}}\mathcal{P}(x)^2\\ &=|\mathrm{Supp}\mathcal{P}|\cdot\mathbb{P}_{X,X'\sim\mathcal{P}}[X=X'],\end{aligned}$$

as $\mathcal{Q}=U(\mathrm{Supp}(\mathcal{P}))$. We then use the fact that the Kullback-Leibler divergence D_{KL} (which matches the limit of the log-Rényi divergence when the order goes to 1) is bounded above by the second-order Rényi divergence. This means that we have $D_{KL}(\mathcal{P}\|\mathcal{Q})\leq\mathrm{RD}_2(\mathcal{P}\|\mathcal{Q})$. Then, Pinsker's inequality states that $\Delta(\mathcal{P},\mathcal{Q})\leq\frac{1}{2}\sqrt{D_{KL}(\mathcal{P}\|\mathcal{Q})-1}$ which completes the proof.

We also need a couple of results linked to the regularity of Gaussian distributions. The first, used in Chapter 2 is due to Micciancio and Regev [MR07] showing that above the smoothing parameter, a continuous Gaussian coset is statistically close to uniform in all the lattice cosets.

Lemma 1.18 ([MR07, Lem. 4.1])

Let d be a positive integer, and $\mathcal{L}\subset\mathbb{R}^d$ a lattice of rank d. Let ε be a positive real and $s\geq\eta_\varepsilon(\mathcal{L})$. Then, the distribution of the coset $\mathbf{e}+\mathcal{L}$, where $\mathbf{e}\sim D_s$ is within statistical distance $\varepsilon/2$ of the uniform distribution over the cosets of $\mathcal{L}$.

We also need an equivalent of Lemma 1.16 where $\mathbf{z}$ is drawn from a discrete Gaussian distribution. It deals with primitive matrices $\mathbf{A}$ and is taken from [GPV08, Lem. 5.2], albeit generalized to elliptical Gaussians.

Lemma 1.19 (Gaussian Regularity [GPV08, Lem. 5.2])

Let d,k,q be positive integers. Let R be the ring of integers of a number field, and let $\mathbf{A}\in R_q^{k\times d}$ be such that $\mathbf{A}R_q^d=R_q^k$. Then, let $\varepsilon\in(0,1)$ and $\mathbf{S}\in\mathcal{S}_{nd}^{++}$ be such that $\sqrt{\mathbf{S}}\geq\eta_\varepsilon(\mathcal{L}_q^\perp(\mathbf{A}))$. We finally define $\mathcal{P}=\mathbf{A}\mathcal{D}_{R^d,\sqrt{\mathbf{S}}}\bmod qR$. It then holds that $\mathcal{P}\approx_{\delta_1,\delta_2}U(R_q^k)$ with $\delta_1=(1-\varepsilon)/(1+\varepsilon)$ and $\delta_2=1+\varepsilon$.

Proof (Lemma 1.19). It clearly holds that the support of $\mathcal{P}$ is $\mathbf{A}R^d\bmod qR=\mathbf{A}R_q^d=R_q^k$. We now adapt [GPV08, Cor. 2.8] to elliptical Gaussians.

Lemma 1.20 (Adapted from [GPV08, Cor. 2.8])

Let D be a positive integer. Let $\mathcal{L}' \subseteq \mathcal{L} \subset \mathbb{R}^D$ be two full-rank lattices. Then, let $\varepsilon \in (0,1)$, $\mathbf{S} \in \mathbb{S}_D^{++}$ be such that $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathcal{L}')$, and $\mathbf{c} \in \mathbb{R}^D$. If we call $\mathcal{P} = \mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}} \bmod \mathcal{L}'$ and $\mathcal{Q} = U(\mathcal{L} \bmod \mathcal{L}')$, we have $\mathcal{P}_0 \approx_{\alpha_1,\alpha_2} \mathcal{Q}$ with $\alpha_1 = (1-\varepsilon)/(1+\varepsilon)$ and $\alpha_2 = \alpha_1^{-1}$. When $\mathbf{c} = \mathbf{0}$, α_2 can be improved to $1+\varepsilon$.

Proof (Lemma 1.20). Let $\mathbf{z}$ be distributed according to $\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}$. Let $\mathbf{v}+\mathcal{L}'$ be a coset of $\mathcal{L}/\mathcal{L}'$. Then, it holds that

$$\mathbb{P}_{\mathbf{z}\sim\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}}[\mathbf{z} = \mathbf{v} \bmod \mathcal{L}'] = \frac{\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{v}+\mathcal{L}')}{\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathcal{L})}.$$

By Poisson's summation formula and our condition on $\mathbf{S}$, it holds that $\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathbf{v}+\mathcal{L}') = \rho_{\sqrt{\mathbf{S}},\mathbf{c}-\mathbf{v}}(\mathcal{L}') \in \mathrm{Vol}(\mathcal{L}')^{-1}\sqrt{\det \mathbf{S}} \cdot [1-\varepsilon, 1+\varepsilon]$. Similarly, because $\eta_\varepsilon(\mathcal{L}') \geq \eta_\varepsilon(\mathcal{L})$, we get $\rho_{\sqrt{\mathbf{S}},\mathbf{c}}(\mathcal{L}) \in \mathrm{Vol}(\mathcal{L})^{-1}\sqrt{\det \mathbf{S}} \cdot [1-\varepsilon, 1+\varepsilon]$ (it becomes $[1, 1+\varepsilon]$ when $\mathbf{c} = \mathbf{0}$). As a result, we obtain

$$\mathbb{P}_{\mathbf{z}\sim\mathcal{D}_{\mathcal{L},\sqrt{\mathbf{S}},\mathbf{c}}}[\mathbf{z} = \mathbf{v} \bmod \mathcal{L}'] \in \frac{\mathrm{Vol}(\mathcal{L})}{\mathrm{Vol}(\mathcal{L}')} \cdot \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] = \frac{1}{\left|\mathcal{L}/\mathcal{L}'\right|} \cdot \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right],$$

as desired.

Applying Lemma 1.20 to $\mathcal{L} = R^d$ and $\mathcal{L}' = \mathcal{L}_q^\perp(\mathbf{A})$ (through their embedding to $\mathbb{Z}^{nd}$) yields $\mathcal{P} \approx_{\delta_1,\delta_2} U(\mathcal{L}/\mathcal{L}')$ for $\delta_1 = (1-\varepsilon)/(1+\varepsilon)$ and $\delta_2 = 1+\varepsilon$. Yet $\mathcal{L}/\mathcal{L}'$ is isomorphic to $\mathbf{A}R^d \bmod qR = R_q^k$ which concludes the proof.

1.3.4 Concentration Bounds

In lattice-based cryptography, we are interested in short elements as their shortness hardens the underlying security assumptions and/or improve the efficiency of the cryptographic schemes. Most elements are however drawn from probability distributions over large, and sometimes infinite, supports. The sole argument relies on the fact that these distributions are narrow, which means that the norms are concentrated around specific values. We can therefore bound these elements with a good probability. In particular, several results bounding the tail of discrete Gaussians can be found in the literature. In this thesis, we need to bound the ℓ_2 and ℓ_∞ norms. Notice that we give them for zero-centered distributions which removes the need for a smoothing condition.

Lemma 1.21 (Gaussian Tail Bound ([Ban93, Lem. 1.5][Pei08, Cor. 5.3])

Let d be a positive integer, $\mathcal{L} \subset \mathbb{R}^d$ a lattice of rank d, and $s > 0$. It holds that for all $c > 1/\sqrt{2\pi}$,

$$\forall c > \frac{1}{\sqrt{2\pi}}, \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{L},s}}[\|\mathbf{x}\|_2 > c \cdot s\sqrt{d}] < \left(c\sqrt{2\pi e}e^{-\pi c^2}\right)^d,$$
$$\forall t \geq 0, \mathbb{P}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{L},s}}[\|\mathbf{x}\|_\infty > t \cdot s] < 2de^{-\pi t^2}.$$

Another very important concentration bound is that of the spectral norm of sub-Gaussian matrices. It represents the main metric for the quality of the gadget-based samplers in Chapter 4 which are used in all our constructions. It also intervenes in Part I to assess the quality of our reductions in terms of parameter constraints. Such results from non-asymptotic random matrix theory are proven for example by Vershynin [Ver12]. These results however require strong conditions on the distribution of the entries, in particular independence. As we consider matrices $\mathbf{R}$ of ring elements, we actually look at the spectral norm of $M_\iota(\mathbf{R})$ for an embedding ι. Said structured matrices do not satisfy the independence requirement unfortunately. By using the union bound and Lemma 1.3, we are able to prove bounds using [Ver12]. It turns out however, that they are interesting only with sub-Gaussian distributions defined with respect to the canonical or Minkowski

embedding. It is the case in Part I where we consider Gaussian matrices with respect to σ_H. In our subsequent constructions on the other hand, they are naturally defined with respect to τ, then resulting in loose bounds. For this reason, we give in Lemma 1.22 the spectral concentration bound needed for Part I, and then formulate a heuristical bound to be used in our constructions.

Lemma 1.22 (Gaussian Spectral Bound)

Let n, d, k be positive integers, and $s > 0$. Let R be the ring of integers of a number field of degree n. Then, there exists a universal constant $C > 0$ such that for any $t \geq 0$, it holds

$$\mathbb{P}_{\mathbf{N} \sim \mathcal{D}_{R^{k \times d}, s}} \left[\left\| M_{\sigma_H}(\mathbf{N}) \right\|_2 \geq C \frac{s}{\sqrt{2\pi}} (\sqrt{k} + \sqrt{d} + t) \right] \leq 2ne^{-\pi t^2}.$$

We recall that here $\mathbf{N}$ is drawn from a discrete Gaussian with respect to σ_H. Also, empirically, it holds that $C \approx 1$.

Note that by the conjugation symmetry, the n factor coming from the union bound can be replaced by $t_1 + t_2$ defined in Section 1.1.2. For distributions with respect to the coefficient embedding, one can always consider the transformed distribution by applying $\mathbf{V}$ and study the resulting one with respect to σ or σ_H. The problem is that it naturally involves $\|\mathbf{V}\|_2$ which can be quite large in some cases. In power-of-two cyclotomic fields, the situation is more favorable as $\|\mathbf{V}\|_2 = \sqrt{n}$. This would essentially have the effect of multiplying the bound by $\sqrt{n}$, getting $Cs(\sqrt{nk} + \sqrt{nd} + t\sqrt{n})$. Note that this bound extends to sub-Gaussian distributions by considering s to be the sub-Gaussian moment. This bound is not tight for many distributions where we empirically find generally better conditions. Even though the following situations do not fit the exact requirements of [Ver12], the bounds have extensively used, e.g., [MP12, GMPW20, LNP22], and verified by our own experiments. The constants in Heuristic 1.1 and 1.2 are roughly $C \approx 2$.

Heuristic 1.1 (Uniform Spectral Bound)

Let n, d, k be positive integers with n a power of two. Let R be the cyclotomic ring of conductor $2n$. It heuristically holds that

$$\mathbb{P}_{\mathbf{R} \sim U(S_1^{k \times d})} \left[\left\| M_\tau(\mathbf{R}) \right\|_2 \geq \sqrt{nk} + \sqrt{nd} + t \right] \leq 2ne^{-\pi t^2}.$$

It also heuristically holds that

$$\mathbb{P}_{\mathbf{R} \sim U(S_1^{k \times d})} \left[\left\| M_\tau(\mathbf{R}) \right\|_2 \leq \sqrt{nk} + \sqrt{nd} \right] = 1/C,$$

for a small constant $C = \Theta(1)$ (in particular giving a non-negligible probability).

Heuristic 1.2 (Binomial Spectral Bound)

Let n, d, k be positive integers with n a power of two. Let R be the cyclotomic ring of conductor $2n$. It heuristically holds that

$$\mathbb{P}_{\mathbf{R} \sim \mathcal{B}_1^{k \times d}} \left[\left\| M_\tau(\mathbf{R}) \right\|_2 \leq \frac{7}{10} \left(\sqrt{nk} + \sqrt{nd} + 6 \right) \right] = 1/C,$$

for a small constant $C = \Theta(1)$ (in particular giving a non-negligible probability).

These spectral bound not only intervenes in the quality of our reductions and samplers, but they can also be used to bound norms in our security proofs, i.e., $\|\mathbf{Rz}\|_2 \leq \|\mathbf{R}\|_2 \|\mathbf{z}\|_2$. As part of our optimizations provided in Section 6.4, we notice that this inequality is also not tight in our case where all the involved coefficients are integers[2]. More precisely, we can use the following Johnson-Lindenstrauss-type bound stating that for an arbitrary vector $\mathbf{z}$ and a random short matrix $\mathbf{R}$, then $\mathbf{Rz}$ is not significantly larger than $\mathbf{z}$ except with negligible probability. We can prove such a bound in the non-structured case, but independence also plays a key role. We thus provide a proven bound in Lemma 1.23 and then again give a tighter (heuristical) bound which is backed up

[2] By definition of the spectral norm, the inequality is tight for real-valued vectors.

by experiments.

Lemma 1.23

Let d, k be two positive integers and $\lambda > 0$. Let $\mathbf{z} \in \mathbb{Z}^d$ and $\mathcal{P}$ be a distribution with subgaussian moment $s > 0$. Then it holds that

$$\mathbb{P}_{\mathbf{R}\sim\mathcal{P}^{k\times d}}\left[\|\mathbf{R}\mathbf{z}\|_2 \geq \sqrt{4 + 2\sqrt{\frac{\lambda}{k}}\left(\sqrt{\frac{\lambda}{k}} + \sqrt{\frac{8}{\ln 2} + \frac{\lambda}{k}}\right)\ln 2}\cdot s\sqrt{k}\|\mathbf{z}\|_2\right] \leq 2^{-\lambda}.$$

Proof (Lemma 1.23). Define $\beta = \|\mathbf{z}\|_2$, and s be the subgaussian moment of $\mathcal{P}$. Let $\mathbf{R} \sim \mathcal{P}^{k\times d}$. Let $i \in [\![k]\!]$ and $t \in \mathbb{R}$. Then,

$$\begin{aligned}
\mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}[\exp(t\mathbf{r}_i^T\mathbf{z})] &= \mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}\left[\prod_{j\in[\![d]\!]} e^{tz_jr_{i,j}}\right] \\
&= \prod_{j\in[\![d]\!]} \mathbb{E}_{r_{i,j}\sim\mathcal{P}}[\exp(tz_jr_{i,j})] \\
&\leq \prod_{j\in[\![d]\!]} \exp(s^2(tz_j)^2/2) \\
&= \exp((\beta s)^2t^2/2).
\end{aligned}$$

So $x_i = \mathbf{r}_i^T\mathbf{z}_i$ is βs-subgaussian for each $i \in [\![k]\!]$. Let $y_i = x_i^2$ and $\mu_i = \mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}[y_i]$. Because x_i is βs-subgaussian, it means that

$$\forall p \geq 1, \mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}[|x_i|^p] \leq p(\sqrt{2}\beta s)^p\Gamma(p/2).$$

In particular, $\mu_i \leq 2(\sqrt{2}\beta s)^2\Gamma(1) = 4\beta^2s^2$. As a consequence, it holds that

$$\begin{aligned}
\mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}[\exp(t(y_i-\mu_i))] &= 1 + t\mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}[y_i-\mu_i] + \sum_{p\geq 2}\frac{t^p}{p!}\mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}[(x_i^2-\mu_i)^p] \\
&\leq 1 + \sum_{p\geq 2}\frac{t^p}{p!}\mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}[x_i^{2p}] \\
&\leq 1 + \sum_{p\geq 2}\frac{t^p}{p!}(2p(\sqrt{2}\beta s)^{2p}\Gamma(p)) \\
&= 1 + 2\sum_{p\geq 2}(2t\beta^2s^2)^p \\
&= 1 + 2\left(\frac{1}{1-2\beta^2s^2t} - (1+2\beta^2s^2t)\right) \\
&= 1 + \frac{8t^2\beta^4s^4}{1-2\beta^2s^2t}.
\end{aligned}$$

where the second to last equality holds if $|t| \leq 1/(2\beta^2s^2)$. So for some $\alpha \geq 1$, and for all t such that $|t| < 1/(2\alpha\beta^2s^2)$, we have

$$\mathbb{E}_{\mathbf{r}_i\sim\mathcal{P}^d}[\exp(t(y_i-\mu_i))] \leq \exp\left(\frac{16\beta^4s^4\alpha}{\alpha-1}\cdot\frac{t^2}{2}\right),$$

meaning that $y_i - \mu_i$ is a centered sub-exponential random variable with parameters $\gamma = 4\beta^2s^2\sqrt{\alpha/(\alpha-1)}$ and $\delta = 2\beta^2s^2\alpha$. Thence, $y - \mu = \sum_{i\in[\![k]\!]} y_i - \mu_i$ is subexponential with parameters $\gamma' = \gamma\sqrt{k}$ and $\delta' = \delta$. Using the sub-exponential tail bound, we obtain that for all $r \in (0, \gamma'^2/\delta')$,

$$\mathbb{P}_{\mathbf{R}\sim\mathcal{P}^{k\times d}}[y - \mu \geq r] \leq \exp(-r^2/(2\gamma'^2)).$$

This can be re-written as follows. For all $\lambda \in (0, \frac{2k}{\alpha(\alpha-1)\ln 2})$, it holds that

$$\mathbb{P}_{\mathbf{R}\sim\mathcal{P}^{k\times d}}\left[\|\mathbf{Rz}\|_2^2 \geq 4k\beta^2 s^2\left(1+\sqrt{\frac{2\alpha\ln 2}{\alpha-1}\cdot\frac{\lambda}{k}}\right)\right] \leq 2^{-\lambda}.$$

We now fix λ and k and optimize over α. More precisely, we need to maximize $\alpha > 1$ while ensuring that $\lambda < 2k/(\alpha(\alpha-1)\ln 2)$. The optimal value is then

$$\alpha^* = \frac{1}{2}\left(1+\sqrt{1+\frac{8k}{\lambda\ln 2}}\right),$$

We then obtain a bound on $\|\mathbf{Rz}\|_2^2/(k\beta^2 s^2)$ as

$$\gamma = 4\left(1+\sqrt{\frac{2\alpha^*\ln 2}{\alpha^*-1}\cdot\frac{\lambda}{k}}\right) = 4+2\ln 2\cdot\sqrt{\frac{\lambda}{k}}\cdot\left(\sqrt{\frac{\lambda}{k}}+\sqrt{\frac{8}{\ln 2}+\frac{\lambda}{k}}\right).$$

We then conclude that $\mathbb{P}_{\mathbf{R}\sim\mathcal{P}^{k\times d}}[\|\mathbf{Rz}\|_2 \geq \sqrt{\gamma}\cdot s\sqrt{k}\|\mathbf{z}\|_2] \leq 2^{-\lambda}$ as desired. In general, k is much larger than λ, meaning that the factor in front of $s\sqrt{k}\|\mathbf{z}\|_2$ can be bounded by a constant, and goes to 2 for smaller ratios λ/k.

The bound on $\|\mathbf{Rz}\|_2$ from Lemma 1.23 is only needed in the proof of unforgeability of our signature of Section 6.4. As a result, it only needs to be verified with a probability that is non-negligible, say a constant, but it does not have to be overwhelming[3]. For example, if the bound is verified only with a probability of 1/2, it only entails a couple of extra bits in the security loss. This allows us to obtain tighter bounds and in turn tighter parameter constraints. We note that such results are obtained with overwhelming probability in [GHL22, LNP22] based on the normal-distribution heuristic but the latter is not verified for structured matrices. This is why we provide the following bound which is empirically verified in the structured case. The constant in Heuristic 1.3 is roughly $C \approx 2$.

Heuristic 1.3 (Johnson-Lindenstrauss Bound)

Let n, d, k be positive integers with n a power of two. Let R be the cyclotomic ring of conductor $2n$. For any arbitrary $\mathbf{z} \in R^d$, it heuristically holds that

$$\mathbb{P}_{\mathbf{R}\sim\mathcal{B}_1^{k\times d}}\left[\|\mathbf{Rz}\|_2 \leq \frac{1}{\sqrt{2}}\sqrt{nk}\|\mathbf{z}\|_2\right] = 1/C,$$

for a small constant $C = \Theta(1)$ (in particular giving a non-negligible probability).

1.3.5 Rejection Sampling

Rejection sampling is a powerful tool in probability theory that allows to sample from a target distribution $\mathscr{D}_t$ by first sampling from a source distribution $\mathscr{D}_s$ and rejecting in order to smooth it to fit $\mathscr{D}_t$. In cryptography, it also serves the purpose of making the output sample independent of secret values. In this thesis, the source distribution is publicly known and allows one to sample *masks* $\mathbf{p}$. The mask is then shifted by a secret value $\mathbf{s}$. Outputting $\mathbf{p}+\mathbf{s}$ would leak some information on $\mathbf{s}$ which is why we reject it according to a specific condition so that the output distribution corresponds to the publicly known target distribution. We start by giving the general version resulting from [DFPS22, Lem. 2.2 & 2.4] that we will used throughout Part II.

[3]Similar bounds for unstructured matrices are used in the zero-knowledge proof system we use. The (heuristical) bound of [LNP22, Lem. 2.8] is $\sqrt{337}\|\mathbf{z}\|_2$ for $\mathcal{P} = \psi_1$ and $(d, \lambda) = (256, 128)$. In this case, we need an overwhelming probability. For the same parameters, our proven result yields a bound of $\sqrt{1037}\|\mathbf{z}\|_2$ instead.

Lemma 1.24 (Generalized Rejection Sampling (adapted from [DFPS22, Lem. 2.2, Lem. 4.1]))

Let d, m be positive integers. Let $\mathscr{D}_s, \mathscr{D}_t, \mathscr{D}_r, \mathscr{D}_z$ be distributions on $\mathbb{R}^d, \mathbb{R}^d, \mathbb{R}^{d\times m}, \mathbb{R}^m$ respectively. Let $\mathbf{R}$ be drawn from $\mathscr{D}_r$. Then, let $Y \subseteq \mathbb{R}^d$ be the support of the distribution of $\mathbf{R}\cdot\mathscr{D}_z$. We assume they are such that $\mathrm{Supp}(\mathscr{D}_t) \subseteq \mathrm{Supp}(\mathscr{D}_s^{+\mathbf{Rz}})$ for all $\mathbf{Rz} \in Y$, where $\mathscr{D}_s^{+\mathbf{Rz}}$ is the distribution corresponding to sampling $\mathbf{p}$ from $\mathscr{D}_s$ and outputting $\mathbf{p}+\mathbf{Rz}$. Let $M > 1$ and $\varepsilon \in [0, 1/2]$ such that $\max_{\mathbf{Rz}\in Y} RD^{\varepsilon}_{\infty}(\mathscr{D}_t \| \mathscr{D}_s^{+\mathbf{Rz}}) \leq M$. We then define two distributions

$\mathcal{P}_1$: Sample $\mathbf{z} \hookleftarrow \mathscr{D}_z$, $\mathbf{p} \hookleftarrow \mathscr{D}_s$ and set $\mathbf{v} \leftarrow \mathbf{p}+\mathbf{Rz}$. Then sample a continuous $u \hookleftarrow U([0,1])$. If $u > \min(1, \mathscr{D}_t(\mathbf{v})/(M\cdot\mathscr{D}_s(\mathbf{p})))$, restart, otherwise output $(\mathbf{v}, \mathbf{z})$.

$\mathcal{P}_2$: Sample $\mathbf{z} \hookleftarrow \mathscr{D}_z$, $\mathbf{v} \hookleftarrow \mathscr{D}_t$. Then sample a continuous $u \hookleftarrow U([0,1])$. If $u > 1/M$, restart, otherwise output $(\mathbf{v}, \mathbf{z})$.

Then, $\Delta(\mathcal{P}_1, \mathcal{P}_2) \leq \varepsilon$ and for all $\alpha \in (1, +\infty]$, $\mathrm{RD}_{\alpha}(\mathcal{P}_1 \| \mathcal{P}_2) \leq 1/(1-\varepsilon)^{\alpha/(\alpha-1)}$.

Notice that in this version we re-sample masks until one is accepted. In other contexts, it may be acceptable to simply abort the process when the first rejection occurs. In particular, this is what we use in the security proof of Section 6.4, and the subsequent zero-knowledge arguments [LNP22]. In this cases, the masks are drawn from a spherical Gaussian distribution and we thus formulate these results as such. Lemma 1.25 simply consists in instantiating Lemma 1.24 with the smooth Rényi divergence bound of Lemma 1.11 [DFPS22].

Lemma 1.25 (Gaussian Rejection Sampling (adapted from [DFPS22, Lem. 2.2, 4.1 & C.2]))

Let d be a positive integer. Let $S \subset R^d$ be a set of vectors of ℓ_2 norm at most $T > 0$, and $\mathscr{D}_S$ be a distribution over S. Let $M > 1$, $\varepsilon \in (0, 1/2]$ and let $\gamma = \frac{\sqrt{\pi}}{\ln M}(\sqrt{\ln \varepsilon^{-1} + \ln M} + \sqrt{\ln \varepsilon^{-1}})$. Then, let $s \geq \gamma T$. We define the following distributions.

$\mathcal{P}_1$: Sample $\mathbf{s} \hookleftarrow \mathscr{D}_S$, $\mathbf{y} \hookleftarrow \mathcal{D}_{R^d,s}$ and set $\mathbf{z} = \mathbf{y}+\mathbf{s}$. Then, sample $u \hookleftarrow U([0,1))$. If $u > \frac{1}{M}\exp\left(\frac{\pi}{s^2}\left(\|\tau(\mathbf{s})\|_2^2 - 2\langle\tau(\mathbf{z}),\tau(\mathbf{s})\rangle\right)\right)$, output $\perp$. Else output $(\mathbf{s}, \mathbf{z})$.

$\mathcal{P}_2$: Sample $\mathbf{s} \hookleftarrow \mathscr{D}_S$ and $\mathbf{z} \hookleftarrow \mathcal{D}_{R^d,s}$. Then sample a continuous $u \hookleftarrow U([0,1))$. If $u \leq 1/M$, output $(\mathbf{s}, \mathbf{z})$, and $\perp$ otherwise.

Then, $\Delta(\mathcal{P}_1, \mathcal{P}_2) \leq \varepsilon/M$, and $RD_{\infty}(\mathcal{P}_1 \| \mathcal{P}_2) \leq 1 + \varepsilon/(M-1)$.

We also need another rejection sampling result from [LNS21] which leaks at most one bit of information if it is to hide ephemeral randomness. It is similar to the previous one except that it also rejects based on the direction of $\mathbf{z}$ with respect to $\mathbf{s}$. Note it cannot be used for long-term secrets as leakage would increase with repetition.

Lemma 1.26 ([LNS21, Lem. 3.2])

Let d be a positive integer. Let $S \subset R^d$ be a set of vectors of ℓ_2 norm at most $T > 0$, and $\mathscr{D}_S$ be a distribution over S. Let $M > 1$ and $\gamma = \sqrt{\pi/\ln M}$. Then, let $s \geq \gamma T$. We define the following distributions.

$\mathcal{P}_1$: Sample $\mathbf{s} \hookleftarrow \mathscr{D}_S$, $\mathbf{y} \hookleftarrow \mathcal{D}_{R^d,s}$ and set $\mathbf{z} = \mathbf{y}+\mathbf{s}$. Then, sample $u \hookleftarrow U([0,1))$. If $\langle\tau(\mathbf{z}),\tau(\mathbf{s})\rangle < 0$ or if $u > \frac{1}{M}\exp\left(\frac{\pi}{s^2}\left(\|\tau(\mathbf{s})\|_2^2 - 2\langle\tau(\mathbf{z}),\tau(\mathbf{s})\rangle\right)\right)$, output $\perp$. Else output $(\mathbf{s}, \mathbf{z})$.

$\mathcal{P}_2$ Sample $\mathbf{s} \hookleftarrow \mathscr{D}_S$ and $\mathbf{z} \hookleftarrow \mathcal{D}_{R^d,s}$. Then sample $u \hookleftarrow U([0,1))$. If $\langle \tau(\mathbf{z}), \tau(\mathbf{s}) \rangle < 0$ or if $u > \frac{1}{M}$, output $\perp$. Otherwise output $(\mathbf{s}, \mathbf{z})$.

Then, $\mathcal{P}_1$ outputs $(\mathbf{s}, \mathbf{z}) \neq \perp$ with probability at least $1/2M$, and conditioned on not aborting it holds that $\mathcal{P}_1$ and $\mathcal{P}_2$ are identical.

1.4 Hardness Assumptions

The lattice problems introduced in Section 1.2 form a solid base to prove the security of cryptographic systems. However, the latter are usually called *worst-case* problems which makes it difficult to directly base cryptographic designs on them. Indeed, for every d, there exist lattices of dimension d in which these problems are easily solvable. On the contrary, there also exists lattices in which these problems are proven (or conjectured) to be exponentially hard. To obtain secure cryptographic constructions, one would need to have an efficient way of finding these hard instances, while finding secret information on them to design public-key cryptography. As this is no easy task, this motivated the introduction of more flexible fundamental problems characterized as *average-case*, for which the instances can be sampled randomly. Their attractive feature, which is unique to lattice-based cryptography, is that they are proven to be at least as hard as the worst instances of $\text{SVP}_\gamma, \text{CVP}_\gamma$ or their variants for γ polynomial in the dimension.

We now introduce these main theoretical assumptions that form the security foundations of our cryptographic primitives and protocols. Although there is a plethora of security assumptions to chose from, we focus on the *Short Integer Solution* [Ajt96, MR07] and the *Learning With Errors* [Reg05] problems which are now considered the most common assumptions. They have been studied in depth over the past few decades, including in their structured variants. In this thesis, we only consider the structured versions of these problems, that is with algebraic integers instead of regular ones. As such, we only define the *module* versions [LS15] and explain how it interpolates between the standard and ring formulations. We still note that other algebraically structured variants exist and refer to [PP19] for more details.

1.4.1 Short Integer Solution

We start by introducing the *Short Integer Solution* (SIS) which was formulated by Ajtai [Ajt96] and later formalized by Micciancio and Regev [MR07]. It relies on the observation mentioned above that hard lattices may be impractical to find and use in cryptography. One would instead try to find random instances of certain lattice problems, say SVP_γ, by essentially sampling random lattices that would make the problem hard. In Section 1.2, we introduce the family of q-ary lattices $\mathcal{L}_q^\perp(\mathbf{A})$ which can be sampled by simply sampling a matrix $\mathbf{A}$ at random. We can then consider SVP_γ over $\mathcal{L}_q^\perp(\mathbf{A})$, that is finding a non-zero vector $\mathbf{x}$ such that $\mathbf{A}\mathbf{x} = \mathbf{0} \bmod q$ and such that $\|\mathbf{x}\|_2 \leq \beta = \gamma\lambda_1(\mathcal{L}_q^\perp(\mathbf{A}))$. This is exactly the SIS problem, which was proven to be at least as hard as the worst instances of some variant of SVP_γ. One can also define it over lattice cosets $\mathcal{L}_q^{\mathbf{u}}(\mathbf{A})$ and remove the non-zero condition to obtain an inhomogeneous version of SIS called ISIS.

These problems can be defined in several variants and in several algebraic contexts. We only work with the module setting studied by Langlois and Stehlé [LS15], but still use different variants. In particular, we consider the Hermite Normal Form of the problem where the matrix $\mathbf{A}$ is of the form $[\mathbf{I}_d|\mathbf{A}']$, at the exception of Section 6.2 where $\mathbf{A}$ is fully uniform. Also, in some occasions, we consider two bounds, one for the Euclidean norm and the other for the infinity norm.

Definition 1.14 (M-ISIS and M-SIS)

Let n, d, m, q be positive integers with $m > d$, and $\beta, \beta_\infty > 0$. Let K be a number field of degree n, and R its ring of integers. The *Module Inhomogeneous Short Integer Solution* problem $\text{M-ISIS}_{n,d,m,q,\beta,\beta_\infty}$ asks to find $\mathbf{x} \in \mathcal{L}_q^{\mathbf{u}}(\mathbf{A})$ such that $\|\mathbf{x}\|_2 \leq \beta$ and $\|\mathbf{x}\|_\infty \leq \beta_\infty$, given $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$ with $\mathbf{A}' \hookleftarrow U(R_q^{d\times m-d})$ and $\mathbf{u} \hookleftarrow U(R_q^d)$.

When $\mathbf{u} = \mathbf{0}$, we call it $\text{M-SIS}_{n,d,m,q,\beta,\beta_\infty}$ and expect the solution $\mathbf{x}$ to be non-zero. We sometimes use a matrix $\mathbf{A}$ that is fully random, i.e., $\mathbf{A} \hookleftarrow U(R_q^{d\times m})$, but do not differentiate the notations of the problems as they are equivalent (with high probability over $\mathbf{A}$).

When only considering the Euclidean norm bound, we remove the subscript β_∞.

The advantage of a *probabilistic polynomial time* (PPT) adversary $\mathcal{A}$ against M-ISIS$_{n,d,m,q,\beta,\beta_\infty}$ is defined by

$$\mathrm{Adv}_{\text{M-ISIS}}[\mathcal{A}] = \mathbb{P}\left[\mathbf{x} \in \mathcal{L}_q^{\mathbf{u}}(\mathbf{A}) \wedge \|\mathbf{x}\|_2 \leq \beta \wedge \|\mathbf{x}\|_\infty \leq \beta_\infty : \mathbf{x} \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{u})\right],$$

where the probability is over the randomness of $(\mathbf{A}, \mathbf{u})$ and the random coins of $\mathcal{A}$. When the parameters are clear from the context, we define the hardness bound as $\varepsilon_{\text{M-ISIS}} = \sup_{\mathcal{A}\ \text{PPT}} \mathrm{Adv}_{\text{M-ISIS}}[\mathcal{A}]$. We define similar quantities for the M-SIS problem.

We note that the original formulation [Ajt96, MR07, LS15] considers a single bound β' on the Euclidean norm. There is a trivial reduction from the latter to the version with (β, β_∞) by setting $\beta' = \min(\beta_\infty\sqrt{nm}, \beta)$. As discussed by Micciancio and Peikert [MP13, Thm. 1.1], using both norm bounds leads to more precise hardness results, and sometimes smaller approximation factors when related to worst-case problems on lattices. Moreover, it seems to be relevant to counter certain cryptanalytic methods used to solve the problem. Indeed, most lattice reduction algorithms aim at finding vectors in the ball of radius β but without constraining the magnitude of the coefficients. Finding a lattice vector that is also in the hypercube of half-side β_∞ is at least as hard as the same task without the β_∞ bound. But when, $\beta_\infty \ll \beta$, it may even be substantially harder. This relates to the observation recently made by Ducas et al. [DEP23] in the sense that a smaller β_∞ bound would invalidate candidate solutions found by their lifting method as it reduces the size of the box. We give more details in Chapter 9.

Also, we mention that our definition encompasses the unstructured version SIS [Ajt96] by selecting $n = 1$, and also its ring version Ring-SIS [PR06, LM06] by selecting $d = 1$.

1.4.2 Learning With Errors

We now introduce a second fundamental problem called *Learning With Errors* (LWE) defined by Regev [Reg05] and that has proven to be as versatile as SIS. It also benefits from worst-case to average-case reductions from lattice problems such as (variants of) the SVP$_\gamma$ problem. Just like the Short Integer Solution problem, the purpose of LWE is to provide a more flexible assumption to design cryptography upon. Nevertheless, it can still be interpreted as a lattice problem on random lattices. Given a random matrix $\mathbf{A}$, one can consider the CVP$_\gamma$ problem on the q-ary lattice $\mathcal{L}_q(\mathbf{A})$. It corresponds to finding the closest vector $\mathbf{As}$ to a vector $\mathbf{t}$. By writing $\mathbf{t} = \mathbf{As}+\mathbf{e}$, the problem becomes finding $\mathbf{e}$ (or $\mathbf{s}$) given $\mathbf{A}$ and $\mathbf{t}$. It can thus be seen as solving a noisy linear system modulo q. If the *error* $\mathbf{e}$ is too close to q, the problem becomes extremely hard, whereas if it is too close to $\mathbf{0}$, the problem becomes easier. In its seminal paper, Regev showed the applicability of LWE in cryptography through a bit encryption scheme. Its use has very much flourished since then and the assumption is now considered as standard in lattice-based cryptography, intervening in the design of encryptions, signatures, zero-knowledge proofs, and more.

In this thesis, we focus on the module variant of LWE, denoted M-LWE, which was first defined by Brakerski et al. [BGV12] and thoroughly studied by Langlois and Stehlé [LS15]. We analyze the hardness of some variants of the problem where the secret $\mathbf{s}$ (see Chapter 2) or the error $\mathbf{e}$ (see Chapter 3) are drawn from bounded uniform distributions. These regimes reflect the practical usage of the M-LWE assumption, which we leverage in our cryptographic designs. We give in Definition 1.15 a version that encompasses all the versions covered in this thesis, both in the more theoretical part than in the practical one.

Definition 1.15 (M-LWE)

Let n, d, m, k, q be positive integers with $m \geq d$. Let K be a number field of degree n, and R its ring of integers. Then, let $\mathscr{D}_s$ be a distribution of secrets over R, and $\mathscr{D}_e$ a distribution of errors over $K_\mathbb{R}$ (possibly with a discrete support included in R). We define $\mathbb{S}_q = \mathrm{Supp}(\mathscr{D}_e)/qR \cap R_q$ (e.g., $\mathbb{S}_q = \mathbb{T}_q$ when $\mathrm{Supp}(\mathscr{D}_e) = K_\mathbb{R}$ and R_q when $\mathrm{Supp}(\mathscr{D}_e) \subseteq R$).

Decision: The (decision) *Module Learning With Errors* problem M-LWE$^k_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ asks to distinguish between the following distributions:

$\mathcal{P}_1$: Sample $\mathbf{A} \hookleftarrow U(R_q^{m\times d})$, $\mathbf{S} \hookleftarrow \mathscr{D}_s^{d\times k}$ and $\mathbf{E} \hookleftarrow \mathscr{D}_e^{m\times k}$. Compute $\mathbf{B} = \mathbf{AS} + \mathbf{E} \bmod qR$.
Output: $(\mathbf{A}, \mathbf{B})$.

$\mathcal{P}_2$ Sample $\mathbf{A} \hookleftarrow U(R_q^{m\times d})$, and $\mathbf{B} \hookleftarrow U(\mathbb{S}_q^{m\times k})$.
Output: $(\mathbf{A}, \mathbf{B})$.

Search: The *search Module Learning With Errors* problem $\text{sM-LWE}^k_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ asks to find $(\mathbf{S}, \mathbf{E})$ given $(\mathbf{A}, \mathbf{B} = \mathbf{AS} + \mathbf{E} \bmod qR) \hookleftarrow \mathcal{P}_1$.

The advantage of a *probabilistic polynomial time* (PPT) adversary $\mathcal{A}$ against $\text{M-LWE}^k_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ is defined by

$$\text{Adv}_{\text{M-LWE}}[\mathcal{A}] = \left|\mathbb{P}[\mathcal{A}(\mathcal{P}_1) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{P}_2) = 1]\right|,$$

while that against $\text{sM-LWE}^k_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ is

$$\text{Adv}_{\text{sM-LWE}}[\mathcal{A}] = \mathbb{P}_{(\mathbf{A},\mathbf{B})\sim\mathcal{P}_1}[\mathbf{B} = \mathbf{AS}^\star + \mathbf{E}^\star \bmod qR : (\mathbf{S}^\star, \mathbf{E}^\star) \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{B})],$$

When the parameters are clear from the context, we define the hardness bounds as $\varepsilon_{\text{M-LWE}} = \sup_{\mathcal{A}\ \text{PPT}} \text{Adv}_{\text{M-LWE}}[\mathcal{A}]$, and similarly for $\varepsilon_{\text{sM-LWE}}$. We note that as for M-SIS, it interpolates the standard formulation LWE from [Reg05] by setting $n = 1$ (and $\mathscr{D}_s = U(\mathbb{Z}_q)$ and $\mathscr{D}_e = \mathcal{D}_{\mathbb{Z},s}$), and also its ring version R-LWE [LPR10] by setting $d = 1$. When considering either, we remove the subscripted quantity that is set to 1.

Discussion on Variants

Definition 1.15 gives a rather abstract way of encompassing several of the usual variants of M-LWE. We briefly explain here how the different variants we consider in this thesis correspond to specific parameter selections in the above definition.

Multiple Secrets. We note that the usual definition considers only $k = 1$ which corresponds to one secret vector and one error vector. Our constructions however rely on this multiple secrets variant which is why we define it in the most general way. We note however that a standard hybrid argument shows that $\text{M-LWE}^k_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ is at least as hard as $\text{M-LWE}^1_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ at the expense of a loss factor k in the reduction. The same holds for the search variant. It remains acceptable as long as k is polynomial in the security parameter. When $k = 1$, we omit the superscript. In that case the instance is some $(\mathbf{A}, \mathbf{b})$ where $\mathbf{b}$ is either uniformly distributed or of the form $\mathbf{As} + \mathbf{e} \bmod qR$ depending on the version we consider.

Discrete versus Continuous. The problem is generally defined with discrete error distributions, i.e., $\text{Supp}(\mathscr{D}_e) \subseteq R$, in cryptographic constructions. We can however consider a continuous version where the error is supported over $K_{\mathbb{R}}$. This version is handy in theoretical results on M-LWE. It can be discretized a posteriori with a carefully chosen rounding function $\lfloor\cdot\rceil : K_{\mathbb{R}} \to R$, which has the effect of changing the error distribution to $\mathscr{D}'_e = \lfloor\mathscr{D}_e\rceil$. We refer to [LPR13b, Sec. 2.6] for a detailed discussion.

Hermite Normal Form. The Hermite Normal Form is generally defined by defining $\mathbf{B} = [\mathbf{I}_m|\mathbf{A}]\mathbf{R} \bmod qR$ for some secret $\mathbf{R}$ drawn from some $\mathscr{D}_r^{m+d\times k}$ with $\text{Supp}(\mathscr{D}_r) \subseteq R$. This corresponds to setting $\mathscr{D}_e = \mathscr{D}_s = \mathscr{D}_r$ in Definition 1.15. In this case, we simply write $\text{M-LWE}^k_{n,d,m,q,\mathscr{D}_r}$.

Worst-case Error. We also define the problem with worst-case error distributions when the error is $D_{\mathbf{r}} \in \Psi_{\leq r}$ for some possibly unknown (and possibly secre-dependent) $\mathbf{r} \in (\mathbb{R}^+)^n$. By abuse of notation, we change the subscript $\mathscr{D}_e$ by $\Psi_{\leq r}$. We reduce to this variant in Section 2.3.3. Note that one can use the reduction of [PRS17, Lem. 7.2], generalizing [LPR10, Lem. 5.16], to move to an average-case error with a spherical Gaussian $D_{r'}$. The reduction increases the noise from r to $r' = r(nm/\log_2 nm)^{1/4}$. The result is stated for R-LWE but naturally extends to the module setting, and we therefore do not consider it new and do not include this extra step in the overall reduction.

1.5 Signatures and Security Models

We end this chapter by giving the security models of the different types of signatures covered in this thesis. Security models are defined to capture real-life scenarios and to theoretically model the behavior of real-life attackers. If a security model is meaningless with respect to actual situations, it should dismissed or overhauled.

A security model is the combination of (1) a threat model, and (2) an objective. The *threat model* (1) encompasses the overall capabilities of the adversary. This includes an *attack model*, that is the adversary's resources in terms of computation time, computational power, memory availability, energy consumption, etc. Compared to the real-world, it allows modeling the difference between a hacker with limited capabilities and a government mustering much more resources. In cryptography, we almost always consider *probabilistic polynomial-time* (PPT) adversaries, that can be modeled as Turing machines. By polynomial-time, we mean that the adversary has a limited amount of time to perform the attack which is polynomial in a security parameter λ. A threat model also includes an *attack setting* establishing what the adversary has access to, e.g., only the public key, or the public key and passively obtained signatures, etc. Finally, the (2) *objective* fixes the goal of the adversary, e.g., recover the secret key, forge a signature, learn some private information, etc.

We note that the threat model has to be changed when assessing quantum security. Indeed, the adversary may have access to quantum queries, quantum random access memory, and so on.

1.5.1 Digital Signatures

We now introduce *digital signatures* which represent one of the most widely used primitive in cryptography, and which can be traced back to the seminal work of Diffie and Hellman [DH76]. They act as a certificate that the signed data is authentic, and they represent a digital version of hand-written signatures.

Syntax of Digital Signatures

Informally, a signature is produced on a message using the secret key so that only the owner of said key can certify data, and the signature can be verified using the message and the public key, making it verifiable by everybody. We give the formal definition in Definition 1.16

Definition 1.16 (Digital Signature)

A digital signature scheme is defined by four algorithms Setup, KeyGen, Sign and Verify which are described as follows.

- Setup: Takes the security parameter λ and outputs public parameters pp (includes λ).
- KeyGen: Takes the public parameters pp and outputs a public key pk and the associated secret key sk.
- Sign: Takes a secret key sk, a message $\mathbf{m}$, and possibly the public parameters pp and public key pk, and outputs a signature $\mathsf{sig} = \mathsf{Sign}(\mathsf{sk}, \mathbf{m}, (\mathsf{pk}, \mathsf{pp}))$.
- Verify: Takes a public key pk, a message $\mathbf{m}$, a signature sig and possibly the public parameters pp, and outputs a bit $b = \mathsf{Verify}(\mathsf{pk}, \mathbf{m}, \mathsf{sig}, \mathsf{pp})$ which is 1 if the signature is valid and 0 otherwise.

The signature scheme must be *correct*, that is verifying $\forall\lambda, \forall \mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda), \forall(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\mathsf{pp}), \forall \mathbf{m}$

$$\mathsf{Verify}(\mathsf{pk}, \mathbf{m}, \mathsf{Sign}(\mathsf{sk}, \mathbf{m}, \mathsf{pk}, \mathsf{pp}), \mathsf{pp}) = 1$$

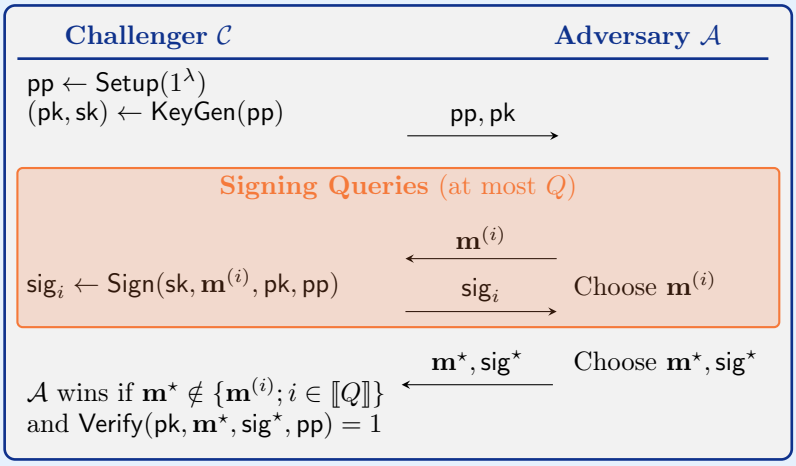
The signer can also maintain a state st which is used to keep track of some information necessary for the signing procedure. The state can be as simple as a counter, but can also be more complex like a table storing all the previously emitted signatures. When a state is used, we generally call the signature scheme *stateful*, and *stateless* otherwise.

EUF-CMA Security Model

The most widely used notion of security for a signature scheme is the *Existential Unforgeability against Chosen Message Attacks* (EUF-CMA) security. It captures the idea that an adversary that only knows the public key and that can obtain signatures on messages of its choosing is incapable of producing a valid signature on a new message. It thus guarantees that nobody is able to usurp the identity of a signer and certify data in their name. The security model is formally defined a three-stage game given in Definition 1.17.

Definition 1.17 (EUF-CMA Security)

Let $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be a signature scheme. We define the following experiment.



The advantage of the adversary $\mathcal{A}$ is its probability of winning the above game, that is

$$\mathrm{Adv}_{\text{EUF-CMA}}[\mathcal{A}] = \mathbb{P}[\mathsf{Verify}(\mathsf{pk}, \mathbf{m}^\star, \mathsf{sig}^\star, \mathsf{pp}) = 1 \wedge \forall i \in [\![Q]\!], \mathbf{m}^\star \neq \mathbf{m}^{(i)}].$$

We say that the signature scheme is ε-EUF-CMA secure if for all PPT adversary $\mathcal{A}$, it holds that $\mathrm{Adv}_{\text{EUF-CMA}}[\mathcal{A}] \leq \varepsilon$.

One can also consider a stronger notion called *Strong Existential Unforgeability against Chosen Message Attacks* (sEUF-CMA). It follows the same security model except for the winning conditions of the adversary. In particularly, in this case, we accept forgeries for which $\mathbf{m}^\star$ was queried to the signing oracle as long as the associated signature $\mathsf{sig}^\star$ is not one of the issued signatures. The advantage is then defined by

$$\mathrm{Adv}_{\text{sEUF-CMA}}[\mathcal{A}] = \mathbb{P}[\mathsf{Verify}(\mathsf{pk}, \mathbf{m}^\star, \mathsf{sig}^\star, \mathsf{pp}) = 1 \wedge \forall i \in [\![Q]\!], (\mathsf{sig}^\star, \mathbf{m}^\star) \neq (\mathsf{sig}^{(i)}, \mathbf{m}^{(i)})].$$

1.5.2 Anonymous Credentials

Anonymous credentials (AC), sometimes known as attribute-based credentials, is a generic term covering a wide spectrum of privacy-preserving systems considering essentially two main use cases. One where an *organization* generates credentials on possibly concealed attributes for *users* through an interactive process $\mathsf{Issue}_{O,U}$. Another where the credentials can thus be shown to *verifiers* through an interactive protocol $\mathsf{Show}_{U,V}$ while limiting leakage of information depending on the specific application. In other words, $\mathsf{Issue}_{O,U}$ and $\mathsf{Show}_{U,V}$ can be seen as the interactive counterparts of Sign and Verify in a regular digital signature with extra privacy requirements, and handling *credentials* and *attributes* instead of *signatures* and *messages*.

Definition 1.18 (Anonymous Credentials)

An anonymous credential system is defined by three algorithms Setup, $\mathsf{OKeyGen}$, $\mathsf{UKeyGen}$, and two interactive protocols Issue and Show, which are described as follows.

- Setup: Takes the security parameter λ and outputs public parameters pp (includes λ).
- $\mathsf{OKeyGen}$: Takes the public parameters pp and outputs an organization public key opk and the associated secret key osk.
- $\mathsf{UKeyGen}$: Takes the public parameters pp and outputs a user public key upk and the associated secret key usk.
- $\mathsf{Issue}_{O,U}$: Protocol between an organization O with $(\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{pp})$ and the user's disclosed attributes $(\mathbf{m}_i)_{i \in \mathcal{I}}$ and a user U holding $(\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp})$ and their attributes $\mathbf{m}$. The user U either obtains a credential cred on its attributes or $\perp$ if the protocol failed, while O simply gets notified of whether or not the execution was successful.

- $\mathsf{Show}_{U,V}$: Protocol between a user U with $(\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathsf{pp}, \mathbf{m}, \mathsf{cred})$ and a verifier V with $(\mathsf{opk}, \mathsf{pp})$ and the user's disclosed attributes $(\mathbf{m}_i)_{i\in\mathcal{I}'}$. The protocol outputs $b = 1$ to V if the credential cred is valid with respect to the disclosed attributes and $b = 0$ otherwise, and U gets no output.

The anonymous credential must be *correct*, meaning that honest executions of the issuance protocol do not fail, and that honestly obtained credentials can be shown successfully.

The system can also be stateful in which case the organizations each maintain a state during the issuance process.

Security Notations

Although there is no unanimous security model for anonymous credentials, in this thesis we consider a yet popular model by FUCHSBAUER et al. [FHS19]. In their definition, the attributes are all revealed to the signer during the issuance phase (except for the user's secret key which must be part of the signed attributes). Rather than an artifact specific to the construction from [FHS19], this peculiarity stems from the difficulty of formally defining a notion of unforgeability when the signed message is hidden. Regardless, many use cases in practice would have credentials emitted on known attributes, e.g., an electronic passport. The attributes would however be hidden when showing a credential. The anonymous credential systems we design are proven in this model although they additionally offer the feature of hiding attributes during issuance if necessary.

The model from [FHS19] stipulates that the anonymous credentials system following Definition 1.18 must be *anonymous* and *unforgeable*. These two notions require introducing the following variables and oracles.

- HU: Set of user indices of honest users ($\varnothing$ at the outset).
- CU: Set of user indices of corrupt users ($\varnothing$ at the outset).
- ctr: Issuance counter (0 at the outset).
- A: Set of triplets $(j, j', \mathbf{m})$ filled after a successful issuance of credentials for user j on attributes $\mathbf{m}$ and issuance index j' ($\mathcal{O}_{\mathsf{ObtIss}}$ or $\mathcal{O}_{\mathsf{Issue}}$).
- $\mathcal{O}_{\mathsf{HU}}(j)$: Given a user index j, it returns $\perp$ if $j \in \mathsf{HU}\cup\mathsf{CU}$. Otherwise, it samples $(\mathsf{upk}_j, \mathsf{usk}_j) \leftarrow \mathsf{UKeyGen}(\mathsf{pp})$, adds j to HU and returns upk_j.
- $\mathcal{O}_{\mathsf{CU}}(j, \mathsf{upk})$: Given a user index j and optionally a public key upk, it registers a new user with public key upk if $j \notin \mathsf{HU}$. Otherwise, it returns usk_j and sets $\mathsf{HU} \leftarrow \mathsf{HU} \setminus \{j\}$. Either way, it adds j to CU. The former case models the ability to register users with malformed keys, i.e., who do not know the associated secret key.
- $\mathcal{O}_{\mathsf{ObtIss}}(j, \mathbf{m})$: Given some honest user index $j \in \mathsf{HU}$ and attributes $\mathbf{m}$, it runs the protocol $\mathsf{Issue}_{O,U}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}_j, \mathsf{pp}, \mathbf{m}); (\mathsf{usk}_j, \mathsf{upk}_j, \mathsf{opk}, \mathsf{pp}, \mathbf{m}))$ assuming the roles of both O and user j. If successful, it increments the issuance counter ctr, stores the resulting credential and stores $(j, \mathsf{ctr}, \mathbf{m})$ in A. It returns $\top$ if the execution succeeded. If $j \notin \mathsf{HU}$, it simply returns $\perp$.
- $\mathcal{O}_{\mathsf{Obtain}}(j, \mathbf{m})$: Given a user index j and attributes $\mathbf{m}$, it returns $\perp$ if $j \notin \mathsf{HU}$. Otherwise, it runs $\mathsf{Issue}_{\mathcal{A},U}(\cdot, (\mathsf{usk}_j, \mathsf{upk}_j, \mathsf{opk}, \mathsf{pp}, \mathbf{m}))$ with the adversary $\mathcal{A}$ posing as the organization.
- $\mathcal{O}_{\mathsf{Issue}}(j, \mathbf{m})$: Given a user index$j$ and attributes $\mathbf{m}$, it returns $\perp$ if $j \notin \mathsf{CU}$. Else, it runs $\mathsf{Issue}_{O,\mathcal{A}}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}_j, \mathsf{pp}, \mathbf{m}), \cdot)$ with the adversary assuming the role of the user. If successful, it increments the issuance counter ctr, stores the credential and adds $(j, \mathsf{ctr}, \mathbf{m})$ in A.
- $\mathcal{O}_{\mathsf{Show}}(j', \mathbf{m}_{\mathcal{I}}^{(j')})$: It takes an issuance index j' and disclosed attributes $\mathbf{m}_{\mathcal{I}}^{(j')}$. The issuance index corresponds to a successfully issued credential $\mathsf{cred}^{(j')}$ on $\mathbf{m}_{\mathcal{I}}^{(j')}$ for a user j during the j'-th query to $\mathcal{O}_{\mathsf{ObtIss}}$ or $\mathcal{O}_{\mathsf{Obtain}}$. If $j \in \mathsf{HU}$, it runs $\mathsf{Show}_{U,\mathcal{A}}((\mathsf{usk}_j, \mathsf{opk}, \mathsf{pp}, \mathbf{m}_{\mathcal{I}}^{(j')}, \mathsf{cred}^{(j')}, \mathcal{I}), \cdot)$ with the adversary posing as the verifier, and returns $\perp$ if $j \notin \mathsf{HU}$.

Anonymity

The anonymity property captures the fact that a user showing its credential cred obtained on their attributes $\mathbf{m}$ remains anonymous among all users who have the same disclosed attributes $(\mathbf{m}_i)_{i\in\mathcal{I}}$. In means that no one, even the organization, can identify the user running the $\mathsf{Show}_{U,V}$ protocol unless the set of disclosed attributes trivially allows to do so. Thence, no information leaks on the credential nor on the concealed attributes. Additionally, different showings of the same credential with the same revealed attributes should be *unlinkable*. This last property is captured by the anonymity, which is formalized by the game presented in Figure 1.1. The anonymous credentials system is *anonymous* if for all PPT adversary $\mathcal{A}$, its advantage in the anonymity game defined by

$$\left|\mathbb{P}[b^\star = b \wedge \mathcal{O}_{\mathsf{CU}} \text{ was not queried on } j_0 \text{ nor } j_1] - \frac{1}{2}\right|$$

is negligible. To avoid overloading the protocols, we assume that $(\mathsf{opk}, \mathsf{osk})$ is an honestly generated key pair, but this assumption is not necessary if one includes a proof that they know the secret osk linked to opk.

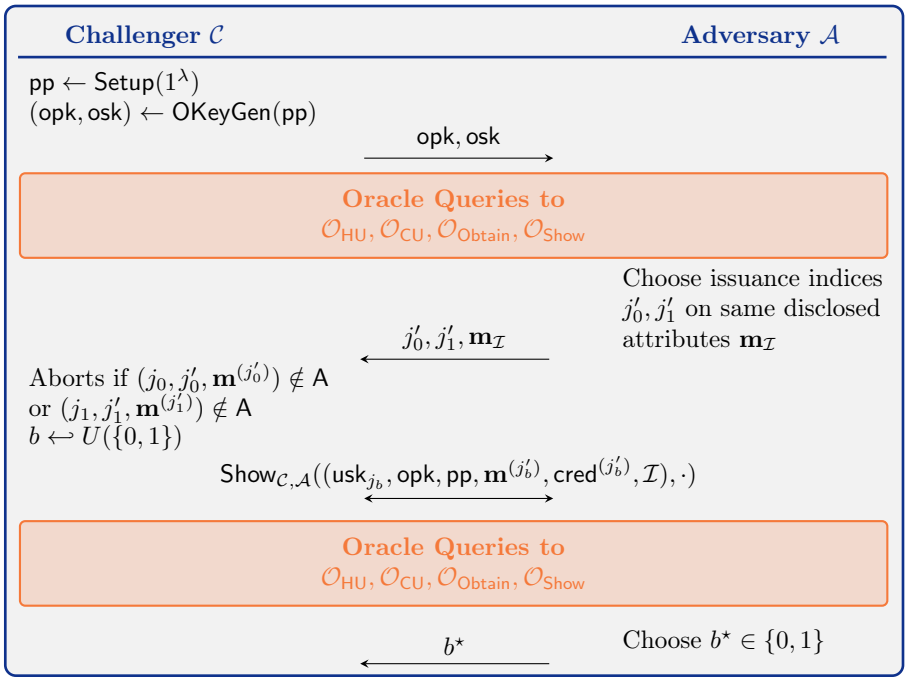


Figure 1.1: Anonymity Game for an Anonymous Credentials System. The index j_α is the user index associated to the issuance index j'_α. The attribute vector $\mathbf{m}^{(j'_\alpha)}$ is the attribute vector used in the j'_α issuance, and must satisfy $\mathbf{m}_\mathcal{I}^{(j'_\alpha)} = \mathbf{m}_\mathcal{I}$.

Unforgeability

The unforgeability property of anonymous credentials ensures that a user cannot show attributes for which it does not own a valid credential. It means that it cannot impersonate an honest user (as it would mean knowing its secret key) which thwarts replay attacks, and that it cannot forge fresh credentials that have not been issued by the Issue protocol. Additionally, malicious users cannot collude and use their legitimate credentials to obtain a new one on a set of attributes that has not been used in a successful issuance. It therefore encompasses forgeries where an adversary would (1) impersonate an honest user, (2) trick the verifier with a falsified proof, and (3) forge a fresh credential, i.e., signature. We formalize it as a game in Figure 1.2. The adversary wins the game if the challenger does not abort and if the challenger's output of the execution of Show is 1. We say that the anonymous credentials system is *unforgeable* if for all PPT adversary $\mathcal{A}$, its probability of winning is negligible.

1.5.3 Random Oracle Model

Despite the efforts of defining security models that both capture real-life threats and make it possible to mathematically prove the security of various schemes, some constructions sometimes

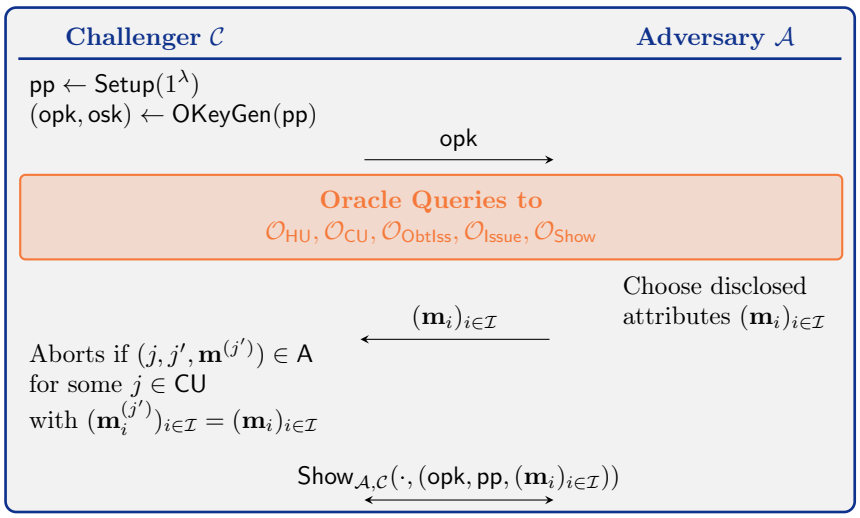


Figure 1.2: Unforgeability Game for the Anonymous Credentials System.

use complex tools that make these proofs harder to carry, if not impossible, from the sole model description. In particular, these complex tools, such as hash functions, are usually employed to make the constructions more practical. To find a middle ground between provable security and efficiency, it is sometimes beneficial to resort to idealized models for proving security. Albeit not a thorough representation of reality, they still allow us to gain confidence in the actual security.

A popular idealized model is called the *Random Oracle Model* (ROM). It assumes the existence of a perfectly random function $\mathcal{H}$ that can be publicly queried as a black-box. Its output is fully uniform in its range, and consistent with previous queries, i.e., if x is queried twice, the random oracle gives the same output. It also enjoys properties that make security proofs easier. We refer to [KL14, Sec. 5.5] for a detailed presentation.

Definition 1.19 (Random Oracle Model)

The Random Oracle Model (ROM) states that there exists a function $\mathcal{H}$ with output space Y and input-output register IOR that can be publicly queried as a black-box verifying the following properties.

- **Regularity.** On a new input x, $\mathcal{H}(x)$ is drawn from $U(Y)$, and IOR is updated to keep track of $(x, \mathcal{H}(x))$.
- **Consistency.** On input x, if $(x, y) \in IOR$ for some y, $\mathcal{H}$ outputs $\mathcal{H}(x) = y$.
- **Extractability.** In a reduction, the challenger can see the queries to $\mathcal{H}$ made by the adversary, and thus learn the inputs x.
- **Programmability.** In a reduction, the challenger can set the output $\mathcal{H}(x)$ or a random oracle query to a value of its choice, as long as this value is uniformly distributed over Y.

Schemes resorting to the random oracle model to prove their security cannot be implemented per se as random oracles are not known to exist. In that case, we *instantiate* the random oracle by an appropriate cryptographic hash function $\mathcal{H}$. The random oracle model thus states that cryptographic hash functions act as random oracles. Constructions that do not resort to random oracles are usually referred to as *standard model* constructions.

Part I

Foundations

The first part of this thesis explores one of the main theoretical hardness assumption of practical lattice-based cryptography, namely the Module Learning With Errors problem. All our subsequent constructions are based directly on some variant of this fundamental problem where the distributions of secret and error are supported over short vectors. We therefore study, from a theoretical perspective, the hardness of such variants.

2

Hardness of Module Learning With Errors with Small Secret

In this chapter, we look into more details at the hardness of Module Learning With Errors (M-LWE) when the secret is uniformly chosen in a small interval, as is frequently the case for practical applications like our signature schemes of Chapter 5 and Section 6.2. We provide a first simple reduction dedicated to the computational version of the problem, before handling the decisional version in a second more involved reduction.

The work presented in this chapter is based on three papers with my co-authors Katharina Boudgoust, Adeline Roux-Langlois and Weiqiang Wen.

[BJRW20] **Towards Classical Hardness of Module-LWE: The Linear Rank Case**. Published at Asiacrypt 2020.

[BJRW21] **On the Hardness of Module-LWE with Binary Secret**. Published at CT-RSA 2021.

[BJRW23] **On the Hardness of Module Learning With Errors with Short Distributions**. Published at IACR Journal of Cryptology 2023.

Contents

2.1 Introduction

The Learning With Errors problem introduced by Regev [Reg05] now represents one of the core security assumptions of lattice-based cryptography. Algebraically structured variants such as the one presented in Section 1.4.2 have been proposed [SSTX09, LPR10, BGV12, LS15] to yield more efficient constructions. These variants still benefit from strong hardness guarantees, namely worst-case to average-case reductions from structured lattice problems. The module version in particular has attracted more and more interest since its introduction, mostly thanks to the fine-grained trade-off it offers between concrete security and efficiency of the corresponding systems. The problem is also extremely versatile in the sense that it allows for constructing a wide variety of cryptographic schemes. As a first example, the recently published post-quantum cryptography

standards chosen by NIST [NISa], namely the signature scheme Dilithium [DKL+18] and the key encapsulation mechanism Kyber [BDK+18], rely on the hardness of M-LWE. All the signature designs presented in this thesis also assume the hardness of M-LWE. However, in order to be efficient, these schemes use parameter settings different from the ones that are covered by the aforementioned reductions. They leverage distributions of secret and error that are supported over sets of short elements. The hardness of M-LWE, and thus the security of the constructions, is then not yet encompassed by theoretical proofs of hardness and is argued based on the state-of-the-art cryptanalysis and attacks using for example the *lattice estimator* [APS15].

The standard formulation of LWE considers a large secret uniform in $\mathbb{Z}_q$ and a Gaussian error, but in practice we tend to consider a short secret, i.e., with coefficients bounded by $\eta \ll q$. This corresponds to choosing the secret $\mathbf{s}$ to be over $\{0, \dots, \eta - 1\}$ (or $\{-\eta, \dots, \eta\}$) instead of $\mathbb{Z}_q$. Typically, efficient designs use η between 1 and 4. Besides gaining in efficiency, choosing a small secret plays an important role in some applications like fully homomorphic encryption [DM15] or modulus switching techniques [BLP+13, AD17, WW19] as it keeps the noise blowup to a minimum. The LWE problem with a uniform bounded secret has been well studied in the binary case (i.e., uniform in $\{0,1\}^d$), but the different approaches easily generalize to slightly larger distributions. A first study of this binary secret variant was provided by Goldwasser et al. [GKPV10] in the context of leakage-resilient cryptography. Although their proof structure has the advantage of being easy to follow, their result suffers from a large error increase. Informally, they show a reduction from $\mathrm{LWE}_{k,m,q,U(\mathbb{Z}_q),D_r}$ to $\mathrm{LWE}_{d,m,q,U(\{0,1\}),D_s}$, where $s/r = d^{\omega(1)}$ (super-polynomial) and $d \geq k \log_2 q + \omega(\log_2 d)$. It was later improved by Brakerski et al. [BLP+13] and Micciancio [Mic18] using more technical proofs. Both of them achieve a similar dimension increase between k and d, but only increase the error by roughly $s/r = \Omega(\sqrt{d})$ (polynomial). The dimension increase from k to roughly $k \log_2 q$ is natural as it essentially preserves the entropy of the secret distribution. A line of work initiated by Brakerski and Döttling [BD20] extended the hardness results to more general secret distributions based on entropic arguments. The question of whether these hardness results carry over to structured variants, and in particular to the module case, was left open, even though they serve as hardness assumptions for most efficient M-LWE-based schemes.

2.1.1 Our Contributions

In this chapter, we provide the first two results on the hardness of M-LWE with a uniform centered η-bounded secret, i.e., with secret drawn from $U(S_\eta^d)$. More precisely, we prove a first reduction from the standard formulation of M-LWE with a secret uniform in R_q to the search variant with secret distribution $U(S_\eta)$. Then, we give a more involved reduction from the standard formulation to the decision variant with secret distribution $U(S_\eta)$. They are generalizations of the results published in our previous conference papers [BJRW20] and [BJRW21] respectively, only dealing with the special case of secret coefficients in $\{0,1\}$, which is already mentioned in one of the author's thesis [Bou21]. We also mention that another result based on entropic arguments has been published in [BJRW22] but is not presented in this thesis. As opposed to [BJRW20, BJRW21, Bou21], we decide to work in the primal ring R and with a centered representation of the secret with coefficients in $\{-\eta, \dots, \eta\}$[1] to match practical uses of the M-LWE assumption like [BDK+18, DKL+18] and that of Parts II and III. The results apply to all cyclotomic fields, but most of the intermediate results are proven in more general number fields.

We warn the reader that all the error distributions in this chapter are Gaussians with respect to the Minkowski embedding σ_H, and can be discrete or continuous. We refer to Section 1.3.2 for more details.

Contribution 1: Computational hardness

We show a first reduction in Section 2.2 for the hardness of the search version sM-LWE with a secret in S_η^d. The formal statement can be found in Theorem 2.1. It follows the original proof structure of Goldwasser et al. [GKPV10] in the case of LWE, while achieving a much better noise parameter by using the Rényi divergence instead of the statistical distance to measure the distance between two distributions. The improvement on the noise rate compared to [GKPV10] stems from the fact that the Rényi divergence only needs to be constant for the reduction to work, and not necessarily

[1] Setting $\eta = 1$ gives ternary secrets instead of binary. We however observe that the parameters covered by the reductions for $\eta = 1$ in the centered representation match those of [BJRW20, BJRW21], and it has the upside of a larger secret space. This leads to smaller ranks d by a factor of $\log_2 3$.

negligibly close to 1 (compared to negligibly close to 0 for the statistical distance). More precisely, as we use the leftover hash lemma (Lemma 1.16) with respect to the Rényi divergence, we can have a rank that is logarithmic in the ring degree n, instead of super-logarithmic. However, using the Rényi divergence as a measure of distribution closeness only allows us to prove the hardness of the *search* variant. Additionally, its use asks to fix the number of samples a priori.

The result consists in a reduction from $\text{sM-LWE}_{n,k,m,q,U(R_q),\mathcal{D}_{R,s}}$ and $\text{M-LWE}_{n,k,m,q,U(R_q),\mathcal{D}_{R,r}}$ with rank k and Gaussian error width r and a secret uniform in R_q^k to $\text{sM-LWE}_{n,d,m,q,U(S_\eta),\mathcal{D}_{R,s}}$ with rank d and Gaussian error width s. The reduction preserves the ring degree n, the number of samples m and the modulus q, where q only needs to be prime. The ranks must satisfy $d \geq k \log_{2\eta+1} q + \log_{2\eta+1} \Omega(n)$, which is due to the use of the leftover hash lemma over rings. The Gaussian noise parameter r is also increased to s by a factor $s/r = \eta \cdot n^{3/2}(\sqrt{m} + \sqrt{d} + \log_2 \lambda)\sqrt{d}$ in general cyclotomic fields, which can be further improved by a factor of $\sqrt{n}$ in the specific case of power-of-two cyclotomic fields.

Contribution 2: Pseudorandomness

We then provide a more involved proof of hardness for the *decision* version in Section 2.3 through a reduction from $\text{M-LWE}_{n,k,m,q,U(R_q),D_r}$ to $\text{M-LWE}_{n,d,m,q,U(S_\eta),\Psi_{\leq s}}$. The thorough statement is provided in Theorem 2.2. Although the noise rate s/r is slightly larger than that of Contribution 1, it no longer depends on the number of samples m which can be preferable in some cases. Also, because there is no search-to-decision reductions for short secret distributions, being able to prove the hardness of the decision variant directly is meaningful regardless. The technique follows the idea of [BLP+13] by introducing the two intermediate problems first-is-errorless M-LWE and ext-M-LWE. We start by reducing the standard M-LWE problem to the first-is-errorless M-LWE variant, where the first sample is not perturbed by an error. We then reduce the latter to ext-M-LWE, which can be seen as M-LWE with an extra information on the error vector $\mathbf{e}$ given by $\langle \mathbf{e}, \mathbf{z} \rangle$ for a uniformly chosen $\mathbf{z}$ in S_η. Three other formulations of ext-M-LWE were proposed by Alperin-Sheriff and Apon [AA16], Lyubashevsky et al. [LNS21] and very recently by Kim et al. [KLSS23], but none of them suits our reduction due to our lossy argument in Lemma 2.5. We discuss further these differences in Section 2.3.2. Then, to reduce ext-M-LWE to M-LWE with a short secret, we use a lossy argument similar to that of Contribution 1 but now relying on the newly derived ext-M-LWE hardness assumption, as well as the leftover hash lemma.

The main challenge is the use of matrices composed of ring elements. The proof in [BLP+13, Lem. 4.7] requires the construction of unimodular matrices which is not straightforward to adapt in the module setting because of invertibility issues. The construction in Lemma 2.2 relies on units of the quotient ring R/qR, which are much harder to explicitly describe than the units of $\mathbb{Z}/q\mathbb{Z}$ in the sense that we do not have practical closed-form expressions. This is the reason why we need to control the splitting structure of the cyclotomic polynomial modulo q. Lemma 1.4 [LS18, Thm. 1.1] solves this issue but requires q to satisfy certain number-theoretic properties and to be sufficiently large so that all the non-zero small norm ring elements are units of R_q.

In the whole reduction, the ring degree n, number of samples m and modulus q are preserved, where m needs to be larger than d and q needs to be a prime satisfying the said number-theoretic properties. With the help of the modulus-switching technique of Langlois and Stehlé [LS15, Thm 4.8], we can then relax the restriction on the modulus q to be any polynomially large modulus, at the expense of a loss in the Gaussian noise parameter. As we again rely on the leftover hash lemma, we obtain a rank condition that is similar to that of Contribution 1. However, we work with decision variants which requires us to use it in the statistical distance rather than the Rényi divergence. As a result, the ranks must satisfy $d \geq (k+1) \log_{2\eta+1} q + \omega(\log_{2\eta+1} n)$, where the asymptotic term is now super-logarithmic. The noise rate is now given by $n\eta\sqrt{2d}\sqrt{4n^2\eta^2+1} = \Theta(\eta^2 n^2 \sqrt{d})$ for cyclotomic fields. In the special case of $\eta = 1$ and $n = 1$, we recover the same noise-ratio $\Theta(\sqrt{d})$ as in the original LWE result from Brakerski et al. [BLP+13].

Throughout this chapter, we use for simplicity the notations $\text{M-LWE}_{\mathscr{D}_s}$ and $\text{sM-LWE}_{\mathscr{D}_s}$ to denote the M-LWE problem with secret distribution $\mathscr{D}_s$, all other parameters being implicit.

2.2 Computational Hardness

We start by proving the hardness of sM-LWE with a short secret with a quite direct reduction. To facilitate the understanding, we illustrate the high level idea of the proof in Figure 2.1. Given an algorithm for instance $(\mathbf{A}, \mathbf{A}\mathbf{z} + \mathbf{e})$ of $\text{sM-LWE}_{U(S_\eta)}$, our goal is to transform it into an algorithm for a related instance of $\text{sM-LWE}_{U(R_q)}$ defined by $(\mathbf{B}, \mathbf{B}\mathbf{s} + \mathbf{e}')$. Note that the secret $\mathbf{z}$ is in S_η^d,

while the secret $\mathbf{s}$ is in R_q^k. At the core of the proof lies a lossy argument, where the public matrix $\mathbf{A}$ is replaced by a lossy matrix $\mathbf{BC} + \mathbf{N}$, which corresponds to the second part of some multiple-secrets M-LWE sample. Note that the rank of the matrix $\mathbf{B}$ is smaller than the one of $\mathbf{A}$, motivating the description *lossy*. Here, we can see that this argument does not work for R-LWE (which corresponds to M-LWE with rank $d = 1$) as it is not possible to replace the public matrix consisting of one column by a matrix of smaller rank. To argue that an adversary cannot distinguish between the two cases, we need to assume the hardness of the *decision* M-LWE$_{U(R_q)}$ problem as well. In a second step, the term $\mathbf{Nz} + \mathbf{e}$ is replaced by the new noise $\mathbf{e}'$, where the Rényi divergence between both expressions can be bounded by a constant using properties of the Rényi divergence of Gaussian distributions, i.e., Lemma 1.11. Finally, the product $\mathbf{Cz}$ is replaced by the uniform secret $\mathbf{s}$, where the Rényi divergence between both elements can be bounded by a constant using Lemma 1.16. The use of the leftover hash lemma is also the reason why our reduction only works for module ranks larger than $\log_{2\eta+1} q + \log_{2\eta+1} \Omega(n)$. Informally speaking, it requires the ratio between the number of rows of $\mathbf{C}$ and its number of columns to be logarithmic in order to bound the Rényi divergence by a constant. We end up with some standard M-LWE instance, which is hard to solve due to our hardness assumption.

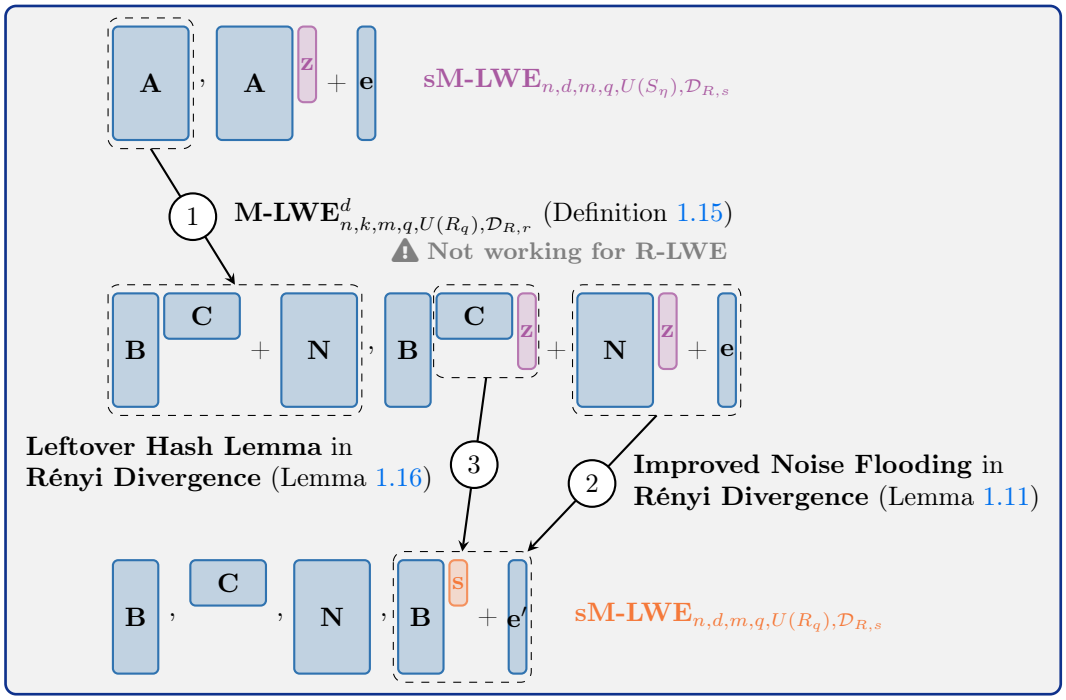


Figure 2.1: Summary of the proof of Theorem 2.1

Overall, this reduction is restricted to monogenic number fields. Furthermore, the norm of the Vandermonde matrix $\|\mathbf{V}\|_2$ is better understood in cyclotomic fields. Note that in this section, all the variants of M-LWE involve discrete Gaussian error supported over R and not $K_{\mathbb{R}}$.

Theorem 2.1 (Computational Hardness of M-LWE with Short Secret)

Let $\lambda, n, k, d, m, \eta, q$ be positive integers, and $t, t' > 0$ arbitrary reals. Let K be a monogenic number field of degree n and R its ring of integers. We assume that q is prime, that m and d are polynomial in λ, and that $d \geq k \cdot \log_{2\eta+1} q + \log_{2\eta+1} t'n$. Further, let r and s be positive reals such that $s \geq r \cdot \eta \|\mathbf{V}\|_2 (\sqrt{m} + \sqrt{d} + t)\sqrt{nd}$, and $s \geq \eta_\varepsilon(R^m)$ for some $\varepsilon \in (0, 1/2)$. There is a PPT reduction from sM-LWE$_{n,k,m,q,U(R_q),\mathcal{D}_{R,s}}$ and M-LWE$^d_{n,k,m,q,U(R_q),\mathcal{D}_{R,r}}$ to sM-LWE$_{n,d,m,q,U(S_\eta),\mathcal{D}_{R,s}}$. More precisely, if $\varepsilon_{\text{sM-LWE}}, \varepsilon_{\text{M-LWE}}$ are the hardness bounds of the formers and $\varepsilon_{\text{sM-LWE},\eta}$ that of the latter, it holds that

$$\varepsilon_{\text{sM-LWE},\eta} \leq \frac{1+\varepsilon}{1-\varepsilon} \exp\left(\frac{1}{2} + \frac{1}{4t'}\right) \cdot \varepsilon_{\text{sM-LWE}}^{1/4} + d \cdot \varepsilon_{\text{M-LWE}} + 2ne^{-\pi t^2}$$

The degree n of K, the number of samples m and the modulus q are preserved. The reduction increases the rank of the module from k to $k \log_{2\eta+1} q + \log_{2\eta+1} \Omega(n)$ and the Gaussian width

from r to roughly $r \cdot \eta\|\mathbf{V}\|_2 (\sqrt{m} + \sqrt{d} + \log_2 \lambda)\sqrt{nd}$. In power-of-two cyclotomic fields, $\|\mathbf{V}\|_2 = \sqrt{n}$, while in the p^k-th cyclotomic field with p an odd prime, we have $\|\mathbf{V}\|_2 = \sqrt{p^k}$. In general cyclotomic fields, we have $\|\mathbf{V}\|_2 \leq \|\mathbf{V}\|_F = (\sum_{i,j} |\alpha_i^{j-1}|^2)^{1/2} \leq n$ (as α_i is a root of unity). Also, M-LWE$_{n,k,m,q,U(R_q),\mathcal{D}_{R,r}}$ trivially reduces to sM-LWE$_{n,k,m,q,U(R_q),\mathcal{D}_{R,s}}$, as $t = \sqrt{s^2 - r^2}$ is above $\eta_\varepsilon(R)$ for a negligible ε, and sufficiently large so that $\mathcal{D}_{R,t}$ is efficiently sampleable. As a result, the limiting assumption in the reduction is the decision variant with the Gaussian error width r.

Proof (Theorem 2.1). Fix any $\lambda, n, k, d, m, \eta, q, r, s$ and ε as in the statement of the theorem. Given an instance $(\mathbf{A}, \mathbf{Az} + \mathbf{e} \bmod qR) \in R_q^{m\times d} \times R_q^m$, with $\mathbf{z} \hookleftarrow U(S_\eta^d)$ and $\mathbf{e} \hookleftarrow \mathcal{D}_{R^m,s}$, the sM-LWE$_{n,d,m,q,U(S_\eta),\mathcal{D}_{R,s}}$ asks to find $\mathbf{z}$ and $\mathbf{e}$. In order to prove the statement, we define different hybrid distributions:

$\mathcal{H}_1$ Sample $(\mathbf{A}, \mathbf{b} = \mathbf{Az} + \mathbf{e} \bmod qR)$ as in the sM-LWE$_{n,d,m,q,U(S_\eta),\mathcal{D}_{R,s}}$ problem.
Output: $(\mathbf{A}, \mathbf{b})$.

$\mathcal{H}_2$ Sample $\mathbf{B} \hookleftarrow U(R_q^{m\times k}), \mathbf{C} \hookleftarrow U(R_q^{k\times d})$, $\mathbf{N} \hookleftarrow \mathcal{D}_{R^{m\times d},r}$ and $\mathbf{z}, \mathbf{e}$ as in $\mathcal{H}_1$.
Output: $(\mathbf{A}' = \mathbf{BC} + \mathbf{N} \bmod qR, \mathbf{A}'\mathbf{z} + \mathbf{e} \bmod qR)$.

$\mathcal{H}_3$ Sample $\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{z}, \mathbf{e}$ as in $\mathcal{H}_2$.
Output: $(\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{BCz} + \mathbf{Nz} + \mathbf{e} \bmod qR)$.

$\mathcal{H}_4$ Sample $\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{z}, \mathbf{e}$ as in $\mathcal{H}_3$, until $\left\|M_{\sigma_H}(\mathbf{N})\right\|_2 > \frac{r}{\sqrt{2\pi}}(\sqrt{m} + \sqrt{d} + t)$.
Output: $(\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{BCz} + \mathbf{Nz} + \mathbf{e} \bmod qR)$.

$\mathcal{H}_5$ Sample $\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{z}$ as in $\mathcal{H}_4$ until $\left\|M_{\sigma_H}(\mathbf{N})\right\|_2 > \frac{r}{\sqrt{2\pi}}(\sqrt{m} + \sqrt{d} + t)$, and $\mathbf{e}' \hookleftarrow \mathcal{D}_{R^m,s}$.
Output: $(\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{BCz} + \mathbf{e}' \bmod qR)$.

$\mathcal{H}_6$ Sample $\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{e}'$ as in $\mathcal{H}_5$ until $\left\|M_{\sigma_H}(\mathbf{N})\right\|_2 > \frac{r}{\sqrt{2\pi}}(\sqrt{m} + \sqrt{d} + t)$, and $\mathbf{s} \hookleftarrow U(R_q^k)$.
Output: $(\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{Bs} + \mathbf{e}' \bmod qR)$.

For $i \in \{1, \ldots, 6\}$, we denote by P_i the problem of finding the secret $\mathbf{z}$ (resp. $\mathbf{s}$ in $\mathcal{H}_6$), given a sample of the distribution $\mathcal{H}_i$. Recall that problem P_i is hard if for any PPT attacker $\mathcal{A}$ the advantage of solving P_i is negligible in λ, thus $\mathrm{Adv}_{P_i}[\mathcal{A}] = \mathbb{P}_{X\sim\mathcal{H}_i}[\mathcal{A}(X) = \mathbf{z}] \leq \mathsf{negl}(\lambda)$, where λ is the security parameter. The overall idea is to show that if P_6 is hard, then P_1 is hard as well.

From P_1 to P_2: Assuming the hardness of M-LWE$^d_{n,k,m,q,U(R_q),\mathcal{D}_{R,r}}$, the distributions $\mathcal{H}_1$ and $\mathcal{H}_2$ are computationally indistinguishable. We recall that the hardness of the latter can be obtained by a hybrid argument, e.g., Lemma 2.4, from that of M-LWE$_{n,k,m,q,U(R_q),\mathcal{D}_{R,r}}$ with a reduction loss factor of d in the advantage. It then holds that

$$\mathrm{Adv}_{P_1}[\mathcal{A}] \leq \mathrm{Adv}_{P_2}[\mathcal{A}] + d \cdot \varepsilon_{\text{M-LWE}},$$

where d is the number of secret vectors, i.e., the columns of the matrix $\mathbf{C}$, and $\varepsilon_{\text{M-LWE}}$ is the hardness bound of M-LWE$_{n,k,m,q,U(R_q),\mathcal{D}_{R,r}}$.

From P_2 to P_3: Since more information is given in distribution $\mathcal{H}_3$ than in distribution $\mathcal{H}_2$, the problem P_2 is harder than P_3. From P_3 onwards the adversary is given more elements (namely $\mathbf{B}, \mathbf{C}, \mathbf{N}$ instead of $\mathbf{A}'$) but can simply reconstruct the M-LWE matrix from these elements. Hence, we have

$$\mathrm{Adv}_{P_2}[\mathcal{A}] \leq \mathrm{Adv}_{P_3}[\mathcal{A}].$$

From P_3 to P_4: Note that conditioned on $\left\|M_{\sigma_H}(\mathbf{N})\right\|_2 \leq \frac{r}{\sqrt{2\pi}}(\sqrt{m} + \sqrt{d} + t)$, the two distributions are identical. Yet, Lemma 1.22 yields the spectral bound with overwhelming probability. We insist again that the Gaussian $\mathbf{N}$ is drawn with respect to σ_H as expected by the lemma. Combined with Lemma 1.7, we have $\Delta(\mathcal{H}_3, \mathcal{H}_4) \leq \mathbb{P}[\left\|M_{\sigma_H}(\mathbf{N})\right\|_2 > \frac{r}{\sqrt{2\pi}}(\sqrt{m} + \sqrt{d} + t)] \leq 2ne^{-\pi t^2}$, resulting in

$$\mathrm{Adv}_{P_3}[\mathcal{A}] \leq \mathrm{Adv}_{P_4}[\mathcal{A}] + 2ne^{-\pi t^2}$$

As t is arbitrary, one can adjust it to make this probability negligible, by choosing $t = \log_2 \lambda$ for example.

From P_4 to P_5: By the probability preservation property of the Rényi divergence (Lemma 1.8), we have

$$\mathrm{Adv}_{P_4}[\mathcal{A}]^2 \leq \mathrm{Adv}_{P_5}[\mathcal{A}] \cdot \mathrm{RD}_2(\mathcal{H}_4 \| \mathcal{H}_5).$$

We first explain how to bound the Rényi divergence between $\mathbf{Nz}+\mathbf{e}$ and $\mathbf{e}'$ for a fixed $(\mathbf{N}, \mathbf{z})$. First note that $\mathbf{Nz}+\mathbf{e}$ follows the distribution $\mathcal{D}_{R^m+\mathbf{Nz},\mathbf{Nz},s}$. Since we have $\mathbf{Nz} \in R^m$, this distribution is exactly $\mathcal{D}_{R^m,\mathbf{Nz},s}$. Then, as σ and σ_H only differ by the unitary transformation $\mathbf{U}_H$, we have that $\left\|\sigma_H(\mathbf{z})\right\|_2 = \left\|\sigma(\mathbf{z})\right\|_2 \leq \|\mathbf{V}\|_2 \left\|\tau(\mathbf{z})\right\|_2 \leq \|\mathbf{V}\|_2 \cdot \eta\sqrt{nd}$, as $\mathbf{z} \in S_\eta^d$. Finally, because of our conditioning, we have $\left\|M_{\sigma_H}(\mathbf{N})\right\|_2 \leq \frac{r}{\sqrt{2\pi}}(\sqrt{m}+\sqrt{d}+t)$. It then holds that $\left\|\sigma_H(\mathbf{Nz})\right\|_2 = \left\|M_{\sigma_H}(\mathbf{N})\sigma_H(\mathbf{z})\right\|_2 \leq \left\|M_{\sigma_H}(\mathbf{N})\right\|_2 \left\|\sigma_H(\mathbf{z})\right\|_2 \leq \frac{r}{\sqrt{2\pi}}(\sqrt{m}+\sqrt{d}+t)\|\mathbf{V}\|_2\,\eta\sqrt{nd}$. Then, using that $s \geq \eta_\varepsilon(R^m)$, Lemma 1.11 yields

$$\mathrm{RD}_2\left(\mathcal{D}_{R^m,\mathbf{Nz},s} \| \mathcal{D}_{R^m,s}\right) \leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 \cdot \exp\left(\frac{2\pi\left\|\sigma_H(\mathbf{Nz})\right\|_2^2}{s^2}\right).$$

However, it holds that $\exp(2\pi\left\|\sigma_H(\mathbf{N}\widetilde{\mathbf{z}})\right\|_2^2/s^2) \leq e^1$ because of how we chose s with respect to r. Without loss of generality, we assume $\varepsilon < \frac{1}{2}$ resulting in $\mathrm{RD}_2(\mathcal{D}_{R^m,\mathbf{Nz},s}\|\mathcal{D}_{R^m,s}) = O(1)$.

Next, the data processing inequality of Lemma 1.8 gives $\mathrm{RD}_2(\mathcal{H}_4\|\mathcal{H}_5) \leq \mathrm{RD}_2((\mathbf{N},\mathbf{z},\mathbf{e}+\mathbf{Nz})\|(\mathbf{N},\mathbf{z},\mathbf{e}'))$. We now bound this divergence by a constant using the previous calculation.

$$\begin{aligned}
\mathrm{RD}_2((\mathbf{N},\mathbf{z},\mathbf{e}+\mathbf{Nz})\|(\mathbf{N},\mathbf{z},\mathbf{e}')) &= \sum_{(\overline{\mathbf{N}},\overline{\mathbf{z}},\overline{\mathbf{e}})} \frac{\mathbb{P}[(\mathbf{N},\mathbf{z},\mathbf{e}+\mathbf{Nz}) = (\overline{\mathbf{N}},\overline{\mathbf{z}},\overline{\mathbf{e}})]^2}{\mathbb{P}[(\mathbf{N},\mathbf{z},\mathbf{e}') = (\overline{\mathbf{N}},\overline{\mathbf{z}},\overline{\mathbf{e}})]} \\
&= \sum_{(\overline{\mathbf{N}},\overline{\mathbf{z}},\overline{\mathbf{e}})} \frac{\mathbb{P}[(\mathbf{N},\mathbf{z}) = (\overline{\mathbf{N}},\overline{\mathbf{z}})]^2 \mathbb{P}[\mathbf{e}+\overline{\mathbf{N}}\overline{\mathbf{z}} = \overline{\mathbf{e}}]^2}{\mathbb{P}[(\mathbf{N},\mathbf{z}) = (\overline{\mathbf{N}},\overline{\mathbf{z}})]\mathbb{P}[\mathbf{e}' = \overline{\mathbf{e}}]} \\
&= \sum_{(\overline{\mathbf{N}},\overline{\mathbf{z}})} \mathbb{P}[(\mathbf{N},\mathbf{z}) = (\overline{\mathbf{N}},\overline{\mathbf{z}})]\mathrm{RD}_2(\mathbf{e}+\overline{\mathbf{N}}\overline{\mathbf{z}}\|\mathbf{e}') \\
&\leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 \cdot e \sum_{(\overline{\mathbf{N}},\overline{\mathbf{z}})} \mathbb{P}[(\mathbf{N},\mathbf{z}) = (\overline{\mathbf{N}},\overline{\mathbf{z}})] \\
&= \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2 \cdot e \\
&< 9e,
\end{aligned}$$

which is constant as desired.

From P_5 to P_6: At this stage, we use Lemma 1.16 to argue the switch from $\mathcal{H}_5$ to $\mathcal{H}_6$. As we deal with search problems, we use its formulation in the Rényi divergence which gives tighter parameters, albeit at the expense of a larger security loss. We get

$$\begin{aligned}
\mathrm{Adv}_{P_5}[\mathcal{A}]^2 &= \mathbb{P}_{(\mathbf{B},\mathbf{C},\mathbf{N},\mathbf{b})\sim\mathcal{H}_5}[\mathcal{A}(\mathbf{B},\mathbf{C},\mathbf{N},\mathbf{b}) = \mathbf{z}]^2 \\
&\leq \mathbb{P}_{(\mathbf{B},\mathbf{C},\mathbf{N},\mathbf{b})\sim\mathcal{H}_5}[\mathcal{A}(\mathbf{B},\mathbf{C},\mathbf{N},\mathbf{b}) = \mathbf{Cz}]^2 \\
&\leq \mathbb{P}_{(\mathbf{B},\mathbf{C},\mathbf{N},\mathbf{b})\sim\mathcal{H}_6}[\mathcal{A}(\mathbf{B},\mathbf{C},\mathbf{N},\mathbf{b}) = \mathbf{s}] \cdot \mathrm{RD}_2(\mathcal{H}_5\|\mathcal{H}_6) \\
&\leq \mathrm{Adv}_{P_6}[\mathcal{A}] \cdot \mathrm{RD}_2((\mathbf{C},\mathbf{Cz})\|(\mathbf{C},\mathbf{s})).
\end{aligned}$$

The first inequality follows from the fact that if $\mathcal{A}$ can find $\mathbf{z}$ from $(\mathbf{B},\mathbf{C},\mathbf{N},\mathbf{b})$, then they can also find $\mathbf{Cz}$, hence the inclusion of events. The second and third inequalities come from the probability preservation and data processing inequality of Lemma 1.8 respectively. By the leftover hash lemma stated in Lemma 1.16, the Rényi divergence between the distribution $(\mathbf{C},\mathbf{Cz})$ and the distribution $(\mathbf{C},\mathbf{s})$ is bounded above by $(1+q^k/(2\eta+1)^d)^n$. As we require $d \geq k\log_{2\eta+1} q + \log_{2\eta+1} t'n$, we obtain $\mathrm{RD}_2(\mathcal{H}_5\|\mathcal{H}_6) \leq (1+1/t'n)^n \leq e^{1/t'}$. As t' is an arbitrary constant, one can tweak t' to trade better parameters for a larger loss and vice-versa. t' must however be a constant asymptotically in λ.

Problem P_6: This problem is exactly the sM-LWE$_{n,k,m,q,U(R_q),\mathcal{D}_{R,s}}$ problem, as $\mathbf{C}$ and $\mathbf{N}$ are independent of $\mathbf{B}, \mathbf{s}$ and $\mathbf{e}'$. It thus holds that

$$\mathrm{Adv}_{P_5}[\mathcal{A}] \leq \varepsilon_{\text{sM-LWE}}$$

where $\varepsilon_{\text{sM-LWE}}$ is the hardness bound of sM-LWE$_{n,k,m,q,U(R_q),\mathcal{D}_{R,s}}$. Putting all equations from above together, we obtain

$$\begin{aligned}
\mathrm{Adv}_{P_1}[\mathcal{A}] &\leq \mathrm{Adv}_{P_2}[\mathcal{A}] + d \cdot \varepsilon_{\text{M-LWE}} \\
&\leq \mathrm{Adv}_{P_3}[\mathcal{A}] + d \cdot \varepsilon_{\text{M-LWE}} \\
&\leq \mathrm{Adv}_{P_4}[\mathcal{A}] + 2ne^{-\pi \log_2^2 \lambda} + d \cdot \varepsilon_{\text{M-LWE}} \\
&\leq \sqrt{\mathrm{Adv}_{P_5}[\mathcal{A}] \cdot \mathrm{RD}_2(\mathcal{H}_4 \| \mathcal{H}_5)} + 2ne^{-\pi t^2} + d \cdot \varepsilon_{\text{M-LWE}} \\
&\leq \sqrt{\sqrt{\varepsilon_{\text{sM-LWE}} \cdot \mathrm{RD}_2(\mathcal{H}_5 \| \mathcal{H}_6)} \cdot \mathrm{RD}_2(\mathcal{H}_4 \| \mathcal{H}_5)} + 2ne^{-\pi t^2} + d \cdot \varepsilon_{\text{M-LWE}} \\
&\leq \frac{1+\varepsilon}{1-\varepsilon} e^{\frac{1}{2}+\frac{1}{4t'}} \cdot \varepsilon_{\text{sM-LWE}}^{1/4} + d \cdot \varepsilon_{\text{M-LWE}} + 2ne^{-\pi t^2}
\end{aligned}$$

By carefully choosing t and t' and because our base assumptions give $\varepsilon_{\text{M-LWE}}, \varepsilon_{\text{sM-LWE}} \leq \mathsf{negl}(\lambda)$, it proves that $\mathrm{Adv}_{P_1}[\mathcal{A}(H_0) = \mathbf{z}] \leq \mathsf{negl}(\lambda)$. Maximizing over $\mathcal{A}$ gives $\varepsilon_{\text{sM-LWE},\eta} \leq \mathsf{negl}(\lambda)$, where $\varepsilon_{\text{sM-LWE},\eta}$ is the hardness bound of sM-LWE$_{n,d,m,q,U(S_\eta),\mathcal{D}_{R,s}}$.

As explained in the introduction of this manuscript, an attractive feature of lattice-based cryptography is that one can relate the hardness of average-case problems such as M-LWE to that of worst-case lattice problems. We can thus plug our reduction to the existing ones that reduce said lattice problems to M-LWE$_{n,k,m,q,U(R_q),\mathcal{D}_{R,r}}$. Typically, we can use the quantum reduction from [LS15, Thm. 4.7] in combination with [BJRW20, Lem. 13] to move to a discrete error distribution. This however reduces to the search version of M-LWE. To avoid a search-to-decision reduction, we directly use the classical worst-case to average-case reduction provided in our paper [BJRW20, Thm. 4]. Although it works for q exponential in nk, we obtain the pseudorandomness directly. We then plug our reduction to small uniform secret, and then the modulus switching reduction from [AD17][BJRW20, Cor. 1] to reach a polynomial modulus. Concretely, we get that if the (gap) decisional variant of SVP$_\gamma$ is hard over module lattices of rank k with approximation factor γ, then sM-LWE$_{n,d,m,q,U(S_\eta),\mathcal{D}_{R,s}}$ is hard with[2]

$$\frac{s}{q} = \omega(\log_2 n) \cdot \sqrt{2\left(\frac{nk\sqrt{k}}{\gamma} \cdot \omega(\log_2 n) \cdot \eta \|\mathbf{V}\|_2 (\sqrt{m}+\sqrt{d}+t)\sqrt{nd}\right)^2 + \boldsymbol{\Delta}},$$

where the factor highlighted in purple originates from Theorem 2.1. The additive term $\boldsymbol{\Delta}$ stems from the modulus switching reduction, and is expressed as $\boldsymbol{\Delta} = 2\alpha^2 B_\eta^2 d$ with α depending on the polynomial modulus q and the ring R. For example, in power-of-two cyclotomic fields, we can use $\alpha = \sqrt{2n}\eta_\varepsilon(\mathbb{Z}^{nd})/q$. In that case, it yields

$$s \approx \omega(\log_2^2 n) \cdot n^2 \eta \sqrt{2d} \sqrt{\frac{q^2 k^3 (\sqrt{m}+\sqrt{d})^2}{\gamma^2} + 2}$$

This means the underlying lattice assumption must be secure for an approximation factor

$$\gamma \approx \frac{k^{3/2}(\sqrt{m}+\sqrt{d}) \cdot n^2 \eta \sqrt{2d}}{s/q} \cdot \omega(\log_2^2 n).$$

2.3 Pseudorandomness

We now provide a more involved proof of hardness for the *decision* version M-LWE with a short uniform secret. The latter is paramount in the security guarantees of M-LWE-based cryptosystems, including the ones presented in this thesis, as it essentially allows to argue that the public key $\mathbf{b} = \mathbf{As} + \mathbf{e}$ does not provide any information on the secret key $(\mathbf{s}, \mathbf{e})$. Regardless of the specific parameter constraints, the result of Section 2.2 is insufficient in that regard. Theoretically proving this statement is however more complex due to the stronger assurance it provides.

[2] We note that we use the relative error width s/q as this is the one considered in the aforementioned reductions.

Our proof follows the same idea as in [BLP+13] that we extend to modules. More precisely, we show a reduction from M-LWE$_{U(R_q)}$ with rank k to M-LWE$_{U(S_\eta)}$ with rank d satisfying $d \geq (k+1)\log_{2\eta+1} q + \omega(\log_{2\eta+1} n)$. The reduction preserves the modulus q, that needs to be a prime satisfying number-theoretic restrictions, the ring degree n and the number of samples m, but the noise is increased by a factor $n\eta\sqrt{2d}\sqrt{4n^2\eta^2+1} \approx \eta^2 n^2\sqrt{8d}$. For the reduction, m also needs to be larger than the target module rank d, and at most polynomial in n because of the hybrid argument used in Lemma 2.4. As mentioned in Section 2.1.1, the reduction requires the explicit construction of a unimodular matrix in R_q. The one we propose in Lemma 2.2 is so far restricted to cyclotomic fields, but we note that all the other steps necessary to prove Theorem 2.2 work in all monogenic fields.

Observe that the noise rate is slightly worse than the one obtained in Theorem 2.1, albeit still a small polynomial, and the rank condition is also stronger. Nonetheless, this reduction allows for proving the hardness of the decision version of M-LWE$_{U(S_\eta)}$ which is needed for the security of cryptographic applications like the ones we present in Part II and III. An alternative to get tighter reductions for the decision variant would be to use a search-to-decision reduction such as the one from [LPR10, LS15]. Unfortunately, no such reduction exists for short secret distributions as they all, at some point, need to re-randomize the secret in R_q. This is why we provide a direct reduction to the decision problem, which is so far the only known method.

Let us give an overview of the full reduction in Figure 2.2. The main idea of this reduction is to avoid the noise flooding technique. Indeed, we were able to use such a technique in the reduction of Section 2.2 by using the Rényi divergence for tighter parameter constraints, with the caveat that it only tackles the search variant. For the decision variant, one would need to either use the statistical distance leading to a super-polynomial modulus [GKPV10], or satisfy the public sampleability property [BLR+18, Sec. 4.1] which is not our case. To bypass the noise flooding, we instead argue the hardness under an assumption called *Extended Module Learning With Errors* (ext-M-LWE) which gives additional hints on the error vector/matrix. The majority of the reduction then consists in linking this ext-M-LWE assumption to the original M-LWE assumption. To do so, we start from the latter and progressively add hints in a controlled way so as to still be able to prove reductions. We note that the hints given in ext-M-LWE are different from those given in similar variants [AA16, LNS21, KLSS23], which we discuss in Section 2.3.2. Also, in this section, we are dealing with *continuous* Gaussian errors in the Minkowski embedding σ_H, but one could discretize the error as mentioned in Section 1.4.2.

Theorem 2.2 (Pseudorandomness of M-LWE with Short Secret)

Let $\nu = \prod_i p_i^{e_i}$, K be the ν-th cyclotomic field of degree $n = \varphi(\nu)$, and R its ring of integers. Let $\mu = \prod_i p_i^{f_i}$ for some $f_i \in [\![e_i]\!]$, and q be a prime number such that $q = 1 \bmod \mu$, $ord_\nu(q) = \nu/\mu$, $q > (\eta\mathfrak{s}_1(\mu))^{\varphi(\mu)}$, and $q^{\nu/\mu} \geq n^{\omega(1/(k+1))}$, where $\mathfrak{s}_1(\mu)$ denotes the largest singular value of the Vandermonde matrix of the μ-th cyclotomic field, and η a positive integer. Further, let k, d, m be positive integers such that $d \geq (k+1)\log_{2\eta+1} q + \omega(\log_{2\eta+1} n)$, and $d \leq m \leq \mathsf{poly}(\lambda)$. Let $r \geq \sqrt{n \cdot \ln(2nm(1+\varepsilon^{-1}))/\pi}$ for some $\varepsilon \in (0, 1/2)$, and $s \geq r \cdot n\eta\sqrt{2d}\sqrt{4n^2\eta^2+1}$. Then there is a PPT reduction from M-LWE$_{n,k,m,q,U(R_q),D_r}$ to M-LWE$_{n,d,m,q,U(S_\eta),\Psi_{\leq s}}$. More precisely, if $\varepsilon_{\text{M-LWE}}, \varepsilon_{\text{M-LWE},\eta}$ are their hardness bound respectively, it holds that

$$\varepsilon_{\text{M-LWE},\eta} \leq (2m+1)\varepsilon_{\text{M-LWE}} + 35m\varepsilon + \frac{1}{2}\sqrt{\left(1+\frac{q^{k+1}}{(2\eta+1)^d}\right)^n - 1} + 2m \prod_{i\in[\![\varphi(\mu)]\!]}\left(1-\frac{1}{q^{(k+1)\frac{\nu}{\mu}}}\right).$$

When $\nu = 2^{\ell+1}, n = 2^\ell$, one can take any prime q such that $q = 2\kappa+1 \bmod 4\kappa$ for some $\kappa = 2^l$ with $l \in [\![\ell]\!]$, and such that $q > (\eta\sqrt{\kappa})^\kappa$ and $q^{\nu/\kappa} \geq n^{\omega(1/(k+1))}$.

The modulus is constrained in terms of its splitting behavior. The conditions essentially mean that q splits into $\varphi(\mu)$ factors, each having algebraic norm $q^{\nu/\mu}$. This norm must be at least $n^{\omega(1/(k+1))}$ for Lemma 2.1 to go through, and q must exceed $(\eta\mathfrak{s}_1(\mu))^{\varphi(\mu)}$ so that every element of S_η is a unit in R_q. Then, the noise ratio s/r contains three main terms. The factor $n\eta$ encapsulates the norm distortion between the coefficient and the canonical embedding, as well as the actual length of the η-bounded vectors. The second term $\sqrt{2d}$ stems from the masking of $\mathbf{z}$ when introduced in the first hybrid in the proof of Lemma 2.5. The last factor $\sqrt{4n^2\eta^2+1}$ solely represents the impact of giving information on the error in the ext-M-LWE problem.

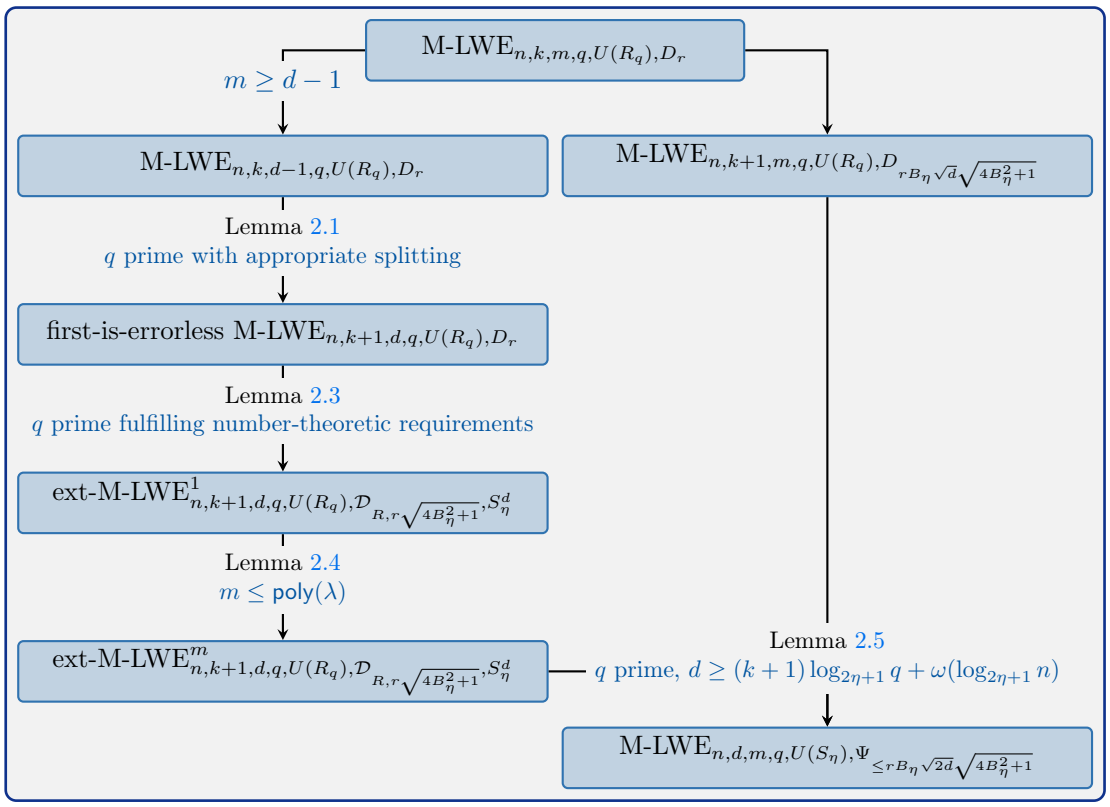


Figure 2.2: Summary of the proof of Theorem 2.2, where $B_\eta = \max_{x\in S_\eta} \left\|\sigma(x)\right\|_\infty$ from Lemma 1.1. In cyclotomic fields, we have $B_\eta \leq n\eta$. Note that Lemma 2.5 uses d samples from ext-M-LWE, where d is the module rank in the final M-LWE problem. The assumptions on q concern the splitting behavior of the cyclotomic polynomial in $\mathbb{Z}_q[x]$, and are discussed in Section 2.3.2.

Proof (Theorem 2.2). We now detail how to combine Lemmas 2.1, 2.3, 2.4 and 2.5. For clarity, we define δ_1 as the loss incurred by the leftover hash lemma, namely $\delta_1 = \frac{1}{2}\sqrt{\left(1+\frac{q^{k+1}}{(2\eta+1)^d}\right)^n - 1}$, and $\delta_2 = \prod_{i\in[\![\varphi(\mu)]\!]}\left(1-q^{-(k+1)\nu/\mu}\right)$.

First, by Lemma 2.5, it holds that

$$\varepsilon_{\text{M-LWE},\eta} \leq \varepsilon^{(m,\alpha)}_{\text{ext-M-LWE}} + \varepsilon^{(1,\gamma)}_{\text{M-LWE}} + \varepsilon^{(m,\alpha)}_{\text{M-LWE}} + 2m\varepsilon + \delta_1,$$

where $\varepsilon^{(m,\alpha)}_{\text{ext-M-LWE}}$ is the hardness bound of ext-M-LWE$^m_{n,k+1,d,q,U(R_q),\mathcal{D}_{R,\alpha},S_\eta^d}$ for $\alpha = r\sqrt{4B_\eta^2+1}$, $\varepsilon^{(1,\gamma)}_{\text{M-LWE}}$ is that of M-LWE$_{n,k+1,m,q,U(R_q),D_\gamma}$ with $\gamma = rB_\eta\sqrt{d}\sqrt{4B_\eta^2+1}$, and $\varepsilon^{(m,\alpha)}_{\text{M-LWE}}$ is that of M-LWE$^m_{n,k+1,d,q,U(R_q),\mathcal{D}_{R,\alpha}}$. By a trivial reduction (which simply discards the hint) we have $\varepsilon^{(m,\alpha)}_{\text{M-LWE}} \leq \varepsilon^{(m,\alpha)}_{\text{ext-M-LWE}}$. Then, using the hybrid argument from Lemma 2.4, it holds that $\varepsilon^{(m,\alpha)}_{\text{ext-M-LWE}} \leq m\varepsilon^{(1,\alpha)}_{\text{ext-M-LWE}}$. We can then use Lemma 2.3 and obtain that $\varepsilon^{(1,\alpha)}_{\text{ext-M-LWE}} \leq \varepsilon^{(r)}_{\text{fie-M-LWE}} + 33\varepsilon/2$, where $\varepsilon^{(r)}_{\text{fie-M-LWE}}$ is the hardness bound of first-is-errorless M-LWE$_{n,k+1,d,q,U(R_q),D_r}$. Finally, we use Lemma 2.1 to get that $\varepsilon^{(r)}_{\text{fie-M-LWE}} \leq \varepsilon^{(1,r)}_{\text{M-LWE}} + \delta_2$, where $\varepsilon^{(1,r)}_{\text{M-LWE}}$ is the hardness bound of M-LWE$_{n,k,d-1,q,U(R_q),D_r}$. This proves that

$$\varepsilon^{(m,\alpha)}_{\text{M-LWE}} \leq \varepsilon^{(m,\alpha)}_{\text{ext-M-LWE}} \leq m(\varepsilon^{(1,r)}_{\text{M-LWE}} + 33\varepsilon/2 + \delta_2).$$

We finish by relating the hardness bounds of the different M-LWE to identify the limiting assumption. We first note that if $m \geq d-1$, which is the case for the proper definition of M-LWE$_{n,d,m,q,U(S_\eta),\Psi_{\leq s}}$, we have a trivial reduction from M-LWE$_{n,k,m,q,U(R_q),D_r}$ to M-LWE$_{n,k,d-1,q,U(R_q),D_r}$. The reduction simply consists in giving only a subset of the m samples of size $d-1$ to the oracle. Hence, if $\varepsilon_{\text{M-LWE}}$ is the hardness bound of M-LWE$_{n,k,m,q,U(R_q),D_r}$, we have $\varepsilon^{(1,r)}_{\text{M-LWE}} \leq \varepsilon_{\text{M-LWE}}$.

Another simple reduction shows that $\varepsilon^{(1,\gamma)}_{\text{M-LWE}} \leq \varepsilon_{\text{M-LWE}}$. The reduction receives $(\mathbf{A}, \mathbf{b})$, samples $(\mathbf{a}, s_{k+1})$ uniformly in $R_q^m \times R_q$, and $\mathbf{e}' \hookleftarrow D^m_{\sqrt{\gamma^2 - r^2}}$, and sends $(\mathbf{A}', \mathbf{b}') = ([\mathbf{A}|\mathbf{a}], \mathbf{b} + s_{k+1}\mathbf{a} + \mathbf{e}' \bmod qR)$ to the oracle. We have that $\gamma^2 - r^2 = r^2(B_\eta^2 d(4B_\eta^2 + 1) - 1) > 0$ as $B_\eta \geq 1$ by Lemma 1.1, which is sufficient to sample $\mathbf{e}'$. Then, if $\mathbf{b}$ is uniform, so is $\mathbf{b}'$. And if $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod qR$, then $\mathbf{b}' = \mathbf{A}'[\mathbf{s}^T | s_{k+1}]^T + (\mathbf{e} + \mathbf{e}') \bmod qR$ where $\mathbf{e} + \mathbf{e}'$ is correctly distributed by the sum of independent continuous Gaussians.

Combining it all gives

$$\begin{aligned}\varepsilon_{\text{M-LWE},\eta} &\leq 2m\left(\varepsilon_{\text{M-LWE}} + \delta_2 + \frac{33\varepsilon}{2}\right) + \varepsilon_{\text{M-LWE}} + 2m\varepsilon + \delta_1 \\ &= (2m+1)\varepsilon_{\text{M-LWE}} + 35m\varepsilon + \delta_1 + 2m\delta_2,\end{aligned}$$

as claimed. The parameter selection in the theorem statement gives that δ_1, δ_2 are negligible in λ.

Following the chain of reductions of [BJRW20] discussed in the previous section, we can perform the same reasoning to link $\text{M-LWE}_{n,d,m,q,U(S_\eta),\Psi_{\leq s}}$ to worst-case module lattice problems, in particular by linking the parameters to the approximation factor γ. As we do not need to re-discretize the Gaussian error after the modulus switching reduction, we have that if the (gap) decisional variant of SVP_γ is hard over module lattices of rank k with approximation factor γ, then $\text{M-LWE}_{n,d,m,q,U(S_\eta),\Psi_{\leq s}}$ is hard with

$$\frac{s}{q} = \sqrt{2\left(\frac{nk\sqrt{k}}{\gamma} \cdot \omega(\log_2 n) \cdot n\eta\sqrt{2d}\sqrt{4n^2\eta^2 + 1}\right)^2 + \boldsymbol{\Delta}},$$

in general cyclotomic fields. The term highlighted in purple stems from Theorem 2.2. In power-of-two cyclotomic fields, we then obtain that the module lattice assumption must be secure for an approximation factor

$$\gamma \approx \frac{4k^{3/2} \cdot n^3\eta^2\sqrt{d}}{s/q} \cdot \omega(\log_2 n).$$

which is larger than with the one obtained with the reduction of Section 2.2 by a factor of roughly $n\eta\sqrt{8}/((\sqrt{m} + \sqrt{d})\omega(\log_2 n))$.

2.3.1 First-Is-Errorless M-LWE

We follow the same idea as Brakerski et al. [BLP+13] by gradually giving more information to the adversary while proving that this additional information does not increase the advantage too much. We define the module version of *first-is-errorless* LWE, from [BLP+13], where the first equation is given without error. A similar definition and reduction from M-LWE are given in [AA16]. We only define the decision variant with a single secret vector as this is the one needed in our case.

Definition 2.1 (First-is-Errorless M-LWE)

Let n, k, m, q be positive integers with $m \geq k$. Let K be a number field of degree n, and R its ring of integers. Then, let $\mathscr{D}_s$ be a distribution of secrets over R, and $\mathscr{D}_e$ a distribution of errors over $K_\mathbb{R}$ (possibly with a discrete support included in R). We define $\mathbb{S}_q = \text{Supp}(\mathscr{D}_e)/qR \cap R_q$ (e.g., $\mathbb{S}_q = \mathbb{T}_q$ when $\text{Supp}(\mathscr{D}_e) = K_\mathbb{R}$ and R_q when $\text{Supp}(\mathscr{D}_e) \subseteq R$). The first-is-errorless $\text{M-LWE}_{n,k,m,q,\mathscr{D}_s,\mathscr{D}_e}$ problem asks to distinguish between the following distributions:

$\mathcal{P}_1$: Sample $\mathbf{A} \hookleftarrow U(R_q^{m\times k})$, $\mathbf{s} \hookleftarrow \mathscr{D}_s^k$ and $\mathbf{e} \hookleftarrow \{0\} \otimes \mathscr{D}_e^{m-1}$. Compute $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod qR$
Output: $(\mathbf{A}, \mathbf{b})$.

$\mathcal{P}_2$: Sample $\mathbf{A} \hookleftarrow U(R_q^{m\times k})$, and $\mathbf{b} \hookleftarrow U(R_q \times \mathbb{S}_q^{m-1})$.
Output: $(\mathbf{A}, \mathbf{b})$.

where $\{0\}$ is the distribution that is deterministically 0. We denote by $\varepsilon_{\text{fie-M-LWE}}$ its hardness bound, defined similarly to $\varepsilon_{\text{M-LWE}}$ in Section 1.4.2.

In this thesis, we decide to define the variants of M-LWE in matrix form which fixes the number of samples a priori. One could consider a sample-per-sample definition which could be used if one does not know the number of samples needed in advance. The latter case is covered in our original papers [BJRW21, BJRW23].

Following the high-level blueprint of [BLP+13, Lem. 4.3], with a pre-processing step specific to the module setting, we now show that at the expense of an extra sample and an extra secret dimension, the first-is-errorless M-LWE problem is no easier than the original M-LWE problem from Definition 1.15. A similar reduction is provided in [AA16]. The only difference with our reductions comes from the pre-processing step, which is performed before receiving the M-LWE samples. In our case, this step is simplified and extended to general number fields, provided that the modulus q verifies certain splitting conditions. Further restrictions on q in our reduction encompasses these conditions.

Lemma 2.1 (M-LWE **to** first-is-errorless M-LWE)

Let λ, n, k, m, q be positive integers with $m \geq k > 1$. Let K be a number field of degree n, and R its ring of integers. Then, let $\mathscr{D}_s$ be a distribution of secrets over R, and $\mathscr{D}_e$ a distribution of errors over $K_{\mathbb{R}}$ (possibly with a discrete support included in R). We define $\mathbb{S}_q = \mathrm{Supp}(\mathscr{D}_e)/qR \cap R_q$ (e.g., $\mathbb{S}_q = \mathbb{T}_q$ when $\mathrm{Supp}(\mathscr{D}_e) = K_{\mathbb{R}}$ and R_q when $\mathrm{Supp}(\mathscr{D}_e) \subseteq R$). We assume that q is an unramified prime such that the smallest norm of its prime ideal factors is at least $\lambda^{\omega(1/k)}$. There is a PPT reduction from $\text{M-LWE}_{n,k-1,m-1,q,U(R_q),\mathscr{D}_e}$ to the variant first-is-errorless $\text{M-LWE}_{n,k,m,q,U(R_q),\mathscr{D}_e}$. More precisely, it holds that

$$\varepsilon_{\text{fie-M-LWE}} \leq \varepsilon_{\text{M-LWE}} + \left(1 - \prod_{i \in [\![\kappa]\!]} \left(1 - \frac{1}{N(\mathfrak{p}_i)^k}\right)\right),$$

where the $\mathfrak{p}_i$'s are the prime ideal factors of $\langle q \rangle$.

Notice that the increase of 1 in the rank, i.e., from $k-1$ to k, seems natural to ensure the hardness of first-is-errorless M-LWE. Indeed, if one defines the first-is-errorless M-LWE with rank $k = 1$ to obtain a ring version, the first sample is $b_1 = a_1 \cdot s \bmod qR$. Thence, if $a_1 \in R_q^\times$, one can recover s. More generally, one can recover all the field embeddings $\sigma_i(s)$ for which $\sigma_i(a_1) \neq 0$, which already gives out too much information on s. From the candidate $s^\star$, the adversary can then use the subsequent samples and compute $e_i^\star = b_i - a_i s^\star$ and be able to distinguish $\mathcal{P}_1$ and $\mathcal{P}_2$ with noticeable probability.

Proof (Lemma 2.1). *Pre-processing:* The reduction first samples $\mathbf{a}' \hookleftarrow U(R_q^k)$ such that $\mathbf{a}'$ is R_q-linearly independent. As a result, $\mathbf{a}'$ is uniform among the R_q-linearly independent vectors. We first show that under the conditions of the lemma, the distribution of $\mathbf{a}'$ is statistically close to $U(R_q^k)$. Recall that if $A \subseteq B$, then $\Delta(U(A), U(B)) = \mathbb{P}_{x \sim U(B)}[x \notin A]$. Applied to the set of vectors of R_q^k that are R_q-linearly independent, and combined with Lemma 1.5 for $\ell = 0$, it yields

$$\Delta(\mathbf{a}', U(R_q^k)) == 1 - \prod_{i \in [\![\kappa]\!]} \left(1 - \frac{1}{N(\mathfrak{p}_i)^k}\right) \leq \frac{n}{\min_{i \in [\![\kappa]\!]} N(\mathfrak{p}_i)^k},$$

the latter quantity being assumed negligible. Hence $\mathbf{a}'$ is within negligible statistical distance of $U(R_q^k)$. Then, from $\mathbf{a}'$, one can efficiently complete it with $\mathbf{b}_2, \ldots, \mathbf{b}_k \in R_q^k$ such that the matrix $\mathbf{U} = [\mathbf{a}'|\mathbf{b}_2|\ldots|\mathbf{b}_k]$ is invertible in R_q. For example, this can be done by successively sampling the $\mathbf{b}_i$'s uniformly at random in R_q^k. By Lemma 1.5, the probability that the newly drawn $\mathbf{b}_{\ell+1}$ is kept is $\prod_{i \in [\![\kappa]\!]} 1 - N(\mathfrak{p}_i)^{-(k-\ell)} \geq 1 - \lambda^{-\omega(1)}$. It would thus require at most a polynomial number of sampled vectors.

Reduction: Then, the reduction samples $s_0 \hookleftarrow U(R_q)$ and $\mathbf{a}'' \hookleftarrow U(R_q^{m-1})$ and proceeds as

follows. Upon receiving the M-LWE instance $(\mathbf{A}, \mathbf{b}) \in R_q^{m-1\times k-1} \times \mathbb{S}_q^{m-1}$, it constructs

$$\overline{\mathbf{A}} = \begin{bmatrix} \mathbf{a}'^T \\ [\mathbf{a}''|\mathbf{A}]\mathbf{U}^T \end{bmatrix} \bmod qR, \quad \text{and} \quad \overline{\mathbf{b}} = \begin{bmatrix} s_0 \\ \mathbf{b} + s_0 \cdot \mathbf{a}'' \end{bmatrix} \bmod qR,$$

and calls the first-is-errorless M-LWE oracle on $(\overline{\mathbf{A}}, \overline{\mathbf{b}})$ and returns the same answer as solution to the M-LWE instance. Let us now analyze the correctness of the reduction. First note that $\overline{\mathbf{A}}$ is statistically close to $U(R_q^{m\times k})$. Indeed, we have proven that $\mathbf{a}'$ is close to uniform over R_q^k for the first sample, and since $\mathbf{A}$ is uniform over $R_q^{m-1\times k-1}$, $\mathbf{a}''$ is uniform over R_q^{m-1}, and $\mathbf{U}$ is invertible in $R_q^{k\times k}$, then $[\mathbf{a}''|\mathbf{A}]\mathbf{U}^T$ is uniform over $R_q^{m-1\times k}$ as well. We now look at the distribution of $\overline{\mathbf{b}}$.

Assume $\mathbf{b}$ is uniform in $\mathbb{S}_q^{m-1}$. The first sample is s_0 which is uniform in R_q. For the other samples, $\mathbf{b} + s_0 \cdot \mathbf{a}'' \bmod qR$ is uniform over $\mathbb{S}_q$ and independent of $[\mathbf{a}''|\mathbf{A}]\mathbf{U}^T$ but also independent from the first sample because $\mathbf{b}$ perfectly masks $s_0 \cdot \mathbf{a}''$. Now, if $b = \mathbf{A}\mathbf{s}+\mathbf{e} \bmod qR$ for some $\mathbf{s} \hookleftarrow U(R_q^{k-1})$ and $e \hookleftarrow \mathscr{D}_e^{m-1}$, then $s_0 = \langle \mathbf{e}_1, [s_0|\mathbf{s}^T]^T\rangle = \langle \mathbf{U}\mathbf{e}_1, \mathbf{U}^{-T}[s_0|\mathbf{s}^T]^T\rangle = \langle \mathbf{a}', \mathbf{U}^{-T}[s_0|\mathbf{s}^T]^T\rangle$, where $\mathbf{e}_1 = [1|0|\dots|0]^T$. For the other samples, we have

$$\begin{aligned} \mathbf{b} + s_0 \cdot \mathbf{a}'' \bmod qR = \mathbf{A}\mathbf{s} + s_0 \cdot \mathbf{a}'' + \mathbf{e} \bmod qR &= [\mathbf{a}''|\mathbf{A}][s_0|\mathbf{s}^T]^T + \mathbf{e} \bmod qR \\ &= [\mathbf{a}''|\mathbf{A}]\mathbf{U}^T \cdot \mathbf{U}^{-T}[s_0|\mathbf{s}^T]^T + \mathbf{e} \bmod qR. \end{aligned}$$

This shows that $\overline{\mathbf{b}} = \overline{\mathbf{A}}\overline{\mathbf{s}}+\overline{\mathbf{e}} \bmod qR$ for $\overline{\mathbf{s}} = \mathbf{U}^{-T}[s_0|\mathbf{s}^T]^T$ and $\overline{\mathbf{e}} = [0|\mathbf{e}^T]^T$. Note that $[s_0|\mathbf{s}^T]^T$ is uniform over R_q^k, which implies that $\overline{\mathbf{s}}$ is also uniform over R_q^k because $\mathbf{U}^{-T}$ is invertible in R_q. Therefore the reduction outputs samples according to first-is-errorless M-LWE with secret $\overline{\mathbf{s}}$. Finally, by Lemma 1.7, for a PPT adversary $\mathcal{A}$ against M-LWE, we have

$$\varepsilon_{\text{M-LWE}} \geq \text{Adv}_{\text{M-LWE}}[\mathcal{A}] \geq \varepsilon_{\text{fie-M-LWE}} - \left(1 - \prod_{i\in[\![\kappa]\!]} \left(1 - \frac{1}{N(\mathfrak{p}_i)^k}\right)\right),$$

as claimed.

Later in the reduction, we restrict the modulus q to be a prime that splits into few prime factors in the underlying cyclotomic field to maximize the number of invertible elements, and more precisely to be able to use Lemma 1.4 without having to take superpolynomial q. In this case, one could use moduli that split into say κ factors such that $q^{-n/\kappa} \leq \mathsf{negl}(\lambda)$.

2.3.2 Extended M-LWE

We now define the module version of the *Extended Learning With Errors* problem introduced in [BLP+13], where the adversary is allowed a hint on the errors. A first definition of ext-M-LWE was introduced by Alperin-Sheriff and Apon [AA16] in which the hints were of the form $Tr(\langle \mathbf{z}_i, \mathbf{e}\rangle)$ for a single error vector $\mathbf{e}$ and several *hint vectors* $\mathbf{z}_i$, and where $Tr(\cdot)$ is the field trace. In our case, we allow for multiple secrets (and thus errors) and one single hint vector $\mathbf{z}$, as required by our final reduction of Lemma 2.5. Additionally, as the field trace does not provide enough information to reconstruct $\langle \mathbf{z}, \mathbf{e}\rangle$ from the hint, we instead directly give $\langle \mathbf{z}, \mathbf{e}\rangle$ as the hint. We prove that it does not make the problem easier. Another version of ext-M-LWE was recently introduced in [LNS21] in the context of lattice-based zero-knowledge proofs, where they only provide the sign $\text{Sign}(\langle \mathbf{z}, \mathbf{e}\rangle)$ as an additional hint for the attacker. Again, this is not sufficient for our lossy argument in Lemma 2.5. Finally, a recent work by Kim et al. [KLSS23] defined a problem called *Hint* M-LWE. The hints are however given on both the secret and error, and perturbed by an additional Gaussian noise as $\langle \mathbf{z}_i, [\mathbf{s}^T|\mathbf{e}^T]\rangle + e'_i$. The presence of this noise e'_i unfortunately prevents the reduction from going through.

We only define the version with discrete error distribution so as to clarify the space in which the hints lie, and because this is the version we need in our reduction. One could however define a more general version and most likely prove its hardness similarly as we do.

Definition 2.2 (Extended M-LWE)

Let n, ℓ, k, m, q be positive integers with $m \geq k$. Let K be a number field of degree n, and R its ring of integers. Then, let $\mathscr{D}_s$ be a distribution of secrets over R, and $\mathscr{D}_e$ a distribution of errors over R. We also let $\mathcal{Z}$ a subset of R^m. The *Extended Module Learning With Errors* problem ext-M-LWE$^{\ell}_{n,k,m,q,\mathscr{D}_s,\mathscr{D}_e,\mathcal{Z}}$ asks to distinguish between the following distributions after the adversary chose some $\mathbf{z} \in \mathcal{Z}$:

$\mathcal{P}_1$: Sample $\mathbf{A} \hookleftarrow U(R_q^{m\times k})$, $\mathbf{S} \hookleftarrow \mathscr{D}_s^{k\times \ell}$ and $\mathbf{E} \hookleftarrow \mathscr{D}_e^{m\times \ell}$. Compute $\mathbf{B} = \mathbf{AS} + \mathbf{E} \bmod qR$, and $\mathbf{h} = \mathbf{E}^T\mathbf{z} \in R^\ell$.
Output: $(\mathbf{A}, \mathbf{B}, \mathbf{h})$.

$\mathcal{P}_2$: Sample $\mathbf{A} \hookleftarrow U(R_q^{m\times k})$, $\mathbf{E} \hookleftarrow \mathscr{D}_e^{m\times \ell}$, and $\mathbf{B} \hookleftarrow U(R_q^{m\times \ell})$. Compute $\mathbf{h} = \mathbf{E}^T\mathbf{z} \in R^\ell$.
Output: $(\mathbf{A}, \mathbf{B}, \mathbf{h})$.

The parameter ℓ represents the number of given hints on independent noise vectors, and therefore the number of secret vectors (which generalizes the multiple secret version of M-LWE of Definition 1.15). The set $\mathcal{Z}$ represents the set of allowed hint vectors $\mathbf{z}$. The ℓ hints are given in form of the inner product of such a fixed *hint vector* $\mathbf{z} \in \mathcal{Z}$ and the corresponding columns of $\mathbf{E}$. Later, we are interested in the case where $\mathcal{Z} = S_\eta^m$ which is actually the set of secrets for the targeted M-LWE assumption. Also, note that if $\mathcal{Z} = \{\mathbf{0}\}$, then we recover the definition of the multiple secrets version of M-LWE from Definition 1.15.

Similarly to [Mic18], for a matrix $\mathbf{A} \in R^{m\times m}$, we denote by $\mathbf{A}^{]1[} \in R^{m\times(m-1)}$ the submatrix of $\mathbf{A}$ obtained by removing the leftmost column. Our reduction from first-is-errorless M-LWE to ext-M-LWE in Lemma 2.3 requires the construction of a matrix $\mathbf{U}_\mathbf{z} \in R^{m\times m}$, for all vectors $\mathbf{z} \in \mathcal{Z} = S_\eta^m$, satisfying several properties. This matrix allows us to transform samples from a first-is-errorless M-LWE challenger into samples that we can give to an oracle for ext-M-LWE. The spectral norm of its submatrix $\mathbf{U}_\mathbf{z}^{]1[}$ (when embedded with M_σ), controls the increase in the Gaussian parameter. We propose a construction for which we bound the spectral norm above by a quantity independent on $\mathbf{z}$, as needed in the reduction.

Lemma 2.2

Let $\nu = \prod_i p_i^{e_i}$, K be the ν-th cyclotomic field of degree $n = \varphi(\nu)$, and R its ring of integers. Let $\mu = \prod_i p_i^{f_i}$ for some $f_i \in [\![e_i]\!]$, η a positive integer and q be a prime such that $q = 1 \bmod \mu$, $ord_\nu(q) = \nu/\mu$ and $q > (\eta \mathfrak{s}_1(\mu))^{\varphi(\mu)}$, where $\mathfrak{s}_1(\mu)$ denotes the spectral norm of the Vandermonde matrix of the μ-th cyclotomic field. Finally, let m be a positive integer, and $\mathcal{Z} = S_\eta^m$. For all $\mathbf{z} \in \mathcal{Z}$, there is an efficiently computable matrix $\mathbf{U}_\mathbf{z} \in R^{m\times m}$ such that

1. $\mathbf{U}_\mathbf{z} \bmod qR \in GL_m(R_q)$,
2. $\mathbf{z}^T\mathbf{U}_\mathbf{z}^{]1[} = \mathbf{0}$ in R,
3. $\left\| M_{\sigma_H}\left(\mathbf{U}_\mathbf{z}^{]1[}\right)\right\|_2 \leq 2B_\eta$, where $B_\eta = \max_{x\in S_\eta} \|\sigma(x)\|_\infty$.

When $\nu = 2^{\ell+1}, n = 2^\ell$, one can take any prime q such that $q = 2\kappa + 1 \bmod 4\kappa$ for some $\kappa = 2^l$ with $l \in [\![\ell]\!]$, and such that $q > (\eta\sqrt{\kappa})^\kappa$.

Proof (Lemma 2.2). Let $\mathbf{z} \in \mathcal{Z}$. First, we construct $\mathbf{U}_\mathbf{z}$ in the case where all the z_i are non-zero. To do so, we define the intermediate matrices $\mathbf{A}$, and $\mathbf{B}$ of $R^{m\times m}$, all unspecified entries being zeros.

$$\mathbf{U_z} = \begin{bmatrix} 1 & & & \\ & -z_2 & & \\ & z_1 & \ddots & \\ & & & -z_m \\ & & z_{m-1} & \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & z_1 & & \\ & & \ddots & \\ & & & z_{m-1} \\ & & & \end{bmatrix} + \begin{bmatrix} 0 & & & \\ & -z_2 & & \\ & 0 & \ddots & \\ & & & -z_m \\ & & 0 & \end{bmatrix}.$$

$\mathbf{U}_{\mathbf{z}}^{]1[}$ $\quad$ $\mathbf{A}^{]1[}$ $\quad$ $\mathbf{B}^{]1[}$

The matrix $\mathbf{U_z}$ is invertible modulo qR only if all the z_i (except z_m) are in $R_q^\times$. Yet, since they are all non-zero and η-bounded elements, we have that for all i in $[\![m]\!]$, $0 < \left\|\tau(z_i)\right\|_\infty \leq \eta$, where τ is the coefficient embedding. By Lemma 1.4, since q verifies the algebraic conditions taking all $f_i = 1$ and $q^{1/\varphi(\mu)}/\mathfrak{s}_1(\mu) > \eta$, all the z_i are in $R_q^\times$.

By construction, the last $m-1$ columns of $\mathbf{U_z}$ are orthogonal to $\mathbf{z}$. Let $\mathbf{U}_{\mathbf{z}}^{]1[}$ be the submatrix of $\mathbf{U_z}$ obtained by removing the leftmost column as shown above. As observed in Lemma 1.3 for example, it holds that $\left\|M_{\sigma_H}(\mathbf{U}_{\mathbf{z}}^{]1[})\right\|_2 = \left\|M_\sigma(\mathbf{U}_{\mathbf{z}}^{]1[})\right\|_2$. Then, using the fact that M_σ is a ring homomorphism, we have $M_\sigma(\mathbf{U}_{\mathbf{z}}^{]1[}) = M_\sigma(\mathbf{A}^{]1[}) + M_\sigma(\mathbf{B}^{]1[})$. We now need to bound the spectral norm of these two matrices, and use the triangle inequality to conclude. For any vector $\mathbf{x} \in \mathbb{C}^{(m-1)n}$, we have that $\left\|M_\sigma(\mathbf{A}^{]1[})\mathbf{x}\right\|_2 = \sqrt{\sum_{i\in[\![m-1]\!]}\sum_{j\in[\![n]\!]}\left|\sigma_j(z_i)\right|^2\left|x_{j+n(i-1)}\right|^2} \leq B_\eta\|\mathbf{x}\|_2$, because each z_i is in S_η. This yields $\left\|M_\sigma(\mathbf{A}^{]1[})\right\|_2 \leq B_\eta$. A similar calculation on $\mathbf{B}^{]1[}$ leads to $\left\|M_\sigma(\mathbf{B}^{]1[})\right\|_2 \leq B_\eta$, thus resulting in $\left\|M_\sigma(\mathbf{U}_{\mathbf{z}}^{]1[})\right\|_2 \leq 2B_\eta$.

Now assume that $z_{i_0}, \ldots, z_m$ are zeros for some i_0 in $[\![m]\!]$. If the zeros do not appear last in the vector $\mathbf{z}$, we can replace $\mathbf{z}$ with $\mathbf{Sz}$, where $\mathbf{S} \in R^{m\times m}$ swaps the coordinates of $\mathbf{z}$ so that the zeros appear last. Since $\mathbf{S}$ is unitary, it preserves the singular values as well as invertibility. Then, the construction remains the same except that the $z_{i_0}, \ldots, z_m$ on the diagonal are replaced by 1. The orthogonality is preserved, and $\left\|M_\sigma(\mathbf{U}_{\mathbf{z}}^{]1[})\right\|_2$ can still be bounded above by $2B_\eta$.

Notice that when the ring is of degree 1 and $\eta = 1$, the constructions in the different cases match the ones from [BLP+13, Claim 4.6]. So do the singular values as $B_\eta \leq n\eta = 1$ by Lemma 1.1. Also, the construction differs from the notion of quality in [AA16] due to the discrepancies between the two definitions of ext-M-LWE. The following lemma shows that the extended variant of M-LWE with one hint ($\ell = 1$) is at least as hard as the first-is-errorless variant of M-LWE, for carefully chosen parameters. This corresponds to the module equivalent of [BLP+13, Lem. 4.7]. The algebraic setting however introduces complications mostly due to the construction of Lemma 2.2.

Lemma 2.3 (first-is-errorless M-LWE **to** ext-M-LWE)

Let $\nu = \prod_i p_i^{e_i}$, K be the ν-th cyclotomic field of degree $n = \varphi(\nu)$, and R its ring of integers. Let $\mu = \prod_i p_i^{f_i}$ for some $f_i \in [e_i]$, η a positive integer and q be a prime such that $q = 1 \bmod \mu$, $ord_\nu(q) = \nu/\mu$ and $q > (\eta\mathfrak{s}_1(\mu))^{\varphi(\mu)}$, where $\mathfrak{s}_1(\mu)$ denotes the spectral norm of the Vandermonde matrix of the μ-th cyclotomic field. Let m, k be positive integers, $\mathcal{Z} = S_\eta^m$, $\varepsilon \in (0, 1/2)$ and $r \geq \sqrt{n} \cdot \sqrt{\ln(2nm(1+\varepsilon^{-1}))/\pi}$. Finally, let $s = r\sqrt{4B_\eta^2 + 1}$, where $B_\eta = \max_{x\in S_\eta}\left\|\sigma(x)\right\|_\infty$. There is a PPT reduction from the variant first-is-errorless M-LWE$_{n,k,m,q,U(R_q),D_r}$ to ext-M-LWE$^1_{n,k,m,q,U(R_q),\mathcal{D}_{R,s},\mathcal{Z}}$. More precisely, it holds that

$$\varepsilon_{\text{ext-M-LWE}} \leq \varepsilon_{\text{fie-M-LWE}} + 33\varepsilon/2.$$

When $\nu = 2^{\ell+1}, n = 2^\ell$, one can take any prime q such that $q = 2\kappa + 1 \bmod 4\kappa$ for some $\kappa = 2^l$ with $l \in [\![\ell]\!]$, and such that $q > (\eta\sqrt{\kappa})^\kappa$.

Proof (Lemma 2.3). First, we clarify the condition on r. Since K is a cyclotomic field, $\mathbf{B} = \mathbf{I}_m \otimes [\sigma_H(1)|\ldots|\sigma_H(\zeta^{n-1})]$ is a basis of the lattice $\sigma_H(R^m)$, and each vector has norm

$\sqrt{n}$. As a result, the max-Euclidean norm of the Gram-Schmidt orthogonalization of $\mathbf{B}$ is at most $\sqrt{n}$. By [GPV08, Lem. 3.1, Thm. 4.1], our condition on r ensures that $r \geq \eta_\varepsilon(R^m)$ and that $\mathcal{D}_{R^m,r}$ is efficiently sampleable.

Now, assume we have access to an oracle $\mathcal{O}$ for ext-M-LWE$^1_{n,k,m,q,U(R_q),\mathcal{D}_{R,s},\mathcal{Z}}$. We take m samples from the first-is-errorless challenger, resulting in

$$(\mathbf{A}, \mathbf{b}) \in R_q^{m\times k} \times (R_q \times \mathbb{T}_q^{m-1}).$$

Assume we need to provide samples to $\mathcal{O}$ for some $\mathbf{z} \in \mathcal{Z}$. By Lemma 2.2 we can efficiently compute a matrix $\mathbf{U_z} \in R^{m\times m}$ that is invertible modulo qR, such that its submatrix $\mathbf{U}_\mathbf{z}^{]1[}$ is orthogonal to $\mathbf{z}$, and that $\left\|M_\sigma(\mathbf{U}_\mathbf{z}^{]1[})\right\|_2 \leq 2B_\eta$. The reduction first samples $\mathbf{f} \in K_\mathbb{R}^m$ from the continuous Gaussian distribution of covariance matrix $r^2(4B_\eta^2\mathbf{I}_{mn} - M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})^T) \in \mathbb{R}^{mn\times mn}$. The covariance matrix is indeed in $\mathcal{S}_{nm}^+$ because $\left\|M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})\right\|_2 \leq 2B_\eta$. The reduction then computes $\mathbf{b}' = \mathbf{U_z}\mathbf{b}+\mathbf{f}$ and samples $\mathbf{c}$ from $\mathcal{D}_{R^m-\mathbf{b}',r}$ (as it is efficiently sampleable), and finally gives the following to $\mathcal{O}$

$$(\mathbf{A}' = \mathbf{U_z}\mathbf{A} \bmod qR, \mathbf{b}' + \mathbf{c} \bmod qR, \langle \mathbf{z}, \mathbf{f}+\mathbf{c}\rangle) \in R_q^{m\times k} \times R_q^m \times R.$$

From the way $\mathbf{c}$ is sampled, $\mathbf{b}' + \mathbf{c} \in R^m$ and thus the second component is indeed in R_q^m. For the third component, we use the fact that $\mathbf{z}^T\mathbf{U_z}\mathbf{b} = b_1\mathbf{z}^T\mathbf{U}_{\mathbf{z},1}$ where $\mathbf{U}_{\mathbf{z},1}$ is the first column of $\mathbf{U_z}$. This is because $\mathbf{z}^T\mathbf{U}_\mathbf{z}^{]1[} = \mathbf{0}$ by construction of $\mathbf{U_z}$. As a result, we have that $\langle \mathbf{z}, \mathbf{f}+\mathbf{c}\rangle = \langle \mathbf{z}, \mathbf{b}'+\mathbf{c}\rangle - b_1\mathbf{z}^T\mathbf{U}_{\mathbf{z},1}$. The first term is in R because of the sampling of $\mathbf{c}$. The second term is also in R because $b_1 \in R_q$. So the third component is indeed in R. We now prove the correctness of the reduction.

First, consider the case where $\mathbf{A} \hookleftarrow U(R_q^{m\times k})$ and $\mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e} \bmod qR$ for some $\mathbf{s} \hookleftarrow U(R_q^k)$, and $\mathbf{e} \hookleftarrow \{0\} \otimes D_r^{m-1}$ where $\{0\}$ denotes the distribution that is deterministically 0. Since $\mathbf{U_z}$ is invertible modulo qR, $\mathbf{A}' = \mathbf{U_z}\mathbf{A}$ is also uniform over $R_q^{m\times k}$ as required. From now on we condition on an arbitrary $\mathbf{A}'$ and analyze the distribution of the remaining components. We have

$$\mathbf{b}' = \mathbf{U_z}\mathbf{A}\mathbf{s} + \mathbf{U_z}\mathbf{e} + \mathbf{f} \bmod qR = \mathbf{A}'\mathbf{s} + \mathbf{U_z}\mathbf{e} + \mathbf{f} \bmod qR.$$

Since the first coefficient of $\mathbf{e}$ is deterministically 0, $\sigma_H(\mathbf{e})$ is distributed according to the degenerate Gaussian $D_{\sqrt{\mathbf{S}}}$ where $\mathbf{S} = \mathrm{diag}(\mathbf{0}_{n\times n}, r^2\mathbf{I}_{n(m-1)})$. It thus holds that $\sigma_H(\mathbf{U_z}\mathbf{e})$ is distributed according to $D_{\sqrt{\mathbf{S}'}}$ where $\mathbf{S}' = M_{\sigma_H}(\mathbf{U_z})\mathbf{S}M_{\sigma_H}(\mathbf{U_z})$. Due to the specific form of $\mathbf{S}$, we observe that $\mathbf{S} = M_{\sigma_H}(\mathrm{diag}(0, r^2\mathbf{I}_{m-1}))$. Using the ring homomorphism property and the form of $\mathbf{S}$, it holds that $\mathbf{S}' = M_{\sigma_H}(r^2\mathbf{U}_\mathbf{z}^{]1[}(\mathbf{U}_\mathbf{z}^{]1[})^T) = r^2M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})^T$. Hence the vector $\mathbf{U_z}\mathbf{e} + \mathbf{f}$ is distributed as the Gaussian over $K_\mathbb{R}^m$ of covariance matrix $r^2M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})^T + r^2(4B_\eta^2\mathbf{I}_{mn} - M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})M_{\sigma_H}(\mathbf{U}_\mathbf{z}^{]1[})^T)$ which is identical to $D_{r\cdot 2B_\eta}^m$. Since $\mathbf{A}'\mathbf{s} \in R^m$, the coset $R^m - \mathbf{b}'$ is the same as $R^m - (\mathbf{U_z}\mathbf{e} + \mathbf{f})$, which yields that $\mathbf{c}$ can be seen as being sampled from $\mathcal{D}_{R^m-(\mathbf{U_z}\mathbf{e}+\mathbf{f}),r}$. Since $r \geq \eta_\varepsilon(R^m)$, Lemma 1.13 gives that the distribution of $\mathbf{U_z}\mathbf{e}+\mathbf{f}+\mathbf{c}$ is within statistical distance 8ε of $\mathcal{D}_{R^m,r\sqrt{4B_\eta^2+1}}$, which shows that the second component is correctly distributed up to 8ε. Note that $\mathbf{U_z}\mathbf{e} = \sum_{i\in[\![m]\!]} e_i \cdot \mathbf{u}_i$ is in the space spanned by the columns of $\mathbf{U}_\mathbf{z}^{]1[}$ because $e_1 = 0$. This yields $\langle \mathbf{z}, \mathbf{U_z}\mathbf{e}\rangle = 0$ as $\mathbf{z}$ is orthogonal to the columns of $\mathbf{U}_\mathbf{z}^{]1[}$, proving that the third component equals $\langle \mathbf{z}, \mathbf{U_z}\mathbf{e}+\mathbf{f}+\mathbf{c}\rangle$ and is thus correctly distributed.

Now consider the case where both $\mathbf{A}$ and $\mathbf{b}$ are uniform in their respective spaces. Using that $r \geq \eta_\varepsilon(R^m)$, Lemma 1.18 shows that the distribution of $(\mathbf{A}, \mathbf{b})$ is within statistical distance $\varepsilon/2$ of the distribution of $(\mathbf{A}, \mathbf{e}' + \mathbf{e} \bmod qR)$ where $\mathbf{e}' \in R_q^m$ is uniform and $\mathbf{e}$ is distributed from $\{0\} \otimes D_r^{m-1}$. So we can assume our input is $(\mathbf{A}, \mathbf{e}' + \mathbf{e} \bmod qR)$. $\mathbf{A}'$ is uniform as before, and clearly independent of the other two components. Moreover, since $\mathbf{b}' = \mathbf{U_z}\mathbf{e}'+\mathbf{U_z}\mathbf{e}+\mathbf{f} \bmod qR$ and $\mathbf{U_z}\mathbf{e}' \in R^m$, then the coset $R^m - \mathbf{b}'$ is identical to $R^m - (\mathbf{U_z}\mathbf{e}+\mathbf{f})$. For the same reasons as above, $\mathbf{U_z}\mathbf{e} + \mathbf{f} + \mathbf{c}$ is distributed as $\mathcal{D}_{R^m,r\sqrt{4B_\eta^2+1}}$ within statistical distance of at most 8ε, and in particular independent of $\mathbf{e}'$. So the third component is correctly distributed again because $\langle \mathbf{z}, \mathbf{U_z}\mathbf{e}\rangle = 0$. Finally, since $\mathbf{e}'$ is independent of the first and third components, and that $\mathbf{U_z}\mathbf{e}'$ is uniform over R_q^m as $\mathbf{U_z}$ is invertible modulo qR, it yields that the second component is uniform and independent of the other ones as required.

Combining it all with the help of Lemma 1.7 gives that for a PPT adversary $\mathcal{A}$ against first-is-errorless M-LWE, we have

$$\varepsilon_{\text{fie-M-LWE}} \geq \text{Adv}_{\text{fie-M-LWE}}[\mathcal{A}] \geq \varepsilon_{\text{ext-M-LWE}} - 33\varepsilon/2,$$

which concludes the proof.

The condition on the modulus q in Lemma 2.2 and 2.3 stems from the invertibility result by Lyubashevsky and Seiler [LS18] stated in Lemma 1.4. Recall that these conditions can be simplified in the case of power-of-two cyclotomic fields as discussed in Remark 1.2.

We now use a standard hybrid argument to show that ext-M-LWE with ℓ hints is at least as hard as ext-M-LWE with one hint, at the expense of reducing the advantage by a factor of ℓ. By setting $\mathcal{Z} = \{\mathbf{0}\}$, it also applies to the M-LWE problem from Definition 1.15.

Lemma 2.4 (ext-M-LWE^1 **to** ext-M-LWE$^\ell$)

Let $\lambda, n, k, m, \ell, q$ be positive integers such that $\ell \leq \mathsf{poly}(\lambda)$. Let K be a number field of degree n, and R its ring of integers. Let $\mathscr{D}_s, \mathscr{D}_e$ be two discrete distributions over R, and $\mathcal{Z} \subseteq R^m$. There is a PPT reduction from ext-M-LWE$^1_{n,k,m,q,\mathscr{D}_s,\mathscr{D}_e,\mathcal{Z}}$ to ext-M-LWE$^\ell_{n,k,m,q,\mathscr{D}_s,\mathscr{D}_e,\mathcal{Z}}$ such that $\varepsilon_{\text{ext-M-LWE},\ell} \leq \ell\varepsilon_{\text{ext-M-LWE},1}$

Proof (Lemma 2.4). Let $\mathcal{O}$ be an oracle for ext-M-LWE$^\ell_{n,k,m,q,\mathscr{D}_s,\mathscr{D}_e,\mathcal{Z}}$. For each $i \in [\![0, \ell]\!]$, we denote by $\mathcal{H}_i$ the hybrid distribution defined as follows.

$\mathcal{H}_i$: Sample $\mathbf{A} \hookleftarrow U(R_q^{m\times k})$, $\mathbf{S} \hookleftarrow \mathscr{D}_s^{k\times i}$ and $\mathbf{E} \hookleftarrow \mathscr{D}_e^{m\times \ell}$. Parse $\mathbf{E}_1$ to be the submatrix of $\mathbf{E}$ composed of the first i columns. Compute $\mathbf{B}_1 = \mathbf{AS} + \mathbf{E}_1 \bmod qR$ and $\mathbf{h} = \mathbf{E}^T\mathbf{z}$, and sample $\mathbf{B}_2 \hookleftarrow U(R_q^{m\times \ell - i})$.
Output: $(\mathbf{A}, [\mathbf{B}_1|\mathbf{B}_2], \mathbf{h})$.

By definition, we have $\text{Adv}_{\text{ext-M-LWE},\ell}[\mathcal{O}] = \left|\mathbb{P}[\mathcal{O}(\mathcal{H}_\ell) = 1] - \mathbb{P}[\mathcal{O}(\mathcal{H}_0) = 1]\right|$. The reduction $\mathcal{A}$ works as follows.

1. Sample $\mathbf{z} \hookleftarrow U(\mathcal{Z})$ and get $(\mathbf{A}, \mathbf{b}, h = \langle \mathbf{z}, \mathbf{e}\rangle)$ as input of ext-M-LWE$^1_{n,k,m,q,\mathscr{D}_s,\mathscr{D}_e,\mathcal{Z}}$.
2. Sample $i^\star \hookleftarrow U([\![\ell]\!])$.
3. Sample $\mathbf{S} \hookleftarrow \mathscr{D}_s^{k\times i^\star - 1}$, $\mathbf{E}_1 \hookleftarrow \mathscr{D}_e^{m\times i^\star - 1}$, $\mathbf{E}_2 \hookleftarrow \mathscr{D}_e^{m\times \ell - i^\star}$ and finally $\mathbf{B}_2 \hookleftarrow U(R_q^{m\times \ell - i^\star})$.
4. Compute $\mathbf{B}_1 = \mathbf{AS} + \mathbf{E}_1 \bmod qR$, $\mathbf{B} = [\mathbf{B}_1|\mathbf{b}|\mathbf{B}_2]$ and $\mathbf{h} = [\mathbf{z}^T\mathbf{E}_1|h|\mathbf{z}^T\mathbf{E}_2]^T$.
5. Return $\mathcal{O}((\mathbf{A}, \mathbf{B}, \mathbf{h}))$.

If $\mathbf{b}$ is uniform, then the distribution in 5. is exactly $\mathcal{H}_{i^\star - 1}$ whereas if $\mathbf{b}$ is of the form $\mathbf{As} + \mathbf{e} \bmod qR$, then the distribution is $\mathcal{H}_{i^\star}$. By a standard hybrid argument, the oracle can distinguish between the two for some $i^\star$ if it can distinguish between $\mathcal{H}_0$ and $\mathcal{H}_\ell$. So the output is correct over the randomness of $i^\star$. Since $i^\star$ is uniformly chosen we have

$$\text{Adv}_{\text{ext-M-LWE},1}[\mathcal{A}] = \left|\sum_{i^\star \in [\![\ell]\!]} \frac{1}{\ell}\mathbb{P}[\mathcal{A}(\mathcal{H}_{i^\star}) = 1] - \sum_{i^\star \in [\![\ell]\!]} \frac{1}{\ell}\mathbb{P}[\mathcal{A}(\mathcal{H}_{i^\star - 1}) = 1]\right| = \frac{1}{\ell}\text{Adv}_{\text{ext-M-LWE},\ell}[\mathcal{O}].$$

By optimizing over $\mathcal{O}$, we can prove that $\varepsilon_{\text{ext-M-LWE},1} \geq \text{Adv}_{\text{ext-M-LWE},1}[\mathcal{A}] = \varepsilon_{\text{ext-M-LWE},\ell}/\ell$.

2.3.3 Reduction to the Decision Version

We now provide the final step of the overall reduction, by reducing to the M-LWE problem with η-bounded secret using a sequence of hybrids. The idea is to use the set $\mathcal{Z}$ of the ext-M-LWE problem as our set of secrets.

To facilitate understanding, we start by illustrating the high level idea of the proof of Lemma 2.5 in Figure 2.3. Given an instance $(\mathbf{A}, \mathbf{Az} + \mathbf{e})$ of M-LWE$_{U(S_\eta)}$, our goal is to show that it is computationally indistinguishable from $(\mathbf{A}, \mathbf{b})$, where $\mathbf{b}$ is a uniformly random vector. To do so, we first decompose the continuous Gaussian error vector $\mathbf{e}$ into $-\mathbf{Nz} + \mathbf{e}'$, by using properties of Gaussian distributions. We then make use of a similar lossy argument as for the previous reduction of Section 2.2 by replacing the random matrix $\mathbf{A}$ by a lossy matrix $\mathbf{A}' = \mathbf{BC} + \mathbf{N}$. As opposed

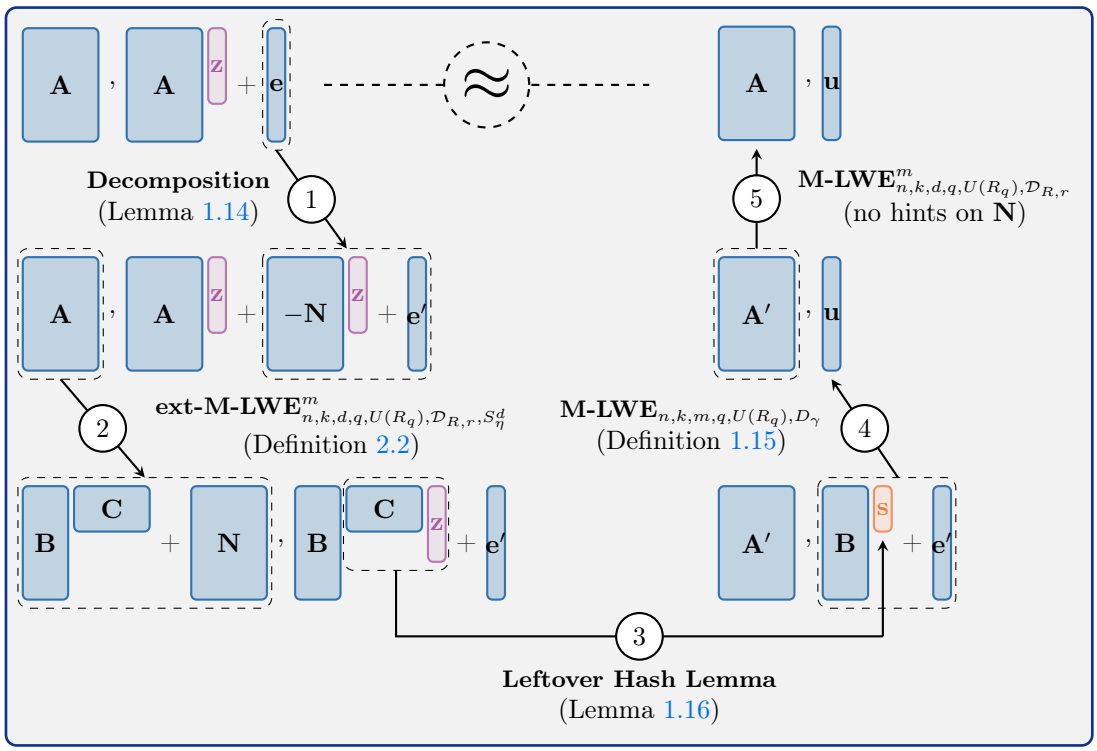


Figure 2.3: Summary of the proof of Lemma 2.5

to the proof from Section 2.2, we can't simply argue with the hardness of multiple-secrets M-LWE as the second part of the sample depends on the noise matrix $\mathbf{N}$. This is the motivation for introducing the ext-M-LWE problem, where we allow for additional information with respect to the noise. We then use the same leftover hash lemma as before to replace the product $\mathbf{Cz}$ by a uniformly random vector $\mathbf{s}$. Assuming the hardness of M-LWE, the term $\mathbf{Bs}+\mathbf{e}'$ is computationally indistinguishable form a uniform vector $\mathbf{u}$. We conclude the proof by re-replacing the lossy matrix $\mathbf{A}'$ by the original uniform matrix $\mathbf{A}$. We also insist on the fact the lossy matrix decomposition $\mathbf{BC}+\mathbf{N}$ does not encompass the ring case ($d=1$) as for the reduction of Section 2.2.

Lemma 2.5 (ext-M-LWE **to** M-LWE$_{U(S_\eta)}$)

Let $\lambda, n, k, d, m, \eta, q$ be positive integers, and $t, t' > 0$ arbitrary reals. Let K be a monogenic number field of degree n and R its ring of integers. We assume that q is prime, and that $d \geq k \cdot \log_{2\eta+1} q + \omega(\log_{2\eta+1} n)$. Further, let r, s, γ be positive reals such that $s = rB_\eta\sqrt{2d}$, $\gamma = rB_\eta\sqrt{d}$ and $r \geq \sqrt{2}\eta_\varepsilon(R^d)$ for some $\varepsilon \in (0, 1/2)$, where $B_\eta = \max_{x\in S_\eta} \|\sigma(x)\|_\infty$. There is a PPT reduction from ext-M-LWE$^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r},S^d_\eta}$, M-LWE$^1_{n,k,m,q,U(R_q),D_\gamma}$, and M-LWE$^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r}}$ to M-LWE$_{n,d,m,q,U(S_\eta),\Psi_{\leq s}}$. More precisely, if $\varepsilon_{\text{ext-M-LWE}}, \varepsilon^{(1)}_{\text{M-LWE}}, \varepsilon^{(m)}_{\text{M-LWE}}$ are the hardness bounds of the formers respectively, and $\varepsilon_{\text{M-LWE},\eta}$ that of the latter, it holds that

$$\varepsilon_{\text{M-LWE},\eta} \leq \varepsilon_{\text{ext-M-LWE}} + \varepsilon^{(1)}_{\text{M-LWE}} + \varepsilon^{(m)}_{\text{M-LWE}} + 2m\varepsilon + \frac{1}{2}\sqrt{\left(1 + \frac{q^k}{(2\eta+1)^d}\right)^n - 1}.$$

Note that the problem M-LWE$^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r}}$ is exactly ext-M-LWE$^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r},\{\mathbf{0}\}}$ where the set of allowed hints is $\{\mathbf{0}\}$, meaning no hints is given. As a result, it is trivially harder than ext-M-LWE$^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r},S^d_\eta}$, which is also why it is not specified in Figure 2.2.

Proof (Lemma 2.5). We are given an instance $(\mathbf{A}, \mathbf{Az}+\mathbf{e} \bmod qR)$ of M-LWE$_{n,d,m,q,U(S_\eta),\Psi_{\leq s}}$, with $\mathbf{A} \hookleftarrow U(R_q^{m\times d})$, $\mathbf{z} \hookleftarrow U(S_\eta^d)$, and $\mathbf{e} \hookleftarrow D^m_{\mathbf{r}}$ with width vector $\mathbf{r}$ defined by $r_j^2 = \gamma^2 +$

$r^2 \sum_{i\in[\![d]\!]} |\sigma_j(z_i)|^2$. We have $\|\mathbf{r}\|_\infty = \sqrt{\gamma^2 + r^2\|\mathbf{z}\|_{2,\infty}^2}$, as well as $\|\mathbf{z}\|_{2,\infty}^2 \le \sum_{i\in[\![d]\!]} \|\sigma(z_i)\|_\infty^2$. Recalling the parameter $B_\eta = \max_{x\in S_\eta} \|\sigma(x)\|_\infty$, that can be upper-bounded by $n\eta$ for cyclotomics by Lemma 1.1, we get $\|\mathbf{r}\|_\infty \le \sqrt{\gamma^2 + r^2 d B_\eta^2} = rB_\eta\sqrt{2d} = s$. The objective is now to show that $(\mathbf{A}, \mathbf{Az} + \mathbf{e} \bmod qR)$ is computationally indistinguishable from uniform. To do so, we define different hybrid distributions as follows, and prove that each one is indistinguishable from the next.

$\mathcal{H}_1$ Sample $(\mathbf{A}, \mathbf{b} = \mathbf{Az}+\mathbf{e} \bmod qR)$ as in the M-LWE$_{n,d,m,q,U(S_\eta),D_\mathbf{r}}$ problem as described above.
Output: $(\mathbf{A}, \mathbf{b})$.

$\mathcal{H}_2$ Sample $\mathbf{N} \hookleftarrow \mathcal{D}_{R^{m\times d},r}$, $\mathbf{e}' \hookleftarrow D_\gamma^m$, and $\mathbf{A}, \mathbf{z}$ as in $\mathcal{H}_1$.
Output: $(\mathbf{A}, \mathbf{Az} - \mathbf{Nz} + \mathbf{e} \bmod qR)$.

$\mathcal{H}_3$ Sample $\mathbf{B} \hookleftarrow U(R_q^{m\times k})$, $\mathbf{C} \hookleftarrow U(R_q^{k\times d})$ and $\mathbf{N}, \mathbf{z}, \mathbf{e}'$ as in $\mathcal{H}_2$.
Output: $(\mathbf{A}' = \mathbf{BC} + \mathbf{N} \bmod qR, \mathbf{A}'\mathbf{z} - \mathbf{Nz} + \mathbf{e}' \bmod qR = \mathbf{BCz} + \mathbf{e}' \bmod qR)$.

$\mathcal{H}_4$ Sample $\mathbf{s} \hookleftarrow U(R_q^k)$ and $\mathbf{B}, \mathbf{C}, \mathbf{N}, \mathbf{e}'$ as in $\mathcal{H}_3$.
Output: $(\mathbf{A}' = \mathbf{BC} + \mathbf{N} \bmod qR, \mathbf{Bs} + \mathbf{e}' \bmod qR)$.

$\mathcal{H}_5$ Sample $\mathbf{u} \hookleftarrow U(\mathbb{T}_q^m)$ and $\mathbf{B}, \mathbf{C}, \mathbf{N}$ as in $\mathcal{H}_4$.
Output: $(\mathbf{A}' = \mathbf{BC} + \mathbf{N} \bmod qR, \mathbf{u})$.

$\mathcal{H}_6$ Sample $\mathbf{A} \hookleftarrow U(R_q^{m\times d})$ and $\mathbf{u}$ as in $\mathcal{H}_5$.
Output: $(\mathbf{A}, \mathbf{u})$.

From $\mathcal{H}_1$ to $\mathcal{H}_2$: We first claim that $\Delta([-\mathbf{Nz} + \mathbf{e}']_i, \mathbf{e}_i) \le 2\varepsilon$ for all $i \in [\![m]\!]$. Indeed, $(1/r^2 + \|\mathbf{z}\|_{2,\infty}^2/\gamma^2)^{-1/2} \ge r/\sqrt{2} \ge \eta_\varepsilon(R^d)$. If $\mathbf{n}_i \in R^d$ denotes the i-th row of $\mathbf{N}$, Lemma 1.14 yields the claim since we have $[-\mathbf{Nz} + \mathbf{e}']_i = \langle \mathbf{n}_i, -\mathbf{z}\rangle + e'_i$, thus giving $\Delta(-\mathbf{Nz} + \mathbf{e}', \mathbf{e}) \le 2m\varepsilon$. Lemma 1.7 gives

$$\left|\mathbb{P}[\mathcal{A}(\mathcal{H}_0) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{H}_1) = 1]\right| \le 2m\varepsilon. \tag{2.1}$$

From $\mathcal{H}_2$ to $\mathcal{H}_3$: We argue that a distinguisher between $\mathcal{H}_2$ and $\mathcal{H}_3$ can be used to derive an adversary $\mathcal{B}_1$ for ext-M-LWE$^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r},S_\eta^d}$ with the same advantage. To do so, $\mathcal{B}_1$ transforms the samples from the challenger of the ext-M-LWE problem into samples defined in $\mathcal{H}_1$ or the ones in $\mathcal{H}_2$ depending on whether or not the received samples are uniform. In the uniform case, $(\mathbf{C}^T, \mathbf{A}^T, \mathbf{Nz})$ can be efficiently transformed into a sample from $\mathcal{H}_1$. Note that $\mathbf{A}^T$ indeed corresponds to the uniform case of ext-M-LWE, because $\mathbf{A}^T$ is uniform over $R_q^{d\times m}$ Additionally, the transpose operator comes from the fact that the hints are $\mathbf{Nz}$, which corresponds to m error vectors of size d. So the second component is indeed of size $d \times m$ as required. In the other case, if we apply the same transformation to the ext-M-LWE sample $(\mathbf{C}^T, \mathbf{C}^T\mathbf{B}^T + \mathbf{N}^T \bmod qR, \mathbf{Nz})$ where $\mathbf{B}^T$ and $\mathbf{N}^T$ are the secret and error matrix respectively, it leads to a sample from $\mathcal{H}_2$. The (randomized) transformation can be described by sampling $\mathbf{e}'$ from D_γ^m and outputting $f(\mathbf{X}_1, \mathbf{X}_2, \mathbf{x}_3) = (\mathbf{X}_2^T, \mathbf{X}_2^T\mathbf{z} - \mathbf{x}_3 + \mathbf{e}' \bmod qR)$. Hence, $\mathcal{B}_1$ is a distinguisher for ext-M-LWE$^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r},S_\eta^d}$, and

$$\left|\mathbb{P}[\mathcal{A}(\mathcal{H}_1) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{H}_2) = 1]\right| = \mathrm{Adv}[\mathcal{B}_1] \le \varepsilon_{\text{ext-M-LWE}}. \tag{2.2}$$

From $\mathcal{H}_3$ to $\mathcal{H}_4$: The leftover hash lemma stated in Lemma 1.16 yields that $(\mathbf{C}, \mathbf{Cz})$ is within statistical distance at most $\delta = \frac{1}{2}\sqrt{(1 + q^k/(2\eta+1)^d)^n - 1}$ from $(\mathbf{C}, \mathbf{s})$. Note that the condition $d \ge k\log_{2\eta+1} q + \omega(\log_{2\eta+1} n)$ implies $\delta \le \mathsf{negl}(\lambda)$. Lemma 1.7 yields

$$\left|\mathbb{P}[\mathcal{A}(\mathcal{H}_3) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{H}_4) = 1]\right| \le \delta. \tag{2.3}$$

From $\mathcal{H}_4$ to $\mathcal{H}_5$: A distinguisher between $\mathcal{H}_4$ and $\mathcal{H}_5$ can be used to derive an adversary $\mathcal{B}_2$ for M-LWE$_{n,k,m,q,U(R_q),D_\gamma}$. For that, $\mathcal{B}_2$ applies the efficient transformation to the samples from the M-LWE challenger, which turns $(\mathbf{B}, \mathbf{u})$ into a sample from $\mathcal{H}_5$ in the uniform case,

and $(\mathbf{B}, \mathbf{Bs} + \mathbf{e}' \bmod qR)$ into a sample from $\mathcal{H}_4$ in the M-LWE case. The transformation is given by $g(\mathbf{X}_1, \mathbf{x}_2) = (\mathbf{X}_1\mathbf{C} + \mathbf{N} \bmod qR, \mathbf{x}_2)$, where $\mathbf{C}, \mathbf{N}$ are sampled as in $\mathcal{H}_3$. Therefore, $\mathcal{B}_2$ is a distinguisher for $\text{M-LWE}_{n,k,m,q,U(R_q),D_\gamma}$ such that

$$\left|\mathbb{P}[\mathcal{A}(\mathcal{H}_4) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{H}_5) = 1]\right| = \text{Adv}[\mathcal{B}_2] \leq \varepsilon^{(1)}_{\text{M-LWE}}. \tag{2.4}$$

From $\mathcal{H}_5$ to $\mathcal{H}_6$: We now change $\mathbf{A}'$ back to uniform. With the same argument as before, we can construct an adversary $\mathcal{B}_3$ for $\text{ext-M-LWE}^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r},\{\mathbf{0}\}} = \text{M-LWE}^m_{n,k,d,q,U(R_q),\mathcal{D}_{R,r}}$ based on a distinguisher between $\mathcal{H}_5$ and $\mathcal{H}_6$. It transforms $(\mathbf{C}^T, \mathbf{A}'^T, \mathbf{N} \cdot \mathbf{0})$ into a sample from $\mathcal{H}_5$ (M-LWE case) and $(\mathbf{C}^T, \mathbf{A}^T, \mathbf{N} \cdot \mathbf{0})$ into a sample from $\mathcal{H}_6$ (uniform case). The transformation samples $\mathbf{u} \hookleftarrow U(\mathbb{T}_q^m)$ as in $\mathcal{H}_5$ and outputs $h(\mathbf{X}_1, \mathbf{X}_2, \mathbf{x}_3) = (\mathbf{X}_2^T, \mathbf{u})$. We then get

$$\left|\mathbb{P}[\mathcal{A}(\mathcal{H}_5) = 1] - \mathbb{P}[\mathcal{A}(\mathcal{H}_6) = 1]\right| = \text{Adv}[\mathcal{B}_3] \leq \varepsilon^{(m)}_{\text{M-LWE}}. \tag{2.5}$$

Putting Equations (2.1), (2.2), (2.3), (2.4), (2.5) altogether yields the result.

2.4 Conclusion

The two reductions we presented are fairly similar in the sense that both aim at replacing the short secret in S_η by a large secret in R_q with the help of a lossy matrix transformation. The main difference lies in how to deal with the error distribution in the meantime. One relies on a noise flooding argument while the other uses hints. We summarize these results in Table 2.1.

	Section 2.2	**Section 2.3**
Variant	Search	Decision
Rank Condition	$k \log_{2\eta+1} q + \log_{2\eta+1} \Omega(n)$	$(k+1) \log_{2\eta+1} q + \omega(\log_{2\eta+1} n)$
Noise Rate s/r	$\Theta(\eta n^{3/2}\sqrt{d}(\sqrt{m} + \sqrt{d}))^{(+)}$	$\Theta(\eta^2 n^2 \sqrt{d})$
Number Fields	Monogenic	Cyclotomic
Modulus	Prime	Prime with low splitting
Noise Argument	Flooding (RD_2)	Hints (ext-M-LWE)
Unstructured Reduction	[GKPV10]	[BLP+13]
Approx. factor $\gamma^{(\star)}$	$\widetilde{O}\left(\frac{k^{3/2}(\sqrt{m} + \sqrt{d}) \cdot n^2 \eta \sqrt{2d}}{s/q}\right)$	$\widetilde{O}\left(\frac{4k^{3/2} \cdot n^3 \eta^2 \sqrt{d}}{s/q}\right)$

Table 2.1: Comparison of the two reductions establishing the hardness of M-LWE with a short uniform secret.
$^{(+)}$ Noise rate for general cylotomic fields. It is improved by $\sqrt{n}$ for power-of-two cyclotomic fields.
$^{(\star)}$ For $\text{Module-GapSVP}_\gamma$ over modules of power-of-two cyclotomic rings.

Even though concrete primitives use even smaller parameters chosen through cryptanalysis, which are not covered by our proofs, our work still proves the robustness of the assumption in certain regimes. As a result, if heuristically chosen parameters in cryptographic constructions turn out to be weaker than expected, one could always increase them to match our reduction and get strong hardness guarantees. The results of this chapter thus inspire confidence in the Module LWE assumption with short secret distributions, showing that the algebraic structure does not fundamentally weaken its hardness.

From a theoretical perspective, our results point out two main caveats. The most concerning one is the rank condition under which these reductions work. Although the increase from k to $k \log_{2\eta+1} q$ seems reasonable as it preserves the entropy of the secret distribution, reaching lower ranks is of high importance in order to close the gap between theory and practice. As an example, when looking at the result of Section 2.2, choosing $\eta = 1$, $n = 128$ and $q \approx 2^{47}$ to match the parameters of our signature in Section 6.2, we would need $d \geq 34$ while the rank we actually use is $d = 10$. The second limitation is the use of Gaussian error distributions, which are much larger than short uniform distributions, regardless of the fact that they are taken in the Minkowski embedding. The choice for Gaussian distributions is that they have good geometric and probabilistic properties which make the reductions easier. Departing from this choice turns out to be challenging, especially while keeping a short secret distribution.

We thus investigate another direction in Chapter 3 by analyzing the hardness of M-LWE by first changing the error distribution, the goal being to bypass the aforementioned limitations.

3

Hardness of Module Learning With Errors with Small Error

We now focus on the hardness of Module Learning With Errors (M-LWE) with small uniform of secret *and* error. We first establish the hardness with standard uniform secret (modulo q) and an error that is uniformly chosen in a small interval by adopting the perspective of function families. We conclude by a Hermite Normal Form transformation to move to a small uniform secret.

The work presented in this chapter is based on a paper with my co-authors Katharina Boudgoust, Adeline Roux-Langlois and Weiqiang Wen.

[BJRW23] **On the Hardness of Module Learning With Errors with Short Distributions**. Published at IACR Journal of Cryptology 2023.

Contents

3.1 Introduction

In response to the results of Chapter 2, we adopt a new perspective in order to prove the hardness of M-LWE with short distributions of secret *and* error. For that we temporarily forget about the secret distribution, which we set to be $\mathscr{D}_s = U(R_q)$, and focus on the error. The main drawbacks of the previous approach was the rank increase and the use of Gaussian distributions for the error. The goal is then to move to error in S_η^m while minimizing the rank increase.

The hardness of LWE with error uniformly distributed below η with $\eta \ll q$ was first studied by Micciancio and Peikert [MP13]. They proved that the LWE function $(\mathbf{s}, \mathbf{e}) \mapsto \mathbf{A}\mathbf{s}+\mathbf{e} \bmod q\mathbb{Z}$ is one-way with respect to inputs $\mathbf{e}$ uniform over $\{0, \dots, \eta-1\}^m$, provided that the number of samples

m is at most $d(1 + O(\log_2 \eta / \log_2 d))$. The one-wayness is proven under the hardness of general lattice problems over lattices of rank $O(d \log_2 \eta / \log_2 d)$. It was then extended to non-uniform binary distributions by Sun et al. [STA20], proving that the maximum number of samples must be $m = d(1 + O(p(d)/\log_2 d))$, where $p(d)$ is the probability of getting 1 from the error distribution. The proof of [MP13] corresponds to $p(d) = 1/2$. However, no results on the hardness of M-LWE with small uniform error were known, even though the assumption is extensively used in efficient cryptographic constructions. For example, our signature schemes of Chapter 5 and Section 6.2 use errors drawn from $U(S_1)$, and the standardized signature scheme Dilithium [DKL+18] uses $\mathscr{D}_e = U(S_2)$ or $U(S_4)$. Making progress in this direction would thus increase our confidence in the security of such schemes.

3.1.1 Our Contributions

This chapter focuses on the module setting with short uniform secret *and* error distributions and provides, to the best of our knowledge, the first such hardness result on an algebraic form of LWE. We start by establishing the hardness of the search version sM-LWE with a large secret and a uniform centered η-bounded error, under specific restrictions on η. From there, we obtain the hardness of sM-LWE with short secret *and* error using a Hermite Normal Form transformation. Let us now give a more technical overview of the proof method and implications.

Our goal being to limit the rank increase, we depart from the proof method used in Chapter 2. For that, we adopt the approach of Micciancio and Peikert [MP13] consisting in proving that the M-LWE function $(\mathbf{s}, \mathbf{e}) \mapsto \mathbf{A}\mathbf{s} + \mathbf{e} \bmod qR$ is one-way, with respect to $\mathbf{s} \sim U(R_q^d)$ and $\mathbf{e} \sim U(S_\eta^m)$. To do so, we prove the *one-wayness* of the M-ISIS function $\mathbf{e} \mapsto \mathbf{A}'^T\mathbf{e} \bmod qR$ and use the duality between M-LWE and M-ISIS to conclude, which we also formalize in the module setting. The one-wayness of the function is ensured by two properties, namely the uninvertibility and the second preimage resistance, which we prove using statistical arguments.

We obtain similar results to [MP13] in terms of the number of samples using asymptotic arguments. However, the asymptotic approach is not optimal for very small values of d. The security of practical schemes is indeed driven by the ring degree n as we wish to use a small rank d for efficiency. An asymptotic analysis is then not suited for achieving very small ranks d and very small error bounds η simultaneously. To overcome this problem, we use a more fine-grained approach using tighter calculations rather than hiding constants in asymptotic notations. This leads to more complicated conditions on the parameters, and we thus evaluate these conditions numerically to determine some concrete parameters that are encompassed by the result. We observe that even with our approach, we cannot set d and η arbitrarily small independently of each other. Our work still highlights a trade-off between η and d, albeit less restrictive than the one stemming from the leftover hash lemma. It shows that in order to reach a very small error, e.g. ternary, the module rank d has to be somewhat logarithmic, but independently of q. In particular, we can still reach a small error size η for constant module ranks, but η will not be arbitrarily small. The result also gives a condition on the maximal number of samples m we can provide with such small uniform error. In particular, we get $m \leq d(1 + o(\log_2 \eta))$ which is similar to what is obtained in [MP13]. Then, to prove the hardness of sM-LWE with small error *and* secret with m' samples, we need to have the hardness of sM-LWE with small error and $m' + d$ samples. This restriction makes it difficult to achieve small error and secret at the same time for a large enough m'. We discuss this transformation in more details in Sections 3.4 and 3.5.

The sM-LWE problem can be seen as a linear system of equations (d variables and m equations over R_q or nd variables and nm equations over $\mathbb{Z}_q$) with noise. The presence of noise or error is what makes the problem difficult to solve. The motivation is therefore to determine the threshold of noise to add to the equations above which the problem is proven hard. Note that the number of equations characterized by m and the distribution of the error need to be chosen carefully with respect to one another. For example, an attack by Arora and Ge [AG11] uses the m samples to build noiseless polynomial equations of degree η, where η is a bound on the error coefficients. If m is sufficiently large, root finding algorithms can perform well on the latter. In particular, if $\eta = 1$ (ternary), then $m \approx d^3$ samples is enough to solve LWE in polynomial time. The attack can also be applied to M-LWE as one equation over R_q gives n equations over $\mathbb{Z}_q$. Our proof independently provides conditions on the number of samples under which these attack do not apply. We discuss it further in Sections 3.5 and 9.2.3.

We note that our results apply to general number fields, which can be of independent theoretical interest.

We again warn the reader that the Gaussian error distributions in this chapter are chosen with respect to the Minkowski embedding σ_H. We refer to Section 1.3.2 for more details.

3.2 Duality between M-LWE and M-ISIS

Although the Short Integer Solution and Learning With Errors problems were introduced somewhat independently, they are very much similar in essence. More precisely, it is well known that they can be seen as *dual* of each other. Recalling the interpretation of SIS as the SVP_γ problem over $\mathcal{L}_q^\perp(\mathbf{A}^T)$ and that of LWE as the CVP_γ problem over $\mathcal{L}_q(\mathbf{A})$ for some $\mathbf{A} \in \mathbb{Z}_q^{m\times d}$, we can see the duality between SIS and LWE as a consequence of the duality between $\mathcal{L}_q^\perp(\mathbf{A}^T)$ and $\mathcal{L}_q(\mathbf{A})$. It indeed holds that $\mathcal{L}_q(\mathbf{A})^\star = q^{-1}\mathcal{L}_q^\perp(\mathbf{A}^T)$ and that $\mathcal{L}_q^\perp(\mathbf{A}^T)^\star = q^{-1}\mathcal{L}_q(\mathbf{A})$. We can therefore switch between the two problems to identify specific properties. This duality between SIS and LWE was formalized by Micciancio and Mol [MM11] using an interpretation with function families. Although the generalization to the module setting seems natural because of the duality of the module lattices $\mathcal{L}_q^\perp(\mathbf{A}^T)$ and $\mathcal{L}_q(\mathbf{A})$ for $\mathbf{A} \in R_q^{m\times d}$, we formalize it in this section.

3.2.1 M-LWE and M-ISIS as Function Families

The purpose of this section is to interpret the hardness of the M-LWE and M-ISIS problems as some security properties of the corresponding function families being satisfied. For example, we can expect that certain families of functions are hard to invert, or have outputs indistinguishable from uniformly random ones. We thus recall the notion of *function families* as well as the standard security properties that we desire from them in this chapter. A function family $\mathcal{F}$ over a set of functions F is a probability distribution over F, where each function of F has domain X and range Y. In our work, we only deal with functions that have an unambiguous and public description in some specified format, e.g., they can be represented by a public matrix $\mathbf{A}$. Hence, we say that an adversary is given a function f as input when it is given its public representation.

Definition 3.1 (Security Properties of Function Families)

Let X, Y be two sets, and F a set of functions from X to Y. Let $\mathcal{F}, \mathcal{G}$ be two function families over F. Let $\mathcal{X}$ be a probability distribution over X, and $\varepsilon \in (0,1)$.

Indistinguishability. $\mathcal{F}$ and $\mathcal{G}$ are ε-indistinguishable if for all PPT algorithm $\mathcal{A}$, it holds
$$\left|\mathbb{P}_{f\sim\mathcal{F}}[\mathcal{A}(f)=1] - \mathbb{P}_{g\sim\mathcal{G}}[\mathcal{A}(g)=1]\right| \leq \varepsilon.$$

Pseudorandomness. $(\mathcal{F},\mathcal{X})$ is ε-pseudorandom if for all PPT algorithm $\mathcal{A}$, it holds
$$\left|\mathbb{P}_{(f,x)\sim\mathcal{F}\times\mathcal{X}}[\mathcal{A}(f,f(x))=1] - \mathbb{P}_{(f,y)\sim\mathcal{F}\times U(Y)}[\mathcal{A}(f,y)=1]\right| \leq \varepsilon.$$

Second preimage resistance. $(\mathcal{F},\mathcal{X})$ is ε-second preimage resistant if for all PPT algorithm $\mathcal{A}$, it holds
$$\mathbb{P}_{\substack{(f,x)\sim\mathcal{F}\times\mathcal{X}\\ x'\leftarrow\mathcal{A}(f,x)}}[x \neq x' \wedge f(x) = f(x')] \leq \varepsilon.$$

Uninvertibility. $(\mathcal{F},\mathcal{X})$ is ε-uninvertible if for all PPT algorithm $\mathcal{A}$, it holds that
$$\mathbb{P}_{(f,x)\sim\mathcal{F}\times\mathcal{X}}[\mathcal{A}(f,f(x))=x] \leq \varepsilon.$$

One-wayness. $(\mathcal{F},\mathcal{X})$ is ε-one-way if for all PPT algorithm $\mathcal{A}$, it holds that
$$\mathbb{P}_{(f,x)\sim\mathcal{F}\times\mathcal{X}}[f(\mathcal{A}(f,f(x))) = f(x)] \leq \varepsilon.$$

We now give useful sufficient conditions to ensure some of these security properties.

Lemma 3.1 ([MP13, Lem. 2.2])

Let $\mathcal{F}$ be a family of functions computable in polynomial time. Let $\mathcal{X}$ be a distribution on X. If $(\mathcal{F},\mathcal{X})$ is ε-uninvertible and ε'-second preimage resistant, then it is also $(\varepsilon+\varepsilon')$-one-way.

Lemma 3.2 ([MP13, Lem. 2.4])

Let $\mathcal{F}$ be a function family with finite domain X. For $\varepsilon = \mathbb{E}_{f\sim\mathcal{F}}\left[\left|f(X)\right|\right]/|X|$, it holds that $(\mathcal{F}, U(X))$ is ε-uninvertible, even against unbounded adversaries.

Lemma 3.3 ([MP13, Lem. 2.5])

Let $\mathcal{F}$ be a function family with domain X and range Y, and $\mathcal{G}$ be an efficiently sampleable family of efficiently computable functions with domain $X' \supseteq Y$. Let $\mathcal{X}$ be a distribution on X. If $(\mathcal{F}, \mathcal{X})$ is ε-uninvertible, then so is $(\mathcal{G}\circ\mathcal{F}, \mathcal{X})$.

The M-ISIS and M-LWE problems can then be interpreted as function families whose distribution rely on the distribution of $\mathbf{A}$. Their uninvertibility or one-wayness therefore captures the hardness of the corresponding search problem, while their pseudorandomness captures the hardness of the decision problem. Note that we can then define a decision variant of M-ISIS which in fact corresponds to a regularity result as the ones from Lemma 1.16 or Lemma 1.19, but that is formulated as a computational assumption.

Definition 3.2 (M-ISIS and M-LWE Function Families)

Let n, d, m, q be positive integers. Let R be the ring of integers of a number field of degree n, and $X \subseteq R^m$. The M-ISIS(n, d, m, q, X) function family is the distribution obtained by sampling a matrix $\mathbf{A} \hookleftarrow U(R_q^{m\times d})$, and outputting $f_{\mathbf{A}}$ defined by $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A}^T\mathbf{x} \bmod qR$ for all $\mathbf{x} \in X$.

The M-LWE(n, d, m, q, X) function family is the distribution obtained by sampling $\mathbf{A} \hookleftarrow U(R_q^{m\times d})$ and outputting $g_{\mathbf{A}}$ defined by $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod qR$ for all $(\mathbf{s}, \mathbf{e}) \in R_q^d \times X$.

We attract the attention of the reader to the fact that the M-ISIS function is defined with a transpose compared to Definition 1.14. We only define them with discrete inputs (i.e., discrete error for M-LWE) because this is the only version needed in this chapter. Also, when using the M-LWE function family, we always implicitly assume that the distribution on the first input $\mathbf{s}$ is always $U(R_q^d)$ and omit it from the notations.

The objective of this chapter is then to show that $(\text{M-LWE}(n, d, m, q, S_\eta^m), U(S_\eta^m))$ is ε-one-way for a negligible ε. This would then show that $\text{sM-LWE}_{n,d,m,q,U(R_q),U(S_\eta)}$ verifies $\varepsilon_{\text{sM-LWE}} \leq \varepsilon$ thus proving the hardness of the search variant. In most M-LWE-based schemes, the secret key is $(\mathbf{s}, \mathbf{e})$ and the public key is $(\mathbf{A}, \mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod qR)$. It is therefore important to prove one-wayness and not just uninvertibility because an adversary breaking one-wayness could compute a different secret key for the same public key, which would allow them to decrypt messages, or forge signatures. It turns out that if the parameters are chosen appropriately so that the function is second preimage resistant, the uninvertibility is then equivalent to the one-wayness by Lemma 3.1.

3.2.2 Duality

In the following, we adapt the duality results from [MM11, Sec. 4.2] to the module setting. To the best of our knowledge, this hasn't been formally done before. The idea when going from M-LWE to M-ISIS is to cancel the secret part via a *parity check* matrix $\mathbf{B}$, i.e., such that $\mathbf{A}^T\mathbf{B} = \mathbf{0} \bmod qR$. The M-LWE error distribution $\mathbf{e}$ then becomes the input distribution of the M-ISIS instance with matrix $\mathbf{B}' = \mathbf{B}\mathbf{U}$ where $\mathbf{U}$ simply randomizes $\mathbf{B}$. Note that in this work we are considering a parameter regime such that the function family of M-ISIS is injective. In other words, solutions to M-ISIS are with a very high probability unique. This regime is sometimes referred to as *low-density* ISIS [Lyu12] or even more generally as a knapsack problem [MM11]. For $\mathbf{B}'$ to be well distributed, we need $\mathbf{A}$ to be non-singular which is characterized by the function $\delta(\cdot,\cdot)$ from Section 1.1.6. The upper bound derived from Lemma 1.6 for this singularity probability requires q to be unramified in order to have an easier characterization of units of R_q. Also, note that the following lemmas are only meaningful if the extra losses incurred by $\delta(\cdot,\cdot)$ are negligible, which may require to restrict the splitting of q. We elaborate on the matter in Section 3.5.

Lemma 3.4 (M-LWE to M-ISIS **(Adapted from [MM11, Lem. 4.8])**)

Let n, d, m, q be positive integers. Let R be the ring of integers of a number field of degree n. We assume that q is prime and unramified in R, and that $m \geq d+1$. Let $\mathcal{X}$ be a probability distribution on R^m. If (M-LWE$(n, d, m, q, R^m), \mathcal{X})$ is ε-uninvertible (resp. one-way, pseudorandom), then (M-ISIS$(n, m-d, m, q, R^m), \mathcal{X})$ is ε'-uninvertible (resp. one-way, pseudorandom), with $\varepsilon' = \delta(m, m-d) + \varepsilon/(1-\delta(m,d))$ (resp. $\varepsilon' = 2\delta(m, m-d) + \varepsilon/(1-\delta(m,d))$ for pseudorandomness).

Proof (Lemma 3.4). We start by describing the transformation T of [MM11] to move from M-LWE to M-ISIS. Given $(\mathbf{A}, \mathbf{b}) \in R_q^{m\times d} \times R_q^m$, where $\mathbf{A}$ is uniformly sampled, T first checks if the rows of $\mathbf{A}$ generate R_q^d. If not, T returns $\perp$. By the quantity defined in Section 1.1.6, T aborts at this step with probability $\delta(m,d)$ (which can be upper bounded from Lemma 1.6). We now condition on $\mathbf{A}$ being non-singular. From $\mathbf{A}$, T computes $\mathbf{B} \in R_q^{m\times(m-d)}$ whose columns generate the set of vectors $\mathbf{x} \in R_q^m$ that verify $\mathbf{A}^T\mathbf{x} = \mathbf{0} \bmod qR$. T samples $\mathbf{U} \hookleftarrow GL_{m-d}(R_q)$, and defines $\mathbf{B}' = \mathbf{BU}$. As $\mathbf{A}$ is uniform in the set of non-singular matrices, $\mathbf{B}'$ is uniform in the set of matrices whose rows generate R_q^{m-d}. Again, by definition of $\delta(\cdot,\cdot)$, we get $\Delta(\mathbf{B}', U(R_q^{m\times(m-d)})) \leq \delta(m, m-d)$. Finally, T computes $\mathbf{c} = \mathbf{B}'^T\mathbf{b} \bmod qR$, and returns $(\mathbf{B}', \mathbf{c})$.

Assume that there exists an adversary $\mathcal{A}$ that attacks the ε'-uninvertibility of M-ISIS. We construct $\mathcal{B}$ that breaks the ε-univertibility of M-LWE by calling $\mathcal{A}$ on the instance transformed by T. Consider $(\mathbf{A}, \mathbf{As}+\mathbf{e} \bmod qR)$, with $(\mathbf{s}, \mathbf{e}) \hookleftarrow U(R_q^d) \times \mathcal{X}$. We let E be the event $\{\mathcal{B}(\mathbf{A}, \mathbf{As}+\mathbf{e}+qR) = (\mathbf{s}, \mathbf{e})\}$. Then, it holds that

$$\begin{aligned}
\mathbb{P}[E] &= \mathbb{P}[\mathbf{A} \text{ non-singular}]\mathbb{P}[E|\mathbf{A} \text{ non-singular}] + \mathbb{P}[\mathbf{A} \text{ singular}] \underbrace{\mathbb{P}[E|\mathbf{A} \text{ singular}]}_{0 \text{ (abort)}} \\
&= (1-\delta(m,d))\mathbb{P}[\mathcal{A}(\mathbf{B}', \mathbf{c}) = \mathbf{e}|\mathbf{A} \text{ non-singular}] \\
&> (1-\delta(m,d)) \cdot (\varepsilon' - \delta(m, m-d)) \\
&= \varepsilon.
\end{aligned}$$

Indeed, by the transformation, we have

$$\begin{aligned}
(\mathbf{B}')^T\mathbf{b} \bmod qR &= (\mathbf{B}')^T\mathbf{As} + (\mathbf{B}')^T\mathbf{e} \bmod qR \\
&= (\mathbf{A}^T\mathbf{B}' \bmod qR)^T\mathbf{s} + (\mathbf{B}')^T\mathbf{e} \bmod qR \\
&= (\mathbf{B}')^T\mathbf{e} \bmod qR.
\end{aligned}$$

Then, $\mathcal{B}$ uses linear algebra to recover $\mathbf{s}$ from $\mathbf{b}-\mathbf{e}$. The proof for one-wayness is the same where $E = \{g_{\mathbf{A}}(\mathcal{B}(\mathbf{A}, \mathbf{As}+\mathbf{e} \bmod qR)) = \mathbf{As}+\mathbf{e} \bmod qR\}$ (recalling that $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{As}+\mathbf{e} \bmod qR$). For the pseudorandomness, we define $E = \{\mathcal{B}(\mathbf{A}, \mathbf{b} \text{ uniform}) = 1\}$, $E' = \{\mathcal{B}(\mathbf{A}, \mathbf{b} = \mathbf{As}+\mathbf{e} \bmod qR) = 1\}$, and F the event $\{\mathbf{A} \text{ non singular}\}$. It then holds that

$$\begin{aligned}
&|\mathbb{P}[E] - \mathbb{P}[E']| \\
&= \mathbb{P}[\mathbf{A} \text{ non-singular}] \cdot \left|\mathbb{P}[E|\mathbf{A} \text{ non-singular}] - \mathbb{P}[E'|\mathbf{A} \text{ non singular}]\right| \\
&= (1-\delta(m,d))\left|\mathbb{P}[\mathcal{A}(\mathbf{B}', \mathbf{c} \text{ uniform}) = 1|F] - \mathbb{P}[\mathcal{A}(\mathbf{B}', (\mathbf{B}')^T\mathbf{e} \bmod qR) = 1|F]\right| \\
&> (1-\delta(m,d)) \cdot (\varepsilon' - 2\delta(m, m-d)) \\
&= \varepsilon,
\end{aligned}$$

concluding the proof.

Lemma 3.5 (M-ISIS to M-LWE **(Adapted from [MM11, Lem. 4.9])**)

Let n, d, m, q be positive integers. Let R be the ring of integers of a number field of degree n. We assume that q is prime and unramified in R, and that $m \geq d+1$. Let $\mathcal{X}$ be a probability distribution on R^m. If (M-ISIS$(n, m-d, m, q, R^m), \mathcal{X})$ is ε-uninvertible (resp. one-way, pseudorandom), then (M-LWE$(n, d, m, q, R^m), \mathcal{X})$ is ε'-uninvertible (resp. one-way,

pseudorandom), with $\varepsilon' = \delta(m,d)+\varepsilon/(1-\delta(m,m-d))$ (resp. $\varepsilon' = 2\delta(m,d)+\varepsilon/(1-\delta(m,m-d))$ for pseudorandomness).

Proof (Lemma 3.5). The transformation T now works as follows. Given $(\mathbf{B},\mathbf{c}) \in R_q^{m\times(m-d)} \times R^{m-d}$ with $\mathbf{B}$ uniformly distributed, T checks whether the rows of $\mathbf{B}$ generate R_q^{m-d}. If not, it aborts, and that with probability $\delta(m, m-d)$. Conditioning on $\mathbf{B}$ being non-singular, T computes $\mathbf{A} \in R_q^{m\times d}$ which generates $\{\mathbf{x} \in R_q^m\colon \mathbf{B}^T\mathbf{x} = \mathbf{0} \bmod qR\}$. The transformation then randomizes $\mathbf{A}$ by a random matrix $\mathbf{U} \in GL_d(R_q)$ to obtain $\mathbf{A}' = \mathbf{AU}$. Similarly as in the proof of Lemma 3.4, $\Delta(\mathbf{A}', U(R_q^{m\times d})) \le \delta(m,d)$. Then, T finds a vector $\mathbf{b}$ such that $\mathbf{B}^T\mathbf{b} = \mathbf{c} \bmod qR$, and returns $(\mathbf{A}', \mathbf{b})$. Note that if $\mathbf{c} = \mathbf{B}^T\mathbf{e} \bmod qR$ for some $\mathbf{e} \hookleftarrow \mathcal{X}$, then $\mathbf{b}-\mathbf{e}$ is in the span of the columns of $\mathbf{A}'$ and therefore, there exists $\mathbf{s} \in R_q^d$ such that $\mathbf{b}-\mathbf{e} = \mathbf{A}'\mathbf{s} \bmod qR$. If $\mathbf{c}$ is uniform, we can argue that $\mathbf{b}$ is also uniform. Using the same calculations as before, we get that

$$\mathrm{Adv}[\mathcal{B}] > (1-\delta(m,m-d))\cdot(\varepsilon'-\delta(m,d)) = \varepsilon,$$

where $\mathrm{Adv}[\mathcal{B}]$ denotes the probability of breaking uninvertibility or one-wayness, or the absolute difference of probability in the case of pseudorandomness.

3.3 Computational Hardness with Small Errors

We now proceed to proving the one-wayness of the M-LWE function family with respect to a short uniform input (i.e., error) distribution, assuming the pseudorandomness of the M-LWE function family with Gaussian input. It therefore implies the hardness of sM-LWE with small uniform error from that of the decision version of M-LWE with Gaussian error. To prove the one-wayness of the M-LWE function, we prove the result in terms of M-ISIS and use Lemma 3.5 to conclude. Recall that by Lemma 3.1, it suffices to prove that M-ISIS is uninvertible and second preimage resistant with respect to this specific input distribution. We actually prove the second preimage resistance of the M-ISIS function, and the uninvertibility of a decomposition of the M-ISIS function. We then argue that these two function families are indistinguishable based on the pseudorandomness of M-ISIS (or M-LWE equivalently). The idea of the proof is summarized in Figure 3.1.

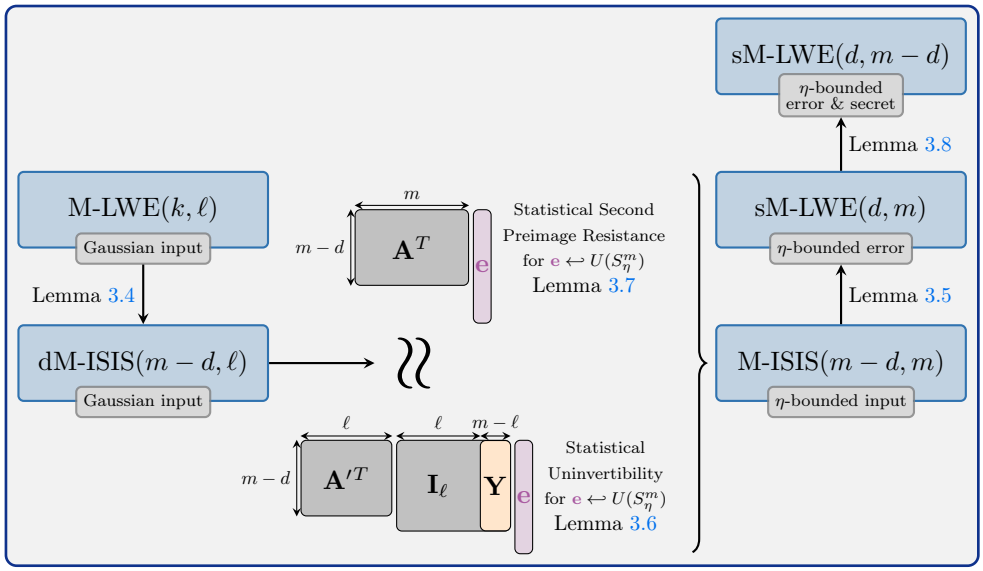


Figure 3.1: Summary of the proof of Theorem 3.1. As we only introduced the notation M-ISIS as a search problem, we write dM-ISIS to specify the decision problem which corresponds to the pseudorandomness of the M-ISIS function family. For clarity, we only mention the parameters of the function families that change in the proof, namely the rank (first parameter) and the number of samples (second parameter). Also, we have $\ell = m-d+k$, and thus $m-\ell = d-k$ which can be as small as 1.

3.3.1 Uninvertibility

In order to prove the uninvertibility of the function family $(\text{M-ISIS}(n, m-d, m, q, R^m), U(S_\eta^m))$, we decompose it into a linear (Gaussian) function family $\mathcal{G}$ and a smaller $\text{M-ISIS}(n, m-d, \ell, q, R^\ell)$ function family with $\ell \leq m$. By Lemma 3.3, it suffices to prove the uninvertibility of $(\mathcal{G}, U(S_\eta^m))$. We first define what we mean by linear (Gaussian) function family.

Definition 3.3 (Linear Gaussian Function Family)

Let n, ℓ, m be positive integers such that $m \geq \ell$ and $s > 0$. Let R be the ring of integers of a number field of degree n, and $X \subseteq R^m$. We define the function family $\mathcal{G}(n, \ell, m, s, X)$ obtained by sampling $\mathbf{Y}$ from $\mathcal{D}_{R^{\ell \times m-\ell}, s}$ and outputting $h_{\mathbf{Y}} : X \to R^\ell$ defined by $\forall \mathbf{x} \in X$, $h_{\mathbf{Y}}(\mathbf{x}) = [\mathbf{I}_\ell | \mathbf{Y}]\mathbf{x}$.

We now use Lemma 3.2 to prove that $(\mathcal{G}(n, \ell, m, s, X), U(X))$ is statistically uninvertible with uniform inputs for carefully chosen parameters. In particular, the result is only meaningful when ε_3 is negligible. This leads to involved conditions on the parameters, which we discuss in Section 3.5.

Lemma 3.6

Let n, ℓ, m, d, η be positive integers such that $m \geq \max(d, \ell)$, and $s > 0$. Let R be the ring of integers of a number field of degree n, and $X \subseteq S_\eta^m$. We define the function family $\mathcal{F} = \text{M-ISIS}(n, m-d, \ell, q, R^\ell) \circ \mathcal{G}(n, \ell, m, s, X)$. Then, for any $t \geq 0$, $(\mathcal{F}, U(X))$ is (statistically) ε_3-uninvertible for

$$\varepsilon_3 = \frac{1}{|X|\sqrt{\pi n \ell}} \left(\eta \sqrt{2\pi e} \left(1 + s \sqrt{\frac{m-\ell}{2\pi \ell}} \left(\sqrt{\ell} + \sqrt{m-\ell} + t \right) \right) \right)^{n\ell} + 2ne^{-\pi t^2}.$$

When $t = \omega(\sqrt{\log_2 \lambda})$, the second term is negligible.

Proof (Lemma 3.6). We first bound $\mathbb{E}_{h_{\mathbf{Y}} \sim \mathcal{G}}[|h_{\mathbf{Y}}(X)|]$ and use Lemma 3.2 to conclude. Let $h_{\mathbf{Y}}$ be sampled from $\mathcal{G}(n, \ell, m, s, X)$. Let $\mathbf{x} = [\mathbf{x}_1^T | \mathbf{x}_2^T]^T \in X$, with $\mathbf{x}_1 \in S_\eta^\ell$, and $\mathbf{x}_2 \in S_\eta^{m-\ell}$. Then, $h_{\mathbf{Y}}(\mathbf{x}) = \mathbf{x}_1 + \mathbf{Y}\mathbf{x}_2$. As seen in Section 1.1.3, it holds that $\tau(h_{\mathbf{Y}}(\mathbf{x})) = \tau(\mathbf{x}_1) + M_\tau(\mathbf{Y})\tau(\mathbf{x}_2)$, and therefore

$$\left\| \tau(h_{\mathbf{Y}}(\mathbf{x})) \right\|_2 \leq \left\| \tau(\mathbf{x}_1) \right\|_2 + \left\| M_\tau(\mathbf{Y}) \right\|_2 \cdot \left\| \tau(\mathbf{x}_2) \right\|_2 .$$

Since $\mathbf{x}_1$ and $\mathbf{x}_2$ are vectors over S_η, it holds that $\left\| \tau(\mathbf{x}_1) \right\|_2 \leq \eta\sqrt{n\ell}$ and that $\left\| \tau(\mathbf{x}_2) \right\|_2 \leq \eta\sqrt{n(m-\ell)}$. By Lemma 1.22 and Lemma 1.3, we also have

$$\mathbb{P}_{\mathbf{Y} \sim \mathcal{D}_{R,s}^{\ell \times (m-\ell)}} \left[\left\| M_\tau(\mathbf{Y}) \right\|_2 > \frac{s}{\sqrt{2\pi}} (\sqrt{\ell} + \sqrt{m-\ell} + t) \right] \leq 2n \cdot e^{-\pi t^2}.$$

For $t = \omega(\sqrt{\log_2 \lambda})$, the bound becomes negligible. Hence, with probability at least $1 - 2ne^{-\pi t^2}$, we have that $\tau(h_{\mathbf{Y}}(\mathbf{x}))$ is bounded by

$$r = \sqrt{n}\eta \left(\sqrt{\ell} + \frac{s}{\sqrt{2\pi}} \sqrt{m-\ell} (\sqrt{\ell} + \sqrt{m-\ell} + t) \right).$$

The number of integer points in the $n\ell$-dimensional ball of radius r is given by the volume of the ball which is $(\sqrt{\pi} r)^{n\ell} / \Gamma(n\ell/2 + 1)$. Yet, it holds that $\Gamma(x+1) > \sqrt{2\pi x}(x/e)^x$. Therefore, we have that

$$\begin{aligned} |h_{\mathbf{Y}}(X)| &\leq \frac{1}{\sqrt{\pi n \ell}} \left(\sqrt{\frac{2\pi e}{n\ell}} \cdot r \right)^{n\ell} \\ &\leq \frac{1}{\sqrt{\pi n \ell}} \left(\eta \sqrt{2\pi e} \left(1 + \frac{s}{\sqrt{2\pi}} \sqrt{\frac{m-\ell}{\ell}} \left(\sqrt{\ell} + \sqrt{m-\ell} + t \right) \right) \right)^{n\ell}. \end{aligned}$$

As the bound is independent of $\mathbf{Y}$, let us temporarily denote it by B. We also define $S = \{\mathbf{Y} \in R^{\ell\times(m-\ell)} : \|M_\tau(\mathbf{Y})\| \leq \frac{s}{\sqrt{2\pi}}(\sqrt{\ell} + \sqrt{m-\ell} + t)\}$, and S' its complement in $R^{\ell\times(m-\ell)}$. We then have

$$\begin{aligned}\mathbb{E}\left[\left|h_{\mathbf{Y}}(X)\right|\right] &= \sum_{\mathbf{Y}'\in S} \mathbb{P}_{\mathbf{Y}}[\mathbf{Y}=\mathbf{Y}']\left|h_{\mathbf{Y}'}(X)\right| + \sum_{\mathbf{Y}'\in S'} \mathbb{P}_{\mathbf{Y}}[\mathbf{Y}=\mathbf{Y}']\left|h_{\mathbf{Y}'}(X)\right| \\ &\leq B\cdot\mathbb{P}_{\mathbf{Y}}[\mathbf{Y}\in S] + |X|\cdot\mathbb{P}_{\mathbf{Y}}[\mathbf{Y}\in S'] \\ &\leq B + |X|\cdot 2ne^{-\pi t^2},\end{aligned}$$

where the first inequality follows from the above calculations and the fact that for $\mathbf{Y}' \in S'$, we have the trivial bound $\left|h_{\mathbf{Y}'}(X)\right| \leq |X|$. Lemma 3.2 then yields the ε_3-uninvertibility of $\mathcal{G}$, with $\varepsilon_3 = B/|X| + 2ne^{-\pi t^2}$. By Lemma 3.3, we thus obtain the ε_3-uninvertibility of $\mathcal{F}$.

3.3.2 Second Preimage Resistance

We now prove the (statistical) second preimage resistance of the M-ISIS function family with respect to the uniform distribution over an η-bounded domain.

Lemma 3.7

Let n, k, q, m, η be positive integers such that q is prime. Let R be the ring of integers of a number field of degree n, and $X \subseteq S_\eta^m$. Then (M-ISIS$(n,k,m,q,X), U(X)$) is (statistically) ε_4-second preimage resistant for

$$\varepsilon_4 = (|X| - 1)\cdot\left(\frac{B_{2\eta}}{q}\right)^{nk},$$

where $B_{2\eta} = \max_{x\in S_{2\eta}}\left\|\sigma(x)\right\|_\infty$.

Proof (Lemma 3.7). To prove it statistically, we show that for $\mathbf{A}, \mathbf{x}$ uniformly chosen, the probability that there exists $\mathbf{x}' \neq \mathbf{x}$ such that $\mathbf{A}^T\mathbf{x}' = \mathbf{A}^T\mathbf{x} \bmod qR$ is less than ε_4, namely

$$p := \mathbb{P}_{\substack{\mathbf{A}\hookleftarrow U(R_q^{m\times k}) \\ \mathbf{x}\hookleftarrow U(X)}}[\exists \mathbf{x}' \in X\setminus\{\mathbf{x}\},\ \mathbf{A}^T\mathbf{x}' = \mathbf{A}^T\mathbf{x} \bmod qR].$$

Using the total probability formula and the union bound on $\mathbf{x}'$, we have the following.

$$\begin{aligned}p &= \sum_{\mathbf{x}^*\in X} \mathbb{P}_{\mathbf{x}}[\mathbf{x}=\mathbf{x}^*]\cdot\mathbb{P}_{\mathbf{A},\mathbf{x}}[\exists \mathbf{x}' \in X\setminus\{\mathbf{x}\},\ \mathbf{A}^T\mathbf{x}' = \mathbf{A}^T\mathbf{x} \bmod qR | \mathbf{x}=\mathbf{x}^*] \\ &= \sum_{\mathbf{x}^*\in X} |X|^{-1}\cdot\mathbb{P}_{\mathbf{A}}[\exists \mathbf{x}' \in X\setminus\{\mathbf{x}^*\},\ \mathbf{A}^T(\mathbf{x}'-\mathbf{x}^*) = \mathbf{0} \bmod qR] \\ &\leq |X|^{-1}\sum_{\mathbf{x}^*\in X}\sum_{\mathbf{x}'\in X\setminus\{\mathbf{x}^*\}} \mathbb{P}_{\mathbf{A}}[\mathbf{A}^T(\mathbf{x}'-\mathbf{x}^*) = \mathbf{0} \bmod qR].\end{aligned}$$

Let $\mathbf{x}^* \in X$, $\mathbf{x}' \in X\setminus\{\mathbf{x}^*\}$, and set $\mathbf{z} = \mathbf{x}' - \mathbf{x}^*$. Then, by [Mic07, Lem. 4.4], $\mathbf{A}^T\mathbf{z} \bmod qR$ is uniformly distributed in $(\mathcal{I}_\mathbf{z}/qR)^k$ over the randomness of $\mathbf{A}$, where $\mathcal{I}_\mathbf{z} = \langle z_1, \ldots, z_m, q\rangle$. Hence the probability that $\mathbf{A}^T\mathbf{z} = \mathbf{0} \bmod qR$ is $\left|\mathcal{I}_\mathbf{z}/qR\right|^{-k}$. As $\mathcal{I}_\mathbf{z}$ and qR are ideals of R, we have $\left|\mathcal{I}_\mathbf{z}/qR\right| = N(qR)/N(\mathcal{I}_\mathbf{z}) = q^n/N(\mathcal{I}_\mathbf{z})$. Yet, for all $i \in [\![m]\!]$, $\langle z_i\rangle \subseteq \mathcal{I}_\mathbf{z}$, meaning that $N(\mathcal{I}_\mathbf{z})$ divides $N(\langle z_i\rangle)$. Similarly, $N(\mathcal{I}_\mathbf{z})$ divides $N(\langle q\rangle) = q^n$. Hence

$$N(\mathcal{I}_\mathbf{z}) \leq \gcd\left(q^n, N(\langle z_1\rangle), \ldots, N(\langle z_m\rangle)\right),$$

which yields the (loose) bound

$$N(\mathcal{I}_\mathbf{z}) \leq \min\left(q^n, \min_{i\in[\![m]\!]:z_i\neq 0} N(\langle z_i\rangle)\right).$$

Since $\mathbf{z} \neq \mathbf{0}$, there exists $i \in [\![m]\!]$ such that $z_i \neq 0$. Note that we have $\mathbf{z} \in \{\mathbf{a} - \mathbf{b}; (\mathbf{a}, \mathbf{b}) \in X^2\} \subseteq S_{2\eta}^m$. It thus holds

$$N(\langle z_i \rangle) = |N(z_i)| = \prod_{j \in [\![n]\!]} |\sigma_j(z_i)| \leq B_{2\eta}^n,$$

where $B_{2\eta} = \max_{x \in S_{2\eta}} \|\sigma(x)\|_\infty$. Recall that in cyclotomic fields we have $B_{2\eta} \leq 2\eta n$ by Lemma 1.1. Hence $\mathbb{P}_{\mathbf{A}}[\mathbf{A}^T\mathbf{z} = \mathbf{0} \bmod qR] \leq (B_{2\eta}/q)^{nk}$. Going back to our original calculation, we then have $p \leq |X|^{-1}|X|\,(|X| - 1) \cdot (B_{2\eta}/q)^{nk} = \varepsilon_4$ which concludes the proof.

For common choices of n, m and prime q, we heuristically observe that the ideals $\langle z_1 \rangle, \ldots, \langle z_m \rangle, \langle q \rangle$ are relatively prime with high probability, which means that $\mathcal{I}_{\mathbf{z}} = R$ in the proof above. In this case, $N(\mathcal{I}_{\mathbf{z}}) = 1$ which yields a much better bound on the probability. Since the probability sums over all the possible $\mathbf{x}'$, one would need to evaluate the proportion of $\mathbf{z}$ generated as above that verify $\mathcal{I}_{\mathbf{z}} = R$. We leave it as an open problem.

Consider the example of cyclotomic fields. By Lemma 1.4 (or Remark 1.2 for power-of-two conductors), if q splits into few factors and is large enough with respect to η so that $S_{2\eta} \setminus \{0\} \bmod qR \subset R_q^\times$, then we have that for all $\mathbf{z} \in S_{2\eta}^m \setminus \{\mathbf{0}\}$, $\mathcal{I}_{\mathbf{z}} = R$. Indeed, for such $\mathbf{z}$, there exists $i \in [\![m]\!]$ such that $z_i \in S_{2\eta} \setminus \{0\}$ and therefore $z_i \bmod qR \in R_q^\times$. This implies that $\langle z_i \rangle + \langle q \rangle = \langle 1 \rangle = R$ and as a result $\mathcal{I}_{\mathbf{z}} = R$. Hence, we can improve the bound on the probability to $\varepsilon_4 = (|X| - 1)/q^{nk}$ if we accept to enforce a specific splitting on q.

3.3.3 One-Wayness of the M-LWE Function

Using the results from Sections 3.3.1 and 3.3.2, we can give the main theorem of this chapter. Under the assumption that the M-LWE function family is pseudorandom with respect to a Gaussian error distribution, it proves that the M-LWE function family is one-way with respect to a small uniform error distribution. Recall that if a function is one-way, then it is also uninvertible. Hence, this shows that the search version sM-LWE with small uniform error is at least as hard as the decision version M-LWE with Gaussian error.

Theorem 3.1 (Computational Hardness of M-LWE with Short Error)

Let n, d, m, k, q, η be positive integers such that $m > d \geq k \geq 1$ and let $\ell = m - d + k$. Let R be the ring of integers of a number field of degree n, and $X \subseteq S_\eta^m$. We assume that q is prime and unramified in R such that $\min_{i \in [\![\kappa]\!]} N(\mathfrak{p}_i)^{\min(m-d,k)+1} \geq \lambda^{\omega(1)}$, where the $\mathfrak{p}_i$'s are the prime ideal factors of $\langle q \rangle$. It holds that if the function family $(\text{M-LWE}(n, k, \ell, q, R^\ell), \mathcal{D}_{R,s}^\ell)$ is ε_1-pseudorandom for some $s > 0$, then $(\text{M-LWE}(n, d, m, q, X), U(X))$ is ε-one-way for

$$\varepsilon = \delta(m, d) + \frac{(m - \ell)(2\delta(\ell, m - d) + \varepsilon_1/(1 - \delta(\ell, k))) + \varepsilon_3 + \varepsilon_4}{1 - \delta(m, m - d)} = (d - k)\varepsilon_1 + \varepsilon_3 + \varepsilon_4 + \mathsf{negl}(\lambda),$$

where $\varepsilon_3, \varepsilon_4$ are defined in the statement of Lemma 3.6 and 3.7 respectively. In particular, we have $\varepsilon_{\text{sM-LWE}} \leq (d - k)\varepsilon_{\text{M-LWE}} + \varepsilon_3 + \varepsilon_4 + \mathsf{negl}(\lambda)$.

Proof (Theorem 3.1). Define the function families $\mathcal{F} = \text{M-ISIS}(n, m - d, \ell, q, R^\ell) \circ \mathcal{G}(n, \ell, m, s, X)$, and $\mathcal{F}' = \text{M-ISIS}(n, m - d, m, q, X)$.
Indistinguishability: Using Lemma 3.4, the pseudorandomness of the M-LWE function family implies that $(\text{M-ISIS}(n, m - d, \ell, q, R^\ell), \mathcal{D}_{R^\ell, s})$ is ε_2-pseudorandom with

$$\varepsilon_2 = 2\delta(\ell, \ell - k) + \frac{\varepsilon_1}{1 - \delta(\ell, k)}.$$

Take $f_{\mathbf{A}} \circ h_{\mathbf{Y}}$ according to $\mathcal{F}$, and $f_{\mathbf{A}'}$ according to $\mathcal{F}'$. Then $f_{\mathbf{A}} \circ h_{\mathbf{Y}}$ is the linear map $\mathbf{x} \mapsto [\mathbf{A}^T | \mathbf{A}^T\mathbf{Y}]\mathbf{x}$. Decomposing $\mathbf{A}'^T$ into $[\mathbf{A}_1'^T | \mathbf{A}_2'^T]$, with $\mathbf{A}_1' \in R_q^{\ell \times (m-d)}$, $\mathbf{A}_2' \in R_q^{(m-\ell) \times (m-d)}$, we have that $f_{\mathbf{A}'} = \mathbf{x} \mapsto [\mathbf{A}_1'^T | \mathbf{A}_2'^T]\mathbf{x}$. By the ε_2-pseudorandomness of M-ISIS with respect to $\mathcal{D}_{R,s}^\ell$, a hybrid argument yields that $\mathcal{F}$ and $\mathcal{F}'$ are $(m - \ell)\varepsilon_2$-indistinguishable.
Uninvertibility: By Lemma 3.6, it holds that $(\mathcal{F}, U(X))$ is ε_3-uninvertible, where ε_3 is defined in Lemma 3.6.

Second Preimage Resistance: By Lemma 3.7, it holds that $(\mathcal{F}', U(X))$ is ε_4-second preimage resistant for

$$\varepsilon_4 = (|X| - 1) \cdot \left(\frac{B_{2\eta}}{q}\right)^{n(m-d)}.$$

By indistinguishability, the properties of $\mathcal{F}$ and $\mathcal{F}'$ transfer to one another with an additive loss of $(m-\ell)\varepsilon_2$. As such, $(\mathcal{F}', U(X))$ is $((m-\ell)\varepsilon_2+\varepsilon_3)$-uninvertible. Lemma 3.1 then yields that $(\mathcal{F}', U(X))$ is ε_0-one-way with $\varepsilon_0 = (m-\ell)\varepsilon_2 + \varepsilon_3 + \varepsilon_4$. Using Lemma 3.5, it gives that $(\text{M-LWE}(n, d, m, q, X), U(X))$ is ε-one-way with

$$\varepsilon = \delta(m, d) + \frac{\varepsilon_0}{1 - \delta(m, m-d)}.$$

Combining everything, we get

$$\varepsilon = \delta(m, d) + \frac{(m-\ell)(2\delta(\ell, m-d) + \varepsilon_1/(1-\delta(\ell, k))) + \varepsilon_3 + \varepsilon_4}{1 - \delta(m, m-d)},$$

which yields the claim. The condition on q ensures that all the $\delta(\cdot,\cdot)$ are negligible. Indeed, as noted after Lemma 1.6, if the smallest norm N of the prime ideal factors is such that $N^{a-b+1} \geq \lambda^{\omega(1)}$ for $a \geq b$, then $\delta(a, b) \leq \lambda^{-\omega(1)}$. The condition on q thus yields that $\delta(m, d), \delta(m, m-d), \delta(\ell, m-d), \delta(\ell, k)$ are negligible. Thence, we get that

$$\varepsilon = (m-\ell)\varepsilon_1 + \varepsilon_3 + \varepsilon_4 + \mathsf{negl}(\lambda).$$

We observe that the *factors* $1/(1-\delta(\cdot,\cdot))$ correspond to abort conditions in Lemma 3.4 and 3.5, and thus do not have to be negligible. The additive components $\delta(\cdot,\cdot)$ however originate from a statistical divergence between the expected and ideal distributions, which have to be negligible. In our case, the multiplicative δ will also be negligible if the additive ones are.

3.4 Hardness with Small Secret and Error

Before discussing which parameters are covered by Theorem 3.1, i.e., what q, η, m, d make ε_3 and ε_4 negligible, we leverage it to obtain the hardness $\text{sM-LWE}_{n,d,m',q,U(S_\eta),U(S_\eta)}$. To do so, we use a Hermite Normal Form transformation which more generally allows to go from secret and error distributions $(\mathscr{D}_s, \mathscr{D}_e)$ to $(\mathscr{D}_e, \mathscr{D}_e)$. Langlois and Stehlé [LS15, Lem. 4.24] proposed an immediate generalization of the reduction from LWE to its Hermite Normal Form by Applebaum et al. [ACPS09] to modules. In particular, it relies on the fact that if one has access to sufficiently many M-LWE samples $(\mathbf{a}_i, b_i)$, they can find a subset of the $\mathbf{a}_i$ that form a matrix in $GL_d(R_q)$. As our proof of Theorem 3.1 seemingly limits the number of available samples, it is relevant for us to understand the trade-off between the quality of the reduction (in terms of loss in advantage) and the number of initial samples. More precisely, if one is limited to use $m > d$ samples to construct this invertible matrix, it comes down to evaluating $\delta'(m, d)$.

Lemma 3.8 (Hermite Normal Form Transform (Adapted from [ACPS09, LS15]))

Let n, d, q, m be positive integers. Let R be the ring of integers of a number field of degree n. We assume that q is prime and unramified in R, and that $m > d \geq 1$. Let $\mathscr{D}_s, \mathscr{D}_e$ be two distribution over R. There is a PPT reduction from $\text{sM-LWE}_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ to $\text{sM-LWE}_{n,d,m-d,q,\mathscr{D}_e,\mathscr{D}_e}$ (and also from $\text{M-LWE}_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ to $\text{M-LWE}_{n,d,m-d,q,\mathscr{D}_e,\mathscr{D}_e}$). More precisely, if $\varepsilon_{\text{sM-LWE},m}$ and $\varepsilon_{\text{sM-LWE},m-d}$ denote the corresponding hardness bounds, it holds that

$$\varepsilon_{\text{sM-LWE},m-d} \leq \frac{1}{1-\delta'(m,d)}\varepsilon_{\text{sM-LWE},m}$$

Proof (Lemma 3.8). We perform the reduction for the search versions, and explain the different arguments for the decision versions which are very similar. Let $(\mathbf{A}, \mathbf{b}) \in R_q^{m\times d} \times R_q^m$ be an instance of $\text{sM-LWE}_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$.

Construction: The reduction first checks if there is a subset $S \subseteq [\![m]\!]$ of size d such that the rows of $\mathbf{A}$ indexed by S are R_q-linearly independent. If not the reduction aborts. Because $\mathbf{A}$ is uniformly random, the quantity defined in Section 1.1.6 captures the abort probability which is exactly $\delta'(m,d)$. So now, we assume that there exists a set $S \subseteq [\![m]\!]$ of size d such that the rows $(\mathbf{a}_i^T)_{i\in S}$ are R_q-linearly independent. Consider the matrix $\overline{\mathbf{A}} \in R_q^{d\times d}$ whose rows are the $(\mathbf{a}_i^T)_{i\in S}$, and $\overline{\mathbf{b}} \in R_q^d$ whose coefficients are the $(b_i)_{i\in S}$. By construction, $\overline{\mathbf{A}}$ is invertible in $R_q^{d\times d}$. Additionally, when $\mathbf{b} = \mathbf{A}\mathbf{s}+\mathbf{e} \bmod qR$, it holds that $\overline{\mathbf{b}} = \overline{\mathbf{A}}\mathbf{s}+\overline{\mathbf{e}} \bmod qR$ for $\mathbf{s} \hookleftarrow \mathscr{D}_s^d$, and $\overline{\mathbf{e}} \hookleftarrow \mathscr{D}_e^d$. On the other hand, if $\mathbf{b}$ is uniform in R_q^m, then $\overline{\mathbf{b}}$ is uniform in R_q^d. We now define $\mathbf{A}'$ the matrix whose rows are $(\mathbf{a}_i^T)_{i\in[\![m]\!]\setminus S}$, and $\mathbf{b}', \mathbf{e}'$ similarly.

Reduction: The reduction then transforms $\mathbf{A}', \mathbf{b}'$ by defining $\mathbf{A}'' = -\mathbf{A}'\overline{\mathbf{A}}^{-1} \bmod qR$ and $\mathbf{b}'' = \mathbf{b}' + \mathbf{A}''\overline{\mathbf{b}} \bmod qR$, and sends $(\mathbf{A}'', \mathbf{b}'')$ to the sM-LWE$_{n,d,m-d,q,\mathscr{D}_e,\mathscr{D}_e}$ oracle. It obtains $(\overline{\mathbf{e}}^\star, \mathbf{e}'^\star)$ and then computes $\mathbf{s}^\star = \overline{\mathbf{A}}^{-1}(\overline{\mathbf{b}} - \overline{\mathbf{e}}^\star) \bmod qR$ and $\mathbf{e}^\star = \mathbf{b} - \mathbf{A}\mathbf{s}^\star \bmod qR$. It finally returns $(\mathbf{s}^\star, \mathbf{e}^\star)$ as the solution to the instance $(\mathbf{A}, \mathbf{b})$. For the decision version, the reduction simply calls the oracle on $(\mathbf{A}'', \mathbf{b}'')$ and outputs the same answer.

Let us now analyze the correctness of the reduction. Since $\overline{\mathbf{A}}$ is in $GL_d(R_q)$, it holds that $\mathbf{A}''$ is uniform in $R_q^{m-d\times d}$ as expected. Then, we have that

$$\begin{aligned}\mathbf{b}'' = \mathbf{A}'\mathbf{s} + \mathbf{e}' + \mathbf{A}''(\overline{\mathbf{A}}\mathbf{s} + \overline{\mathbf{e}}) \bmod qR &= \mathbf{A}'\mathbf{s} - \mathbf{A}'\overline{\mathbf{A}}^{-1}\overline{\mathbf{A}}\mathbf{s} + \mathbf{A}''\overline{\mathbf{e}} + \mathbf{e}' \bmod qR \\ &= \mathbf{A}''\overline{\mathbf{e}} + \mathbf{e}' \bmod qR,\end{aligned}$$

which is correctly distributed as $\overline{\mathbf{e}} \sim \mathscr{D}_e^d$ and $\mathbf{e}' \sim \mathscr{D}_e^{m-d}$ independently of one another. So the oracle, if successful, returns $(\overline{\mathbf{e}}^\star, \mathbf{e}'^\star) = (\overline{\mathbf{e}}, \mathbf{e}')$. As a result, $\mathbf{s}^\star = \overline{\mathbf{A}}^{-1}(\overline{\mathbf{b}} - \overline{\mathbf{e}}) \bmod qR = \mathbf{s}$, and $\mathbf{e}^\star = \mathbf{b} - \mathbf{A}\mathbf{s} \bmod qR = \mathbf{e}$ as desired. For the decision version, if $\mathbf{b}$ is uniform, then $\mathbf{b}'' = \mathbf{b}' + \mathbf{A}''\overline{\mathbf{b}}$ is also clearly uniform.

So for an oracle $\mathcal{O}$ for sM-LWE$_{n,d,m-d,q,\mathscr{D}_e,\mathscr{D}_e}$, we have constructed a PPT adversary $\mathcal{A}$ such that

$$\begin{aligned}\mathbb{P}[\mathcal{A}(\mathbf{A}, \mathbf{b}) = (\mathbf{s}, \mathbf{e})] &= \mathbb{P}[E]\mathbb{P}[\mathcal{A}(\mathbf{A}, \mathbf{b}) = (\mathbf{s}, \mathbf{e})|E] + \mathbb{P}[\neg E]\mathbb{P}[\mathcal{A}(\mathbf{A}, \mathbf{b}) = (\mathbf{s}, \mathbf{e})|\neg E] \\ &= (1 - \delta'(m,d))\mathbb{P}[\mathcal{O}(\mathbf{A}'', \mathbf{b}'') = (\overline{\mathbf{e}}, \mathbf{e}')],\end{aligned}$$

where $E = \{\exists S \subseteq [\![m]\!], |S| = d, (\mathbf{a}_i^T)_{i\in S}$ are R_q-linearly independent$\}$. Optimizing over $\mathcal{O}$ gives $\varepsilon_{\text{sM-LWE},m} \geq \mathrm{Adv}_{\text{sM-LWE},m}[\mathcal{A}] = (1 - \delta'(m,d))\varepsilon_{\text{sM-LWE},m-d}$, as claimed.

This reduction essentially allows to discard the secret distribution and use a subvector of the error to be the new secret. All the parameters are preserved except for the number of samples because we need d extra samples to constitute the new secret by identifying an invertible submatrix of $\mathbf{A}$. Hence, to prove the hardness of sM-LWE$_{\mathscr{D}_e,\mathscr{D}_e}$ with m' samples, we thus need to assume the hardness of sM-LWE$_{\mathscr{D}_s,\mathscr{D}_e}$ with $m = m' + d$ samples. Let us now discuss the loss $\delta'(m,d)$. In the case of integers, $\mathbb{Z}_q$ is generally a field which yields a closed-form expression of this probability. Unfortunately, in the case of R_q, it becomes anything but trivial as explained in Section 1.1.6. We can still obtain the following bound

$$\delta'(m,d) \leq \delta'(d,d)^{\lfloor m/d \rfloor} = \left(1 - \prod_{\ell\in[\![0,d[\![}\prod_{i\in[\![\kappa]\!]}\left(1 - \frac{1}{N(\mathfrak{p}_i)^{d-\ell}}\right)\right)^{\lfloor m/d \rfloor},$$

which simply consists in looking at $[\![m]\!]$ in blocks of size d and checking if they give an invertible submatrix. We note that $\delta'(d,d)$ highly depends on the size and splitting of q as it is essentially dominated by $\frac{1}{\min_{i\in[\![\kappa]\!]} N(\mathfrak{p}_i)}$. Hence, depending on the splitting of q, we would need to take $m = Cd$ with C sufficiently large to make $\delta'(m,d)$ negligible. Our bound is however not tight and we expect $\delta'(m,d)$ to decrease much faster when m grows. Unfortunately, we were not able to find a better bound on $\delta'(m,d)$ which would support this conjecture. We leave it as an interesting open problem. Regardless, the loss $\delta'(m,d)$ is only featured in a multiplicative term as it is due to an abort condition. Indeed, when constructing $\mathbf{A}''$, we notice that it is *perfectly* uniform over $R_q^{m-d\times d}$ because we do not impose restrictions on $\mathbf{A}'$. As such, there is no additive loss. Thence, the term $\delta'(m,d)$ only has to be non-overwhelming, i.e., $1 - \delta'(m,d)$ non-negligible. For example, if $\delta'(m,d) = 1/C$, then this only reduces the hardness bound by $\log_2 C/(C-1)$ bits. In most cases,

$\delta'(m,d)$ is not overwhelming which means that it incurs almost no loss through Lemma 3.8. We discuss the concrete parameter selection in Section 3.5. Nevertheless, we then obtain the following corollary on the hardness of sM-LWE with short secret and error by combining Theorem 3.1 with Lemma 3.8.

Corollary 3.1 (Computational Hardness of M-LWE with Short Secret and Error)

Let n, d, m, k, q, η be positive integers such that $m > d \geq k \geq 1$ and let $\ell = m - d + k$. Let R be the ring of integers of a number field of degree n, and $X \subseteq S_\eta^m$. We assume that q is prime and unramified in R such that $\min_{i \in [\![\kappa]\!]} N(\mathfrak{p}_i)^{\min(m-d,k)+1} \geq \lambda^{\omega(1)}$, where the $\mathfrak{p}_i$'s are the prime ideal factors of $\langle q \rangle$. Assuming that M-LWE$_{n,k,\ell,q,U(R_q),\mathcal{D}_{R,s}}$ is hard for some $s > 0$, then sM-LWE$_{n,d,m-d,q,U(S_\eta),U(S_\eta)}$ is also hard. More precisely, it holds that

$$\varepsilon_{\text{sM-LWE}} \leq \frac{1}{1-\delta'(m,d)} \left(\delta(m,d) + \frac{(d-k)\left(2\delta(\ell, m-d) + \frac{\varepsilon_{\text{M-LWE}}}{1-\delta(\ell,k)}\right) + \varepsilon_3 + \varepsilon_4}{1-\delta(m, m-d)} \right),$$

where $\varepsilon_3, \varepsilon_4$ are defined in the statement of Lemma 3.6 and 3.7 respectively.

3.5 Parameter Selection

Let us now discuss the various conditions that are needed to apply Theorem 3.1 and Corollary 3.1, in the context of cyclotomic fields, i.e., such that $\varepsilon_3, \varepsilon_4, \delta(\cdot,\cdot), \delta'(\cdot,\cdot)$ are negligible.

3.5.1 M-LWE with Small Error

We first look at Theorem 3.1 dealing with a small uniform error and a large secret uniform in R_q. First, ensuring $\varepsilon_4 = \mathsf{negl}(\lambda)$ is fairly straighforward by placing a lower bound on q. Indeed, we have $X = S_\eta^m$ and thus $|X| = (2\eta+1)^{nm}$. In order to obtain $\varepsilon_4 \leq 2^{-\lambda} = \mathsf{negl}(\lambda)$, it suffices to have

$$(2\eta+1)^m \left(\frac{2n\eta}{q}\right)^{m-d} < 2^{-\lambda/n}, \tag{3.1}$$

which can be written as $q > 2^{\lambda/(n(m-d))} \cdot 2n\eta \cdot (2\eta+1)^{m/(m-d)}$. Once this lower bound on q is set, one can easily find the closest prime q with an appropriate splitting as required by the theorem. We note that the splitting of q may provide a better expression for ε_4 as explained in Section 3.3.2, and thus a better lower bound for q. For example, in a cyclotomic field of power-of-two conductor, if $\langle q \rangle$ splits into κ prime ideals and q verifies $q > (2\eta\sqrt{\kappa})^\kappa$ and $q > 2^{\lambda/(n(m-d))} \cdot (2\eta+1)^{m/(m-d)}$, then $\varepsilon_4 \leq 2^{-\lambda}$.

The expression of ε_3 is more involved, but the idea is the same. For it to be negligible, we need $t = \omega(\sqrt{\log \lambda})$, say $t = \log_2 \lambda$, and

$$\frac{\eta^\ell}{(2\eta+1)^m (\pi n \ell)^{1/2n}} \left(\sqrt{2\pi e} \left(1 + s\sqrt{\frac{m-\ell}{2\pi\ell}} \left(\sqrt{\ell} + \sqrt{m-\ell} + t \right) \right) \right)^\ell < 2^{-\lambda/n}, \tag{3.2}$$

Due to the many dependencies in m, k, d and η, it is harder to extract a closed-form inequality on m given k, d and η. As we aim at proving the hardness of sM-LWE with small parameters, one can numerically evaluate Equations (3.1) and (3.2) with the goal of minimizing η, q and d, while maximizing m and making sure that $m > d \geq k \geq 1$ ($k \geq 2$ being preferable to rely on modules). It turns out that the condition is not met for all parameter sets, and η cannot be arbitrarily small for arbitrary ranks k, d. Nonetheless, we can find settings in which η is a small constant, but this might require to take d slightly larger. As expected, when $m-d$ grows for a fixed d, the error bound η must be larger as well. Table 3.1 gives two example parameter sets that verify the conditions of Theorem 3.1, along with the losses $\varepsilon_3, \varepsilon_4$, one relying on ring assumptions ($k = 1$).

3.5.2 M-LWE with Small Secret and Error

From Table 3.1, we observe that we can now reach parameters verifying $d < k \log_{2\eta+1} q$, whereas the results from Chapter 2 required at least $d \geq k \log_{2\eta+1} q + \log_2 \Omega(n)$. In particular, for a fixed k and η, the minimal rank does not seem to depend on q. We also note that the loss $\delta'(m,d)$ is bounded

λ	n	k	d	m	η	q	κ	ε_3	ε_4	$\delta(\cdot,\cdot)$
128	256	1	10	11	1	$2^{26.9}$	64	$2^{-199.7}$	2^{-128}	$2^{-209.4}$
128	256	1	10	11	1	$2^{17.9}$	4	$2^{-199.7}$	2^{-128}	$2^{-2293.6}$
128	256	2	10	12	9	$2^{37.9}$	128	$2^{-213.1}$	2^{-128}	$2^{-220.4}$
128	256	2	10	12	9	$2^{25.7}$	4	$2^{-213.1}$	2^{-128}	$2^{-4939.6}$

Table 3.1: Example parameter sets for M-LWE with small error, reaching the conditions of Theorem 3.1. We take $t = \log_2 \lambda$, and $s \approx \log_2 n$ if $k = 1$ and $s \approx 2\sqrt{k}\log_2 n$ if $k > 1$. The column $\delta(\cdot,\cdot)$ corresponds to the maximum of all the $\delta(\cdot,\cdot)$ for the dimensions involved in the theorem statement. The highlighted rows correspond to a low splitting of q (small κ).

above by $2^{-101.7}$ and $2^{-68.8}$ respectively for the high splitting sets (κ large). It is sufficiently small to incur no noticeable loss during the Hermite Normal Form transformation reduction. But these parameters are still not sufficient to use Corollary 3.1 as it would give $m - d < d$, resulting in a peculiar regime for the M-LWE problem where the solution is likely not unique. We now give example parameter sets in Table 3.2 with $m = 2d$. Also, to give another perspective, we give examples with smaller ranks d at the expense of a possibly larger η.

λ	n	k	d	m	η	q	κ	ε_3	ε_4	$\delta(\cdot,\cdot)$
128	256	1	4	8	486	$2^{20.8}$	2	$2^{-130.0}$	$2^{-1020.9}$	$2^{-5335.5}$
128	256	1	5	10	275	$2^{19.2}$	2	$2^{-128.3}$	$2^{-1273.3}$	$2^{-4914.8}$
128	256	1	6	12	197	$2^{18.2}$	2	$2^{-133.8}$	$2^{-1524.7}$	$2^{-4668.4}$
128	256	1	7	14	158	$2^{17.6}$	2	$2^{-133.5}$	$2^{-1775.6}$	$2^{-4505.5}$

Table 3.2: Example parameter sets for M-LWE with small secret *and* error, reaching the conditions of Corollary 3.1. We take $t = \log_2 \lambda$, and $s \approx \log_2 n$ if $k = 1$ and $s \approx 2\sqrt{k}\log_2 n$ if $k > 1$. The column $\delta(\cdot,\cdot)$ corresponds to the maximum of all the $\delta(\cdot,\cdot)$ and $\delta'(\cdot,\cdot)$ for the dimensions involved in the corollary statement.

Although the minimal reachable η is much smaller than q and can seem satisfying, it is insightful to isolate a trend between η, d and m. Intuitively, a larger η would allow for more samples m. Also, as d increases, the problem should also become harder as it essentially increases the lattice dimension, thus allowing for a smaller η. We depict these trends in Figure 3.2. The concrete results present in the figure showcase an exponential growth of η as a function of m for a fixed d. This matches the observation from the hardness proof of the non-structured variant [MP13] where $m \leq d(1 + O(\log_2 \eta))$.

3.5.3 Asymptotic Analysis

Note that we can provide the asymptotic behavior $\varepsilon_3 = O(s \cdot m \cdot \eta \cdot t/\sqrt{\ell})^{n\ell}/|X| + 2ne^{-\pi t^2}$, but this approach makes it unclear how to choose the parameters. In particular, as we can use low ranks like $d = O(1)$, we have to make sure that $k \geq 1$ and $m \geq d + 1$, which is not always possible for low values of η. Regardless, we can still derive the asymptotic behavior of m with respect to η which explains Figure 3.2. Indeed, taking $s = 2\sqrt{k}t$, and denoting by C' the asymptotic constant, we have

$$O(s \cdot m \cdot \eta \cdot t/\sqrt{\ell})^{n\ell}/|X| \leq \left(\frac{(2C'mt^2\eta\sqrt{k/\ell})^{\ell}}{(2\eta+1)^m}\right)^n.$$

Since $\ell > k$, we can choose the parameters to have $(2C'mt^2\eta)^{\ell}/(2\eta+1)^m \leq 1/2$ to have a exponentially small loss. This leads to a condition on m which is

$$d < m \leq (d-k)\left(1 + \frac{\log_2(2\eta+1)}{\log_2(2C' \cdot m \cdot t^2/3)}\right),$$

which is much similar to the condition in [MP13]. The main difference stems from the fact that m is no longer our asymptotic parameter, which explains the presence of $t^2 = \omega(\log_2 \lambda)$. It still remains difficult to see which parameter sets meet this condition, mostly because the constant C' can be rather large while we wish d and k to be small constants. Regardless, if we aim at very

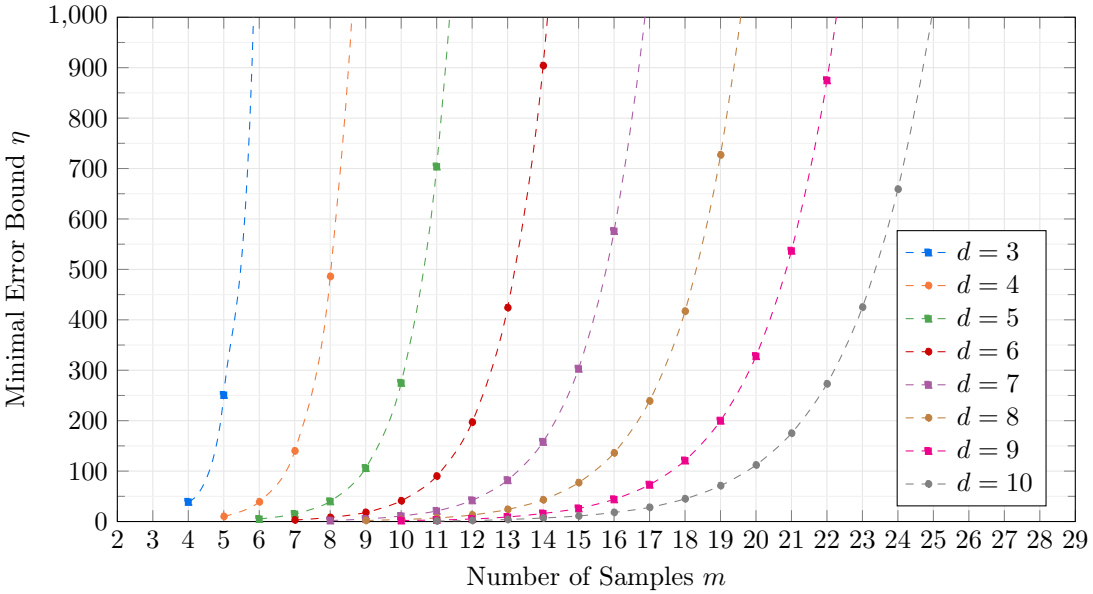


Figure 3.2: Concrete behavior of the minimal error bound η with respect to the number of samples m for different module ranks d. We take $\lambda = 128$, $n = 256$, $k = 1$ and low splitting $\kappa = 2$.

small values of η, we see that we obtain $m \leq d(1 + o(1))$. This bound seems true even with our non-asymptotic analysis.

3.6 Conclusion

The intricacies between the different parameters and the scope of our proof become clearer when looking at Figure 3.2. Choosing a particularly small η, e.g., $\eta = 1$, requires to increase d, while keeping d small and $m \geq 2d$ requires a much larger η.

Yet, the results of this chapter somewhat bypass the limitations of Chapter 2. The blow-up in the module rank d is still present for arbitrarily small η, albeit less restrictive. Indeed, the parameters in Table 3.1 typically verify $d < k \log_{2\eta+1} q$, whereas the result from Chapter 2 required $d \geq k \log_{2\eta+1} q + \log_2 \Omega(n)$ at best. However new restrictions arise on the number of samples m. Aiming for $m = 2d$ to obtain small secret *and* error drastically impacts the reachable error bound η as it grows exponentially with m. On the bright side, our result still provides insights on the hardness of M-LWE with short secret *and* error which is what we set out to achieve. Although the parameters selected through direct cryptanalysis are smaller, our work narrows the gap between provable hardness and cryptanalysis.

Part I helped gaining confidence in the M-LWE assumption with small secret and/or error by proving its robustness in a variety of close to practical regimes. For efficiency reasons, in the rest of this thesis, the parameters chosen to make M-LWE hard will be selected through concrete security analyses relying on the celebrated lattice estimator [APS15]. Further details are provided in Chapter 9.

Part II

Samplers and Signatures

Before looking at the design of advanced lattice signatures, we focus in this second part on the design of regular signature schemes based on trapdoor preimage samplers. In particular, we investigate in depth the potential of gadget-based samplers and signatures.

4

Optimizing Gadget-Based Samplers

This chapter focuses on the optimization of lattice gadget-based samplers that are prominent in the design of lattice-based signatures and most importantly advanced ones. We revisit the original sampler from [MP12] in several ways to improve its efficiency in different application ranges.

The work presented in this chapter is based on two papers with my co-authors Sven Argo, Tim Güneysu, Georg Land, Adeline Roux-Langlois and Olivier Sanders.

[JRS24] **Phoenix: Hash-And-Sign with Aborts from Lattice Gadgets**. Published at PQCrypto 2024. Co-authored only with Adeline Roux-Langlois, and Olivier Sanders.

[AGJ+24] **Practical Post-Quantum Signatures for Privacy**. Published at ACM CCS 2024.

Contents

4.1 Introduction

Trapdoor functions and preimage sampling represent fundamental tools in the design of lattice-based cryptographic schemes. The former, which has been known for decades, has been used in the construction of digital signature schemes over lattices in a rather compact way: given a short basis $\mathbf{B}_{\mathsf{sk}}$ of a lattice, representing the trapdoor, Babai's nearest plane or round-off algorithms [Bab85, Bab86] can be used to solve the CVP_γ problem for a target $\mathbf{u} = \mathcal{H}(\mathbf{m})$ corresponding to a hashed

message in the ambient space. This represents the inversion of the trapdoor function. A long basis $\mathbf{B}_{\mathsf{pk}}$ can then be used to verify the CVP solution, corresponding to the forward calculation of the trapdoor function. This idea led to the design of the GGH signature by Goldreich et al. [GGH97] and the NTRUSign signature by Hoffstein et al. [HHP+03]. These designs however suffered from security issues [NR06, DN12] because signatures leaked information that revealed the geometry of $\mathbf{B}_{\mathsf{sk}}$. In order to circumvent this leakage, Gentry et al. [GPV08] proposed a way to randomize the decoding process so that it becomes independent of the secret key. They abstracted this idea into the notion of *preimage sampleable trapdoor functions*, leading to the celebrated procedure of *preimage sampling*.

This framework was successfully instantiated over lattices by the authors themselves [GPV08][1] using a discrete Gaussian sampler [Kle00, GPV08] over arbitrary lattices, whose quality depends on the size of the input basis $\mathbf{B}_{\mathsf{sk}}$. Many works followed in their footsteps to design such trapdoor functions and preimage samplers in the most efficient way possible [AP09, Pei10, MP12, DLP14, LW15, DP16]. Among them, Micciancio and Peikert [MP12] proposed a way of generating trapdoors with very interesting features that have been leveraged in many cryptographic designs, especially advanced ones. The efficiency of these primitives however strongly relies on the quality of the associated preimage sampling procedure: sampling from a narrower distribution leads to better efficiency and security guarantees. At a high level, the authors propose a sampler (later called MP sampler) for linear functions of the form $\mathbf{A}_{\mathbf{T}} = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}] \bmod qR$ where $\mathbf{R}$ is a short matrix representing the trapdoor. More precisely, $\mathbf{A}$ is a uniform matrix, $\mathbf{T}$ a tag matrix, and $\mathbf{G}$ is a so-called *gagdet* matrix which enables *efficient* and *public* sampling. Concretely, we use $\mathbf{G} = \mathbf{I}_d \otimes [1|b|\dots|b^{k-1}]$ which is the base-b recomposition matrix. Their algorithm then uses the knowledge of $\mathbf{R}$ to sample $\mathbf{v}$ according to a spherical discrete Gaussian of parameter s such that $\mathbf{A}_{\mathbf{T}}\mathbf{v} = \mathbf{u} \bmod qR$ for an input syndrome $\mathbf{u}$. The technique first relies on the observation that if $\mathbf{z}$ is a Gaussian with width $s_{\mathbf{G}}$ such that $\mathbf{TGz} = \mathbf{u}$, then the vector $\mathbf{v}' = [(\mathbf{Rz})^T|\mathbf{z}^T]^T$ is a valid candidate. This naive approach leaks information on the trapdoor $\mathbf{R}$, which is why the authors perturb this solution $\mathbf{v}'$ into $\mathbf{v} = \mathbf{p} + \mathbf{v}'$, for some suitable perturbation vector $\mathbf{p}$, while adjusting $\mathbf{z}$ to verify $\mathbf{TGz} = \mathbf{u} - \mathbf{A}_{\mathbf{T}}\mathbf{p}$. By carefully choosing the covariance of the Gaussian $\mathbf{p}$, one can indeed ensure that $\mathbf{v}$ follows a spherical Gaussian distribution of width s that is independent of the trapdoor. The original sampler is actually more general, encompassing elliptical Gaussians of covariance $\mathbf{S}$ but most works use the spherical case. A specific elliptical instantiation has been studied by Jia et al. [JHT22] to compress the size of $\mathbf{v}$, i.e., having $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{Rz}$ with width s_1 and $\mathbf{v}_2 = \mathbf{p}_2 + \mathbf{z}$ with width $s_2 \approx s_1/\|\mathbf{R}\|_2$. However, the security analysis they provide only considers uniformly random targets $\mathbf{u}$. Situations where the inverted syndrome $\mathbf{u}$ is not perfectly uniform or cannot be *programmed* would require a worst-case analysis.

Although the approach above perfectly fulfils the security expectations of preimage sampling, it remains unsatisfactory in a number of aspects. First, the information on $\mathbf{R}$ in $\mathbf{v}' = [(\mathbf{Rz})^T|\mathbf{z}^T]^T$ that needs to be hidden only affects the first component. One would expect to only have to perturb the first part to ensure security. Additionally, the sampler is quite rigid as it requires sampling perturbations $\mathbf{p}$ from highly non-spherical Gaussians, and is limited to Gaussian preimages. To address these problems, Lyubashevsky and Wichs [LW15] break the symmetry between $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{Rz}$ and $\mathbf{v}_2 = \mathbf{p}_2 + \mathbf{z}$ by setting $\mathbf{p} = [\mathbf{p}_1^T|\mathbf{0}]^T$ and $\mathbf{z} = \mathbf{G}^{-1}(\mathbf{u} - \mathbf{Ap}_1)$ where $\mathbf{G}^{-1}(\cdot)$ is the base-b decomposition. Directly outputting $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{Rz}$ and $\mathbf{v}_2 = \mathbf{z}$ again leaks information on $\mathbf{R}$ because of $\mathbf{v}_1$ and they thus need to adjust this approach. By identifying $\mathbf{Ap}_1$, $\mathbf{z}$ and $\mathbf{v}_1$ with (respectively) the commitment, the challenge and the response of a zero-knowledge proof of knowledge of $\mathbf{R}$, this problem is very similar to the one of Fiat-Shamir signatures in [Lyu12]. They then resort to the same workaround, namely rejection sampling: before outputting $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{Rz}$ and $\mathbf{v}_2 = \mathbf{z}$, one performs rejection sampling on $\mathbf{v}_1$ to make its distribution independent of $\mathbf{R}$ and $\mathbf{z}$. We later refer to this sampling method as the LW sampler.

However, to thoroughly show that the preimages do not leak information on $\mathbf{R}$, they provide a simulation result which suffers from parameter constraints that make it less efficient than the MP sampler in terms of preimage size. More concretely, they show that the output distribution of the preimages is statistically close to a distribution that does not depend on the trapdoor $\mathbf{R}$ for an *arbitrary* (potentially adversarial) syndrome $\mathbf{u}$. This worst-case analysis means that nothing can be assumed about its distribution which in turn places strong restrictions on the parameters to compensate. Indeed, in their result, they need to assume that $\mathbf{Av}_1$ (and $\mathbf{Ap}_1$) is *statistically* close to uniform requiring the parameters to be large enough to use a regularity lemma. This requirement in turn leads to much larger preimages unfortunately. This looks like a paradox as one would intuitively expect the method from [LW15] to combine the best of trapdoor-based

[1] We later use the abbreviation GPV to refer to this framework.

mechanisms and rejection sampling.

4.1.1 Our Contributions

The goal of this chapter is to optimize the gadget-based samplers so as to achieve their full potential for the design of lattice-based signature schemes, regular and advanced. We therefore revisit the MP and LW samplers [MP12, LW15] and propose significant improvements that could also be relevant in other cryptographic designs beyond the sole digital signature use-case. We first perform an in-depth analysis of the elliptic sampler from [JHT22] and give the precise performance and security assessment in the case of arbitrary syndromes (worst-case), as needed by several advanced constructions, e.g., those of Part III. Then, we reassess the LW sampler in several ways, first showing that we can significantly alleviate the requirements identified in [LW15], at least when inverting uniform targets. It entails important gains in performance, making it more efficient than the samplers from [MP12] and [JHT22], thus solving the apparent paradox mentioned above. We then push this assessment further by leveraging the works on approximate trapdoors initiated by Chen et al. [CGM19] to again reduce the size of the preimages. Our approach allows to reduce the sampling error, thus yielding either higher security guarantees or better compactness.

Contribution 1: From Spherical to Elliptical, at what cost?

Our first contribution consists in a finer analysis than the one provided in [JHT22]. Let us first rapidly describe the elliptic sampler idea. Going back to the preimage sampling of [MP12], it seemed that only the first component $\mathbf{Rz}$ needed to be perturbed. However, the spherical MP sampler drowns the information by $\mathbf{p}$ symmetrically in both $\mathbf{v}'_1 = \mathbf{Rz}$ and $\mathbf{v}'_2 = \mathbf{z}$ to obtain a spherical Gaussian $\mathbf{v}$ with parameter s. A natural thought is to try perturbing $\mathbf{v}'_2$ less than $\mathbf{v}'_1$. The authors of [JHT22] then considered two Gaussian widths s_1, s_2 for $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{v}'_1$ and $\mathbf{v}_2 = \mathbf{p}_2 + \mathbf{v}'_2$ with the goal of decreasing s_2 as much as possible while retaining the same security guarantees. It shows that s_2 can be smaller than s_1 by a factor $\|\mathbf{R}\|_2$.

The security analysis was however only performed for the case of uniform syndromes $\mathbf{u}$, and assuming ideal samplers for each of the necessary subroutines. Regarding the former, we observe that several constructions, including most advanced signatures, do not invert uniform targets and must therefore be proven secure in the worst-case. Fortunately, the worst-case analysis is already covered by the original paper [MP12], but again assuming ideal subroutine samplers. Among these subroutines, one must provide a way of sampling $\mathbf{p}$ from a highly non-spherical Gaussian, and $\mathbf{z}$ from a spherical Gaussian over the gadget lattice $\mathcal{L}_q^\perp(\mathbf{G})$. In our work, we adapt the perturbation sampler from [GM18, BEP+21] for the former and the Klein sampler [Kle00, GPV08] for the latter. We obtain the precise loss incurred by these imperfect samplers on the final elliptic sampler, which leads to an improved parameter selection. We use this contribution when designing our optimized signature of Chapter 6.

Contribution 2: Re-assessing the Lyubashevsky-Wichs Sampler

We then focus on the rejection sampler provided in [LW15]. Our contribution is then to give a more specific analysis of said sampler to get rid of the restrictive requirements mentioned above and thus obtain more compact preimages. Intuitively, our new analysis stems from the observation that the initial assumption of [LW15], namely the fact that the syndrome can be fully controlled by the adversary, is too strong in some situations. Indeed, in several common cases, the worst-case analysis is not optimal because the syndrome may follow a prescribed distribution, which can be leveraged to simulate preimages in the proof.

For GPV signatures [GPV08] for example, the syndrome $\mathbf{u}$ is the hash output of the message $\mathcal{H}(\mathbf{m})$ where $\mathcal{H}$ is modelled as a random oracle. This means that the syndromes we expect are uniformly distributed and cannot be controlled by the adversary. This allows us to remove this constraint on $\mathbf{Av}_1$ being statistically close to uniform, as we can, at a high level, use the randomness of $\mathbf{u}$ to achieve the same conclusion. As we show in this chapter, getting rid of this constraint removes the need for a regularity lemma (e.g., Lemma 1.16, [MP12, Lem. 2.4]) in turn preventing a dimension blow-up[2]. It then entails significant performance improvements. In the meantime, our result avoids placing restrictions on the underlying algebraic ring R nor the working modulus q, making it suitable for a larger variety of settings and applications.

To put these improvements into perspective, we compare the performance of the different samplers. The perks of [LW15] is that although it led to $\mathbf{v}_1$ being slightly larger, it made $\mathbf{v}_2$ smaller

[2]Recall that such regularity results were the main bottleneck in Chapter 2 requiring $d = \Omega(k \log q)$.

in magnitude than in [MP12, JHT22]. Unfortunately, the constraints on the parameters had the opposite effect annihilating the prior gain vacuous. With our average-case analysis, $\mathbf{v}_2$ can now be smaller in magnitude and actual bit-size. Concretely, the total bit-size of $\mathbf{v}$ for a GPV signature built upon our improved simulation result is reduced by 65% (resp. 35%, 6%) compared to the results of [LW15] (resp. [MP12], [JHT22]). In the process of comparing the different samplers, we also analyze the impact of the gadget base b. We show that the intuition of increasing b to reduce the signature size, that was true for the MP sampler (and the worst-case LW sampler) should be re-assessed when the sampler changes. More precisely, we explain why these perform better with higher bases, and why our new analysis and parameter constraints show that the base leading to the smallest signatures is $b = 2$. As such, $\mathbf{v}_2$ is ternary which is optimal.

Contribution 3: Leveraging Approximate Trapdoors

At this stage, we have shown that the revisited LW sampler can outperform the MP one (and its elliptic version) but the resulting preimage size is still far from competitive. For example, Contribution 2 leads to a hash-and-sign signature of around 6.52 KB, whereas Falcon signatures [PFH+20] are about 0.65 KB, and Dilithium signatures [DKL+18] are 2.36 KB. To fully reinstate LW samplers, we thus need to find other means of reducing this size.

As this approach inherently leads to signatures where most elements are very small (since $\|\mathbf{v}_2\|_\infty < b$), the remaining target to improve performance is essentially the dimension of those signatures. Thanks to our new analysis above, we have already managed to reduce the one of $\mathbf{A}$, and hence of $\mathbf{v}_1$. When it comes to $\mathbf{v}_2$, the situation is more complex as the dimension seems to be dictated by the one of the gadget matrix $\mathbf{G}$. Fortunately, a study initiated by Chen et al. [CGM19] improved the performance of gadget-based constructions through the notion of approximate trapdoors. The idea is to drop the low-order gadget entries and only consider a partial gadget $\mathbf{G}_H = [b^\ell | \dots | b^{k-1}] \otimes \mathbf{I}_d$. It not only reduces the dimension of $\mathbf{v}_2$ (and hence the number of elements in the signature), but it also reduces the public and secret key sizes. Additionally, having a secret key $\mathbf{R}$ with fewer columns allows us to reduce $\|\mathbf{Rz}\|_2$ which defines the quality of our sampler, thus reducing the size of $\mathbf{v}_1$ as well.

The removed low-order entries however introduce an error on the preimage which must be taken into account in the security assessment. Intuitively, the more entries are dropped, the larger the error, and in turn the less secure it gets. Reducing the error is thus critical as it leads to better security, or enables to drop more entries to further improve performance. In this regard, we note that our revisited LW sampler lends itself well to approximate trapdoors since $\mathbf{v}_2$ is binary and not gaussian. This leads to a sampling error that is smaller than the one from [CGM19] and (almost) as small as that of the recent gadget construction of Yu, Jia and Wang [YJW23]. This optimized sampler, which we call *approximate rejection sampler*, will be one of the key component in our signature design of Chapter 5.

4.2 Reminder: The Micciancio-Peikert Sampler

We start by recalling the original gadget-based Gaussian sampler introduced by Micciancio and Peikert [MP12]. In the remainder of this thesis, we will often referred to the latter as the MP sampler. It is based on a notion of gadget trapdoors (which we call MP trapdoors) which are very versatile and enabled more efficient lattice-based designs, including advanced primitives. In particular, it yields the ability to naturally design tag-based constructions, a property leveraged in a number of works such as group signatures [dPLS18, LNPS21] or our constructions of Part III.

Gadget Trapdoors

The authors define a family trapdoor functions that can be efficiently generated and which benefits from an efficient preimage sampling procedure due to its specific form. We present it over a ring of algebraic integers although it was first presented over $\mathbb{Z}$ in [MP12]. More precisely, they generate matrices $\mathbf{A}_\mathbf{T}$ of the form

$$\mathbf{A}_\mathbf{T} = [\mathbf{A} | \mathbf{TG} - \mathbf{AR}] \bmod q\mathbb{Z} \in R_q^{d \times (m_1 + kd)},$$

where $\mathbf{R} \in R^{m_1 \times kd}$ alone represents the trapdoor. As opposed to other examples of trapdoor functions such as that of [AP09, DLP14], the trapdoor here is short but is not exactly a basis of the q-ary lattice $\mathcal{L}_q^\perp(\mathbf{A}_\mathbf{T})$. The matrix $\mathbf{A}$ lies in $R_q^{d \times m_1}$ and is generally uniform or in Hermite Normal Form $\mathbf{A} = [\mathbf{I}_d | \mathbf{A}']$ with $\mathbf{A}' \sim U(R_q^{d \times m_1 - d})$. The latter will be leveraged in Section 4.6

and also to compact the signature size in Chapter 5 and Section 6.4. Then, the tag corresponds to the matrix $\mathbf{T}$ and its sole constraint is $\mathbf{T} \in GL_d(R_q)$. The most common case is to choose $\mathbf{T} = \mathfrak{t} \cdot \mathbf{I}_d$ for some $\mathfrak{t} \in R_q^\times$ as in our constructions of Chapter 5 and 6. We already observe that it also includes the case $\mathbf{T} = \mathrm{diag}(\mathfrak{t}_1, \ldots, \mathfrak{t}_d)$ for $\mathfrak{t}_i \in R_q^\times$, which we leverage in Section 6.3. Finally, the key component of such trapdoor functions is the *gadget matrix* $\mathbf{G} \in R^{d \times kd}$. This part enables efficient lattice decoding and Gaussian sampling over $\mathcal{L}_q^\perp(\mathbf{G})$ due to its specific form. In [MP12], the authors propose $\mathbf{G}$ to be the base-b recomposition matrix, i.e., $\mathbf{G} = \mathbf{I}_d \otimes [1|b|\ldots|b^{k-1}]$ with $k = \lceil \log_b q \rceil$ which allows to decompose every element below q.

Remark 4.1 (Gadget and Centered Modular Representation)

Note that when working with centered modular arithmetic, the gadget needs to invert possibly negative elements. For $w \in (-q/2, q/2] \cap \mathbb{Z}$, the gadget inversion thus takes the base-b decomposition of $|w|$ and multiplies all coefficients by the sign of w. Additionally, the elements have magnitude at most $\lceil (q-1)/2 \rceil$ and not $q-1$. The base-b decomposition thus requires k entries where $b^k - 1 \geq \lceil (q-1)/2 \rceil$ which leads to $k = \lceil \log_b(\lceil (q-1)/2 \rceil + 1) \rceil$ instead of $k = \lceil \log_b q \rceil$. This almost never differs for large bases and moduli except for rare corner cases, but when $b = 2$ for example this saves one dimension in the gadget length and thus d columns for $\mathbf{R}$. We use this centered gadget decomposition in Sections 4.4, 4.5 and 4.6, as well as Chapter 5.

Recently, Yu et al. [YJW23] proposed another gadget to obtain more efficient decoding and shorter signatures in the random oracle model. We elaborate more on this in Chapter 5.

Gadget-Based Gaussian Preimage Sampler

As mentioned in Section 4.1, trapdoor functions are prominent tools in the design of lattice-based signature schemes using preimage sampling techniques. The MP trapdoor function also benefit from such a feature, which we now describe. The sampling algorithm relies on the link between such matrices $\mathbf{A}_\mathbf{T}$ and the gadget $\mathbf{G}$, that is

$$\mathbf{A}_\mathbf{T} \begin{bmatrix} \mathbf{R} \\ \mathbf{I}_{kd} \end{bmatrix} = \mathbf{TG} \bmod qR.$$

Thence, if $\mathbf{z}$ is a short vector in the coset $\mathcal{L}_q^\mathbf{u}(\mathbf{TG})$, we can define $\mathbf{v}' = [(\mathbf{Rz})^T | \mathbf{z}^T]^T$ which is a short vector in $\mathcal{L}_q^\mathbf{u}(\mathbf{A}_\mathbf{T})$. It means that $\mathbf{A}_\mathbf{T}\mathbf{v}' = \mathbf{u} \bmod qR$ and therefore $\mathbf{v}'$ is a *short* preimage of $\mathbf{u}$ by the function $\mathbf{A}_\mathbf{T}$. The knowledge of the secret short trapdoor $\mathbf{R}$ is then sufficient to map lattices points associated to $\mathbf{TG}$ to points associated to $\mathbf{A}_\mathbf{T}$.

It then leaves the question of finding such a short $\mathbf{z}$. Noticing that the tag matrix is invertible, it suffices to find $\mathbf{z} \in \mathcal{L}_q^{\mathbf{T}^{-1}\mathbf{u}}(\mathbf{G})$. Finally, $\mathbf{G}$ is specifically chosen to enable efficient decoding and sampling in $\mathcal{L}_q^\perp(\mathbf{G})$ which makes the latter task easy.

Unfortunately, outputting $\mathbf{v}'$ leaks information on the trapdoor $\mathbf{R}$ which is undesirable in cryptographic applications where $\mathbf{R}$ usually represents the long-term secret key. Indeed, assuming $\mathbf{z}$ is drawn from $\mathcal{D}_{\mathcal{L}_q^\mathbf{u}(\mathbf{TG}), s_\mathbf{G}}$, then it holds that $\mathbf{v}$ is statistically close to $\mathcal{D}_{\mathcal{L}_q^\mathbf{u}(\mathbf{A}_\mathbf{T}), \sqrt{\mathbf{S}'}}$ for

$$\mathbf{S}' = M_\tau\left(\begin{bmatrix} \mathbf{R} \\ \mathbf{I}_{nkd} \end{bmatrix}\right) \cdot s_\mathbf{G}^2 \mathbf{I}_{kd} \cdot M_\tau\left(\begin{bmatrix} \mathbf{R}^* & \mathbf{I}_{kd} \end{bmatrix}\right) = M_\tau\left(s_\mathbf{G}^2 \begin{bmatrix} \mathbf{R}\mathbf{R}^* & \mathbf{R} \\ \mathbf{R}^* & \mathbf{I}_{kd} \end{bmatrix}\right),$$

for example by [GMPW20, Thm. 3.1], where M_τ is the multiplication matrix map for the coefficient embedding τ defined in Section 1.1.3. In particular, it becomes possible to mount a statistical attack to approach the covariance $\mathbf{S}'$ and thus $\mathbf{R}$. To circumvent this issue, the authors use the Gaussian convolution theorem [Pei10, Thm. 3.1] recalled in Lemma 1.13 to perturb $\mathbf{v}'$ into $\mathbf{v} = \mathbf{p} + \mathbf{v}'$, for some suitable perturbation $\mathbf{p}$, while adjusting the computation of $\mathbf{z}$ to verify $\mathbf{TGz} = \mathbf{u} - \mathbf{A}_\mathbf{T}\mathbf{p} \bmod qR$. The role of this perturbation is to make the output $\mathbf{v}$ independent of $\mathbf{R}$ to ensure there is no leakage. In more details, they sample a (highly) non-spherical Gaussian perturbation $\mathbf{p} = [\mathbf{p}_1^T | \mathbf{p}_2^T]^T$ from $\mathcal{D}_{R^{m_1+kd}, \sqrt{\mathbf{S}}}$ with

$$\mathbf{S} = s^2 \mathbf{I}_{n(m_1+kd)} - \mathbf{S}' = M_\tau\left(s^2 \mathbf{I}_{m_1+kd} - s_\mathbf{G}^2 \begin{bmatrix} \mathbf{R}\mathbf{R}^* & \mathbf{R} \\ \mathbf{R}^* & \mathbf{I}_{kd} \end{bmatrix}\right), \tag{4.1}$$

and then compensate it by sampling $\mathbf{z} \hookleftarrow \mathcal{D}_{\mathcal{L}_q^\mathbf{w}(\mathbf{G}), s_\mathbf{G}}$ for $\mathbf{w} = \mathbf{T}^{-1}(\mathbf{u} - \mathbf{A}\mathbf{p}_1 + \mathbf{A}\mathbf{R}\mathbf{p}_2) - \mathbf{G}\mathbf{p}_2 \bmod qR$. The output sample is then $\mathbf{v} = [(\mathbf{p}_1 + \mathbf{Rz})^T | (\mathbf{p}_2 + \mathbf{z})^T]^T$. By [Pei10, Thm. 3.1], $\mathbf{v}$ is statistically

close to a Gaussian distribution over $\mathcal{L}_q^{\mathbf{u}}(\mathbf{A_T})$ with parameter s, which no longer depends on $\mathbf{R}$. In order to sample $\mathbf{p}$, we need $\mathbf{S}$ to be positive definite which requires $s^2 > s_{\mathbf{G}}^2(1+\|\mathbf{R}\|_2^2)$, where $\|\mathbf{R}\|_2 = \left\|M_\tau(\mathbf{R})\right\|_2$. As such, $\|\mathbf{R}\|_2$ drives the parameter selection for the sampler, and is sometimes referred to as the *quality* of the sampler. We summarize this description in Algorithm 4.1.

Algorithm 4.1: MP-Sampler$(\mathbf{R}; \mathbf{A}, \mathbf{u}, \mathbf{T}, s, s_{\mathbf{G}})$

Input: Trapdoor $\mathbf{R} \in R^{m_1 \times dk}$, Matrix $\mathbf{A} \in R_q^{d \times m_1}$, Syndrome $\mathbf{u} \in R_q^d$, Gaussian parameters $s, s_{\mathbf{G}} > 0$, tag $\mathbf{T} \in GL_d(R_q)$.

1. $\mathbf{p} \hookleftarrow \mathcal{D}_{R^{m_1+kd}, \sqrt{\mathbf{S}}}$. ▷ $\mathbf{S}$ defined in Eq. (4.1)
2. $\mathbf{w} \leftarrow \mathbf{T}^{-1}(\mathbf{u} - [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{p}) \bmod qR$.
3. $\mathbf{z} \hookleftarrow \mathcal{D}_{\mathcal{L}_q^{\mathbf{w}}(\mathbf{G}), s_{\mathbf{G}}}$.

Output: $\mathbf{v} = \mathbf{p} + \begin{bmatrix} \mathbf{Rz} \\ \mathbf{z} \end{bmatrix}$ ▷ Statistically close to $\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}(\mathbf{A_T}), s}$

4.3 Elliptic Gaussian Sampler

From the security standpoint, the approach described in Section 4.2 perfectly addresses the problem of preimage sampling for cryptographic applications. However, if we reconsider the unperturbed vector $\mathbf{v}' = [(\mathbf{Rz})^T|\mathbf{z}^T]^T$, we note that the convolution is now applied to both parts. This does not seem optimal as the bottom section of $\mathbf{v}$ is independent of $\mathbf{R}$, and $\mathbf{Rz}$ is almost always larger than $\mathbf{z}$. Unfortunately, this seems inherent to the approach stated in [Pei10, Sec. 1.3] which only considers covariance matrices of the form $s^2\mathbf{I} - \mathbf{S}'$ for some covariance $\mathbf{S}'$. Ideally, we would like to select a perturbation that only affects the top component, typically:

$$\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{0} \end{bmatrix} \sim \mathcal{D}_{R^{m_1+kd}, \sqrt{\mathbf{S}''}}, \quad \text{with} \quad \mathbf{S}'' = M_\tau\left(\begin{bmatrix} s_1^2\mathbf{I}_{m_1} - s_{\mathbf{G}}^2\mathbf{RR}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\right).$$

However, when sampling $\mathbf{z}$ and outputting $\mathbf{p} + [\mathbf{R}^T|\mathbf{I}_{kd}]^T\mathbf{z}$, we end up with a joint probability distribution of covariance (up to applying M_τ)

$$\begin{bmatrix} s_1^2\mathbf{I}_{m_1} - s_{\mathbf{G}}^2\mathbf{RR}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} + \mathbf{S}' = \begin{bmatrix} s_1^2\mathbf{I}_{m_1} & s_{\mathbf{G}}^2\mathbf{R} \\ s_{\mathbf{G}}^2\mathbf{R}^* & s_{\mathbf{G}}^2\mathbf{I}_{kd} \end{bmatrix},$$

which again leaks information about $\mathbf{R}$. This highlights the need to hide both $\mathbf{Rz}$ and $\mathbf{z}$ to rely on the convolution technique. Intuitively, the first component $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{Rz}$ can be seen as a Gaussian distribution with a secret center $\mathbf{Rz}$. Looking at its marginal distribution, one could use standard techniques to hide this secret center, namely convolution when $\mathbf{z}$ is Gaussian or noise flooding (based on either the statistical distance or the Rényi divergence as in Section 2.2) if $\mathbf{z}$ is non-Gaussian. However, giving $\mathbf{v}_2 = \mathbf{z}$ provides side information on this secret center which explains why $\mathbf{z}$ also has to be perturbed for the convolution technique to be meaningful. We therefore need a middle way between this efficient, but insecure, approach and the one from Section 4.2 that seems unnecessarily overstated given the type of vectors we have to perturb.

To do so, we break the symmetry between the top and bottom parts by using different parameters s_1 and s_2. More precisely, we sample a perturbation of covariance

$$\mathbf{S} = M_\tau\left(\begin{bmatrix} s_1^2\mathbf{I}_{m_1} & \mathbf{0} \\ \mathbf{0} & s_2^2\mathbf{I}_{kd} \end{bmatrix} - s_{\mathbf{G}}^2\begin{bmatrix} \mathbf{RR}^* & \mathbf{R} \\ \mathbf{R}^* & \mathbf{I}_{kd} \end{bmatrix}\right), \tag{4.2}$$

where s_2 can hopefully be much smaller than s_1. This approach, which we note is already encompassed by [MP12], has been recently be studied by Jia et al. [JHT22], albeit in the simplified context of simulation with uniform targets. In this section, we not only revisit the sampler with a worst-case analysis following that of [MP12], but we also precise all the intermediate samplers and incurred losses. This detailed analysis will be extremely helpful in our optimized construction of Section 6.4, and its implementation discussed in Chapter 8. More precisely, we need four different samplers:

❶ A gadget sampler for $\mathcal{D}_{\mathcal{L}_q^{\perp}(\mathbf{G}), s_{\mathbf{G}}, \mathbf{c}}$, which itself requires: (Section 4.3.1)
  ❷ A spherical base sampler for $\mathcal{D}_{\mathbb{Z}, s, c}$ or $\mathcal{D}_{R, s, c} = \mathcal{D}_{\mathbb{Z}^n, s, \tau(c)}$, (e.g., [GPV08, BLP+13, Ros20])
❸ A perturbation sampler for $\mathcal{D}_{R^{m_1+kd}, \sqrt{\mathbf{S}}}$, which itself requires: (Section 4.3.2)
  ❹ A ring sampler for $\mathcal{D}_{R, \sqrt{M_\tau(f)}, c}$ for $f \in K_{\mathbb{R}}^{++}$ and $c \in K_{\mathbb{R}}$. (Section 4.3.2 [GM18])

Combining them then gives a preimage sampler similar to that of Algorithm 4.1.

4.3.1 Klein Sampler on the Gadget Lattice

We start by describing the solution chosen for sampler ❶. The specific form of the gadget $\mathbf{G}$ implies that a short basis of the gadget lattice $\mathcal{L}_q^\perp(\mathbf{G})$ can be efficiently computed publicly. As a result, Klein's sampler [Kle00] which was thoroughly formalized by Gentry et al. [GPV08] seems the most relevant. Klein's sampler has been used in a number of popular lattice signatures such as the selected standard Falcon [PFH+20] and was thoroughly analyzed by Prest [Pre15, Pre17]. It relies on the Gram-Schmidt Orthogonalization (GSO) $\widetilde{\mathbf{B}}$ of the basis $\mathbf{B}$ of the sampled lattice. We give a (rigorously equivalent) formulation of Klein's sampler that uses the *scaled* Gram-Schmidt $\widetilde{\mathbf{B}}'$ whose columns are defined as $\widetilde{\mathbf{b}}_i/\|\widetilde{\mathbf{b}}_i\|_2^2$, and for integer centers $\mathbf{c}$. The algorithm also takes scaled widths $s_i = s_{\mathbf{G}}/\|\widetilde{\mathbf{b}}_i\|_2$ in order to output a spherical Gaussian with width $s_{\mathbf{G}}$.

The motivation is that in our case, $\mathcal{L}_q^\perp(\mathbf{G})$ (when embedded with τ) has an integer basis $\mathbf{B}_{\mathbf{G}} = \mathbf{I}_d \otimes \mathbf{B}_{\mathbf{g}} \otimes \mathbf{I}_n$ where $\mathbf{B}_{\mathbf{g}} \in \mathbb{Z}^{k\times k}$ is a basis of $\{\mathbf{x} \in \mathbb{Z}^k : [1|b|\dots|b^{k-1}]\mathbf{x} = \mathbf{0} \bmod q\mathbb{Z}\}$ defined by

$$\mathbf{B}_{\mathbf{g}} = \begin{bmatrix} b & & & & q_0 \\ -1 & b & & & \vdots \\ & \ddots & \ddots & & \vdots \\ & & \ddots & b & \vdots \\ & & & -1 & q_{k-1} \end{bmatrix},$$

with $(q_0, q_1, \dots, q_{k-1})$ is the base-b decomposition of q. As such, we can derive a closed-form expression its GSO $\widetilde{\mathbf{B}_{\mathbf{G}}} = \mathbf{I}_d \otimes \widetilde{\mathbf{B}_{\mathbf{g}}} \otimes \mathbf{I}_n$ and its scaled Gram-Schmidt $\widetilde{\mathbf{B}_{\mathbf{G}}}'$, containing only rationals. Using the scaled Gram-Schmidt slightly simplifies the description of the algorithm and can possibly be stored with fixed-points. We defer this consideration to Chapter 8. Also, the exact expressions of $\widetilde{\mathbf{B}_{\mathbf{G}}}$ and $\widetilde{\mathbf{B}_{\mathbf{G}}}'$ are a bit tedeous and not necessary for this presentation.

Algorithm 4.2: KleinSampler$(s_{\mathbf{G}}, \mathbf{c})$

Input: Gaussian parameter $s_{\mathbf{G}} > 0$, center $\mathbf{c} \in R^{dk}$.

Precomputation: From $b, q, s_{\mathbf{G}}$, precompute $\mathbf{B}_{\mathbf{G}} \in \mathbb{Z}^{ndk\times ndk}$, $\widetilde{\mathbf{B}_{\mathbf{G}}}' \in \mathbb{Q}^{ndk\times ndk}$, and $(s_i)_{i\in[\![ndk]\!]} = (s_{\mathbf{G}}/\|\widetilde{\mathbf{b}}_i\|_2)_{i\in[\![ndk]\!]}$.

1. $\mathbf{v}_{ndk} \leftarrow \mathbf{0}$
2. **for** $i = ndk, \dots, 1$ **do**
3. $\quad d_i \leftarrow \langle \tau(\mathbf{c}) - \mathbf{v}_i, \widetilde{\mathbf{b}}_i' \rangle$.
4. $\quad z_i \hookleftarrow \mathcal{D}_{\mathbb{Z}, s_i, d_i}$. ▷ Base sampler ❷
5. $\quad \mathbf{v}_{i-1} \leftarrow \mathbf{v}_i + z_i \mathbf{b}_i$.

Output: $\tau^{-1}(\mathbf{v}_0)$. ▷ Statistically close to $\mathcal{D}_{\mathcal{L}_q^\perp(\mathbf{G}), s_{\mathbf{G}}}$.

The quality of the sampler is driven by $\max_{i\in[\![ndk]\!]} \|\widetilde{\mathbf{b}}_i\|_2$ which in our case is $\sqrt{b^2+1}$. In this thesis, we consider a perfect base sampler (❷) and only study the divergence between the actually outputted distribution and the ideal distribution. This study was done by Prest [Pre17] which we state here in the case of the gadget lattice.

Lemma 4.1 (Adapted from [Pre17, Lem. 8])

Let $\varepsilon \in (0, 1/4)$ and let $s_{\mathbf{G}} \geq \eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{b^2+1}$. Denote by $\mathcal{P}$ the distribution outputted by KleinSampler$(s_{\mathbf{G}}, \mathbf{c})$ (Algorithm 4.2) for the lattice $\mathcal{L}_q^\perp(\mathbf{G})$ and a center $\mathbf{c}$. Then, it holds that

$$\mathcal{P} \approx_{\delta^{-1},\delta} \mathcal{D}_{\mathcal{L}_q^\perp(\mathbf{G}), s_{\mathbf{G}}, \mathbf{c}}, \quad \text{with} \quad \delta = \left(\frac{1+\varepsilon/ndk}{1-\varepsilon/ndk}\right)^{ndk} \underset{\varepsilon\to 0}{\sim} 1 + 2\varepsilon.$$

4.3.2 Perturbation Sampler

Let us now specify sampler ❸ used to sample perturbations $\mathbf{p}$ as described above. Non-spherical sampling requires more involved tools. Generically, Peikert [Pei10] provides a way to do so based on the convolution theorem by *randomized rounding*. It essentially samples from the *continuous* Gaussian D_1 and uses a Cholesky decomposition of $\mathbf{S} - \mathbf{I}$. Although it has a better running time in general, see [Pre15], Genise and Micciancio [GM18] proposed a way to leverage the tower structure of power-of-two cyclotomic rings to sample more efficiently. It was then extended to the

module case by Bert et al. [BEP+21] to obtain a perturbation sampler for the spherical preimage sampling. In Algorithm 4.3, we slightly adapt their algorithm to our elliptic distribution featuring two Gaussian widths s_1 and s_2 instead of one. The only difference comes in step 3 in the definition of $\mathbf{S}_{m_1}$ as the Schur complement is slighlty different. The first step only involves the spherical base sampler over R^{dk}, while the sampling of p_i has a covariance $M_\tau(f_i)$ for some $f_i \in K_{\mathbb{R}}^{++}$. This is handled by the ring sampler SampleFz from [GM18, Fig. 4] which we recall in Algorithm 4.4.

Algorithm 4.3: SamplePerturb$(\mathbf{R}, s_1, s_2, s_{\mathbf{G}})$

Input: Trapdoor $\mathbf{R} \in R^{2d\times dk}$, Gaussian parameters $s_1, s_2, s_{\mathbf{G}} > 0$.

1. $\mathbf{p}_2 \hookleftarrow \mathcal{D}_{R^{dk},\sqrt{s_2^2-s_{\mathbf{G}}^2}}$. ▷ Base sampler
2. $\mathbf{c}_{m_1} \leftarrow -s_{\mathbf{G}}^2/(s_2^2 - s_{\mathbf{G}}^2)\mathbf{R}\mathbf{p}_2$
3. $\mathbf{S}_{m_1} \leftarrow s_1^2\mathbf{I}_{m_1} - (s_{\mathbf{G}}^{-2} - s_2^{-2})^{-1}\mathbf{R}\mathbf{R}^*$.
4. **for** $i = m_1, \dots, 1$ **do**
5. Write $\mathbf{S}_i, \mathbf{c}_i$ as $\mathbf{S}_i = \begin{bmatrix}\mathbf{S}'_i & \mathbf{s}_i \\ \mathbf{s}_i^* & f_i\end{bmatrix}$ and $\mathbf{c}_i = \begin{bmatrix}\mathbf{c}'_i \\ d_i\end{bmatrix}$.
6. $p_i \leftarrow \mathsf{SampleFz}(n, f_i, d_i)$. ▷ $p_i \sim \mathcal{D}_{R,\sqrt{M_\tau(f_i)},d_i}$ (Algorithm 4.4)
7. $\mathbf{c}_{i-1} \leftarrow \mathbf{c}'_i + f_i^{-1}(p_i - d_i)\mathbf{s}_i$.
8. $\mathbf{S}_{i-1} \leftarrow \mathbf{S}'_i - f_i^{-1}\mathbf{s}_i\mathbf{s}_i^*$.
9. $\mathbf{p}_1 \leftarrow [p_1|\dots|p_{m_1}]^T$.

Output: $\mathbf{p} = \begin{bmatrix}\mathbf{p}_1 \\ \mathbf{p}_2\end{bmatrix}$ ▷ Statistically close to $\mathcal{D}_{R^{m_1+dk},\sqrt{\mathbf{S}}}$

The ring sampler SampleFz (sampler ❹) leverages the tower structure of the power-of-two cyclotomic field, by embedding an element of K_{2n} into K_n^2 where $K_{2n} = \mathbb{Q}(\zeta_{2n})$ is the $2n$-th cyclotomic field and n is a power of two. We also note $R_{2n} = \mathcal{O}_{K_{2n}}$ its ring of integers. As mentioned in Section 1.1.4, for an element a of K_{2n}, we denote by $a^{(e)} = \sum_{j\in[\![0,n/2[\![} a_{2j}\zeta_n^j \in K_n$ and and $a^{(o)} = \sum_{j\in[\![0,n/2[\![} a_{2j+1}\zeta_n^j \in K_n$ its even and odd embeddings in the subfield of degree $n/2$. The algorithm is recursive and performs operations in subrings of degree $n/2, n/4, \dots, 2, 1$.

Algorithm 4.4: SampleFz(n, f, c)

Input: Ring degree n (power-of-two, K_{2n} being the $2n$-th cyclotomic field, i.e., of degree n), Covariance f in $K_{\mathbb{R},2n}^{++}$, Center $c \in K_{\mathbb{R},2n}$.

1. **if** $n = 1$ **then**
2. $p \hookleftarrow \mathcal{D}_{\mathbb{Z},\sqrt{f},c}$. ▷ $f \in \mathbb{R}^{+*}$ and $c \in \mathbb{R}$
3. Split f and c into $(f^{(e)}, f^{(o)}) \in (K_{\mathbb{R},n}^{++})^2$ and $(c^{(e)}, c^{(o)}) \in K_{\mathbb{R},n}^2$ respectively.
4. $p^{(o)} \leftarrow \mathsf{SampleFz}(n/2, f^{(e)}, c^{(o)})$. ▷ $p^{(o)} \in R_n$
5. $c' \leftarrow c^{(e)} + f^{(o)}{f^{(e)}}^{-1}(p^{(o)} - c^{(o)})$.
6. $f' \leftarrow f^{(e)} - f^{(o)}{f^{(e)}}^{-1}f^{(o)*}$.
7. $p^{(e)} \leftarrow \mathsf{SampleFz}(n/2, f', c')$. ▷ $p^{(e)} \in R_n$
8. Recombine $p \leftarrow p^{(e)}(\zeta_{2n}^2) + \zeta_{2n} \cdot p^{(o)}(\zeta_{2n}^2) \in R_{2n}$. ▷ $p \in R_{2n} = R$

Output: p ▷ Statistically close to $\mathcal{D}_{R,\sqrt{M_\tau(f)},c}$

We note that the inverses in Algorithms 4.3 and 4.4 are inverses in $K_{\mathbb{R}}$ and not in R_q. Also, we observe that most of the update materials required in the loop of Algorithm 4.3 can be precomputed. More precisely, we can precompute all the f_i necessary for step 6, and all the $f_i^{-1}\mathbf{s}_i$ used to update the center in step 7. It then simplifies the algorithm by removing steps 3, 5 and 8.

The analysis of Algorithm 4.3 goes through the exact same way than the one from [BEP+21], as their sampler is an extension of that of [GM18], which was already general enough to encompass the elliptic case. We thus follow the proof of [BEP+21] but specifying the loss at each step similarly to [GM18].

Lemma 4.2 (Adapted from [GM18, Thm. 4.1])

Let $\varepsilon \in (0,1)$ be such that $\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})$ where $\mathbf{S}$ is defined in Equation (4.2). Denote by $\mathcal{P}$ the distribution outputted by SamplePerturb. Then, it holds that

$$\mathcal{P} \approx_{\delta^{-1},\delta} \mathcal{D}_{R^{d(2+k)},\sqrt{\mathbf{S}}}, \quad \text{with} \quad \delta = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{6d(n-1)+1} \underset{\varepsilon\to 0}{\sim} 1 + 2(6d(n-1)+1)\varepsilon.$$

4.3.3 Preimage Sampler

We can now describe in more details the elliptic sampler in Algorithm 4.5. The idea is to adapt Algorithm 4.1 by plugging the new covariance $\mathbf{S}$ and the specific samplers necessary for each step. We obtain a sample $\mathbf{v}$ whose distribution is close to $\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}(\mathbf{A_T}),[s_1\mathbf{1}_{nm_1}|s_2\mathbf{1}_{ndk}]}$, where the notation $\mathcal{D}_{\mathcal{L},\mathbf{s}}$ for a lattice of dimension N and a width vector $\mathbf{s} \in (\mathbb{R}^{+*})^N$ is explained in Section 1.3.2

Algorithm 4.5: EllipticSampler$(\mathbf{R}; \mathbf{A}, \mathbf{u}, \mathbf{T}, s_1, s_2, s_{\mathbf{G}})$

Input: Trapdoor $\mathbf{R} \in R^{2d\times dk}$, Matrix $\mathbf{A} \in R_q^{m_1\times d}$, Syndrome $\mathbf{u} \in R_q^d$, Gaussian parameters $s_1, s_2, s_{\mathbf{G}} > 0$, tag $\mathbf{T} \in GL_d(R_q)$.

1. $\mathbf{p} \leftarrow \mathsf{SamplePerturb}(\mathbf{R}, s_1, s_2, s_{\mathbf{G}})$. ▷ $\mathbf{p} \sim \mathcal{D}_{R^{d(2+k)},\sqrt{\mathbf{S}}}$ (Algorithm 4.3)
2. $\mathbf{w} \leftarrow \mathbf{T}^{-1}(\mathbf{u} - [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{p}) \bmod qR$. ▷ Syndrome correction
3. $\mathbf{c} \leftarrow \mathbf{G}^{-1}(\mathbf{w})$. ▷ Arbitrary solution such that $\mathbf{Gc} = \mathbf{w} \bmod qR$
4. $\mathbf{y} \leftarrow \mathsf{KleinSampler}(s_{\mathbf{G}}, -\mathbf{c})$. ▷ $\mathbf{y} \sim \mathcal{D}_{\mathcal{L}_q^{\perp}(\mathbf{G}),s_{\mathbf{G}},-\mathbf{c}}$ (Algorithm 4.2)
5. $\mathbf{z} \leftarrow \mathbf{c} + \mathbf{y}$. ▷ $\mathbf{z} \sim \mathcal{D}_{\mathcal{L}_q^{\mathbf{w}}(\mathbf{G}),s_{\mathbf{G}}}$
6. $\mathbf{v} \leftarrow \mathbf{p} + \begin{bmatrix}\mathbf{Rz}\\ \mathbf{z}\end{bmatrix}$.

Output: $\mathbf{v}$ ▷ Statistically close to $\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}(\mathbf{A_T}),[s_1\mathbf{1}_{nm_1}|s_2\mathbf{1}_{ndk}]}$

We then provide the main security result on the simulatability of preimages, which ensures that no leakage of the secret trapdoor $\mathbf{R}$ occurs. Notice that as opposed to the result from [JHT22] which only applies to uniform targets $\mathbf{u}$, we treat the case of arbitrary syndromes. For that, we simply use the main result from [MP12, Thm. 5.5], which already encompasses the elliptic sampler, but we adapt it to specify the precise loss incurred by using the imperfect samplers SamplePerturb and KleinSampler instead of ideal distributions.

Lemma 4.3 (Adapted from [MP12, Thm. 5.5])

Let $\varepsilon \in (0, 1/4)$ and $s_{\mathbf{G}} \geq \eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{b^2+1}$. Let s_1, s_2 be positive reals such that the matrix $\mathbf{S}$ defined as in Equation (4.2) using s_1, s_2 verifies

$$\sqrt{\mathbf{S}} \geq \eta_\varepsilon(\mathbb{Z}^{n(m_1+dk)}), \quad \text{and} \quad \mathbf{S} - \frac{s_{\mathbf{G}}^2}{s_{\mathbf{G}}^2-1} M_\tau\left(\begin{bmatrix}\mathbf{RR}^* & \mathbf{R}\\ \mathbf{R}^* & \mathbf{I}_{dk}\end{bmatrix}\right) \in \mathcal{S}^{++}_{n(m_1+dk)}$$

Denote by $\mathcal{P}_{\mathbf{u}}$ the distribution outputted by EllipticSampler on syndrome $\mathbf{u}$. Then, for all $\mathbf{u} \in R_q^d$, it holds that $\mathcal{P}_{\mathbf{u}} \approx_{\delta_1,\delta_2} \mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}([\mathbf{A}|\mathbf{TG}-\mathbf{AR}]),[s_1\mathbf{1}_{nm_1}|s_2\mathbf{1}_{ndk}]}$, where

$$\delta_1 = \left(\frac{1-\varepsilon}{1+\varepsilon}\right)^{6d(n-1)+3}\left(\frac{1-\varepsilon/ndk}{1+\varepsilon/ndk}\right)^{ndk} \underset{\varepsilon\to 0}{\sim} 1 - 2(6d(n-1)+4)\varepsilon$$
$$\delta_2 = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{6d(n-1)+2}\left(\frac{1+\varepsilon/ndk}{1-\varepsilon/ndk}\right)^{ndk} \underset{\varepsilon\to 0}{\sim} 1 + 2(6d(n-1)+3)\varepsilon.$$

To guarantee that the sampler is correct, we need to investigate the parameter constraints of Lemma 4.3. First, we directly set $s_{\mathbf{G}} = \eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{b^2+1}$ and then determine the values of s_1, s_2 so that $\mathbf{S}$ verifies the necessary conditions. We thus use the following lemma.

Lemma 4.4

Let m, ℓ be positive integers, $\mathbf{R} \in \mathbb{R}^{m\times\ell}$, and α, β, γ positive reals. The matrix

$$\mathbf{S} = \begin{bmatrix}\alpha^2\mathbf{I}_m & \mathbf{0}\\ \mathbf{0} & \beta^2\mathbf{I}_\ell\end{bmatrix} - \gamma^2\begin{bmatrix}\mathbf{R}\\ \mathbf{I}_\ell\end{bmatrix}\begin{bmatrix}\mathbf{R}^T & \mathbf{I}_\ell\end{bmatrix}$$

is in $\mathcal{S}^{++}_{m+\ell}$ if and only if $\alpha \geq \sqrt{1+1/(c^2-1)}\gamma\|\mathbf{R}\|_2$ and $\beta \geq c\gamma$ for some $c > 1$. For $c = \sqrt{2}$ it yields $\alpha \geq \sqrt{2}\gamma\|\mathbf{R}\|_2$ and $\beta \geq \sqrt{2}\gamma$.

Proof (Lemma 4.4). We can re-write $\mathbf{S}$ as

$$\mathbf{S} = \begin{bmatrix} \alpha^2\mathbf{I}_m - \gamma^2\mathbf{R}\mathbf{R}^T & -\gamma^2\mathbf{R} \\ -\gamma^2\mathbf{R}^T & (\beta^2-\gamma^2)\mathbf{I}_\ell \end{bmatrix} =: \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix}.$$

Then, by using the characterization by Schur complements, it holds that $\mathbf{S} \in \mathcal{S}^{++}_{m+\ell}$ if and only if $\mathbf{C} \in \mathcal{S}^{++}_{\ell}$ and $\mathbf{S}/\mathbf{C} = \mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T \in \mathcal{S}^{++}_{m}$. This means having

$$(\beta^2-\gamma^2)\mathbf{I}_\ell \quad \text{and} \quad \alpha^2\mathbf{I}_m - \left(\gamma^2 + \frac{\gamma^4}{\beta^2-\gamma^2}\right)\mathbf{R}\mathbf{R}^T$$

positive definite. The condition translates to $\beta > \gamma$ and $\alpha^2 > \lambda_{\max}((\gamma^2 + \gamma^4(\beta^2 - \gamma^2)^{-1})\mathbf{R}\mathbf{R}^T)$, where $\lambda_{\max}$ denotes the largest eigenvalue. It comes down to $\beta \geq c\gamma$ and $\alpha \geq \sqrt{1 + 1/(c^2-1)}\gamma\|\mathbf{R}\|_2$ for any $c > 1$ as claimed.

As a result, we have to choose the Gaussian width s_1

$$\sqrt{s_1^2 - \eta_\varepsilon(\mathbb{Z}^{n(m_1+dk)})^2} \geq \sqrt{2}s_{\mathbf{G}}\|\mathbf{R}\|_2, \quad \text{and} \quad s_1 \geq \sqrt{\frac{2s_{\mathbf{G}}^4}{s_{\mathbf{G}}^2-1}}\|\mathbf{R}\|_2, \tag{4.3}$$

and s_2 such that

$$\sqrt{s_2^2 - \eta_\varepsilon(\mathbb{Z}^{n(m_1+dk)})^2} \geq \sqrt{2}s_{\mathbf{G}}, \quad \text{and} \quad s_2 \geq \sqrt{\frac{2s_{\mathbf{G}}^4}{s_{\mathbf{G}}^2-1}}. \tag{4.4}$$

In Equation (4.3), the second condition subsumes the first one whenever $2s_{\mathbf{G}}^2\|\mathbf{R}\|_2^2/(s_{\mathbf{G}}^2-1) \geq \eta_\varepsilon(\mathbb{Z}^{n(m_1+dk)})^2$ which is generally the case for common parameters. On the contrary, the first condition of Equation 4.4 subsumes the second as we usually have $\eta_\varepsilon(\mathbb{Z}^{n(m_1+dk)})^2 \geq 2s_{\mathbf{G}}^2/(s_{\mathbf{G}}^2-1)$. We can therefore set

$$s_1 = \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)}\|\mathbf{R}\|_2, \quad \text{and} \quad s_2 = \sqrt{2s_{\mathbf{G}}^2 + \eta_\varepsilon(\mathbb{Z}^{n(m_1+dk)})^2}$$

and still inherit from the analysis of [MP12]. This allows us to drastically reduce the size of the bottom part by a factor $\|\mathbf{R}\|_2$ for free, while keeping the size of the top part (almost) the same as before. Additionally, the overall norm of $\mathbf{v}$ is smaller which can result in slightly increased concrete security. Using the perturbation sampler of Algorithm 4.3 leads to slightly improved parameters over [JHT22], but, more importantly, a drastic computational efficiency gain over the Peikert sampler [Pei10] implicitly used in [MP12] and in turn [JHT22]. We emphasize that Algorithm 4.1 can be substituted by the elliptic sampler from Algorithm 4.5 without necessitating a new security analysis. The concrete performance gains entailed by splitting the covariance with two parameters s_1 and s_2 should however be assessed on a case-by-case basis, but we expect that it would showcase improvements in most gadget-based designs.

Example 4.1

We take the example of the group signature of [LNP22, Sec. 6.4]. The scheme uses the spherical preimage sampler from Algorithm 4.1 with a parameter $s = 44233$. Our analysis of the elliptic case allows to take $s_1 = 62007$ and $s_2 = 549$ for Algorithm 4.5 which then reduces the size of their preimage by around 28%.

4.4 Rejection Sampler

The elliptic sampler allows to reduce the size of preimages for free by breaking the symmetry between the top and bottom components. It unfortunately still suffers from some of the original limitations. It indeed requires the sampling of a perturbation from a (highly) non-spherical Gaussian distribution. As a result, one needs to resort to the complex perturbation sampler which involves Schur complements on the secret trapdoor $\mathbf{R}$. This makes the implementation of the sampler quite complex. Its structure, or rather that of the ring sampler SampleFz, is actually reminiscing of the *Fast Fourier Orthogonalization* (FFO) sampler from Ducas and Prest [DP16],

used for example in the signature scheme Falcon [PFH+20], due to its recursive structure. Additionally, the elliptic sampler is seemingly limited to Gaussian distributions as its analysis relies on the Gaussian convolution theorem [Pei10, Thm. 3.1]. This in turns limits the possible preimage distributions. To circumvent these shortcomings, Lyubashevsky and Wichs [LW15] proposed a more flexible preimage sampler which we recall in Section 4.4.1. We observe however that as opposed to the MP sampler whose worst-case analysis gives roughly the same results as the one with uniform targets, this is not the case for the sampler of [LW15]. We thus identify the limitations of their worst-case analysis and provide an improved simulatability result in Section 4.4.2.

4.4.1 The Lyubashevsky-Wichs Rejection Sampler

Going back to our original thought of Section 4.2 to only perturb the top component and set $\mathbf{p}_2 = \mathbf{0}$, the issue was that it leaked information on the trapdoor $\mathbf{R}$. A more general way of avoiding such leakage is to perform rejection sampling (see Section 1.3.5) to ensure that the output is independent of $\mathbf{R}$. This is the approach adopted in [LW15].

It can be seen as combining the features of tag-friendly gadget-based preimage sampling with rejection sampling that is extensively used in Fiat-Shamir with Aborts (FSwA) signatures. Let $\mathbf{G}^{-1}(\cdot)$ be the entry-wise base-b decomposition of vectors of R_q^d. As we explain below in Remark 4.1, we consider a centered representation of $\mathbb{Z}_q$ which results in a signed base-b decomposition. Hence, $\mathbf{G}^{-1}$ maps to vectors of S_{b-1}^{dk}. The intuition is to sample a perturbation $\mathbf{p}_1 \in R^{m_1}$ from a source distribution $\mathscr{D}_s$. Further, instead of using Gaussian $\mathbf{G}$-sampling, we simply use $\mathbf{G}^{-1}$ and obtain $\mathbf{v}_2 = \mathbf{G}^{-1}(\mathbf{T}^{-1}(\mathbf{u} - \mathbf{A}\mathbf{p}_1) \bmod qR)$. Then, we can define $\mathbf{v}_1 = \mathbf{p}_1 + \mathbf{R}\mathbf{v}_2$ so that the relation $\mathbf{A}_\mathbf{T}\mathbf{v} = \mathbf{u} \bmod qR$ is verified, and apply rejection sampling to make $\mathbf{v}_1$ independent of $\mathbf{R}\mathbf{v}_2$ and in turn $\mathbf{R}$. This setting is reminiscent of lattice-based zero-knowledge arguments or Lyubashevsky's signature scheme [Lyu12], where $\mathbf{R}$ is the witness, $\mathbf{p}_1$ is the mask, $\mathbf{A}\mathbf{p}_1$ is a commitment to the mask, $\mathbf{v}_2$ is the challenge, and $\mathbf{v}_1$ is the response to the challenge. We now give the description in Algorithm 4.6.

Algorithm 4.6: LW-Sampler$(\mathbf{R}; \mathbf{A}, \mathbf{T}, \mathbf{u}, \mathscr{D}_s, \mathscr{D}_t)$

Input (offline phase)**:** Matrix $\mathbf{A} \in R_q^{d \times m_1}$, Source distribution $\mathscr{D}_s$ over R^{m_1}.
Input (online phase)**:** Trapdoor $\mathbf{R} \in R^{m_1 \times dk}$, Tag $\mathbf{T} \in GL_d(R_q)$, Syndrome $\mathbf{u} \in R_q^d$, Target distribution $\mathscr{D}_t$ over R^{m_1} such that rejection sampling can be performed with respect to the source distribution $\mathscr{D}_s$.

Offline phase
1. $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$.
2. $\mathbf{w} \leftarrow \mathbf{A}\mathbf{p}_1 \bmod qR$.

Online phase
3. $\mathbf{x} \leftarrow \mathbf{T}^{-1}(\mathbf{u} - \mathbf{w}) \bmod qR$. ▷ Syndrome correction
4. $\mathbf{v}_2 \leftarrow \mathbf{G}^{-1}(\mathbf{x}) \in S_{b-1}^{dk}$. ▷ Deterministic
5. $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{v}_2$.
6. Sample a continuous $u \hookleftarrow U([0,1])$.
7. **if** $u > \min\left(1, \frac{\mathscr{D}_t(\mathbf{v}_1)}{M \cdot \mathscr{D}_s(\mathbf{p}_1)}\right)$ **then** go back to 1. ▷ Rejection

Output: $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$

Limitations of the Worst-Case Analysis

At first glance, the approach from [LW15] seems to fully achieve what we wanted to do in Section 4.2, namely to completely break the symmetry between $\mathbf{v}_1$ and $\mathbf{v}_2$ to reduce the size of $\mathbf{v}_2$. However, in practice, the choice of parameters and suitable distributions $\mathscr{D}_s, \mathscr{D}_t$ is conditioned by the security requirements coming from the simulation result of [LW15, Thm. 3.1]. Unfortunately, the latter is too restrictive in most cases, which explains why it does not lead to improvements on the preimage size, as we explain below.

Concretely, in [LW15, Thm. 3.1], it is shown that the output distribution of LW-Sampler is statistically close to some ideal distribution that does not depend on the trapdoor $\mathbf{R}$ for an arbitrary (potentially adversarial) syndrome $\mathbf{u}$. It means that a preimage $\mathbf{v}$ of $\mathbf{u}$ can be simulated without resorting to the trapdoor $\mathbf{R}$, and thus does not leak information on $\mathbf{R}$. There are however some challenges to overcome in order to prove this result. The first one is to identify this ideal distribution that must additionally be close to the one of actual preimages. If we focus on the $\mathbf{v}_2$ component of these preimages, we indeed note that the Algorithm 4.6 generates them as $\mathbf{G}^{-1}(\mathbf{T}^{-1}(\mathbf{u} - \mathbf{w}) \bmod$

qR) where $\mathbf{w} = \mathbf{A}\mathbf{p}_1 \bmod qR$. If $\mathbf{w}$ is non-uniform, then so is $\mathbf{T}^{-1}(\mathbf{u} - \mathbf{w})$, which makes the distribution of $\mathbf{v}_2$ complex to define when $\mathbf{u}$ is arbitrary.

It therefore seems necessary to assume that $\mathbf{A}\mathbf{p}_1$ is close to uniform, but at this stage one could still wonder whether a computational argument is sufficient. Unfortunately we here face a second challenge which is due to the very nature of the perturbation $\mathbf{p}_1$. Indeed, $\mathbf{p}_1$ does not only affect the syndrome (through $\mathbf{A}\mathbf{p}_1$) but also the preimage as it is eventually added to its upper component to form $\mathbf{v}_1$. In a computational argument, one would end up with an intermediate game where $\mathbf{A}\mathbf{p}_1$ would be replaced by some random vector $\mathbf{r}$, but then how to generate $\mathbf{v}_1$? The syndrome would indeed be $\mathbf{u} + \mathbf{r}$, which seems impossible to invert without resorting to the trapdoor since the reduction does not control $\mathbf{u}$.

This is why the authors of [LW15] need to assume that $\mathbf{A}\mathbf{p}_1$ is *statistically* close to uniform requiring $\mathbf{p}_1$ to have a high entropy in order to use a regularity lemma (like that of Lemma 1.16), which in turn leads to large parameters (either in the dimension m_1 of $\mathbf{p}_1$, or in the size of its entries). This in particular prevents them from using a (much more efficient) computational instantiation of MP trapdoors where $m_1 = 2d$ and $\mathbf{AR} \bmod qR$ can be argued to be pseudorandom based on M-LWE rather than the leftover hash lemma. This results in significant performance losses which cancel out the benefits of having a smaller $\mathbf{v}_2$. In addition, regularity lemmas generally require the modulus q to be prime and/or with low splitting [LW20] in the ring R, which may be undesirable for concrete applications.

We give concrete parameter and performance estimates in Table 4.1 following the original result and parameter selection from [LW15, Sec 3.2] in the Gaussian case, i.e., when $\mathscr{D}_s = \mathscr{D}_t = \mathcal{D}_{R^m,s}$ is a spherical discrete Gaussian of width s. Their simulation result leads to choosing $m_1 = dk = d\lceil \log_b q \rceil$ and $s = \gamma \cdot (b-1)\sqrt{ndk}(\sqrt{ndk} + \sqrt{ndk} + t)$, for a slack $\gamma \approx 8$. Overall it yields a signature of around 19 KB whereas the original MP sampler yields signature of approximately 10 KB.

4.4.2 An Improved Simulatability for Uniform Targets

Worst-case analyses of preimage samplers is required in several lattice constructions, including the ones from Part III. The analysis of [LW15] then shows in this case that the LW sampler is outperformed by the MP sampler, and a fortiori by the elliptic sampler of Section 4.3. But we emphasize that this is *not required* in a variety of other designs. When inverting uniformly random targets, one can possibly lighten certain requirements. This is for example the case for GPV signatures [GPV08], where the syndrome $\mathbf{u}$ is the hash output $\mathcal{H}(\mathbf{m})$ of the message $\mathbf{m}$, where $\mathcal{H}$ is modelled as a random oracle. This means that the syndromes we use are uniformly distributed and non-adversarial. We now explain how to get rid of the limitations of the LW sampler we identified above when the syndrome follows a prescribed uniform distribution.

This assumption drastically changes the proof strategy. Indeed, we first note that we no longer have to study the distribution of $\mathbf{v}$ conditioned on some arbitrary $\mathbf{u}$ as we can now consider the joint distribution of $\mathbf{v}$ and $\mathbf{u}$. Put differently, we can now manipulate these two vectors as long as their joint distribution is correct, which offers a lot more flexibility in the proof. In particular, this allows to circumvent the challenges faced in the proof of [LW15] because we can now leverage the randomness of $\mathbf{u}$ to compensate the one introduced by the computational assumption. More precisely, this allows us to specify the expected distribution of $\mathbf{v}_2$ as $\mathbf{T}^{-1}(\mathbf{u} - \mathbf{A}\mathbf{p}_1 \bmod qR)$ is now uniform because $\mathbf{u}$ is uniform and independent of $\mathbf{A}\mathbf{p}_1 \bmod qR$.

This removes the restriction on $\mathbf{A}\mathbf{p}_1$ being statistically uniform, while still being able to simulate the pairs $(\mathbf{v}, \mathbf{u})$ without resorting to the trapdoor $\mathbf{R}$. Note that $\mathbf{p}_1$ still needs to have a sufficient entropy so as to hide $\mathbf{R}\mathbf{v}_2$, which is given by the rejection sampling condition in Lemma 1.24. This trapdoor-independence property of the preimages is necessary for cryptographic applications, e.g., signatures, as an adversary can usually have access to many such preimages (and syndromes) for a single key. As a consequence, we no longer need a large perturbation (either in norm or dimension), which leads to improved performances, as illustrated by the tables in Section 4.5.2. We provide our new simulation result in Theorem 4.1 dealing with uniform targets.

Theorem 4.1 (Simulation with Uniform Targets)

Let R be the ring of integers of a number field. Let d, q, b, m_1 be positive integers with $b \geq 2$, $m_1 \geq d$, and let $k = \lceil \log_b(\lceil (q-1)/2 \rceil + 1) \rceil$. Let $\mathscr{D}_r, \mathscr{D}_s, \mathscr{D}_t$ be three distributions over $R^{m_1 \times dk}$, R^{m_1} and R^{m_1} respectively. Let $\mathbf{A} \in R_q^{d \times m_1}$, $\mathbf{R} \sim \mathscr{D}_r$, $\mathbf{T} \in GL_d(R_q)$. Then, let $Y \subseteq R^{m_1}$ be the support of the distribution of $\mathbf{R} \cdot \mathbf{G}^{-1}(U(R_q^d))$. Let $M > 1, \varepsilon \in [0, 1/2]$ such that $\max_{\mathbf{R}\mathbf{v}_2 \in Y} RD_\infty^\varepsilon(\mathscr{D}_t \| \mathscr{D}_s^{+\mathbf{R}\mathbf{v}_2}) \leq M$. We then define two distributions

$\mathcal{P}_1$: $\mathbf{u} \hookleftarrow U(R_q^d)$, and $\mathbf{v} \leftarrow \mathsf{LW\text{-}Sampler}(\mathbf{R}; \mathbf{A}, \mathbf{T}, \mathbf{u}, \mathscr{D}_s, \mathscr{D}_t)$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{P}_2$:
1. $\mathbf{v}_1 \hookleftarrow \mathscr{D}_t$, $\mathbf{v}_2 \hookleftarrow \mathbf{G}^{-1}(U(R_q^d))$.
2. $\mathbf{v} \leftarrow [\mathbf{v}_1^T | \mathbf{v}_2^T]^T$.
3. $\mathbf{u} \leftarrow [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{v} \bmod qR$.
4. With probability $1 - 1/M$ go back to 1.

Output: $(\mathbf{v}, \mathbf{u})$.

Then, $\Delta(\mathcal{P}_1, \mathcal{P}_2) \leq \varepsilon$ and for all $\alpha \in (1, +\infty]$, $\mathrm{RD}_\alpha(\mathcal{P}_1 \| \mathcal{P}_2) \leq 1/(1-\varepsilon)^{\alpha/(\alpha-1)}$.

Proof (Theorem 4.1). We define the following hybrid distributions from $\mathcal{H}_1$ to $\mathcal{H}_5$, where $\mathcal{H}_1 = \mathcal{P}_1$ and $\mathcal{H}_5 = \mathcal{P}_2$.

$\mathcal{H}_1$: $\mathbf{u} \hookleftarrow U(R_q^d)$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{x}' \leftarrow \mathbf{u} - \mathbf{Ap}_1 \bmod qR$, $\mathbf{v}_2 \leftarrow \mathbf{G}^{-1}(\mathbf{T}^{-1}\mathbf{x}')$, $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{Rv}_2$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1)/(M \cdot \mathscr{D}_s(\mathbf{p}_1)))$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1^T | \mathbf{v}_2^T]^T$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_2$: $\mathbf{x}' \hookleftarrow U(R_q^d)$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{u} \leftarrow \mathbf{x}' + \mathbf{Ap}_1 \bmod qR$, $\mathbf{v}_2 \leftarrow \mathbf{G}^{-1}(\mathbf{T}^{-1}\mathbf{x}')$, $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{Rv}_2$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1)/(M \cdot \mathscr{D}_s(\mathbf{p}_1)))$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1^T | \mathbf{v}_2^T]^T$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_3$: $\mathbf{v}_2 \hookleftarrow \mathbf{G}^{-1}(U(R_q^d))$, $\mathbf{x}' \leftarrow \mathbf{TGv}_2 \bmod qR$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{u} \leftarrow \mathbf{x}' + \mathbf{Ap}_1 \bmod qR$, $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{Rv}_2$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1)/(M \cdot \mathscr{D}_s(\mathbf{p}_1)))$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1^T | \mathbf{v}_2^T]^T$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_4$: $\mathbf{v}_2 \hookleftarrow \mathbf{G}^{-1}(U(R_q^d))$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{v}_1 \leftarrow \mathbf{p}_1 + \mathbf{Rv}_2$, $\mathbf{v} \leftarrow [\mathbf{v}_1^T | \mathbf{v}_2^T]^T$, $\mathbf{u} \leftarrow [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{v} \bmod qR$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1)/(M \cdot \mathscr{D}_s(\mathbf{p}_1)))$. If not go to output.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_5$: $\mathbf{v}_2 \hookleftarrow \mathbf{G}^{-1}(U(R_q^d))$, $\mathbf{v}_1 \hookleftarrow \mathscr{D}_t$, $\mathbf{v} \leftarrow [\mathbf{v}_1^T | \mathbf{v}_2^T]^T$, $\mathbf{u} \leftarrow [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{v} \bmod qR$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > 1/M$. If not go to output.
Output: $(\mathbf{v}, \mathbf{u})$.

Let us now show that these distributions are statistically close to each other.

$\mathcal{H}_1$ - $\mathcal{H}_2$: Here we just change the sampling order of $\mathbf{u}$ and $\mathbf{x}'$. In $\mathcal{H}_2$ the vector $\mathbf{x}'$ is uniform and independent of $\mathbf{Ap}_1$ implying that $\mathbf{u}$ is also uniform, as in $\mathcal{H}_1$. Hence $\mathcal{H}_1$ and $\mathcal{H}_2$ are identically distributed.

$\mathcal{H}_2$ - $\mathcal{H}_3$: We now change the way $\mathbf{x}'$ is generated. Notice that, for correctness, once $\mathbf{x}'$ is fixed then so is $\mathbf{v}_2$ and vice-versa. In $\mathcal{H}_2$, since $\mathbf{T}$ is in $GL_d(R_q)$, $\mathbf{T}^{-1}\mathbf{x}'$ also follows the uniform distribution over R_q^d. As a result, $\mathbf{v}_2$ follows exactly $\mathbf{G}^{-1}(U(R_q^d))$ as in $\mathcal{H}_3$. Also, $\mathbf{x}'$ is coherently set in $\mathcal{H}_3$. Indeed, in $\mathcal{H}_2$, we have $\mathbf{TGv}_2 = \mathbf{T}(\mathbf{T}^{-1}\mathbf{x}') \bmod qR = \mathbf{x}' \bmod qR$. Thence, $\mathcal{H}_2$ and $\mathcal{H}_3$ are identically distributed as well.

$\mathcal{H}_3$ - $\mathcal{H}_4$: $\mathcal{H}_4$ is merely a re-writing of $\mathcal{H}_3$. Indeed, in $\mathcal{H}_3$, $\mathbf{x}'$ only acts as an intermediate vector to define $\mathbf{u}$. Defining $\mathbf{R}' = [\mathbf{R}^T | \mathbf{I}_{kd}]^T$, we have $[\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{R}' = \mathbf{TG} \bmod qR$. In $\mathcal{H}_3$, this yields

$$\begin{aligned}\mathbf{u} = \mathbf{TGv}_2 + \mathbf{Ap}_1 \bmod qR &= [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{R}'\mathbf{v}_2 + \mathbf{Ap}_1 \bmod qR \\ &= [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]\mathbf{v} \bmod qR,\end{aligned}$$

as $\mathbf{v} = [\mathbf{p}_1^T | \mathbf{0}]^T + \mathbf{R}'\mathbf{v}_2$. Again, $\mathcal{H}_3$ and $\mathcal{H}_4$ are identical.

$\mathcal{H}_4$ - $\mathcal{H}_5$: We now change the way $\mathbf{v}_1$ is generated by using the rejection sampling result. In $\mathcal{H}_4$, $\mathbf{Rv}_2$ is distributed according to $\mathbf{R} \cdot \mathbf{G}^{-1}(U(R_q^d))$ with support Y as defined in the theorem

statement. By our assumptions on Y, $\mathscr{D}_s$, $\mathscr{D}_t$, the rejection sampling result from Lemma 1.24 yields that

$$\Delta((\mathbf{v}_1,\mathbf{v}_2)_{\mathcal{H}_4},(\mathbf{v}_1,\mathbf{v}_2)_{\mathcal{H}_5}) \leq \varepsilon \quad \text{and} \quad \mathrm{RD}_\alpha((\mathbf{v}_1,\mathbf{v}_2)_{\mathcal{H}_4}\|(\mathbf{v}_1,\mathbf{v}_2)_{\mathcal{H}_5}) \leq \frac{1}{(1-\varepsilon)^{\frac{\alpha}{\alpha-1}}},$$

for all $\alpha > 1$. By Lemma 1.7 and 1.8, it holds

$$\Delta(\mathcal{H}_4,\mathcal{H}_5) \leq \varepsilon \quad \text{and} \quad \mathrm{RD}_\alpha(\mathcal{H}_4\|\mathcal{H}_5) \leq \frac{1}{(1-\varepsilon)^{\frac{\alpha}{\alpha-1}}}.$$

Since $\mathcal{H}_1 = \mathcal{P}_1$ and $\mathcal{H}_5 = \mathcal{P}_2$, combining the above gives the result.

We can see that there is no requirement on the regularity of $\mathbf{A}\mathbf{p}_1 \bmod qR$. As a consequence, our simiulatability result does not place any restrictions on the modulus q, nor the field K. Typically, regularity lemmas in the module setting are usually restricted to monogenic fields and/or to prime modulus with low splitting (e.g., Lemma 1.16). Our simulation avoids these constraints altogether. Additionally, the result specifies to the integers by choosing the field $K = \mathbb{Q}$, and thus the ring $R = \mathbb{Z}$, of degree $n = 1$. This also allows us to set $m_1 = 2d$ and argue that $\mathbf{A}\mathbf{R} \bmod qR$ is uniform based on M-LWE rather than the leftover hash lemma. Though, for now, we keep the dimension m_1 to remain general. This clearly shows that the regime of uniform targets is much simpler to deal with in the case of the LW sampler, whereas it incurs almost no difference for the MP sampler.

Theorem 4.1 also provides the simulation in Rényi divergence because, as noted for example in [Pre17], it usually leads to tighter constructions. One can indeed take a much larger ε for (almost) the same security guarantees, which in turn relaxes the constraints on other parameters. This follows the same observation as the one made in the reduction of Section 2.2 to argue in favor of a Rényi divergence-based analysis. Because the unforgeability of signatures is a search problem, adopting the same divergence tool makes sense, and actually leads to tighter security reductions.

Remark 4.2

Our proof strategy would still work if the syndrome $\mathbf{u}$ were statistically uniform and not necessarily a hash output. This is for example the case in the construction of Section 6.2 where we simulate one signature query $\mathbf{v}$ along with the public key syndrome $\mathbf{u}$. However, we note that in this construction, we aim for a statistical regime anyway which requires us to use regularity lemmas in other places. We thus fall in the parameter regime of [LW15], i.e., $m_1 \approx kd$, where the original MP sampler performs better. Its optimized version of Section 6.4 then requires a worst-case analysis of the sampler which is not covered by Theorem 4.1. We therefore do not consider the LW sampler in Chapter 6.

4.4.3 Example: Spherical Gaussian

One of the main benefits of the LW sampler is that it can be instantiated with a plethora of distributions. For comparison purposes in Section 4.5.2, we provide the instantiation with discrete Gaussian distributions. We note however that it only involves a spherical Gaussian distribution on $\mathbf{v}_1$ which can be sampled from using a base sampler $\mathcal{D}_{R^{m_1},s}$. We instantiate it with a non-Gaussian distribution in the approximate case in Section 5.4.1.

We choose $\mathscr{D}_r = U(S_1^{m_1\times dk})$ for the trapdoor distribution, and we select either $\mathscr{D}_s = \mathscr{D}_t = \mathcal{D}_{R^{m_1},s}$. For convenience, we write $\mathsf{LW\text{-}Sampler}(\mathbf{R};\mathbf{A},\mathbf{T},\mathbf{u},s)$ instead of specifying $\mathscr{D}_s$ and $\mathscr{D}_t$. In the Gaussian case, we need to derive a bound T on $\|\mathbf{R}\mathbf{v}_2\|_2$ to bound the smooth Rényi divergence with Lemma 1.11. For that, we upper-bound it by $\|\mathbf{R}\|_2\|\mathbf{v}_2\|_2$, and apply Heuristic 1.1 to get the standard inequality $\|\mathbf{R}\|_2 \leq \sqrt{nm_1} + \sqrt{ndk} + t =: B$. To thoroughly match the conditions of the rejection sampling, we need to enforce this spectral bound on $\|\mathbf{R}\|_2$ before the sampling procedure. Since $\mathbf{R}$ represents the secret key, it should be enforced during key generation[3]. As it is verified with overwhelming probability if $t = \log_2\lambda$ say, this only discards a negligible fraction of all the possible keys. We therefore actually apply Theorem 4.1 on $\mathscr{D}_r =$ "$U(S_1^{m_1\times dk})$ conditioned on $\|\mathbf{R}\|_2 \leq B$". We then choose a repetition rate $M > 1$ and a loss ε, which both define the minimal slack $\gamma > 0$ so that $s = \gamma T$. We get the following corollary.

[3]It is also the case for the original MP sampler as it may happen (albeit with negligible probability) that the sampler fails if $\mathbf{R}$ has norm larger than the bound used to set the Gaussian width s.

Corollary 4.1 (Gaussian Rejection Sampler)

Let n, d, q, b be positive integers with n a power of two, $b \geq 2$, and define the gadget dimension $k = \lceil \log_b(\lceil (q-1)/2 \rceil + 1) \rceil$. Let R be the power-of-two cyclotomic ring of degree n. Let $t > 0$, and $T = (b-1)\sqrt{ndk}(\sqrt{nm_1} + \sqrt{ndk} + t)$. Let $M > 1$, $\varepsilon \in (0, 1/2]$ and define $\gamma = \frac{\sqrt{\pi}}{\ln M}(\sqrt{\ln \varepsilon^{-1} + \ln M} + \sqrt{\ln \varepsilon^{-1}})$. Finally $s = \gamma T$. Let $\mathbf{A} \in R_q^{d \times m_1}$, $\mathbf{R} \sim U(S_1^{m_1 \times dk})$ conditioned on $\|\mathbf{R}\|_2 \leq \sqrt{nm_1} + \sqrt{ndk} + t$, and $\mathbf{T} \in GL_d(R_q)$. We define $\mathcal{P}_1$ and $\mathcal{P}_2$ the same way as in Theorem 4.1 but where $\mathscr{D}_s, \mathscr{D}_t$ are replaced with $\mathcal{D}_{R^{m_1}, s}$.
Then, it holds that $\Delta(\mathcal{P}_1, \mathcal{P}_2) \leq \varepsilon$ and $\mathrm{RD}_\alpha(\mathcal{P}_1 \| \mathcal{P}_2) \leq 1/(1-\varepsilon)^{\alpha/(\alpha-1)}$ for all $\alpha \in (1, +\infty]$.

Proof (Corollary 4.1). We simply have to verify that the smooth Rényi divergence condition of Theorem 4.1 holds. In our context, we restrict the matrices $\mathbf{R}$ to have a bounded spectral norm. Following the notations of Theorem 4.1, the distribution $\mathscr{D}_r$ consists in sampling $\mathbf{R}$ from $U(S_1^{m_1 \times dk})$ such that $\|\mathbf{R}\|_2 \leq B$, where $B = \sqrt{2nd} + \sqrt{ndk} + t$. It holds that $\mathscr{D}_r$ is efficiently sampleable because the bound is verified with high probability by Heuristic 1.1. The set Y is the support of $\mathbf{R} \cdot \mathbf{G}^{-1}(U(R_q^d))$. Hence, for all $\mathbf{R}\mathbf{v}_2$ in Y, we have $\|\mathbf{R}\mathbf{v}_2\|_2 \leq \|\mathbf{R}\|_2 \|\mathbf{v}_2\|_2 \leq B \cdot (b-1)\sqrt{ndk} = T$. We note that since $Y \subset R^{m_1}$, we have $\mathcal{D}^{+\mathbf{R}\mathbf{v}_2}_{R^{m_1}, s} = \mathcal{D}_{R^{m_1}, s, \mathbf{R}\mathbf{v}_2}$ for all $\mathbf{R}\mathbf{v}_2 \in Y$. Using Lemma 1.11, it thus holds that

$$\begin{aligned}
\mathrm{RD}^\varepsilon_\infty(\mathcal{D}_{R^{m_1}, s} \| \mathcal{D}^{+\mathbf{R}\mathbf{v}_2}_{R^{m_1}, s}) &\leq \exp\left(\pi \frac{\|\mathbf{R}\mathbf{v}_2\|_2^2}{s^2} + 2\frac{\|\mathbf{R}\mathbf{v}_2\|_2}{s}\sqrt{\pi \ln \varepsilon^{-1}}\right) \\
&\leq \exp\left(\pi \frac{T^2}{s^2} + 2\frac{T}{s}\sqrt{\pi \ln \varepsilon^{-1}}\right) \\
&\leq M,
\end{aligned}$$

where the last inequality follows from the fact that $s = \frac{\sqrt{\pi}}{\ln M}(\sqrt{\ln \varepsilon^{-1} + \ln M} + \sqrt{\ln \varepsilon^{-1}}) \cdot T$. Thence, $\max_{\mathbf{R}\mathbf{v}_2 \in Y} \mathrm{RD}^\varepsilon_\infty(\mathcal{D}_{R^{m_1}, s} \| \mathcal{D}^{+\mathbf{R}\mathbf{v}_2}_{R^{m_1}, s}) \leq M$. Theorem 4.1 then yields the result.

In this specific instantiation with Gaussian distributions, we only reach widths s which are larger than the ones from [MP12]. Indeed, in the latter, $\mathbf{v}_1$ was distributed according to a discrete Gaussian of width $s = \Theta(b\|\mathbf{R}\|_2) = \Theta(b(\sqrt{nm_1} + \sqrt{ndk}))$, while here we obtain a width $s = \Theta(b\sqrt{ndk}(\sqrt{nm_1} + \sqrt{ndk}))$. However, in the meantime, we drastically reduce the size of $\mathbf{v}_2$, which largely compensate for the increase in size of $\mathbf{v}_1$ for typical parameters, as shown in Section 4.5.2.

4.5 Optimal Gadget Base and Sampler Performance

In the computational instantiation of MP trapdoors, the gadget base b is an important parameter to optimize over. Since the base defines the length of the gadget matrix $dk = d\lceil \log_b(\lceil (q-1)/2 \rceil + 1) \rceil$, choosing a larger base results in lower dimensional vectors, at the expense of a larger norm. As the norm only impacts the bitsize logarithmically while the dimension impacts it linearly, one could think that the optimal choice for b is around $\sqrt{q}$, thus resulting in $k = 2$, smaller preimages and in turn smaller signatures. The goal of this section is to show that the optimal base actually depends on the preimage sampler. We illustrate our discussion with the instructive example of GPV signatures [GPV08]. Other applications would need a similar assessment. We compare signatures generated using the MP sampler [MP12], the elliptic sampler from Section 4.3 (thereafter called MP*), the LW sampler [LW15] with the worst-case analysis (denoted by LW) and those resulting from our simulation of Corollary 4.1 (denoted by LW*). In the process, we demonstrate interesting improvement factors on the size of preimages, which represents a step towards concrete practicality of constructions based on MP trapdoors.

In this section, we look at the size of the preimage which matches the signature size when using the preimage sampler in the GPV Hash-and-Sign framework [GPV08]. We recall this signature paradigm in Section 5.2. For the purpose of the present comparison study, we only need to know that the security relies on M-SIS with a bound $\beta \geq \|\mathbf{v} - \mathbf{v}^\star\|_2$ for two preimages $\mathbf{v}, \mathbf{v}^\star$. To evaluate the signature size, we assume that $\mathbf{A} = [\mathbf{I}_d | \mathbf{A}']$ so that the signature is simply $(\mathbf{v}_{1,2}, \mathbf{v}_2)$, where $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T$ with $\mathbf{v}_{1,2} \in R^{m_1 - d}$. This signature compacting trick is commonly used for example in [PFH+20, EFG+22, ETWY22]. Finally, the bitsize of Gaussian vectors is estimated by

the entropy bound, which can be achieved using the rANS encoding as discussed in [ETWY22]. More precisely, for a discrete Gaussian vector of dimension N and width s, the entropy bound is close to $N/2 \cdot (1 + \log_2 s^2) = N(1/2 + \log_2 s)$.

4.5.1 Choosing the Gadget Base

The main difficulty when determining the optimal base for a given sampler is that b impacts both the bitsize evaluation of the signature and the hardness of the underlying computational assumptions. As the latter in turn affects the parameters (and hence the bitsize), this may lead to some counterintuitive situations. Whenever possible, we use the computational instantiation with $m_1 = 2d$.

For a given base, the minimal Gaussian parameter needed for MP signatures $\mathbf{v}$ is[4] $s \approx \gamma_1 b \|\mathbf{R}\|_2$, with γ_1 linked to the smoothing parameter and where $\|\mathbf{R}\|_2$ can be bounded by Heuristic 1.1 by $\sqrt{2nd} + \sqrt{ndk} + t$ for a slack $t \approx 7$ which thus depends on b as $\sqrt{1/\ln(b)}$. The bitsize of a signature is thus

$$|\mathsf{sig}_{\mathsf{MP}}| \approx nd(1/2 + \log_2(\gamma_1 b \|\mathbf{R}\|_2)) + nd \log_b(q)(1/2 + \log_2(\gamma_1 b \|\mathbf{R}\|_2)). \tag{4.5}$$

For MP* signatures, we introduce an asymmetry between $\mathbf{v}_1$ and $\mathbf{v}_2$ and thus have two Gaussian parameters $s_1 = \gamma_2 b \|\mathbf{R}\|_2$ and $s_2 = s_1/\|\mathbf{R}\|_2$. The bitsize of a signature is

$$|\mathsf{sig}_{\mathsf{MP}^\star}| \approx nd(1/2 + \log_2(\gamma_2 b \|\mathbf{R}\|_2)) + nd \log_b(q)(1/2 + \log_2(\gamma_2 b)). \tag{4.6}$$

For the LW sampler from Algorithm 4.6, the Gaussian width for $\mathbf{v}_1$ is given by $s \approx \gamma_3 b \|\mathbf{R}\|_2 \sqrt{ndk}$ where γ_3 defines the repetition rate M. As mentioned in Section 4.4.1, the dimension m_1 for LW signatures is chosen to be $m_1 = dk$ instead of $m_1 = 2d$ for MP* signatures. The corresponding bitsizes are thus given by

$$|\mathsf{sig}_{\mathsf{LW}}| \approx nd(\log_b(q) - 1)(1/2 + \log_2(\gamma_3 b \|\mathbf{R}\|_2 \sqrt{ndk})) + nd \log_2 q \tag{4.7}$$

$$|\mathsf{sig}_{\mathsf{LW}^\star}| \approx nd(1/2 + \log_2(\gamma_3 b \|\mathbf{R}\|_2 \sqrt{ndk})) + nd \log_2 q. \tag{4.8}$$

We already see that the size of $\mathbf{v}_2$, for both LW and LW*, is $nd \log_2 q$, independently of the choice of b. This is because we can equivalently send $\mathbf{x} \in R_q^d$ instead of $\mathbf{v}_2 = \mathbf{G}^{-1}(\mathbf{x})$. For those two schemes, the dependency in b thence only comes from the first component $\mathbf{v}_{1,2}$. In the case of LW*, the situation is simple according to equation 4.8: the bitsize increases with b, which pleads for small base b. Conversely, the bitsize of LW signatures essentially benefits from large bases b. The same holds true for MP signatures. In the latter cases, the optimal base therefore seems to be $b = \lceil \sqrt{q} \rceil$ if we consider this sole metric. Finally, the situation of MP* signatures is bit more complex as it seems to be better for smaller bases up to a certain inflexion point.

We must now evaluate the impact of the base b on the underlying computational assumptions. The security proof is mostly driven by the simulatability of preimages (e.g., Lemma 4.3 and Theorem 4.1), and the final M-SIS assumption with $\beta \geq \|\mathbf{v} - \mathbf{v}^*\|_2$ for two preimages $\mathbf{v}, \mathbf{v}^*$. Lemma 1.21 (with $c = 1$) then yields $\beta_{\mathsf{MP}} = 2s\sqrt{nd(2+k)}$, $\beta_{\mathsf{MP}^\star} = 2\sqrt{nd(2s_1^2 + ks_2^2)}$, $\beta_{\mathsf{LW}} = 2\sqrt{ndk(s^2 + (b-1)^2)}$, and $\beta_{\mathsf{LW}^\star} = 2\sqrt{nd(2s^2 + k(b-1)^2)}$.

For MP signatures, the bound β_{MP} is dominated by the bottom part $\mathbf{v}_2$ as $k \geq 2$. It thus makes sense to increase b in order to reduce the dimension of dk and thus have balanced contributions of $\mathbf{v}_1$ and $\mathbf{v}_2$ to the M-SIS bound. On the contrary, for LW* signatures, $\mathbf{v}_1$ and $\mathbf{v}_2$ have essentially the same dimension but the specificity of this sampler leads to a strong asymmetry between them. This re-balances the contributions of $\mathbf{v}_1$ and $\mathbf{v}_2$ in the bound $\beta_{\mathsf{LW}^\star}$ which is actually already dominated by the former for $b = 2$. In this case, increasing b will only enlarge the gap between the contributions of $\mathbf{v}_1$ and $\mathbf{v}_2$ to the M-SIS bound and thus decrease the security. In parallel, using too large bases such as $b = \sqrt{q}$ impacts the M-SIS bound too drastically, as noted in e.g. [CGM19], and parameters need to be increased to compensate the security accordingly. In particular, one has to ensure that the infinity norm of the M-SIS solution is smaller than q to avoid trivial solutions.

4.5.2 Comparing Samplers

This intricate situation is reflected by the estimated performance of a GPV signature that we describe below, for different samplers and parameter constraints. We aim to achieve $\lambda = 128$ bits of security for the GPV signature using the security assessment methodology described in

[4] For ease of exposition, we simplify the formulas in this paragraph but we stress that the final estimates in Table 4.1 are computed with the exact parameter settings.

Chapter 9. For all the estimates, we fix the maximal number of emitted signatures per key to $Q = 2^{40}$. When relevant, the repetition rate is chosen to be $M \approx 11$ which leads to $\gamma_3 \approx 8.13$ for $\varepsilon = 1/4Q$. We then find the appropriate rank d and modulus q to achieve the security target while minimizing the signature size.

To highlight the importance of the gadget base, we give the performance of MP, MP*, LW, and LW* signatures for several choices of bases. The estimates are given in Table 4.1. The value of $\lambda^\star$ corresponds to the reached classical security for the signature scheme. When the base is said to be $q^{1/k}$, we actually consider $b = \lceil q^{1/k} \rceil$ to have an integer base for which the gadget dimension is dk. The rows with the value of b giving the smallest size (for $n = 256$) are highlighted in the tables. The goal of Tables 4.1 is to highlight the role of b according to each sampler. Different trade-offs in the parameter selection (e.g., changing n) are likely to be possible but we believe they will not change the overall trend.

Base	$\lambda^\star$	q	d	s		$\lvert\mathbf{v}_{1,2}\rvert$	$\lvert\mathbf{v}_2\rvert$	$\lvert\mathsf{sig}\rvert$
				s_1	s_2			
MP Signatures								
$b = 2$	162	$\approx 2^{15.2}$	5	1546		1.73	26.00	27.73
$b = 4$	164	$\approx 2^{15.6}$	5	2285		1.82	14.57	16.39
$b = q^{1/5}$	158	$\approx 2^{16.8}$	5	5264		2.01	10.05	12.06
$b = q^{1/3}$	140	$\approx 2^{19.7}$	5	39007		**2.46**	**7.38**	**9.84**
$b = q^{1/2}$	133	$\approx 2^{26.7}$	6	4212532		4.22	8.44	12.66
MP* Signatures								
$b = 2$	136	$\approx 2^{15.4}$	4	2951	18	1.50	8.70	10.20
$b = 4$	131	$\approx 2^{15.8}$	4	4006	29	**1.56**	**5.35**	**6.91**
$b = q^{1/5}$	163	$\approx 2^{17.2}$	5	10034	74	2.16	5.24	7.40
$b = q^{1/3}$	143	$\approx 2^{20.2}$	5	83932	712	2.63	4.68	7.31
$b = q^{1/2}$	155	$\approx 2^{27.1}$	7	9978807	79722	5.20	7.34	12.54
LW Signatures								
$b = 2$	131	$\approx 2^{23.6}$	6	572109		80.96	4.50	86.46
$b = 4$	130	$\approx 2^{23.8}$	6	901768		41.83	4.50	46.33
$b = q^{1/5}$	130	$\approx 2^{27.3}$	6	5586865		17.19	5.25	22.44
$b = q^{1/3}$	133	$\approx 2^{30.6}$	7	105308864		**11.88**	**6.78**	**18.66**
$b = q^{1/2}$	138	$\approx 2^{40.5}$	9	96061795597		10.40	11.53	21.93
LW* Signatures								
$b = 2$	128	$\approx 2^{22.5}$	5	305614		**2.93**	**3.59**	**6.52**
$b = 4$	151	$\approx 2^{23.2}$	6	651558		3.72	4.5	8.22
$b = q^{1/5}$	134	$\approx 2^{25.6}$	6	3599595		4.18	4.87	9.05
$b = q^{1/3}$	137	$\approx 2^{30.3}$	7	90707115		5.89	6.78	12.67
$b = q^{1/2}$	138	$\approx 2^{40.3}$	9	90722771912		10.38	11.53	21.91

Table 4.1: Parameter and size estimates of MP, MP*, LW, and LW* signatures using different bases b. The sizes are expressed in KB. The ring degree is $n = 256$. The Gaussian parameter is always s except for the MP* which features two distinct widths s_1 and s_2.
We note that for LW we extrapolated the result of [LW15] which is only presented for $b = 2$. In particular, the parameters we give for $b = q^{1/3}$ and $b = q^{1/2}$ do not perfectly meet the regularity condition from their paper, namely $ndk \log_2 s > 3nd \log_2 q + 4\lambda$. For low values of k, one would need to increase s but it would also lead to increasing q to compensate the security loss.

These estimates show that the choice of the base is far from anecdotal, with a 3-4 ratio for the signature size between the best option and the worst one. They also show that there is no generic choice as $b = 2$ is optimal in our case (LW*) whereas it corresponds to the worst case for

MP, and LW. Additionally, because the size $\mathbf{v}_2$ in MP* signatures is much smaller than for regular MP signatures, we observe that the optimal base is also much lower. When plugged into other signature designs [DM14, BFRS18, dPLS18, BEP+21, LNPS21, LNP22], the conclusions may differ as the relative contributions of $\mathbf{v}_1$ and $\mathbf{v}_2$ to the M-SIS bound may evolve compared to the case of GPV signature.

Besides this sole consideration of optimal base, these tables clearly show the benefits of the LW* sampler as it yields signatures that are about 34% (resp. 6%, 65%) smaller than those produced with the MP sampler (resp. MP*, LW). This is of course to be balanced with the fact that LW* only deals with uniform syndromes. When comparing worst-case samplers, the MP* sampler outperforms both the MP and LW samplers by 30% and 63%. It thus shows that breaking the symmetry of preimages can have practical implications. In particular, one can indeed leverage rejection sampling to improve gadget-based sampling, which solves the apparent paradox of the original LW sampler.

4.6 Approximate Rejection Sampler

In Sections 4.3 and 4.4, we revisited the original preimage sampler from [MP12], showing that we could outperform it by breaking the symmetry of preimages. However, when plugged into the GPV framework, one still ends up with signature sizes that are much larger than the state-of-the-art.

Fortunately, a study initiated by Chen et al. [CGM19] improves the performance of gadget-based constructions through the notion of approximate trapdoors. The idea is to drop the low-order gadget entries and only consider a partial gadget $\mathbf{G}_H = \mathbf{I}_d \otimes [b^\ell | \dots | b^{k-1}]$, which reduces the signature dimension and the number of columns in the trapdoor $\mathbf{R}$ from dk to $d(k-\ell)$. Obviously, this introduces an error in the preimage which depends on ℓ. This error must be taken into account in the security assessment. Intuitively, the more entries are dropped, the larger the error, and in turn the less secure it gets. Reducing the error thus leads to better security, or enables to drop more entries to gain on the key and signature sizes.

The preimage error also depends on the specificities of the sampler. In [CGM19], which is based on the MP sampler, the authors generate normally $\mathbf{z} \sim \mathcal{D}_{\mathcal{L}_q^{\mathbf{x}}(\mathbf{G}), s_{\mathbf{G}}}$ for the full gadget matrix $\mathbf{G}$ and some appropriate vector $\mathbf{x}$ and then drop the component $\mathbf{z}_L$ of $\mathbf{z}$ corresponding to $\mathbf{G}_L = \mathbf{I}_d \otimes [1 | \dots | b^{\ell-1}]$. This leads to a Gaussian error $\mathbf{e} = \mathbf{G}_L \mathbf{z}_L$ whose infinity norm is likely to be larger than $b^\ell - 1$, which does not seem optimal. We note that the same phenomenon occurs for the MP* sampler, for which the approximate trapdoor setting was treated in [JHT22][5].

A recent work by Yu et al. [YJW23] pursued this study with the goal of lowering the error to achieve more compact gadget constructions, and thus more efficient hash-and-sign signature schemes. They propose a brand new gadget accompanied with a new sampler, called *semi-random* sampler, which leads to a square gadget instead of short and fat. This allows them to drastically compact the signature size while still keeping enough security due to a smaller preimage error.

The security analysis of these approximate samplers is only done when simulating preimages for uniform targets. It remains open to obtain a worst-case analysis in this framework. Because the LW sampler was discarded for its lack of efficiency, it was not yet adapted to the approximate trapdoor setting. We show that with only minor modifications to our analysis of Section 4.4.2, we can fit the approximate trapdoor setting. The resulting sampler founds the construction of our signature in Chapter 5, which outperforms that of [CGM19] but also closely match (and sometimes outperforms) the one from [YJW23]. We defer this discussion to Chapter 5 and now present what we call the *approximate rejection sampler*.

4.6.1 Approximate Preimage Sampling from General Distribution

The main limitation of [CGM19] is the size of the preimage error. Dropping too many entries significantly impacts security, thus thwarting the full benefits this approach could achieve. In the case of the LW sampler, $\mathbf{z}$ is exactly $\mathbf{G}^{-1}(\mathbf{w})$ for some syndrome $\mathbf{w}$ instead of being Gaussian. Put differently, $\mathbf{z}$ is simply the signed base-b decomposition of $\mathbf{w}$. Applying the approximate trapdoor approach in our case then essentially consists in discarding the lower-order entries $\mathbf{z}_L$ of this decomposition, which leads to an error $\mathbf{e} = \mathbf{G}_L \mathbf{z}_L$, with $\|\mathbf{e}\|_\infty < b^\ell$. Actually, we show afterwards that this error is (almost) uniform over a subset of $S_{b^\ell - 1}$, which also improves the bound on $\|\mathbf{e}\|_2$. This smaller error, having a similar behaviour than the one in [YJW23], allows for dropping more entries than in [CGM19], leading to better performance. In our scheme in

[5] We do not elaborate on the approximate elliptic sampler in this thesis because our only use of the elliptic sampler requires a worst-case security analysis, which is so far unknown for approximate samplers.

Chapter 5, we can in particular drop $\ell = k-1$ entries, yielding a gadget of length 1 as in [YJW23]. The formal description of our approximate sampler is provided in Algorithm 4.7. We present it for $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$ and $\mathbf{A}' \in R_q^{d\times d}$, i.e., with $m_1 = 2d$.

Algorithm 4.7: AppRejSampler$(\mathbf{R}; \mathbf{A}', \mathbf{u}, \mathscr{D}_s, \mathscr{D}_t)$

Input: Trapdoor $\mathbf{R} \in R^{2d\times d(k-\ell)}$, Matrix $\mathbf{A}' \in R_q^{d\times d}$, Syndrome $\mathbf{u} \in R_q^d$, Source and target distributions $\mathscr{D}_s$ and $\mathscr{D}_t$ over R^{2d} such that rejection sampling can be performed.

1. $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$.
2. $\mathbf{w} \leftarrow \mathbf{u} - [\mathbf{I}_d|\mathbf{A}']\mathbf{p}_1 \bmod qR$. ▷ Syndrome correction
3. $\mathbf{z} \leftarrow \mathbf{G}^{-1}(\mathbf{w}) \in S_{b-1}^{dk}$. ▷ Deterministic.
4. Parse $\mathbf{z}$ into $\mathbf{z}_L \in S_{b-1}^{d\ell}$ and $\mathbf{z}_H \in S_{b-1}^{d(k-\ell)}$ so that $\mathbf{Gz} = \mathbf{G}_L\mathbf{z}_L + \mathbf{G}_H\mathbf{z}_H$.
5. $\mathbf{v}_1' \leftarrow \mathbf{p}_1 + \mathbf{Rz}_H$.
6. $u \hookleftarrow U([0,1])$ ▷ Continuous
7. **if** $u > \min(1, \mathscr{D}_t(\mathbf{v}_1')/(M\mathscr{D}_s(\mathbf{p}_1)))$, go back to 1.
8. **else** $\mathbf{v}_1 \leftarrow \mathbf{v}_1' + \begin{bmatrix}\mathbf{G}_L\mathbf{z}_L \\ \mathbf{0}\end{bmatrix}$
9. $\mathbf{v}_2 \leftarrow \mathbf{z}_H$

Output: $\mathbf{v} = \begin{bmatrix}\mathbf{v}_1 \\ \mathbf{v}_2\end{bmatrix}$.

We also need to adapt the Theorem 4.1 on the simulatability of preimages. The proof is very similar to that of the exact version of the sampler but requires a careful treatment of the error $\mathbf{G}_L\mathbf{z}_L$. We provide the result in Theorem 4.2. Note that setting $\ell = 0$ in Algorithm 4.7 and Theorem 4.2 gives exactly Algorithm 4.6 and Theorem 4.1. We slightly abuse notations and denote by $\mathbf{G}_H^{-1}$ (resp. $\mathbf{G}_L^{-1}$) the map that from $\mathbf{w}$ computes $\mathbf{z} = \mathbf{G}^{-1}(\mathbf{w})$ and outputs the vector $\mathbf{z}_H$ (resp. $\mathbf{z}_L$) defined above. We nevertheless recall that $\mathbf{G}_L\mathbf{G}_L^{-1}(\mathbf{w}) = \mathbf{w}$ only holds for some vectors $\mathbf{w}$ and not in general. We also note that $\mathbf{G}_H^{-1}(R_q^d) \subset S_{b-1}^{d(k-\ell)}$ but equality does not hold simply by a counting argument.

Theorem 4.2 (Simulatability of Approximate Rejection Sampler)

Let R be the ring of integers of a number field. Let d, q, b be positive integers with $b \geq 2$. We define $k = \lceil \log_b(\lceil (q-1)/2 \rceil + 1) \rceil$ and let $\ell \in [\![0, k[\![$. Let $\mathscr{D}_r, \mathscr{D}_s, \mathscr{D}_t$ be three distributions over $R^{2d\times d(k-\ell)}$, R^{2d} and R^{2d} respectively. Let $\mathbf{A}' \in R_q^{d\times d}$, $\mathbf{R} \sim \mathscr{D}_r$ and $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}'] \in R_q^{d\times 2d}$. Then, let $Y \subseteq R^{2d}$ be the support of the distribution of $\mathbf{R}\cdot\mathbf{G}_H^{-1}(U(R_q^d))$. Let $M > 1, \varepsilon \in [0, 1/2]$ such that $\max_{\mathbf{Rz}_H \in Y} \mathrm{RD}_\infty^\varepsilon(\mathscr{D}_t \| \mathscr{D}_s^{+\mathbf{Rz}_H}) \leq M$. We also define the error distribution $\mathscr{D}_e = \mathbf{G}_L\mathbf{G}_L^{-1}(U(R_q^d))$ over $S_{b^\ell-1}^d$. We then define two distributions

$\mathcal{P}_1$: $\mathbf{u} \hookleftarrow U(R_q^d)$, and $\mathbf{v} \leftarrow \mathsf{AppRejSampler}(\mathbf{R}; \mathbf{A}, \mathbf{u}, \mathscr{D}_s, \mathscr{D}_t)$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{P}_2$:
1. $\mathbf{v}_1' \hookleftarrow \mathscr{D}_t$, $\mathbf{v}_2 \hookleftarrow \mathbf{G}_H^{-1}(U(R_q^d))$, $\mathbf{e} \hookleftarrow \mathscr{D}_e$.
2. $\mathbf{v} \leftarrow [\mathbf{v}_1'^T + [\mathbf{e}^T|\mathbf{0}]|\mathbf{v}_2^T]^T$.
3. $\mathbf{u} \leftarrow [\mathbf{A}|\mathbf{G}_H - \mathbf{AR}]\mathbf{v} \bmod qR$.
4. With probability $1 - 1/M$ go back to 1.

Output: $(\mathbf{v}, \mathbf{u})$.

Then, $\Delta(\mathcal{P}_1, \mathcal{P}_2) \leq \varepsilon$ and for all $\alpha \in (1, +\infty]$, $\mathrm{RD}_\alpha(\mathcal{P}_1 \| \mathcal{P}_2) \leq 1/(1-\varepsilon)^{\alpha/(\alpha-1)}$.

Proof (Theorem 4.2). We define the following hybrid distributions from $\mathcal{H}_1$ to $\mathcal{H}_6$, where $\mathcal{H}_1 = \mathcal{P}_1$ and $\mathcal{H}_6 = \mathcal{P}_2$.

$\mathcal{H}_1$: $\mathbf{u} \hookleftarrow U(R_q^d)$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{w} \leftarrow \mathbf{u} - \mathbf{Ap}_1 \bmod qR$, $\mathbf{z} \leftarrow \mathbf{G}^{-1}(\mathbf{w})$, $\mathbf{v}_1' \leftarrow \mathbf{p}_1 + \mathbf{Rz}_H$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1')/(M\cdot\mathscr{D}_s(\mathbf{p}_1)))$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1'^T + [(\mathbf{G}_L\mathbf{z}_L)^T|\mathbf{0}]|\mathbf{z}_H^T]^T$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_2$ $\mathbf{w} \hookleftarrow U(R_q^d)$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{u} \leftarrow \mathbf{w} + \mathbf{A}\mathbf{p}_1 \bmod qR$, $\mathbf{z} \leftarrow \mathbf{G}^{-1}(\mathbf{w})$, $\mathbf{v}_1' \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{z}_H$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1')/(M \cdot \mathscr{D}_s(\mathbf{p}_1)))$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1'^T + [(\mathbf{G}_L\mathbf{z}_L)^T|\mathbf{0}]|\mathbf{z}_H^T]^T$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_3$ $\mathbf{z} \hookleftarrow \mathbf{G}^{-1}(U(R_q^d))$, $\mathbf{w} \leftarrow \mathbf{G}\mathbf{z} \bmod qR$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{u} \leftarrow \mathbf{w} + \mathbf{A}\mathbf{p}_1 \bmod qR$, $\mathbf{v}_1' \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{z}_H$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1')/(M \cdot \mathscr{D}_s(\mathbf{p}_1)))$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1'^T + [(\mathbf{G}_L\mathbf{z}_L)^T|\mathbf{0}]|\mathbf{z}_H^T]^T$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_4$ $\mathbf{e} \hookleftarrow \mathscr{D}_e$, $\mathbf{z}_H \leftarrow \mathbf{G}_H^{-1}(U(R_q^d))$, $\mathbf{w} \leftarrow \mathbf{e} + \mathbf{G}_H\mathbf{z}_H \bmod qR$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{u} \leftarrow \mathbf{w} + \mathbf{A}\mathbf{p}_1 \bmod qR$, $\mathbf{v}_1' \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{z}_H$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1')/(M \cdot \mathscr{D}_s(\mathbf{p}_1)))$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1'^T + [\mathbf{e}^T|\mathbf{0}]|\mathbf{z}_H^T]^T$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_5$ $\mathbf{e} \hookleftarrow \mathscr{D}_e$, $\mathbf{z}_H \leftarrow \mathbf{G}_H^{-1}(U(R_q^d))$, $\mathbf{p}_1 \hookleftarrow \mathscr{D}_s$, $\mathbf{v}_1' \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{z}_H$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > \min(1, \mathscr{D}_t(\mathbf{v}_1')/(M \cdot \mathscr{D}_s(\mathbf{p}_1)))$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1'^T + [\mathbf{e}^T|\mathbf{0}]|\mathbf{z}_H^T]^T$, and $\mathbf{u} \leftarrow [\mathbf{A}|\mathbf{G}_H - \mathbf{A}\mathbf{R}]\mathbf{v} \bmod qR$.
Output: $(\mathbf{v}, \mathbf{u})$.

$\mathcal{H}_6$ $\mathbf{e} \hookleftarrow \mathscr{D}_e$, $\mathbf{z}_H \leftarrow \mathbf{G}_H^{-1}(U(R_q^d))$, $\mathbf{v}_1' \hookleftarrow \mathscr{D}_t$. Then, sample $u \hookleftarrow U([0,1])$ and restart if $u > 1 - 1/M$. Otherwise define $\mathbf{v} \leftarrow [\mathbf{v}_1'^T + [\mathbf{e}^T|\mathbf{0}]|\mathbf{z}_H^T]^T$, and $\mathbf{u} \leftarrow [\mathbf{A}|\mathbf{G}_H - \mathbf{A}\mathbf{R}]\mathbf{v} \bmod qR$.
Output: $(\mathbf{v}, \mathbf{u})$.

Let us now show that these distributions are statistically close to each other.

$\mathcal{H}_1$ - $\mathcal{H}_2$: Here we just change the sampling order of $\mathbf{u}$ and $\mathbf{w}$. In $\mathcal{H}_2$ the vector $\mathbf{w}$ is uniform and independent of $\mathbf{A}\mathbf{p}_1$ implying that $\mathbf{u}$ is also uniform, as in $\mathcal{H}_1$. Hence $\mathcal{H}_1$ and $\mathcal{H}_2$ are identically distributed.

$\mathcal{H}_2$ - $\mathcal{H}_3$: We now change the way $\mathbf{w}$ is generated. Notice that, for correctness, once $\mathbf{w}$ is fixed then so is $\mathbf{z}$ and vice-versa. In $\mathcal{H}_2$, $\mathbf{w}$ is uniform over R_q^d which means that $\mathbf{z}$ follows exactly $\mathbf{G}^{-1}(U(R_q^d))$ as in $\mathcal{H}_3$. Also, $\mathbf{w}$ is coherently set in $\mathcal{H}_3$. Thence, $\mathcal{H}_2$ and $\mathcal{H}_3$ are identically distributed as well.

$\mathcal{H}_3$ - $\mathcal{H}_4$: $\mathcal{H}_4$ simply separates the sampling of low-order and high-order parts compared to $\mathcal{H}_3$. When $\mathbf{z}$ is drawn from $\mathbf{G}^{-1}(U(R_q^d))$, the corresponding $\mathbf{z}_L$ and $\mathbf{z}_H$ are independent. So $\mathbf{z}_H \sim \mathbf{G}_H^{-1}(U(R_q^d))$ and $\mathbf{z}_L \sim \mathbf{G}_L^{-1}(U(R_q^d))$. As such, $\mathbf{z}_H$ is identically distributed in $\mathcal{H}_4$ as in $\mathcal{H}_4$ by definition of $\mathbf{G}_H^{-1}$ which samples a whole vector and drops the low-order entries. Since $\mathbf{z}_L$ is not directly used but only as $\mathbf{e} = \mathbf{G}_L\mathbf{z}_L$, and because $\mathbf{z}_L \sim \mathbf{G}_L^{-1}(U(R_q^d))$ in both $\mathcal{H}_3$ and $\mathcal{H}_4$, it holds that $\mathbf{e} \sim \mathbf{G}_L\mathbf{G}_L^{-1}(U(R_q^d)) = \mathscr{D}_e$ in both hybrids. The way $\mathbf{z}_L$ is sampled, recomposing the low-order entries gives $\mathbf{e} \in S_\gamma^d$ where $\gamma = \sum_{i=0}^{\ell-1}(b-1)b^i = b^\ell - 1$, as desired. This shows that $\mathcal{H}_3$ and $\mathcal{H}_4$ are identically distributed.

$\mathcal{H}_4$ - $\mathcal{H}_5$: $\mathcal{H}_5$ is merely a re-writing of $\mathcal{H}_4$. Indeed, in $\mathcal{H}_4$, $\mathbf{w}$ only acts as an intermediate vector to define $\mathbf{u}$. Defining $\mathbf{R}' = [\mathbf{R}^T|\mathbf{I}_{d(k-\ell)}]^T$, we have $[\mathbf{A}|\mathbf{G}_H - \mathbf{A}\mathbf{R}]\mathbf{R}' = \mathbf{G}_H \bmod qR$. In $\mathcal{H}_3$, this yields

$$\begin{aligned}\mathbf{u} = \mathbf{G}_H\mathbf{z}_H + \mathbf{e} + \mathbf{A}\mathbf{p}_1 \bmod qR &= [\mathbf{A}|\mathbf{G}_H - \mathbf{A}\mathbf{R}]\mathbf{R}'\mathbf{z}_H + \mathbf{A}\mathbf{p}_1 + \mathbf{e} \bmod qR \\ &= [\mathbf{A}|\mathbf{G}_H - \mathbf{A}\mathbf{R}]\mathbf{v} \bmod qR,\end{aligned}$$

as $\mathbf{v} = [\mathbf{p}_1^T + [\mathbf{e}^T|\mathbf{0}]|\mathbf{0}]^T + \mathbf{R}'\mathbf{z}_H$. Again, $\mathcal{H}_4$ and $\mathcal{H}_5$ are identical.

$\mathcal{H}_5$ - $\mathcal{H}_6$: We now change the way $\mathbf{v}_1'$ is generated by using the rejection sampling result. In $\mathcal{H}_5$, $\mathbf{R}\mathbf{z}_H$ is distributed according to $\mathbf{R} \cdot \mathbf{G}_H^{-1}(U(R_q^d))$ with support Y as defined in the theorem statement. By our assumptions on Y, $\mathscr{D}_s$, $\mathscr{D}_t$, the rejection sampling result from Lemma 1.24 yields that

$$\Delta((\mathbf{v}_1', \mathbf{z}_H)_{\mathcal{H}_5}, (\mathbf{v}_1', \mathbf{z}_H)_{\mathcal{H}_6}) \leq \varepsilon \quad \text{and} \quad \mathrm{RD}_\alpha((\mathbf{v}_1', \mathbf{z}_H)_{\mathcal{H}_5} \| (\mathbf{v}_1', \mathbf{z}_H)_{\mathcal{H}_6}) \leq \frac{1}{(1-\varepsilon)^{\frac{\alpha}{\alpha-1}}},$$

for all $\alpha > 1$. By the data processing inequality of the statistical distance and Rényi divergence,

it holds

$$\Delta(\mathcal{H}_5, \mathcal{H}_6) \leq \varepsilon \quad \text{and} \quad \mathrm{RD}_\alpha(\mathcal{H}_5 \| \mathcal{H}_6) \leq \frac{1}{(1-\varepsilon)^{\frac{\alpha}{\alpha-1}}}.$$

Since $\mathcal{H}_1 = \mathcal{P}_1$ and $\mathcal{H}_6 = \mathcal{P}_2$, combining the above gives the result.

The study carried in Section 4.5 leads to the same conclusions for the approximate samplers, although the analysis is slightly more complex as one can optimize over the number of dropped entries ℓ as well. Because the sampling error $\mathbf{e}$ is smaller in our case, we can drop more entries and thus increase the performance gap between the approximate MP sampler and ours.

4.7 Conclusion

The gadget-based trapdoors introduced by Micciancio and Peikert [MP12] provide a easy way of generating trapdoor functions and preimages. However, most uses of the preimage sampling procedure were limited to spherical Gaussian distributions which we show is not optimal. In this chapter, we proposed a more practical and in-depth analysis of the elliptic Gaussian sampler from [JHT22], providing it with the detailed worst-case simulatability result needed by most advanced signature constructions such as the one from Part III.

Additionally, after being introduced by Lyubashevsky and Wichs in 2015 [LW15], the rejection sampler, which combines ideas from different lattice techniques such as rejection sampling and MP trapdoors, seemed discarded because of the limitations of its worst-case analysis. We showed that one can considerably relieve the requirements placed on it when used in a context where the inverted syndromes are uniform. In the latter case, it not only results in much better parameters (compared to [LW15]) but also in an interesting middle way between rejection sampling and trapdoor sampling. It indeed borrows the nice features of both approaches and can thus be seen as an interesting alternative that has been overlooked so far. For example, the sampler combines well with the notion of approximate trapdoors, leading to the more efficient approximate rejection sampler. Plugging the latter into the design of a signature scheme yields more attractive sizes already, but we push the performance further in Chapter 5 by providing a new hash-and-sign signature.

5

Phoenix: Hash-and-Sign with Aborts from Lattice Gadgets

This chapter introduces a new signature scheme called Phoenix. It leverages the approximate rejection sampler presented in Chapter 4, and follows the Hash-and-Sign signature rationale. Being also reliant on rejection sampling strategies which are heavily used in Fiat-Shamir with Aborts designs, Phoenix actually bridges both signature paradigms. The scheme is presented in two variants using different signature distributions.

The work presented in this chapter is based on a paper with my co-authors Adeline Roux-Langlois and Olivier Sanders.

[JRS24] **Phoenix: Hash-And-Sign with Aborts from Lattice Gadgets**. Published at PQCrypto 2024.

Contents

5.1 Introduction

Lattice-based cryptography has proven to be a relatively stable and extensively studied candidate to provide post-quantum secure primitives, and has now shifted towards proposing concretely efficient constructions. The NIST standardization [NISa] perfectly reflects this trend as it recently released the first round of standards, which is dominated by lattice schemes [BDK+18, DKL+18, PFH+20], and are moving to practical deployment discussions. Although they provide a first set of solutions for initiating the post-quantum transition, NIST recently called for additional digital signatures [NISb]. The lattice-based candidates to this new competition, along with some recent

publications, e.g., [YJW23, DPS23], show that there is still room for improvement in this area in terms of optimizing bandwidth, ease of implementation, side-channel protection, etc.

If we set aside schemes designed with very specific applications in mind, e.g., the schemes of Part III, lattice-based signature schemes usually follow one of two main paradigms. The first one, called the *hash-and-sign* paradigm, was instantiated by Gentry et al. [GPV08] (GPV) with lattice preimage sampleable trapdoor functions as mentioned in Chapter 4. In such schemes, the signing key consists of a trapdoor for a publicly computable function which allows one to efficiently find short preimages. Signatures are then preimages of seemingly random (and possibly message-dependent) syndromes. Only the signer is able to compute such preimages, but everyone is able to compute the image to ensure they represent valid signatures. Several schemes rely on variants of the above, e.g., [GPV08, MP12, DM14, DLP14], and were successfully pushed towards concrete practicality [PFH+20, EFG+22, YJW23] using an additional assumption. In their general use, trapdoor preimage samplers can however be quite computationally intensive, and most efficient solutions were designed to only support Gaussian-distributed preimages prior to our work of Chapter 4.

An alternative, called the *Fiat-Shamir with Aborts* (FSwA) paradigm, was proposed by Lyubashevsky [Lyu12], building signatures on Schnorr-like proofs made non-interactive with the Fiat-Shamir transform. This framework avoids the use of trapdoors, and uses rejection sampling to control the distribution of signatures while making them independent of the signing key. Even though most applications yield Gaussian-distributed signatures, it is possible to tweak the rejection sampling step to get other distributions that can be more suitable depending on the context. Efficient instantiations of this signature paradigm were proposed, such as qTESLA [ABB+20] and Dilithium [DKL+18].

Reexamining the cleavage between these two signature paradigms may lead to new features in the design of lattice signatures, while hopefully remaining efficient. In particular, in light of the analysis of gadget-based samplers in Chapter 4, it seems that one could benefit from the perks of each one by combining efficient trapdoor generation and efficient preimage sampling based on rejection. The approximate rejection sampler of Section 4.6 seems the natural candidate to obtain rather compact lattice signatures.

5.1.1 Our Contributions

Plugging the approximate rejection sampler in the GPV framework naturally leads to a new hash-and-sign signature scheme, which we call Phoenix. It allows us to assess the benefits of the LW sampler for concrete applications.

One of the most surprising features of Phoenix is arguably its relatively small signatures sizes $|\mathsf{sig}|$ which even outperform those of the future NIST standard Dilithium [DKL+18] and of the very recent gadget-based scheme Eagle [YJW23]. Given the initial performance of the LW sampler, this was clearly unexpected. So far we have only looked at optimizing the size of the preimage through the approximate rejection sampler in Chapter 4, which translates into the signature size. Although dropping gadget entries allows for keys with smaller dimensions which are then more compact size-wise, the public key size remains a bit large. With this straightforward plug-and-play the public key of Phoenix is about twice as big as that of the state-of-the-art M-LWE-based signatures.

Fortunately, we observe that one of the specificities of the approximate rejection sampler is that it produces extremely short $\mathbf{v}_2$. When trying to drop low-order bits of the public key $\mathbf{B} = \mathbf{AR} \bmod qR$, it entails an error $\mathbf{B}_L\mathbf{v}_2$ on the preimage, where $\mathbf{B}_L$ corresponds to the low-order bits of $\mathbf{B}$. From the prior observation, $\mathbf{B}_L\mathbf{v}_2$ actually turns out to be rather small compared to the preimage without compression, meaning that it incurs almost no security loss (depending on the number of dropped bits). This method is done for example in Dilithium [DKL+18] to also reduce the size of the public key. We note however that our situation is much more favorable. Indeed, the verification of Dilithium signatures (and FSwA signatures in general) requires knowledge of (some function of) the full public key in an exact way to be fed into a hash function. Dropping entries of the public key thus requires careful adjustments, which is done by providing *hints* on the carries it incurs, to maintain a correct signature scheme. These hints if not handled correcly could leak private information, which is why they have to perform another rejection sampling step (the one on $\mathbf{r}_0$ so that it does not leak information on $\mathbf{s}_2$). In our case, these countermeasures are not necessary as we do not need to hash elements that depend on the low-order bits of the public key. There is therefore no additional hints to account for, but only a very mild key compression error which will feed through to the preimage constituting the signature. We show that we can indeed cut a little more than half of the public key size $|\mathsf{pk}|$ at almost no cost on the security, allowing us to reach smaller public keys than [DKL+18, YJW23].

Finally, altough our scheme follows the hash-and-sign paradigm, it benefits from the versatility of signature distributions of FSwA signatures. Phoenix indeed benefits from the unique features of our approximate rejection sampler of Section 4.6. The latter can be instantiated with a variety of distributions that are more suited for easy and secure implementations. In particular, Phoenix only involves spherical Gaussians over R which removes the need for complex Gaussian samplers as in previous hash-and-sign schemes (FFO sampler for [PFH+20], hybrid sampler for [EFG+22], perturbation samplers similar to Section 4.3.2 for [CGM19, JHT22, YJW23]). This makes Phoenix easier to protect against side-channel attacks. We also provide a version of Phoenix , called $\text{Phoenix}_{\text{U}}$, which uses uniform distributions over hypercubes to avoid floating points altogether, as described in Section 5.4.

We give a detailed comparison with the other M-LWE-based signatures Dilithium [DKL+18], Haetae [CCD+23], Raccoon [dPEK+], Eagle [YJW23] and G+G [DPS23] in Section 5.5 and Table 5.4. For the sake of completeness, we also discuss and compared our signature to NTRU-based ones (Falcon [PFH+20], Mitaka [EFG+22], Solmae [KTW+22], Robin [YJW23]) which are usually more compact. The performances of Phoenix and $\text{Phoenix}_{\text{U}}$ are summarized in Table 5.1.

	NIST-II			NIST-III			NIST-V		
	|sk|	|pk|	|sig|	|sk|	|pk|	|sig|	|sk|	|pk|	|sig|
Phoenix	512	1184	2190	648	1490	2897	972	2219	4468
$\text{Phoenix}_{\text{U}}$	648	1652	3442	768	1952	4072	1024	2592	5416

Table 5.1: Performance in bytes of Phoenix and $\text{Phoenix}_{\text{U}}$ for NIST-II, NIST-III and NIST-V security.

Our scheme thus combines the benefits of Fiat-Shamir with Aborts schemes and of hash-and-sign schemes, as was originally expected from the LW sampler. This work shows that said sampler is not only of theoretical interest but may have concrete applications that could benefit from its nice performance and implementation features.

5.2 Reminder: The GPV Hash-And-Sign Framework

We start by giving a short reminder of the GPV hash-and-sign paradigm [GPV08] over lattices. As mentioned in Section 4.1, this framework was proposed in response to the security issues in previous lattice signature designs [GGH97, HHP+03]. It gave a rather abstract notion of preimage sampleable trapdoor functions which, given their property, could be used to design hash-and-sign signatures[1]. We only describe the lattice versions.

These schemes are usually defined over q-ary lattices $\mathcal{L}_q^\perp(\overline{\mathbf{A}})$, where the dimensions of $\overline{\mathbf{A}}$ and the space it belongs to differ from one construction to the next. The secret signing key consists in a trapdoor on $\overline{\mathbf{A}}$, which is a short basis of $\mathcal{L}_q^\perp(\overline{\mathbf{A}})$ (or material that can be used to derive one) with good geometric properties. Using the latter, one can efficiently solve CVP_γ over said lattice for a rather small approximation factor γ. It means that given a target $\mathbf{t}$ in $\text{Span}_{\mathbb{R}}(\mathcal{L}_q^\perp(\overline{\mathbf{A}}))$, the knowledge of the secret key allows to find $\mathbf{v} \in \mathcal{L}_q^\perp(\overline{\mathbf{A}})$ such that $\|\mathbf{v}-\mathbf{t}\|_2 \leq \gamma \text{dist}(\mathbf{t}, \mathcal{L}_q^\perp(\overline{\mathbf{A}})) =: \beta$. The preimage sampling procedure associated to GPV-type trapdoor functions allows to find a random solution $\mathbf{v}$ satisfying the above. Prior to our work, signatures based upon this idea [GPV08, MP12, DLP14, PFH+20, EFG+22, YJW23] randomized the solution using a discrete Gaussian distribution, at the exception of [LW15] but which led to rather large signatures as described in Chapter 4.

GPV with Gadgets

More concretely, the signature works as follows. The target $\mathbf{t} \in R_q^d$ is obtained by hashing the message $\mathbf{m}$ to be signed. Then, the signer uses their trapdoor to sample a preimage $\mathbf{v}$ such that $\overline{\mathbf{A}}\mathbf{v} = \mathbf{t} \bmod qR$ and that $\|\mathbf{v}\|_2 \leq \beta$ and outputs $\mathbf{v}$ as the signature. Verification then recomputes $\mathbf{t}$ from the message and checks that $\overline{\mathbf{A}}\mathbf{v} = \mathbf{t} \bmod qR$ and that $\mathbf{v}$ is shorter than β.

A standard optimization is to consider $\overline{\mathbf{A}}$ of the form $\overline{\mathbf{A}} = [\mathbf{I}_d | \overline{\mathbf{A}}']$ so that the signature verification can be split into $\mathbf{v}' + \overline{\mathbf{A}}'\overline{\mathbf{v}}' = \mathbf{t} \bmod qR$, where $\mathbf{v} = [\mathbf{v}'|\overline{\mathbf{v}}']$. In this case, $\mathbf{v}'$ is uniquely

[1] We note that the *hash-and-sign* appelation is not specific to lattices and not due to [GPV08]. Regardless, in the rest of this thesis, we sometimes identify the *hash-and-sign* approach to that of [GPV08].

determined by $\mathbf{t}$ and $\overline{\mathbf{v}}'$ and does not have to be transmitted. One would perform verification by recomputing $\mathbf{t}$, and then $\mathbf{v}' = \mathbf{t} - \overline{\mathbf{A}}'\overline{\mathbf{v}}' \bmod qR$ before checking that $\mathbf{v} = [\mathbf{v}'|\overline{\mathbf{v}}']$ is small.

Recalling the gadget-based trapdoor functions from Chapter 4, $\overline{\mathbf{A}}$ is of the form $[\mathbf{A}|\mathbf{G}-\mathbf{B}]$ mod qR for $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$ and $\mathbf{B} = \mathbf{AR} \bmod qR$. The preimage $[\mathbf{v}_{1,1}|\mathbf{v}_{1,2}|\mathbf{v}_2]$ then constitutes the signature $(\mathbf{v}_{1,2}, \mathbf{v}_2)$ where $\mathbf{v}_{1,1}$ is recovered by $\mathbf{v}_{1,1} = \mathbf{t} - \mathbf{A}'\mathbf{v}_{1,2} - (\mathbf{G} - \mathbf{B})\mathbf{v}_2 \bmod qR$.

Security

The security relies on so-called *simulatability results* like the ones studied in Chapter 4 to argue that these randomized solutions do not leak the trapdoor, nor its geometry. In this framework, the targets are hash outputs and thus uniformly random in the random oracle model. As a result, an average-case simulatability is sufficient to ensure the absence of leakage.

Security also relies on the fact that from the public description of the lattice, i.e., $\overline{\mathbf{A}}$, one cannot easily compute a basis sufficiently short to produce valid solutions $\mathbf{v}$, that is within distance β of some targets $\mathbf{t}$. In the context of gadget-based solutions, this is generally argued under the M-LWE and M-SIS assumption. The most efficient designs (see Table 5.5) are however based on another construction for $\overline{\mathbf{A}}$ based on the NTRU assumption (or iNTRU for [YJW23]) as well as M-SIS.

Notice however that from the generic description of GPV signatures, the security seems to rely on the M-ISIS assumption with matrix $\overline{\mathbf{A}}$ and syndrome $\mathbf{t}$. The complexity of forging a signature is actually assessed through this inhomogeneous assumption as it provides a tighter approximation of the actual security.

5.3 The Phoenix Signature Scheme

Following our goal to design efficient lattice signatures with attractive features, we now introduce Phoenix. It continues our quest of testing the limits of the LW sampler and in particular leverages the approximate rejection sampler from Section 4.6 within the GPV framework recalled in Section 5.2. As mentioned earlier, the gadget base study carried in Section 4.5 gives similar conclusions for the approximate samplers. As a result, we express everything using the optimal base $b = 2$ directly. Also, we choose the modulus to be $q = 2^{k+1} - 1$ so that the representatives of $\mathbb{Z}_q$ are taken in the centered interval $[\![-(q-1)/2, (q-1)/2]\!] = [\![-(2^k - 1), 2^k - 1]\!]$. The resulting gadget dimension is $\lceil \log_2(\lceil (q-1)/2 \rceil + 1) \rceil = k$. We start by describing in Section 5.3.1 the public key compression technique we use in Phoenix. We then account for the compression error directly in the preimage sampler. For that, we instantiate the approximate rejection sampler in its Gaussian version, explaining why we favored this distribution to another, and with the compression error. Then, we give the full description of the scheme in Section 5.3.3, before detailing the security analysis in Section 5.3.4 and give concrete parameter instantiations in Table 5.2.

5.3.1 Adding Public Key Compression

By discarding columns in the gadget matrix, the use of the approximate rejection sampler of Section 4.6 already allows us to reduce the size of $\mathbf{R}$ and thus $\mathbf{B} = [\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR \in R_q^{d \times d(k-\ell)}$ thus reducing the public key size. Despite this significant compression, the public key $\mathbf{B}$ can remain quite large[2] for typical parameters when instantiating the sampler in the GPV hash-and-sign framework. Fortunately, we can use public key compression techniques, like the one used in [DKL+18, DSH21] for example. The unique features of our sampler actually makes these techniques much easier to use in Phoenix, and seem almost optimal. As mentioned in Section 5.1, our advantage over FSwA designs when it comes to compression is that we only need to account for a compression error which very mildly affect the parameters. In particular, it does not require the use of *hints* or additional rejection sampling dedicated to the compression.

Let us now describe how to compress the public key. We let ℓ' be a positive integer in $[\![0, k[\![$. Once the full public key $\mathbf{B}$ has been generated, we interpret it in R as a matrix over $S_{(q-1)/2}$. At this stage notice that due to our choice of q, this essentially means taking the unique centered representative of the equivalence class modulo q. We can then write $\mathbf{B}$ as $\mathbf{B}_L + \mathbf{B}_H$ by separating the low-order and high-order bits, with $\mathbf{B}_L \in S_{2^{\ell'}-1}^{d \times d(k-\ell)}$ and $\mathbf{B}_H \in 2^{\ell'} S_\gamma^{d \times d(k-\ell)}$ for $\gamma = \lfloor 2^{-\ell'} \frac{q-1}{2} \rfloor = 2^{k-\ell'} - 1$. The compression consists in simply using $\mathbf{B}_H$ as the public key (or $2^{-\ell'}\mathbf{B}_H$ equivalently) which can be stored using $nd^2(k-\ell)(1+k-\ell')$ bits, thus saving ℓ' bits per coefficients.

[2] The other public key matrix $\mathbf{A}'$ is stored using a public seed of 256 bits.

The catch in discarding $\mathbf{B}_L$ is that $\mathbf{B}_H$ is only an approximation of the public key. The removed low-order bits then introduce a new error $\mathbf{e}_{\mathsf{pk}}$ which we call *compression error*. Indeed, following the GPV paradigm recalled in Section 5.2, the verification checks that

$$[\mathbf{I}_d|\mathbf{A}'|\mathbf{G}_H - \mathbf{B}_H] \begin{bmatrix} \mathbf{v}_{1,1} \\ \mathbf{v}_{1,2} \\ \mathbf{v}_2 \end{bmatrix} = \mathcal{H}(\mathbf{m}) \bmod qR,$$

but where the preimage is generated using the *full* secret key $\mathbf{R}$. As such, this equation is not verified as it features $\mathbf{B}_H$ instead of $\mathbf{B}$. We need to adjust it as follows.

$$\begin{aligned} \mathcal{H}(\mathbf{m}) &= [\mathbf{I}_d|\mathbf{A}'|\mathbf{G}_H - \mathbf{B}] \begin{bmatrix} \mathbf{v}_{1,1} \\ \mathbf{v}_{1,2} \\ \mathbf{v}_2 \end{bmatrix} \bmod qR, \\ &= \mathbf{v}_{1,1} + \mathbf{A}'\mathbf{v}_{1,2} + (\mathbf{G}_H - (\mathbf{B}_L + \mathbf{B}_H))\mathbf{v}_2 \bmod qR \\ &= (\mathbf{v}_{1,1} - \mathbf{B}_L\mathbf{v}_2) + \mathbf{A}'\mathbf{v}_{1,2} + (\mathbf{G}_H - \mathbf{B}_H)\mathbf{v}_2 \bmod qR, \end{aligned}$$

which can now be verified using only $\mathbf{B}_H$ by including the compression error $\mathbf{e}_{\mathsf{pk}} = \mathbf{B}_L\mathbf{v}_2$ directly into the preimage $\mathbf{v}_{1,1}$. This error can be combined with the approximate sampling error $\mathbf{e}$ during preimage sampling.

The reason why this compression technique is particularly interesting in our situation, as opposed to Eagle [YJW23] for example, is because $\mathbf{v}_2$ is ternary and not Gaussian. As such, the error $\mathbf{e}_{\mathsf{pk}}$ remains moderate compared to $\mathbf{e}$ if ℓ' is carefully chosen with respect to ℓ, as detailed below. Looking ahead, we go up to $\ell' \gtrsim k/2$ at almost no cost on the security.

5.3.2 Approximate Gaussian Rejection Sampler

We now instantiate the approximate rejection sampler of Section 4.6 with the exact distributions we are using in Phoenix. Our main goal is to showcase the concrete potential of the approximate rejection sampler in hash-and-sign signatures and our work only provides a first step which shall foster future improvements. In this thesis, we first focus on the size metrics (signature, keys) while retaining strong security guarantees. In this direction, we follow the results highlighted by Devevey et al. [DFPS22] regarding the compactness of rejection sampling in the unimodal setting: discrete Gaussians entail a rather compact (imperfect) rejection sampling. We note that their work also presents uniform distributions in hyperballs as being as compact as discrete Gaussians in the imperfect unimodal case (and even optimal for exact unimodal sampling). We decide to leave it as a promising future work as it requires a few adjustments, but notice that it leads to an easier rejection sampling step which would be highly relevant for secure implementations.

Algorithm 5.1: AppRejSampler$(\mathbf{R}; \mathbf{A}', \mathbf{u}, s)$

Input: Trapdoor $\mathbf{R} \in R^{2d\times d(k-\ell)}$, Matrix $\mathbf{A}' \in R_q^{d\times d}$, Syndrome $\mathbf{u} \in R_q^d$, Gaussian parameter $s > 0$.

1. $\mathbf{p}_1 \hookleftarrow \mathcal{D}_{R^{2d},s}$. ▷ Base sampler
2. $\mathbf{w} \leftarrow \mathbf{u} - [\mathbf{I}_d|\mathbf{A}']\mathbf{p}_1 \bmod qR$. ▷ Syndrome correction
3. $\mathbf{z} \leftarrow \mathbf{G}^{-1}(\mathbf{w}) \in S_1^{dk}$. ▷ Deterministic
4. Parse $\mathbf{z}$ into $\mathbf{z}_L \in S_1^{d\ell}$ and $\mathbf{z}_H \in S_1^{d(k-\ell)}$ so that $\mathbf{G}\mathbf{z} = \mathbf{G}_L\mathbf{z}_L + \mathbf{G}_H\mathbf{z}_H$.
5. $\mathbf{v}_1' \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{z}_H$.
6. $u \hookleftarrow U([0,1])$ ▷ Continuous
7. **if** $u > \min\left(1, \frac{1}{M}\exp\left(\frac{\pi}{s^2}\left(\|\mathbf{R}\mathbf{z}_H\|_2^2 - 2\langle\tau(\mathbf{v}_1'), \tau(\mathbf{R}\mathbf{z}_H)\rangle\right)\right)\right)$, go back to 1.
8. $\mathbf{e} \leftarrow \mathbf{G}_L\mathbf{z}_L$ ▷ Sampling error
9. $\mathbf{e}_{\mathsf{pk}} \leftarrow (([\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR) - \mathbf{B}_H)\mathbf{z}_H$ ▷ Compression error
10. $\mathbf{v}_1 \leftarrow \mathbf{v}_1' + \begin{bmatrix} \mathbf{e} - \mathbf{e}_{\mathsf{pk}} \\ \mathbf{0} \end{bmatrix}$
11. $\mathbf{v}_2 \leftarrow \mathbf{z}_H$

Output: $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$.

Observe that the compression error can be computed using $\mathbf{R}$ and the public key $\mathbf{B}_H$ by first reconstructing $\mathbf{B} = [\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR$ and then getting $\mathbf{B}_L = \mathbf{B} - \mathbf{B}_H$. One could also just compute the low-order bits from $\mathbf{B}$ without having to substract $\mathbf{B}_H$. Alternatively, $\mathbf{B}_L$ could be stored alongside the secret key to avoid having to recompute it at each sampling procedure. The simulatability of preimages generated by Algorithm 5.1 is formalized in Section 5.3.4.

Remark 5.1 (Bimodal Setting)

As per [DFPS22], Gaussians or uniform distributions in hyperballs yield a fairly compact rejection sampling in the unimodal setting. Better (and optimal) compactness can be achieved in the bimodal case. It would lead to smaller signature sizes but requires updating the modulus and the overall structure of the sampler [DDLL13, CCD+23]. We leave it as part of future optimizations.

5.3.3 Description

We now give the full description of Phoenix which is based on the sampler in Algorithm 5.1, before discussing all the parameters that intervene in the different algorithms. It implicitly work over a cyclotomic ring of integers R of degree n. Later, we only choose cyclotomic whose conductor is 3-smooth.

Algorithm 5.2: Phoenix.Setup

Input: Security parameter λ.

1. Choose positive integers d, k.
2. $q \leftarrow 2^{k+1} - 1$.
3. Choose $\ell, \ell' \in [\![0, k[\![$.
4. $\mathbf{G} = \mathbf{I}_d \otimes [1|\cdots|2^{k-1}] \in R_q^{d \times dk}$.
5. $\mathbf{G}_H = \mathbf{I}_d \otimes [2^\ell|\cdots|2^{k-1}] \in R_q^{d \times d(k-\ell)}$.
6. $\mathbf{G}_L = \mathbf{I}_d \otimes [1|\cdots|2^{\ell-1}] \in R_q^{d \times d\ell}$.
7. $\varepsilon \leftarrow 1/4Q$ ▷ Rejection sampling loss
8. Choose $M > 1$. ▷ Repetition rate
9. $\gamma \leftarrow \frac{\sqrt{\pi}}{\ln M}(\sqrt{\ln \varepsilon^{-1} + \ln M} + \sqrt{\ln \varepsilon^{-1}})$. ▷ Rejection sampling slack
10. $s \leftarrow \gamma\sqrt{nd(k-\ell)}(\sqrt{2nd} + \sqrt{nd(k-\ell)})$. ▷ Gaussian width
11. $\mathbf{A}' \hookleftarrow U(R_q^{d \times d})$.

Output: $\mathsf{pp} = (\mathbf{A}'; \mathbf{G}, \mathbf{G}_L, \mathbf{G}_H; \lambda, n, q, d, k, \ell, s, M)$.

Algorithm 5.3: Phoenix.KeyGen

Input: Public parameters pp as in Algorithm 5.2.

1. $\mathbf{R} \hookleftarrow U(S_1^{2d \times d(k-\ell)})$ such that $\|\mathbf{R}\|_2 \leq \sqrt{2nd} + \sqrt{nd(k-\ell)}$.
2. $\mathbf{B} \leftarrow [\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR \in R_q^{d \times d(k-\ell)}$
3. Parse $\mathbf{B}$ as $\mathbf{B}_L + \mathbf{B}_H$ with $\mathbf{B}_L \in S_{2^{\ell'}-1}^{d \times d(k-\ell)}$ and $\mathbf{B}_H \in 2^{\ell'} S_{2^{k-\ell'}-1}^{d \times d(k-\ell)}$.

Output: $\mathsf{pk} = \mathbf{B}_H$, and $\mathsf{sk} = \mathbf{R}$. ▷ pp stored with pk for simplicity

Algorithm 5.4: Phoenix.Sign

Input: Secret key sk, Message $\mathbf{m} \in \{0,1\}^*$, Public key pk.

1. $\mathsf{salt} \hookleftarrow U(\{0,1\}^{320})$.
2. $\begin{bmatrix} \mathbf{v}_{1,1} \\ \mathbf{v}_{1,2} \\ \mathbf{v}_2 \end{bmatrix} \leftarrow \mathsf{AppRejSampler}(\mathbf{R}; \mathbf{A}', \mathcal{H}(\mathbf{m}, \mathsf{salt}), s)$. ▷ Algorithm 5.1
3. $b_1 \leftarrow (\|\mathbf{v}_{1,1}\|_2 \leq B_{1,1}) \wedge (\|\mathbf{v}_{1,2}\|_2 \leq B_{1,2})$.
4. $b_2 \leftarrow (\|\mathbf{v}_{1,1}\|_\infty \leq B_{1,1}^\infty) \wedge (\|\mathbf{v}_{1,2}\|_\infty \leq B_{1,2}^\infty) \wedge (\|\mathbf{v}_2\|_\infty \leq 1)$.
5. **if** $b_1 \wedge b_2 = 0$, restart.

Output: $\mathsf{sig} = (\mathsf{salt}, \mathbf{v}_{1,2}, \mathbf{v}_2)$.

Algorithm 5.5: Phoenix.Verify

Input: Public key pk, Message $\mathbf{m} \in \{0,1\}^*$, Signature sig.

1. $\mathbf{v}_{1,1} \leftarrow \mathcal{H}(\mathbf{m}, \mathsf{salt}) - \mathbf{A}'\mathbf{v}_{1,2} - (\mathbf{G}_H - \mathbf{B}_H)\mathbf{v}_2 \bmod qR \in R^d$.
2. $b_1 \leftarrow (\|\mathbf{v}_{1,1}\|_2 \leq B_{1,1}) \wedge (\|\mathbf{v}_{1,2}\|_2 \leq B_{1,2})$.
3. $b_2 \leftarrow (\|\mathbf{v}_{1,1}\|_\infty \leq B_{1,1}^\infty) \wedge (\|\mathbf{v}_{1,2}\|_\infty \leq B_{1,2}^\infty) \wedge (\|\mathbf{v}_2\|_\infty \leq 1)$.

Output: $b_1 \wedge b_2$. ▷ 1 if valid, 0 otherwise

Preimage error distribution

Recall that Theorem 4.2 identifies the distribution of $\mathbf{e}$ to be $\mathscr{D}_e = \mathbf{G}_L\mathbf{G}_L^{-1}(U(R_q^d))$ over $S_{2^\ell-1}^d$. Let us define a modified error distribution $\mathscr{D}_e^+$ where we sample $\mathbf{e} \hookleftarrow \mathscr{D}_e$ and output $\mathbf{e}^+$ corresponding

to $\mathbf{e}$ but where the coefficient embeddings of $\mathbf{e}^+$ are the magnitude of that of $\mathbf{e}$. We observe that $\mathscr{D}_e^+$ is almost the uniform distribution over $\tau^{-1}(\{0,\dots,2^\ell-1\}^{nd})$ because of the form of q. Indeed, as $\mathbf{G}_L^{-1}(\cdot)$ gives the signed low-order bit decomposition, recomposing it and taking the magnitude ensures that all entries appear with same probability, except for 0 whose probability is twice as small.

Naturally, the Euclidean norm of $\mathbf{e}$ is distributed the same way as that of $\mathbf{e}^+$. The variance of $U(\llbracket 0, 2^\ell \rrbracket)$ is exactly $(2^{2\ell}-1)/12$, and therefore the norm of $\mathbf{e}$ can be bounded on average by $\sqrt{(2^\ell-1))(2^{\ell+1}-1)/6}\sqrt{nd} \approx 2^\ell\sqrt{nd/3}$ by the central limit theorem. In [CGM19], $\mathbf{e}$ is close to a discrete Gaussian of width $\eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{5(2^{2\ell}-1)/3}$ which is 4 to 12 times larger than ours depending on the smoothing loss ε and Gaussian tailcut parameter. Our error also almost matches that of [YJW23]. In the latter, the sampling error corresponds to the lattice decoding error of a uniform target, which is then uniform over a centered set. The norm of such error can then be bounded by the central limit theorem by $\sqrt{(2^{2\ell}-1)/12}\sqrt{nd} \approx 2^\ell\sqrt{nd/12}$. The factor 2 between ours and theirs essentially comes from the fact that although $\mathscr{D}_e$ gives coefficients in $\llbracket -(2^\ell-1), 2^\ell-1 \rrbracket$, its entropy is about half of that of the uniform distribution over such set (corresponding to the error of [YJW23]) due to the signed decomposition.

Verification bounds

We now explain how the verification bounds are set. First, because $\mathbf{v}_{1,2}$ is statistically close to a discrete Gaussian of parameter s, the bounds on its ℓ_2 and ℓ_∞ norms are simply taken from Lemma 1.21 by adjusting the slack to avoid too many repetitions. As such we set

$$B_{1,2} = 1.048 \cdot s\sqrt{\frac{nd}{2\pi}}, \quad \text{and} \quad B_{1,2}^\infty = \left\lfloor 4.5\frac{s}{\sqrt{2\pi}} \right\rfloor,$$

which gives a probability of 1/10 that each bound is not verified.

Choosing appropriate bounds is more complex for $\mathbf{v}_{1,1}$ because the value recovered by the verifier is $\mathbf{v}'_{1,1} + \mathbf{e} - \mathbf{e}_{\mathsf{pk}}$ which contains the error terms. Bounding each term separately overshoots the actual norm of $\mathbf{v}_{1,1}$. We thus give a more fine-grained analysis based on the following observations. We first notice that the coefficients of $\mathbf{e}_{\mathsf{pk}} = \mathbf{B}_L\mathbf{v}_2$ behave in a similar fashion to the drift of lazy random walks with adaptive steps whose magnitude are at most $2^{\ell'}-1$, up to a slack factor μ depending on the conductor of the cyclotomic field[3]. As such, we can approach the bounds on $\mathbf{v}_{1,1} - \mathbf{e}_{\mathsf{pk}}$ by the Gaussian tail bound with the appropriate variance. Then, $\|\mathbf{e}\|_2$ can be evaluated as described above which also behaves like the Gaussian tail bound, and $\|\mathbf{e}\|_\infty$ is very likely to be close to the worst-case bound $2^\ell-1$. Using these Gaussian approximations, we set

$$B_{1,1} = 1.04\sqrt{\frac{s^2}{2\pi} + \frac{(2^\ell-1)(2^{\ell+1}-1)}{6} + \mu^2\frac{2^{\ell'}(2^{\ell'}-1)}{6}\frac{nd(k-\ell)}{2}}\sqrt{nd}$$
$$B_{1,1}^\infty = \left\lfloor 3.8 \cdot \sqrt{\frac{s^2}{2\pi} + \mu^2\frac{2^{\ell'}(2^{\ell'}-1)}{6}\frac{nd(k-\ell)}{2}} \right\rfloor + (2^\ell-1).$$

which are verified empirically and only entail a small degradation[4] of the average number of repetition M. The term in $nd(k-\ell)/2$ stems from the contribution of $\mathbf{e}_{\mathsf{pk}}$, and naturally comes from the average number of steps in the lazy random walk due to the Hamming weight of $\tau(\mathbf{v}_2)$. As a result, choosing $\ell' \approx \ell$ would not be optimal because it would essentially make $\mathbf{e}_{\mathsf{pk}}$ larger than $\mathbf{e}$ as $\mathbf{e}_{\mathsf{pk}}$ grows faster with ℓ' than $\mathbf{e}$ does with ℓ. For common parameters (see Table 5.2), where ℓ is close to k[5], choosing $\ell' \approx (k+1)/2$ seems to be the best option as it halves the public key size while incurring (almost) no security loss. This is because for such parameters $\mathbf{e}_{\mathsf{pk}}$ is overpowered by the preimage error $\mathbf{e}$.

Remark 5.2

Phoenix shares with [ETWY22] the goal of moving the bulk of the preimage in $\mathbf{v}_{1,1}$ which is not transmitted. Our treatment is howbeit very different from the twisted norm approach of the latter work.

[3] This slack comes from the multiplication $M_\tau(\mathbf{B}_L)\tau(\mathbf{v}_2)$ in the coefficient embedding. Later we choose 3-smooth conductors yielding $\mu = \sqrt{2}$, and $\mu = 1$ for power-of-two conductors.

[4] Experimental results over 1000 signatures show that the median number of signature rejections is 14 and the average is 20.3, instead of $M = 20$ as chosen in Table 5.2.

[5] Choosing $\ell = k-2$ or $\ell = k-1$ is possible as opposed to the approach in [CGM19] because $\mathbf{e}$ is smaller by a factor of $\sqrt{3}\omega(\sqrt{\log_2 nd})$.

5.3.4 Security Analysis

Our scheme follows the GPV framework. One can thus use the simulation result of Theorem 4.2 adapted to Phoenix, which we provide here. As the proof is essentially the same as that of Corollary 4.1, we do not include it.

Corollary 5.1 (Approximate Gaussian Rejection Sampler)

Let n, d, k be positive integers with n a power of two, and define $q = 2^{k+1}-1$. Let $\ell, \ell' \in [\![0, k[\![$. Let R be the power-of-two cyclotomic ring of degree n. Let $T = \sqrt{nd(k-\ell)}(\sqrt{2nd} + \sqrt{nd(k-\ell)})$. Let $M > 1$, $\varepsilon \in (0, 1/2]$ and define $\gamma = \frac{\sqrt{\pi}}{\ln M}(\sqrt{\ln \varepsilon^{-1} + \ln M} + \sqrt{\ln \varepsilon^{-1}})$, adn $s = \alpha T$. Let $\mathbf{A}' \sim U(R_q^{d\times d})$, $\mathbf{R} \sim U(S_1^{2d\times d(k-\ell)})$ conditioned on $\|\mathbf{R}\|_2 \leq \sqrt{2nd} + \sqrt{nd(k-\ell)}$. We define $\mathcal{P}_1$ and $\mathcal{P}_2$ the same way as in Theorem 4.2 but where $\mathscr{D}_s, \mathscr{D}_t$ are replaced with $\mathcal{D}_{R^{2d},s}$.
Then, it holds that $\Delta(\mathcal{P}_1, \mathcal{P}_2) \leq \varepsilon$ and $\mathrm{RD}_\alpha(\mathcal{P}_1 \| \mathcal{P}_2) \leq 1/(1-\varepsilon)^{\alpha/(\alpha-1)}$ for all $\alpha \in (1, +\infty]$.

We can now formally state the strong EUF-CMA security (see Section 1.5.1) of Phoenix for uncompressed public key and then discuss the slight differences stemming from key compression. Compared to the original GPV security result [GPV08, Thm. 5.9], we note that we rely on the version of M-SIS given in Definition 1.14 which adds a norm check in the infinity norm of the candidate solutions. Nevertheless, as it still follows the GPV framework [GPV08], it is also secure in the QROM [BDF+11]. The proof exactly follows that of [GPV08, Prop. 6.2] which is why we do not include it. At a high level, the proof follows two game hops[6]. The first consists in simulating all the signature queries (and random oracle queries accordingly) and thus is argued by Corollary 5.1. The second then simulates the public key under the M-LWE assumption by replacing $\mathbf{AR} \bmod qR$ by a uniform matrix $\mathbf{B}$. The advantage in the final game is then bounded by the M-SIS advantage as we can use this adversary to construct a solution to an M-SIS instance.

Theorem 5.1 (Adapted from [GPV08, Prop. 6.2])

We take the parameters selected according to Phoenix.Setup (Algorithm 5.2). It holds that Phoenix is strongly EUF-CMA-secure in the random oracle model under $\text{M-LWE}_{n,d,d,q,U(S_1),U(S_1)}$ and $\text{M-SIS}_{n,d,d(2+k-\ell),q,\beta,\beta_\infty}$, where $\beta = 2\sqrt{B_{1,1}^2 + B_{1,2}^2 + nd(k-\ell)}$ and $\beta_\infty = 2\max(B_{1,1}^\infty, B_{1,2}^\infty, 1) = 2B_{1,1}^\infty$. More precisely, Phoenix is δ-sEUF-CMA secure with

$$\delta \lesssim \left(\frac{1}{1-\varepsilon}\right)^Q (\varepsilon_{\text{M-SIS}} + d(k-\ell)\varepsilon_{\text{M-LWE}}).$$

Notice that the M-SIS assumption does not tightly match the forgery of Phoenix. Indeed, each preimage is extremely asymmetric which is why we impose different bounds $B_{1,1}, B_{1,2}$ and $B_{1,1}^\infty, B_{1,2}^\infty, (B_2^\infty = 1)$. We could then define a tighter M-SIS assumption that would only accepts solution meeting this specific asymmetrical behavior. The latter would then be trivially harder than the one used in Theorem 5.1. Additionally, as our scheme features key compression, we can use the M-LWE assumption in the security reduction but the public key will not be uniform over R_q but only over the high-order bits. This would give a skewed M-SIS assumption over the instance $[\mathbf{I}_d | \mathbf{A}' | \mathbf{G}_H - \mathbf{B}_H] \bmod qR$ where the third block only has high-order bits (modulo a possible wrap-around on the last component $\mathbf{G}_H - \mathbf{B}_H$). Since solving M-SIS involves discarding columns as described in Section 9.3 to find an optimal subdimension between nd and $2nd$, this skewed assumption could be estimated by $\text{M-SIS}_{n,d,2d,q,\beta',\beta'_\infty}$ where the bounds are set by taking $\mathbf{v}_2 = \mathbf{0}$. We do not elaborate further on the specific M-SIS assumption that underlies the security of Phoenix because we actually set the parameters of Phoenix by estimating the M-ISIS instance corresponding to our signature.

Concrete Security of Phoenix

The concrete security is instead assessed as described in Chapter 9. The key recovery is evaluated via the sM-LWE assumption. In our case, we apply public key compression which means that the

[6] The stateless variant contains a salt so that signature queries on the same message $\mathbf{m}$ will corresponds to independent syndromes $\mathcal{H}(\mathbf{m}, \mathsf{salt})$. The probability of having a collision on salt is bounded by $2^{-320} \cdot Q(Q-1)/2$

adversary only has access to the high-order bits of the sM-LWE instance $\mathbf{B} = [\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR$. Since $\mathbf{B}_H$ contains less information on $\mathbf{R}$ than the full matrix $\mathbf{B}$, we can lower-bound the complexity of the key recovery by assessing the cost of finding $\mathbf{R}$ given $(\mathbf{A}', \mathbf{B})$. The actual assumption that captures key recovery is a hybrid assumption mixing sM-LWE and rounding (as in the *Learning With Rounding* assumption [BPR12]).

The complexity of the forgery is lower-bounded by Theorem 5.1. However, it is best approximated via the inhomogeneous variant M-ISIS as is done in most hash-and-sign schemes [PFH+20, EFG+22, YJW23]. A forgery consists of a vector $\mathbf{v} = [\mathbf{v}_{1,1}^T|\mathbf{v}_{1,2}^T|\mathbf{v}_2^T]^T$ such that $[\mathbf{I}_d|\mathbf{A}'|\mathbf{G}_H - \mathbf{B}_H]\mathbf{v} = \mathbf{u} \bmod qR$ for a seemingly random and non-adversarial syndrome $\mathbf{u} = \mathcal{H}(\mathbf{m}, \mathsf{salt})$. Since the adversary must provide the salt as part of the signature, the best strategy is to select an arbitrary message and salt, compute $\mathbf{u} = \mathcal{H}(\mathbf{m}, \mathsf{salt})$ and find $\mathbf{v}$. Additionally, as $\mathbf{v}_2$ has very strict bounds (ternary), it is unlikely to have such small coefficients for $\mathbf{v}_2$ by solving M-ISIS on $([\mathbf{I}_d|\mathbf{A}'|\mathbf{G}_H - \mathbf{B}_H], \mathbf{u})$, unless they are set to zero. To hope for a valid forgery, one would thus fix a value for $\mathbf{v}_2 \in S_1^{d(k-\ell)}$ and solve the M-ISIS instance $([\mathbf{I}_d|\mathbf{A}'], \mathbf{u}' = \mathbf{u} - (\mathbf{G}_H - \mathbf{B}_H)\mathbf{v}_2)$ with norm bounds set from the signature verification from Algorithm 5.4. Setting $\mathbf{v}_2 = \mathbf{0}$ would discard these columns which is done in concrete attacks on M-ISIS anyway. Due to the asymmetry of our preimages, the solution returned by the adversary should also have a specific form. In particular $\mathbf{v}_{1,1}, \mathbf{v}_{1,2}$ are bounded both in Euclidean and infinity norms. This makes the fine-grained cryptanalysis difficult as current lattice reduction algorithms focus mostly on the Euclidean norm. Our approach is therefore to underestimate the actual cost of the attack by discarding the infinity norm and also the asymmetry of the solution. We believe that a thorough cryptanalysis would show that the forgery is more complex than the approach we described here. More precisely, we simply evaluate the complexity of finding $\mathbf{v}_1$ such that $[\mathbf{I}_d|\mathbf{A}']\mathbf{v}_1 = \mathbf{u}' \bmod q$ and $\|\mathbf{v}_1\|_2 \leq \beta = \sqrt{B_{1,1}^2 + B_{1,2}^2}$. We note that if β is close to or larger than $q\sqrt{nd/12}$, this M-ISIS instance becomes trivial but not the forgery because of our infinity norm checks and asymmetry. The above M-ISIS assumption is then estimated using the methodology from Section 9.3.2

Although our modulus is not particularly small with respect to the dimension and the M-ISIS bound, we also ran the estimator recently proposed by Ducas et al. [DEP23] as a sanity check to make sure it does not lead to a more efficient attack than the previously described approach. Their tool unfortunately suffers from large memory requirements when computing the intersection of a hypercube and a hyperball if the parameters are too large. We also leave this cryptanalysis to future work. The preimage and key compression can easily be reduced, and as a result the M-ISIS bound, to avoid possibly stretched parameter regimes at the expense of slightly larger signatures and/or keys. For example, if one were to take more conservative to achieve a smaller ratio β/q, we could still get signatures of 2412 bytes and a public key of 2592 bytes. Nevertheless, we again insist on the fact that our scheme also places infinity norm bounds which may invalidate the attack or make it much more complex.

Parameters of Phoenix

We now suggest parameter sets to instantiate Phoenix in Table 5.2. Although our scheme is presented over modules of rank d, working over rings offers better key compression. We thus give parameters in the ring setting. As all our tools hold for general number fields, we can use cyclotomic fields of composite conductors. This has been done in Mitaka [EFG+22] to achieve fine-grained security levels where they consider 3-smooth conductors. In this case, it incurs a loss of $\sqrt{2}$ in the quality of our sampler similarly to [EFG+22] due to the spectral bound on $\mathbf{R}$, which we take into account in our parameter selection. An alternative would be to choose a power-of-two cyclotomic ring of smaller degree and a larger rank d so that nd matches the dimension we suggest, the parameters scaling with nd. Although it would deteriorate the key sizes, it can be acceptable in applications where the public key is not sent often.

We see that the forgery security for Phoenix-III, estimated through M-ISIS in Euclidean norm, falls a few bits short of the NIST-III security level. Our estimate is however rather pessimistic because we discard the infinity norms and asymmetry of the preimage. Our cryptanalysis thus underestimates the actual complexity of the forgery. We believe that a thorough cryptanalysis would place the cost of the forgery above the NIST-III requirement, and also yield a better security for Phoenix-II, Phoenix-V, and Phoenix-V+. We however leave this cryptanalysis for future work.

	Phoenix-II	Phoenix-III	Phoenix-V	Phoenix-V$^+$
Security	NIST-II	NIST-III	NIST-V	NIST-V$^+$
Conductor	2^{11}	$2^4 3^5$	$2^3 3^6$	2^{12}
n	1024	1296	1944	2048
d	1	1	1	1
(k, ℓ, ℓ')	(16,15,8)	(17,16,9)	(18,17,10)	(18,17,10)
q	$2^{17}-1$	$2^{18}-1$	$2^{19}-1$	$2^{19}-1$
(M, ε, γ)	$(20, 2^{-66}, 8.13)$	$(20, 2^{-66}, 8.13)$	$(20, 2^{-66}, 8.13)$	$(20, 2^{-66}, 8.13)$
s	20105	35986	53978	40210
$B_{1,1}$	688341.2	1541069.0	3705333.9	3694729.9
$B_{1,2}$	268983.0	541623.4	995025.8	760798.9
$B^\infty_{1,1}$	64537	127114	238760	210427
$B^\infty_{1,2}$	36895	66037	99056	73790
|sk| (B)	512	648	972	1024
|pk| (B)	1184	1490	2219	2336
|sig| (B)	2190	2897	4468	4595
Key Rec. (C/Q)	162/143	203/179	312/275	332/292
Forg. (C/Q)	125/110	161/142	257/226	276/243

Table 5.2: Suggested parameter sets for Phoenix. Sizes are in bytes. The public key includes 32 bytes for the seed that expands to $\mathbf{A}'$. The size of Gaussian vectors is estimated by the entropy bound which can be achieved via the rANS encoding (see [ETWY22]). The bit security is the estimated core-SVP hardness (classical C, quantum Q).

5.4 Phoenix$_U$: A Version without Floats

We now describe a version of the Phoenix signature scheme called Phoenix$_U$ where we instantiate the distribution of signatures with uniform distributions over hypercubes instead of Gaussians. While it suffers from larger signature sizes, it has the advantage of requiring no floating point arithmetic whatsoever. Additionally, the rejection step is deterministic which makes the scheme even easier to implement. Although it follows the hash-and-sign paradigm in the GPV framework, the resulting scheme has many similarities with the Dilithium signature scheme [DKL$^+$18]. As such, further optimizations to Dilithium could also be applied to our scheme to heighten its efficiency.

5.4.1 Approximate Uniform Rejection Sampler

The approximate rejection sampler lends itself well to other families of distributions. One can instantiate it with every pairs of distributions $(\mathscr{D}_s, \mathscr{D}_t)$ that allow for efficient rejection sampling. Certain pairs lead to more compact sizes as depicted by the study of Devevey et al. [DFPS22]. Others, that would be discarded for the sole compactness consideration, can however have features that are relevant in other contexts. For example, ease of implementation or side-channel protection have become increasingly relevant now that post-quantum schemes are moving to range of application and deployment discussions. We therefore propose an alternative relying on uniform distributions over hypercubes which tend to facilitate such features. For completeness, we give the modified sampler tailored for uniform distributions. As for Phoenix, we choose $q = 2^{k+1} - 1$, and the gadget decomposition is centered. The error distribution coming from dropping low-order bits is exactly the same as that of Phoenix, and so is the key compression error.

Algorithm 5.6: AppRejSampler$_U(\mathbf{R}; \mathbf{A}', \mathbf{u}, \eta, B)$

Input: Trapdoor $\mathbf{R} \in R^{2d \times d(k-\ell)}$, Matrix $\mathbf{A}' \in R_q^{d \times d}$, Syndrome $\mathbf{u} \in R_q^d$, Mask bound $\eta > 0$, Shift bound $B > 0$.

1. $\mathbf{p}_1 \hookleftarrow U(S_\eta^{2d})$.
2. $\mathbf{w} \leftarrow \mathbf{u} - [\mathbf{I}_d | \mathbf{A}']\mathbf{p}_1 \bmod qR$. ▷ Syndrome correction
3. $\mathbf{z} \leftarrow \mathbf{G}^{-1}(\mathbf{w}) \in S_1^{dk}$. ▷ Deterministic
4. Parse $\mathbf{z}$ into $\mathbf{z}_L \in S_1^{d\ell}$ and $\mathbf{z}_H \in S_1^{d(k-\ell)}$ so that $\mathbf{Gz} = \mathbf{G}_L\mathbf{z}_L + \mathbf{G}_H\mathbf{z}_H$.
5. $\mathbf{v}_1' \leftarrow \mathbf{p}_1 + \mathbf{R}\mathbf{z}_H$.
6. $u \hookleftarrow U([0,1])$ ▷ Continuous
7. **if** $\|\mathbf{v}_1'\|_\infty > \eta - B$, go back to 1.
8. $\mathbf{e} \leftarrow \mathbf{G}_L\mathbf{z}_L$ ▷ Sampling error
9. $\mathbf{e}_{\mathsf{pk}} \leftarrow (([\mathbf{I}_d | \mathbf{A}']\mathbf{R} \bmod qR) - \mathbf{B}_H)\mathbf{z}_H$ ▷ Compression error

10. $\mathbf{v}_1 \leftarrow \mathbf{v}'_1 + \begin{bmatrix} \mathbf{e} - \mathbf{e}_{\mathsf{pk}} \\ \mathbf{0} \end{bmatrix}$
11. $\mathbf{v}_2 \leftarrow \mathbf{z}_H$

Output: $\mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$.

The bound B should be a bound on $\|\mathbf{R}\mathbf{z}_H\|_\infty$, while the mask bound η is set afterwards to ensure a correct rejection sampling. We discuss it in Sections 5.4.2 and 5.4.3.

5.4.2 Description

The scheme $\mathsf{Phoenix}_\mathsf{U}$ can then be obtained by essentially substituting the preimage sampler. The only difference comes from distribution-specific parameters which then feed through to the verification bounds.

Algorithm 5.7: $\mathsf{Phoenix}_\mathsf{U}$.Setup

Input: Security parameter λ.

1. Choose positive integers d, k.
2. $q \leftarrow 2^{k+1} - 1$.
3. Choose $\ell, \ell' \in [0, k-1]$.
4. $\mathbf{G} = \mathbf{I}_d \otimes [1|\cdots|2^{k-1}] \in R_q^{d \times dk}$.
5. $\mathbf{G}_H = \mathbf{I}_d \otimes [2^\ell|\cdots|2^{k-1}] \in R_q^{d \times d(k-\ell)}$.
6. $\mathbf{G}_L = \mathbf{I}_d \otimes [1|\cdots|2^{\ell-1}] \in R_q^{d \times d\ell}$.
7. Fix B a bound on $\|\mathbf{R}\mathbf{z}_H\|_\infty$.
8. Choose $M > 1$. ▷ Repetition rate
9. $\eta \leftarrow \lceil B \cdot M^{1/2nd}/(M^{1/2nd} - 1) - 1/2 \rceil$
10. $\mathbf{A}' \hookleftarrow U(R_q^{d \times d})$.

Output: $\mathsf{pp} = (\mathbf{A}'; \mathbf{G}, \mathbf{G}_L, \mathbf{G}_H; \lambda, n, q, d, k, \ell, B, \eta, M)$.

Algorithm 5.8: $\mathsf{Phoenix}_\mathsf{U}$.KeyGen

Input: Public parameters pp as in Algorithm 5.7.

1. $\mathbf{R} \hookleftarrow U(S_1^{2d \times d(k-\ell)})$.
2. $\mathbf{B} \leftarrow [\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR \in R_q^{d \times d(k-\ell)}$
3. Parse $\mathbf{B}$ as $\mathbf{B}_L + \mathbf{B}_H$ with $\mathbf{B}_L \in S_{2^{\ell'}-1}^{d \times d(k-\ell)}$ and $\mathbf{B}_H \in 2^{\ell'} S_{2^{k-\ell'}-1}^{d \times d(d-\ell)}$.

Output: $\mathsf{pk} = \mathbf{B}_H$, and $\mathsf{sk} = \mathbf{R}$. ▷ pp stored with pk for simplicity

Algorithm 5.9: $\mathsf{Phoenix}_\mathsf{U}$.Sign

Input: Secret key sk, Message $\mathbf{m} \in \{0,1\}^*$, Public key pk.

1. $\mathsf{salt} \hookleftarrow U(\{0,1\}^{320})$.
2. $(\mathbf{v}_1, \mathbf{v}_2) \leftarrow \mathsf{AppRejSampler}_\mathsf{U}(\mathbf{R}; \mathbf{A}', \mathcal{H}(\mathbf{m}, \mathsf{salt}), \eta, B)$. ▷ Algorithm 5.6
3. $b_1 \leftarrow (\|\mathbf{v}_{1,1}\|_\infty \le B_{1,1}^\infty) \wedge (\|\mathbf{v}_{1,2}\|_\infty \le B_{1,2}^\infty) \wedge (\|\mathbf{v}_2\|_\infty \le 1)$.
4. **if** $b_1 = 0$, restart.

Output: $\mathsf{sig} = (\mathsf{salt}, \mathbf{v}_{1,2}, \mathbf{v}_2)$.

Algorithm 5.10: $\mathsf{Phoenix}_\mathsf{U}$.Verify

Input: Public key pk, Message $\mathbf{m} \in \{0,1\}^*$, Signature sig.

1. $\mathbf{v}_{1,1} \leftarrow \mathcal{H}(\mathbf{m}, \mathsf{salt}) - \mathbf{A}'\mathbf{v}_{1,2} - (\mathbf{G}_H - \mathbf{B}_H)\mathbf{v}_2 \bmod qR \in R^d$.
2. $b_1 \leftarrow (\|\mathbf{v}_{1,1}\|_\infty \le B_{1,1}^\infty) \wedge (\|\mathbf{v}_{1,2}\|_\infty \le B_{1,2}^\infty) \wedge (\|\mathbf{v}_2\|_\infty \le 1)$.

Output: b_1. ▷ 1 if valid, 0 otherwise

We now explain how the different bounds are set. First, the bound B can be derived by studying the distribution of $\mathbf{R}\mathbf{z}_H$. In particular, one can obtain a much tighter bound than the trivial $nd(k-\ell)$ that is still verified with overwhelming probability. Consider the power-of-two cyclotomics case and that $d = k - \ell = 1$. Then, $\mathbf{R}\mathbf{z}_H = [r_1 z | r_2 z]^T$ is a vector of R^2, where $r_i \sim U(S_1)$ and z almost follows the centered binomial distribution. This is because each coefficient of z corresponds to the most significant bit of some $|u|$ for u uniform in $[-(q-1)/2, (q-1)/2]$, mutiplied by the sign of u. As we have $\tau(r_i z) = M_\tau(r_i)\tau(z)$ the i-th coefficient is given by $\sum_j \pm r_{i,j_i} z_j$. Because $U([-1,1])$ is centered, $\pm r_{i,j_i}$ follows the same distribution, and thus $\pm r_{i,j_i} z_j$ follows a centered binomial

distribution of parameter 2/3 which we call $\mathcal{B}_{1,2/3}$. That is 0 with probability 2/3 and ± 1 each with probability 1/6. We can then use Chernoff bound using the cumulant generating function $K(\cdot)$ of $\mathcal{B}_{1,2/3}$ defined by $K(t) = \ln(\mathbb{E}[\exp(t\mathcal{B}_{1,2/3})]) = \ln(2/3 + 1/3\cosh(t))$ for $t \in \mathbb{R}$. The Chernoff bound gives $\mathbb{P}[|\sum_j \pm r_{i,j_i} z_j| \geq B] \leq 2^{-\lambda\alpha(B,n,\lambda)}$ where $\alpha(B,n,\lambda) = \frac{\log_2 e}{\lambda+1}\sup_{t\geq 0}(tB - nK(t))$. Then, the union bound gives $\mathbb{P}[\|\mathbf{R}\mathbf{z}_H\|_\infty \geq B] \leq 2n \cdot 2^{-\lambda\alpha(B,n,\lambda)}$. For a fixed λ, n, we can then solve for B so that the probability is at most $2^{-\lambda}$. For composite conductors, one also has to account for the slack μ. In practice we observe that B can even be slightly smaller than what the Chernoff bound gives. It could theoretically be enforced by rejecting the $\mathbf{v}_2$ (and thus the $\mathbf{p}_1$) that make $\mathbf{R}\mathbf{v}_2$ larger than B.

Then, by adopting the same Gaussian approximation for the compression error than the one we formulated for Phoenix, the verification bounds can be set as

$$B_{1,1}^{\infty} = \eta - B + 2^{\ell} - 1 + \left\lfloor 3.8\mu\sqrt{\frac{2^{\ell'}(2^{\ell'}-1)}{6}\frac{nd(k-\ell)}{2}} \right\rfloor$$
$$B_{1,2}^{\infty} = \eta - B.$$

Remark 5.3

The rejection sampling step in Algorithm 5.6 now only consists of an infinity norm check and all the distributions are uniform distributions. Also, we note that the quality of the sampler (and thus the signature scheme itself) is no longer driven by $\|\mathbf{R}\|_2$. As such, there is no need to enforce a spectral bound at key generation. Looking ahead to Chapter 8, this step can generally be performed efficiently but at the cost of some floating point FFT computations. It is not needed for Phoenix$_U$, which means that there is no need for floating point arithmetic in the entire scheme.

5.4.3 Security Analysis

We start by adapting Theorem 4.2 to ensure the simulatability of preimages. We state it in the following corollary for completeness. Here, the parameter ε for the smooth Rényi divergence is chosen to be $\varepsilon = 0$, which means the rejection sampling is exact.

Corollary 5.2 (Approximate Uniform Rejection Sampler)

Let n, d, k be positive integers and define $q = 2^{k+1} - 1$. Let $\ell, \ell' \in [\![0, k[\![$. Let $\mathbf{A}' \sim U(R_q^{d\times d})$ and $\mathbf{R} \sim U(S_1^{2d\times d(k-\ell)})$. Then, we let B be a bound on $\|\mathbf{R}\mathbf{z}_H\|_\infty$ and $M > 1$ be the average repetition rate. We define $\eta = \lceil B \cdot M^{1/2nd}/(M^{1/2nd} - 1) - 1/2 \rceil$. We define $\mathcal{P}_1$ and $\mathcal{P}_2$ the same way as in Theorem 4.2 but where $\mathscr{D}_s = U(S_\eta^{2d})$ and $\mathscr{D}_t = U(S_{\eta-B}^{2d})$. Then, it holds that $\mathcal{P}_1$ and $\mathcal{P}_2$ are identical.

Proof (Corollary 5.2). We again rely on Theorem 4.2 and simply have to check the smooth Rényi divergence condition. Here we set $\varepsilon = 0$ which means that we actually look at the regular Rényi divergence of infinite order. Consider a shift $\mathbf{R}\mathbf{z}_H \in R^d$, which then verifies $\|\mathbf{R}\mathbf{z}_H\|_\infty \leq B$. We define $\mathcal{P} = U(S_{\eta-B}^{2d})$ and $\mathcal{Q} = U(S_\eta^{2d})$. We then have

$$\begin{aligned}\mathrm{RD}_\infty(\mathcal{P}\|\mathcal{Q}^{+\mathbf{R}\mathbf{z}_H}) = \max_{\mathbf{x}\in S_{\eta-B}^{2d}} \frac{\mathcal{P}(\mathbf{x})}{\mathcal{Q}^{+\mathbf{R}\mathbf{z}_H}(\mathbf{x})} &= \frac{(2(\eta-B)+1)^{-2nd}}{\mathbf{1}(\mathbf{x} - \mathbf{R}\mathbf{z}_H \in S_\eta^{2d}) \cdot (2\eta+1)^{-2nd}} \\ &= \left(\frac{2\eta+1}{2\eta - 2B + 1}\right)^{2nd},\end{aligned}$$

where the last equality follows from the fact that $\|\mathbf{x} - \mathbf{R}\mathbf{z}_H\|_\infty \leq (\eta - B) + B = \eta$. Note that this condition is actually necessary so that $\mathrm{Supp}(\mathcal{P}) \subseteq \mathrm{Supp}(\mathcal{Q}^{+\mathbf{R}\mathbf{z}_H})$. Because of how we chose η, we have

$$\eta \geq \frac{B \cdot M^{1/2nd}}{M^{1/2nd} - 1} - \frac{1}{2},$$

which leads to $\frac{2\eta+1}{2\eta-2B+1} \leq M^{1/2nd}$ and thus $\mathrm{RD}_\infty(\mathcal{P}\|\mathcal{Q}^{+\mathbf{R}\mathbf{z}_H}) \leq M$ as required.

This scheme then again follows the GPV framework and thus inherits the same security analysis. As the simulation result of Corollary 5.2 is a bit different and since the verification only involves the infinity norm, we give the security reduction result for completeness. We note that the version of M-SIS here only performs infinity norm checks. That is, we look for a non-zero vector $\mathbf{x}$ in $\mathcal{L}_q^\perp(\overline{\mathbf{A}})$ such that $\|\mathbf{x}\|_\infty \leq \beta_\infty$. When key compression is applied, the M-SIS assumption is also skewed as for Phoenix due to the block $\mathbf{G}_H - \mathbf{B}_H$

Theorem 5.2 (Adapted from [GPV08, Prop. 6.2])

We take the parameters selected according to $\mathsf{Phoenix_U}.\mathsf{Setup}$ (Algorithm 5.7). It holds that $\mathsf{Phoenix_U}$ is strongly EUF-CMA-secure in the random oracle model under $\text{M-LWE}_{n,d,d,q,U(S_1),U(S_1)}$ and $\text{M-SIS}_{n,d,d(2+k-\ell),q,\beta_\infty}$, where $\beta_\infty = 2\max(B_{1,1}^\infty, B_{1,2}^\infty, 1) = 2B_{1,1}^\infty$. More precisely, $\mathsf{Phoenix_U}$ is δ-sEUF-CMA with

$$\delta \leq \varepsilon_{\text{M-SIS}} + d(k-\ell)\varepsilon_{\text{M-LWE}}.$$

Just like Phoenix, we perform the forgery security assessment and parameter selection by looking at the M-ISIS instance that the scheme describes. We use the same methodology on the same M-ISIS instance but with a Euclidean bound specific to this scheme. As we deal with uniform elements, we can evaluate the expected bounds and thus derive the M-ISIS norm bound from them. We essentially use the same observation based on the Gaussian approximation to derive the M-ISIS bound in Euclidean norm. In the case of $\mathbf{v}'_{1,1}$ (before adding the errors) and $\mathbf{v}_{1,2}$, they follow centered uniform distribution with bounds $\eta - B$. As a result, they can be bounded with high probability by $\sqrt{(\eta-B)(\eta-B+1)/3}\sqrt{nd} \approx (\gamma-B)\sqrt{nd/3}$. Then, just like in Phoenix, the sampling error $\mathbf{e} = \mathbf{G}_L\mathbf{z}_L$ can be bounded by $\sqrt{(2^\ell-1))(2^{\ell+1}-1)/6}\sqrt{nd} \approx 2^\ell\sqrt{nd/3}$. Finally, the key compression error $\mathbf{e}_{\mathsf{pk}}$ can be bounded in Euclidean norm by $\sqrt{2^{\ell'}(2^{\ell'}-1)/3 \cdot nd(k-\ell)/2}\sqrt{nd}$. To be thorough so as to rely on this M-ISIS assumption, we would need to set Euclidean norm checks in the signing and verification process. Concretely, we would set

$$B_{1,1} = 1.04\sqrt{\frac{(\gamma-B)(\gamma-B+1)}{3} + \frac{(2^\ell-1)(2^{\ell+1}-1)}{6} + \mu^2\frac{2^{\ell'}(2^{\ell'}-1)}{6}\frac{nd(k-\ell)}{2}}\sqrt{nd}$$
$$B_{1,2} = 1.04\sqrt{\frac{(\gamma-B)(\gamma-B+1)}{3}}\sqrt{nd},$$

and then define the M-ISIS bound $\beta = \sqrt{B_{1,1}^2 + B_{1,2}^2}$. As for Phoenix, the bound provided by these formulas are verified empirically.

We now suggest parameter sets to instantiate this version in Table 5.3. The public key is reasonable compared to prior constructions but it suffers from slightly larger signatures than Phoenix, as already observed in [DFPS22]. This is to be balanced with the computational benefits of this variant. Additionally, we have not tried to optimize further this version as it is conceptually close to Dilithium [DKL+18]. Future optimizations could consist in re-using tricks from the latter and subsequent improvements to optimize this scheme. It may help further compress the signature size and public key or heighten security. Additionally, a thorough cryptanalysis is required to have a precise estimate of the security as our analysis does not consider the infinity norm at the moment. We expect the actual security to be higher than our current estimates as our approach is rather pessimistic.

As for Phoenix, the repetition rate M is chosen quite high but we observe that the actual number of rejection is much lower. Experimental results show that over 1000 signatures, the median and average number of rejections are respectively 3 and 4.6. Also, these results showed that although $B_{1,2}^\infty$ is tight, the bound $B_{1,1}^\infty$ overshoots the actual value of $\|\mathbf{v}_{1,1}\|_\infty$. A more precise estimate would therefore increase security, or allow one to choose slightly smaller parameters.

5.5 Comparison with Other Signatures

In this section, we compare the previous performance and security estimates of Tables 5.2 and 5.3 to that of the state-of-the-art lattice signatures. Table 5.4 first details the performance and the security levels of Phoenix/$\mathsf{Phoenix_U}$ and the main M-LWE-based signature schemes, namely Dilithium [DKL+18], Haetae [CCD+23], Raccoon [dPEK+], Eagle [YJW23], and G+G [DPS23]. We nevertheless stress that comparing these schemes directly has some limits as all of them, except Eagle, follows the Fiat-Shamir approach which is fundamentally different from the hash-and-sign

	Phoenix$_U$-I$^-$	Phoenix$_U$-II	Phoenix$_U$-III	Phoenix$_U$-V
Security	NIST-I$^-$	NIST-II	NIST-III	NIST-V
Conductor	2^{11}	$2^4 3^5$	$2^9 3^2$	2^{12}
n	1024	1296	1536	2048
d	1	1	1	1
(k, ℓ, ℓ')	(19,18,11)	(20,19,11)	(20,19,11)	(20, 19, 11)
q	$2^{20}-1$	$2^{21}-1$	$2^{21}-1$	$2^{21}-1$
(M, B)	$(20, 131)$	$(20, 186)$	$(20, 186)$	$(20, 131)$
η	89622	160387	190071	179179
$B^{\infty}_{1,1}$	423507	798837	838659	804979
$B^{\infty}_{1,2}$	89491	160201	189885	179048
$B_{1,1}$	5359390.0	11903691.8	13189833.4	15106869.9
$B_{1,2}$	1719509.1	3462930.6	4468500.5	4865286
|sk| (B)	512	648	768	1024
|pk| (B)	1184	1652	1952	2592
|sig| (B)	2600	3442	4072	5416
Key Rec. (C/Q)	134/118	171/150	211/185	299/263
Forg. (C/Q)	105/93	138/121	171/151	248/218

Table 5.3: Suggested parameter sets for the scheme with uniform distribution. Sizes are in bytes. The bit security is the estimated core-SVP hardness (classical C, quantum Q).

one. Our goal here is not to discuss in depth the comparative advantages of each approach but we note that the current state-of-the-art tends to show that schemes based on Fiat-Shamir are easier to implement as they support "simple" distributions, such as the uniform or the spherical Gaussian ones, but they rely on rewinding/forking lemma techniques which makes security in the QROM harder to prove, at least for the proposed parameters [KLS18, JMW23]. On the contrary, security of hash-and-sign constructions in the QROM is better understood [BDF+11] but these constructions require distributions that are harder to implement. In this regard, Phoenix illustrates the benefits of the approximate rejection sampler as it combines the nice features of these two approaches and thus constitutes an interesting alternative for those that do not want to choose between them.

	Distribution	Rejection	|sk| (B)	|pk| (B)	|sig| (B)	$\lambda^\star$ (C/Q)
Dilithium II	$U(\mathcal{B}_\infty)$	Yes	2544	1312	2420	121/110
Haetae-120	$U(\mathcal{B}_2)$	Yes	1376	992	1463	97/85
Raccoon-128	$\sum U(\mathcal{B}_\infty)$	No	14800	2256	11524	133/114
G+G-120	$\mathcal{D}_{R,s}$	No	$480^{(+)}$	1472	1677	121/106
Phoenix-II	$\mathcal{D}_{R,s}{}^{(\star)}$	Yes	512	1184	2190	125/110
Phoenix$_U$-II	$U(\mathcal{B}_\infty)^{(\star)}$	Yes	648	1652	3442	138/121
Dilithium III	$U(\mathcal{B}_\infty)$	Yes	4016	1952	3293	176/159
Haetae-180	$U(\mathcal{B}_2)$	Yes	2080	1472	2337	149/131
Raccoon-192	$\sum U(\mathcal{B}_\infty)$	No	18840	3160	14554	193/166
Eagle-1024	$\mathcal{D}_{R,s}$	No	$512^{(+)}$	1952	3052	176/160
G+G-180	$\mathcal{D}_{R,s}$	No	$640^{(+)}$	1952	2143	178/156
Phoenix-III	$\mathcal{D}_{R,s}{}^{(\star)}$	Yes	648	1490	2897	161/142
Phoenix$_U$-III	$U(\mathcal{B}_\infty)^{(\star)}$	Yes	768	1952	4072	171/151
Dilithium V	$U(\mathcal{B}_\infty)$	Yes	4880	2592	4595	252/229
Haetae-260	$U(\mathcal{B}_2)$	Yes	2720	2080	2908	214/188
Raccoon-256	$\sum U(\mathcal{B}_\infty)$	No	26016	4064	20330	284/243
G+G-260	$\mathcal{D}_{R,s}$	No	$768^{(+)}$	2336	2804	219/193
Phoenix-V	$\mathcal{D}_{R,s}{}^{(\star)}$	Yes	972	2219	4468	257/226
Phoenix$_U$-V	$U(\mathcal{B}_\infty)^{(\star)}$	Yes	1024	2592	5416	248/218

Table 5.4: Comparison of M-LWE-based schemes (sEUF-CMA versions, randomized signing). $U(\mathcal{B}_\infty)$: Uniform over hypercubes, $\sum U(\mathcal{B}_\infty)$: Convolution of uniform over hypercubes, $U(\mathcal{B}_2)$: Uniform over continuous hyperball, $\mathcal{D}_{R,s}$: Gaussian.
$^{(\star)}$ The distribution of $\mathbf{v}_2$ is $U(\mathbf{G}_H^{-1}(R_q^d))$ for Phoenix /Phoenix$_U$.
$^{(+)}$ Does not include the Gaussian perturbation sampling material.

For a full comparison, we also give in Table 5.5 how Phoenix and Phoenix$_U$ place themselves in the landscape of hash-and-sign schemes: Falcon [PFH+20], Mitaka [EFG+22], Solmae [KTW+22], Eagle and Robin [YJW23]. Most of them are based on NTRU lattices rather than M-LWE. They usually achieve a smaller bandwidth (signature + public key) than M-LWE-based schemes, but at the expense of an extra assumption. Designs based on M-LWE may be preferred to those based on NTRU in specific use cases, e.g., with stretched parameters. Such schemes also carry a certain complexity of implementation due to complex Gaussian samplers (FFO sampler for [PFH+20], hybrid sampler for [EFG+22], perturbation samplers for [YJW23], mask sampler for [DPS23]). Specific use-cases where it is best to avoid complex operations may benefit from other designs like Phoenix or Phoenix$_U$, if one is willing to accept a factor $1.6 - 2.2$ in bandwidth of course.

	Assumption	|sk| (B)	|pk| (B)	|sig| (B)	$\lambda^\star$ (C/Q)
Falcon-512	NTRU	$1998^{(\approx)}$	896	666	123/108
Mitaka-648	NTRU	$2421^{(\approx)}$	972	827	136/123
Solmae-512	NTRU	$1998^{(\approx)}$	896	666	127/115
Robin-701	iNTRU	$351^{(+)}$	1227	992	116/105
Phoenix-II	M-LWE	512	1184	2190	125/110
Phoenix$_U$-II	M-LWE	648	1652	3442	138/121
Mitaka-864	NTRU	$3528^{(\approx)}$	1512	1176	192/174
Eagle-1024	M-LWE	$512^{(+)}$	1952	3052	176/160
Robin-1061	iNTRU	$531^{(+)}$	1990	1527	181/165
Phoenix-III	M-LWE	648	1490	2897	161/142
Phoenix$_U$-III	M-LWE	768	1952	4072	171/151
Falcon-1024	NTRU	$3840^{(\approx)}$	1792	1280	272/239
Mitaka-1024	NTRU	$4215^{(\approx)}$	1792	1405	233/211
Solmae-1024	NTRU	$4125^{(\approx)}$	1792	1375	256/232
Robin-1279	iNTRU	$640^{(+)}$	2399	1862	228/207
Phoenix-V	M-LWE	972	2219	4468	257/226
Phoenix$_U$-V	M-LWE	1024	2592	5416	248/218

Table 5.5: Comparison of hash-and-sign schemes (sEUF-CMA versions, randomized signing).
$^{(+)}$ Does not include the Gaussian perturbation sampling material.
$^{(\approx)}$ Approximation $|\mathsf{sk}| \approx 3|\mathsf{sig}|$ taken from the specifications for Falcon, and we use a similar approximation for Mitaka and Solmae although they use different samplers.

5.6 Conclusion

The goal of this chapter and of Part II in general was to optimize preimage samplers and argue their practicality in the design of digital signatures. This led to the design of Phoenix, a signature scheme bridging the two main lattice signature paradigms, namely *Hash-and-Sign* and *Fiat-Shamir with Aborts*. It not only provides a reexamination of the cleavage between the two paradigms, but also provide an efficient proposal blending some of the most interesting traits of both families of signatures.

Not only does Phoenix achieve decently compact signature sizes, but the approximate rejection sampler also yields other attractive characteristics. It allows for a wide variety of signature distributions, some of which being more relevant than others for specific use-cases such as ease of implementation, side-channel resistance, compactness, etc. It also gives the ability to compress the public key of Phoenix in a much simpler way than for other schemes like Dilithium [DKL+18] or Haetae [CCD+23], and at almost no cost on security. These features, among others like the strong asymmetry of preimages, should foster further improvements on the approximate rejection sampler and signatures like Phoenix.

Part III

Advanced Signatures

This third part dives into the design of advanced post-quantum signatures using lattices. Through schemes proven secure in the standard model thought for privacy-enhancing applications, we present the first post-quantum anonymous credentials. We then focus on the next step of the cryptographic pipeline after having analyzed the security foundations, designed the primitives and assembled them to build up protocols: implementation. We present our full-fledged implementation of lattice-based anonymous credentials, showcasing practical post-quantum privacy.

6

Standard Model Signatures for Privacy

We now look at the design of so-called *signatures with efficient protocols* [CL02] on lattices, which constitutes a key building block in privacy applications. This chapter focuses on the signature scheme itself which can be seen as a standalone signature whose security is proven in the standard model, although it is only relevant when plugged into privacy-driven applications. We first present a construction based mostly on statistical arguments, and optimize it further to obtain compact signatures for privacy-enhancing primitives such as *anonymous credentials* in Chapter 7.

The work presented in this chapter is based on two papers with my co-authors Sven Argo, Tim Güneysu, Georg Land, Adeline Roux-Langlois and Olivier Sanders.

[JRS23] **Lattice Signature With Efficient Protocols, Application to Anonymous Credentials**. Published at Crypto 2023. Co-authored only with Adeline Roux-Langlois, and Olivier Sanders.

[AGJ+24] **Practical Post-Quantum Signatures for Privacy**. Published at ACM CCS 2024.

Contents

6.1 Introduction

The transition to post-quantum cryptography has been an enormous challenge and effort for cryptographers over the last decade. It has shown impressive results, in particular in the construction

of efficient and compact digital signatures. However, these efforts have been driven by the urgency of designing and standardizing central cryptographic signatures, and not more advanced ones, e.g., targeting privacy-preserving applications. Beyond the consideration of post-quantum security, cryptographers have indeed questioned the limitations of simple signatures as they may give rise to many privacy issues. Typically, presentation of the same certificate sig each time $\mathbf{m}$ needs to be authenticated allows for tracing sig and hence its owner. Moreover, if $\mathbf{m}$ is a vector of elements m_i, then verification of sig requires knowledge of all these elements even if they are irrelevant for the current authentication. In the context of digital identity, this concretely means that a user must reveal all their attributes, e.g., name, address, date of birth, etc, to prove authenticity of only one of them.

The topical example of age control to access adult-only websites epitomizes these problems. The current debates in France[1] or United Kingdom[2] show the same divide between two groups. One group is obviously unhappy with the current declarative approach, where the user certifies being old enough to access the website, and thus calls for stronger forms of authentication. Digital certificates could easily address this problem but the other group points out the obvious privacy issues resulting from the limitations mentioned above. Actually, unnecessarily providing sensitive information to a website is likely to lead to severe security issues that go well beyond mere privacy concerns: phishing, impersonation, etc.

6.1.1 Related Work

Based on this observation, it is actually logical that standard digital signatures are not best suited for all use-cases. In particular, the fact that electronic data can no longer be controlled once they are revealed calls for solutions disclosing as few information as possible during authentication. This has given rise to countless advanced cryptographic primitives, tailored to very specific use-cases, such as blind signatures [Cha82], group signatures [CvH91, BSZ05], Direct Anonymous Attestations (DAA) [BCC04], Enhanced Privacy IDentification (EPID) [BL07], anonymous credentials [Cha85, CL01, FHS19], e-cash [Cha82], etc. Far from simply being theoretical constructions, these mechanisms can be implemented very efficiently [PS16, CDL16, San21] leading to a small overhead compared to a non-private version built upon standard digital signatures. Some of them have been included in standards [ISO13a, ISO13b] and even embedded in billions of devices [TCG15, Int16]. Very recently, they have been advocated[3] by the GSMA (an organization gathering most industrial actors of the telecommunication ecosystem) for implementing the future European Digital Identity Wallet[4]. Interestingly, this GSMA document depicts privacy as a "positive differentiator", thus contrasting with the usual perception of privacy which was so far seen as a legal constraint. If it reflects an evolution of the industrial position on this topic, then we could see more applications of those privacy-preserving mechanisms in a near future.

Surprisingly, the diversity of use-cases addressed by these privacy-preserving authentication mechanisms contrasts with the very few mathematical settings allowing efficient designs. A closer look at these standards indeed shows that all of them make use of RSA moduli or cyclic groups and thus cannot withstand the power of quantum computing. The emerging success of such systems is thus based on foundations that will crumble as soon as a sufficiently powerful quantum computer appears.

This unsatisfying state of affairs clearly calls for the design of post-quantum alternatives to such systems. However, when we look at the cryptographic literature on this topic, it is striking to see that the existing post-quantum solutions are not only much less efficient than their classical counterparts but also extremely rare. Typically, prior to our first paper on the subject [JRS23], there was no explicit post-quantum anonymous credentials system. Even when we consider popular primitives such as group signatures, we note that the most efficient solutions [dPLS18, LNPS21, LNP22] depart from the traditional model [BSZ05] as they do not achieve non-frameability, a property implying that the certificate issuer does not know users' secret keys and that is thus incompatible with their construction. Although this might seem to be a minor restriction for group signatures, this has very important consequences on their industrial variants such as DAA [CKLL19] and EPID [BEF19]. Indeed, for the latter, the knowledge of the users' secret keys allows one to break anonymity, which would make the resulting construction totally pointless.

To understand the contrasting situations of classical constructions and post-quantum ones in the area of privacy-preserving authentication mechanisms, it is important to recall that all of them

[1] CNIL recommendations for online age verification and user privacy
[2] United Kingdom safety bill strengthening age verification
[3] GSMA Official Response: eIDAS 2.0 and Privacy
[4] European Digital Identity Wallet Architecture and Reference Framework

require, at some point, to prove knowledge [GMR85] of a signature on some (potentially secret) attributes. For example, in an anonymous credential system, the user generally receives a signature on their attributes and some secret key at the time of issuance. To show their credentials, they then reveal the requested attributes and prove knowledge of the signature, the hidden attributes and the secret key so as to remain anonymous. Non-frameable group signatures, DAA or EPID schemes follow the same high-level workflow. Of course, the resulting signatures also contains additional elements that define the specificity of each primitive but the point is that the common core is this proof of knowledge which essentially needs two kinds of building blocks: a "signature scheme with efficient protocols" (SEP) as coined by Camenisch and Lysyanskaya [CL02] and an associated zero-knowledge (ZK) proof system.

The latter notion is well-known and has seen several advances over the past few years, in particular in the lattice setting, e.g., [BLS19, YAZ+19, LNP22]. The former notion is rather informal but it usually refers to a digital signature scheme with some specific features such as the ability to sign committed (hidden) messages and to efficiently prove knowledge of a signature on such messages. This places some restrictions on the design of the signature scheme as it for example proscribes hash functions and hence most popular paradigms such as Hash-and-Sign and Fiat-Shamir discussed in Chapter 5. Yet, several extremely efficient constructions from number theoretic assumptions exist, in particular in bilinear (pairing) environments [CL04, BB08, PS16]. They constitute a very powerful and simple-to-use building block, which explains the countless applications using them.

This situation stands in sharp contrast with the one of post-quantum cryptography where only one lattice-based construction [LLM+16] with such features existed before the beginning of this thesis. Moreover the latter was designed with Stern's proof of knowledge in mind and thus does not leverage the recent advances in the area of lattice-based zero-knowledge proofs. The original paper only provides asymptotic estimation but our thorough analysis (which can be found in [JRS23, App. H]) shows that, even with the more recent ZK protocol from [YAZ+19], a proof of knowledge of a signature is still, at best, 670 MB large, which is far too high for practical applications. This leaves designers of privacy-preserving systems with no other solution than constructing the whole system from scratch, as was done for example in the case of EPID [BEF19] and DAA [CKLL19], which requires skills in many different areas and thus limits the number of contributions.

6.1.2 Our Contributions

The goal of this chapter is to propose the lattice counterpart of [CL04, BB08, PS16], that is, a *signature scheme with efficient protocols* (SEP) that is specifically designed to smoothly and efficiently interact with the most recent lattice-based zero-knowledge proof systems. More precisely, we provide two lattice-based signature schemes for which we can (**P1**) obtain signatures on potentially hidden (in a commitment) messages, and (**P2**) prove in zero-knowledge the possession of a message-signature pair. These protocols will be thoroughly presented in Chapter 7. Compared to the initial construction [LLM+16], our schemes are not only much more efficient but also transposes well to an algebraically structured setting which leads to further performance improvements.

Contribution 1: Statistical SEP

Our natural starting point is [LLM+16] which consists in a Boyen signature [Boy10] on a randomly chosen tag $\mathbf{t} \in \{0,1\}^\ell$ and for a syndrome shifted by the binary decomposition of the commitment $\mathbf{c} = \mathbf{D}_0\mathbf{r} + \mathbf{D}_1\mathbf{m} \bmod q\mathbb{Z}$ to a binary message $\mathbf{m}$, the commitment scheme being implicit in [Ajt96]. At first sight, this scheme perfectly fits the recent zero-knowledge proof system proposed by Yang et al. [YAZ+19] but yet leads to an extremely large proof of knowledge as mentioned above (a thorough complexity analysis is provided in [JRS23, App. F.3 & Tab. H.1]). We then undertake a complete overhaul of this scheme, pointing out at the same time the reasons of such a high complexity.

The main novelty is that we adopt a much more global approach as we look simultaneously at the three components of such systems, namely the commitment scheme (necessary to obtain signature on hidden messages), the signature scheme and the zero-knowledge proof systems, and the possible synergies. We, in particular, emphasize that the design choices we made for each component were not driven by the will to improve the latter individually but rather by their impact on the whole system. Typically, some of the modifications we introduce in the signature scheme itself has almost no impact on its complexity but yet results in very significant gains when it comes to proving knowledge of a signature.

The signature scheme. One of the first consequences of having to sign committed messages is that the signature must now include the randomness added to the commitment by the signer. In [LLM+16], this randomness has the same dimension as the one of the Boyen signature but a much larger width and thus represents the largest part of the signature. This is amplified by the proof of knowledge, which explains in part the high complexity of the latter. One of the reasons of such a large width is that the security proof requires to embed a hidden relation in the matrix $\mathbf{D}$ that is applied to the binary decomposition of the Ajtai commitment $\mathbf{c}$. More precisely, it defines $\mathbf{D} = \mathbf{AS} \bmod q\mathbb{Z}$ for the matrix $\mathbf{A}$ from the Boyen public key and some short matrix $\mathbf{S}$. This (along with other design choices discussed below) deteriorates the quality of the M-SIS solution extracted during the security proof and thus leads to large parameters.

To address this issue, we depart from [LLM+16] by generating conjointly the parameters of the signature scheme and the ones of the commitment scheme and in particular by re-using parts of the former in the latter. More specifically, in our construction, a commitment to $\mathbf{m}$ is $\mathbf{c} = \mathbf{Ar} + \mathbf{Dm} \bmod qR$, for a Gaussian randomness $\mathbf{r}$, where $\mathbf{A}$ is a matrix from the signer's public key and $\mathbf{D}$ is a public random matrix. From the efficiency standpoint, this has two important effects. First, this allows merging the randomness $\mathbf{r}$ with the other parts of the signatures, as we explain below, and thus to reduce the number of elements that we have to prove knowledge of. Second, as $\mathbf{A}$ is no longer hidden by a matrix $\mathbf{S}$, this significantly reduces the discrepancy between the adversary output and the extracted M-SIS solution in the security proof, leading to much better parameters.

Obviously, this has important consequences on the construction as the commitment matrix $\mathbf{A}$ is now selected by the signer, which is usually embodied by the adversary in privacy security games. To ensure that $\mathbf{A}$ is random to make the Ajtai commitment hiding, we need to generate it as a hash output. This solution is then totally incompatible with the approach in [LLM+16] where the signer needs to generate $\mathbf{A}$ together with an associated trapdoor.

Instead of Boyen's signature, we then choose to use the trapdoors of [MP12], recalled in Section 4.2, which interface well with the Ajtai commitment. More precisely, our public key is composed of a random matrix $\mathbf{A}$, a matrix $\mathbf{B} = \mathbf{AR} \bmod qR$ and a random syndrome $\mathbf{u}$, and the secret key is a random ternary matrix $\mathbf{R}$. In order to sign a binary message $\mathbf{m}$ hidden in a commitment $\mathbf{c} = \mathbf{Ar} + \mathbf{Dm} \bmod qR$, we use the MP sampler of Algorithm 4.1 to sample a Gaussian vector $\mathbf{v}'$ such that $[\mathbf{A}|\mathsf{t}\mathbf{G} - \mathbf{B}]\mathbf{v}' = \mathbf{u} + \mathbf{c} \bmod qR$, where t is a tag from a tag space $\mathcal{T} \subseteq R_q^\times$ and $\mathbf{G}$ is the gadget matrix. As $\mathbf{A}$ is involved in both the left hand side of the equation and in $\mathbf{c}$, we can set the signature as $(\mathsf{t}, \mathbf{v} = \mathbf{v}' - [\mathbf{r}^T|\mathbf{0}]^T)$. Verification then consists in checking

$$[\mathbf{A}|\mathsf{t}\mathbf{G} - \mathbf{B}]\mathbf{v} = \mathbf{u} + \mathbf{Dm} \bmod qR, \quad \mathsf{t} \in \mathcal{T}, \quad \text{and} \quad \mathbf{v} \text{ short}. \tag{6.1}$$

One can note that we have removed in the process the binary decomposition of $\mathbf{c}$. We indeed choose a very different approach in the security proof which shows that this step is actually not necessary. Among other things, we rely on a noise flooding argument (in the Rényi divergence) similar to the one used in Section 2.2. Removing this decomposition allows for a direct translation to an algebraic setting, enabling better compactness. We thus describe our construction over a (power-of-two) cyclotomic ring R. This removal of the binary decomposition is also crucial in order to compact the commitment randomness $\mathbf{r}$ with the preimage $\mathbf{v}'$. It avoids further intermediate steps that deteriorate the M-SIS solution extracted from the forgery, as explained above, which leads to better parameters overall. Moreover, when it comes to proving knowledge of the signature, each intermediate step makes the whole statement harder to prove and requires to create additional witnesses, i.e., each bit of $\mathbf{c}$, that must be committed, whose membership in $\{0, 1\}$ must be proven, etc. Our point here is that each seemingly innocent modification is considerably amplified when considering the full protocol and therefore results in major gains.

At this stage, a reader familiar with the construction in [dPLS18] might wonder why we do not try to embed the committed message in the tag t, instead of having this $\mathbf{Dm}$ component in our verification equation. Here, we need to recall that the situation of [dPLS18] is very specific as the signer (the group manager in their application) knows the signed message $\mathbf{m}$, which belongs to some bounded set in their application. In our case, we want to hide this message that may have a very large entropy (this is for example the case in anonymous credentials systems). In all cases, the security reduction must guess, at the setup stage, the value of the tag $\mathsf{t}^\star$ involved in the forgery. Therefore, if t is generated from $\mathbf{m}$ itself, then the reduction would have to guess this message, which would result in an exponential security loss in most scenarios. A workaround could be to construct t from $\mathcal{H}(\mathbf{m})$ for some appropriate function $\mathcal{H}$ (most likely a hash function because of the properties it would have to satisfy) whose image has lower entropy so as to guess $\mathcal{H}(\mathbf{m})$ instead of $\mathbf{m}$. Alternatively, $\mathcal{H}$ could be modelled as a random oracle. The problem with this solution is that

verification would now require to prove that $\mathcal{H}(\mathbf{m})$ has been correctly evaluated. For very specific scenarios (e.g., blind signature [dPK22, BLNS23a]) where $\mathbf{m}$ can be revealed at verification time, this would work with a security loss depending on the entropy of $\mathcal{H}(\mathbf{m})$. For all others (e.g., non-frameable group signatures, anonymous credentials, e-cash, etc), where the message must remain secret, this would not be possible with the zero-knowledge frameworks we target because of the nature of $\mathcal{H}$. As we aim to design a versatile tool, suitable for all applications, we choose to have a tag uncorrelated to the message, hence the $\mathbf{Dm}$ component mentioned above. As per the security proof, there are two constraints in the way to choose tags: generate tags without encountering collisions to only emit one signature per tag, and without enduring an exponential loss in the security proof due to guesses. Given that we essentially target privacy-preserving applications such as group signatures or anonymous credentials, we first focus on a stateful construction that inherently solves these two problems. For all these applications, it is indeed natural for the signer to keep track of the signatures it has issued, for revocation purposes if nothing else. For group signatures, this is even a requirement of the security model [BSZ05]: a registration table must be updated after each addition of a group member. A description of how to tweak the scheme to make it stateless can be found in the original paper [JRS23, App. G], incurring only a mild cost on the signature size.

We note here that to fulfill the requirements of protocol (**P1**), the user provides a commitment $\mathbf{c}_u = \mathbf{A}\mathbf{r}_u + \mathbf{Dm} \bmod qR$ which hides the message $\mathbf{m}$. However, in the security proof, the signer must somewhat control the commitment randomness. It can be done by re-randomizing the commitment into $\mathbf{c} = \mathbf{c}_u + \mathbf{A}\mathbf{r}_s \bmod qR$ for a fresh randomness $\mathbf{r}_s$ chosen by the signer, which is then merged to the signature vector $\mathbf{v}$ sent to the user. The noise $\mathbf{r}_s$ is essentially needed in a noise flooding argument (in the Rényi divergence) similar to the one used in Section 2.2.

Contribution 2: Optimized SEP

So far, we have essentially discussed improvements of both the commitment and the signature schemes. Table 6.1 shows that our resulting signature is around 30 times smaller than that of [LLM+16]. However, this gain is still not sufficient to lead to practical systems. Looking ahead to Chapter 7, the size of a proof of knowledge of a signature neighbors 700 KB. We now introduce several optimizations over the previous construction, without compromising on security. Combined, they provide significant improvements on the sizes of keys, signatures, and, more relevant even, of credentials.

Solving the double trapdoors problem. The inefficiency in Contribution 1 mainly stems from the use of statistical security arguments that requires to increase the number of columns of $\mathbf{A}$ to roughly dk (to use Lemma 1.16) and in turn the size of the signatures (and of the associated zero-knowledge proofs, see Chapter 7). Our first improvement is thus to use computational security arguments based on well-studied assumptions so as to move to more compact elements and in particular smaller matrices $\mathbf{A}$ with only $2d$ columns. Far from being a mere switching of parameters, this move introduces a very technical issue that was already identified in [dPLS18, LNPS21, BLNS23b] but for which no fully satisfactory solution has been proposed so far.

Let us first recall this issue. The core idea of security proofs of such signature schemes is to change the public key so as to have a valid trapdoor for all tags but one, which we denote by $\mathfrak{t}^+$. This is concretely done by replacing $\mathbf{AR}$ in the public key by $\mathbf{AR} + \mathfrak{t}^+\mathbf{G}$. As a result, for this new public key, we have $\mathbf{A}_\mathfrak{t} = [\mathbf{A}|(\mathfrak{t} - \mathfrak{t}^+)\mathbf{G} - \mathbf{AR}]$ where the gadget vanishes for $\mathfrak{t} = \mathfrak{t}^+$. In the computational setting, this change in the public key is done through a series of games where $\mathbf{AR}$ is first replaced by a random matrix $\mathbf{B}$ which is then replaced by $\mathbf{AR} + \mathfrak{t}^+\mathbf{G}$. At first sight, indistinguishability of these games seems to directly follow from the M-LWE assumption. Unfortunately, the proof is not that easy because the reduction must still produce valid signatures in the intermediate game (the one with public key $\mathbf{B}$) even though there is no longer any trapdoor. In [dPLS18, LNPS21], this problem was solved by artificially extending the public key so as to introduce a *second* trapdoor. In the case of MP trapdoors, this concretely means using matrices of the form $\mathbf{A}_\mathfrak{t} = [\mathbf{A}|\mathfrak{t}\mathbf{G} - \mathbf{AR}|\mathbf{G} - \mathbf{AR}'] \in R_q^{2d+2kd}$ where $\mathbf{R}'$ is a second trapdoor whose only purpose is to sample preimages in this intermediate game[5]. In other words, one must almost double the dimension of the signatures because of a peculiarity of the security proof, which is quite frustrating. In [dPLS18] the authors already question the actual need for this second trapdoor whereas the ones of [BLNS23b] see it as an "artifact" of the proof and propose to remove it in one of their instantiations. At this stage, we therefore end up with two unsatisfactory solutions. Either

[5] In the real-world, $\mathbf{R}'$ can be discarded after having generated the public key or, alternatively, one can replace $\mathbf{G} - \mathbf{AR}'$ by a random matrix.

we use this redundant trapdoor to prove security or we remove it to get a more efficient scheme without security proofs.

We instead propose a more satisfactory solution with no compromise on security and with only a very moderate efficiency loss. We indeed leverage the specificities of preimage sampling with MP trapdoors to move from $\mathbf{AR}$ to $\mathbf{AR} + \mathbf{t}^+\mathbf{G}$ by only replacing k columns simultaneously per game hop. More specifically, we ensure that, in each game, at most k columns of the public key have been replaced by random vectors. We therefore have, at all time, a partial trapdoor allowing to invert all components of a syndrome but one. We then only need a way to deal with the missing component, which can be done by only adding a $d \times k$ matrix $\mathbf{A}_3$ to $\mathbf{A}_\mathbf{t}$ instead of a $d \times dk$ matrix $\mathbf{AR}'$ as in the double trapdoors approach. We provide more details on this proof strategy in Sections 6.3 and 6.4. As this new strategy directly decreases the dimension of the signatures, it leads to a significant improvement of their size for most[6] of the parameters we use in practice. We believe it is of independent interest, although it is very specific to MP trapdoors.

Rényi-Based Security Analysis. In the same vein as the previous improvement, we also adopt a finer analysis of the security arguments which remains statistical arguments. More precisely, we need the outputs of the Gaussian samplers to be close enough to their ideal Gaussian distributions. So far, we only considered the statistical distance for such arguments. Other approaches based on the the Rényi divergence (say of order 2λ as suggested in [Pre17]) yield tighter security proofs and in turn more compact parameters. We thus depart from the statistical distance whenever possible. Also, as we are interested in implementing our scheme, such analyses have also proven to be beneficial to reduce the floating-point precision needed. The precision analysis of our samplers is presented in Chapter 8.

Elliptic Sampler. Another improvement comes from leveraging the elliptic sampler from Section 4.3 to further reduce the signature size. As mentioned in Remark 4.2, the security proof of the SEP requires a worst-case analysis of the preimage sampler. This is why we favor the elliptic sampler over the (approximate) rejection sampler.

Removing signer's randomness. Next, we also leverage different security arguments based on rejection sampling, which is inspired from the proof technique of [CKLL19, Lem. 3.1]. The idea is to decrease the reduction loss entailed by the probability preservation property of the Rényi divergence in the noise flooding argument in Contribution 1. We instead use rejection sampling which only suffers a (small) constant reduction loss factor, while tolerating a small leakage to keep compact parameters.

This modified security argument also allows for removing the randomness $\mathbf{r}_s$ added to the syndrome by the signer. This was necessary to prove security in the chosen message setting. Although $\mathbf{r}_s$ can be merged with the first part of the signature, it negatively impacts the parameters as it increases the norm of this first part. Our new security reduction shows that this additional randomness is no longer necessary, which means that we can remove it altogether.

Hermite Normal form. Instead of relying on M-SIS with a fully uniform $\mathbf{A}$, we rely on its Hermite Normal Form. Using a matrix of the form $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$ enables standard tricks [PFH+20, EFG+22, ETWY22] to reduce the signature size without affecting security by sending only part of the signature and recovering the remaining part during verification. Unfortunately, it has no impact on the zero-knowledge proof size because one needs to recompute the full preimage to perform the proof.

Tighter bounds. Finally, we use parameter optimizations by using tighter probabilistic bounds in several places. The first stems from a better use of the Gaussian tail bound of Lemma 1.21. We choose a smaller tailcut c to get a probability of $2^{-\lambda}$ instead of 2^{-2N} in the previous contribution, where N is the dimension of the Gaussian which is usually much bigger than λ. Then, we change the distribution of the secret key from uniform $U(S_1)$ to a centered binomial $\mathcal{B}_1$ as it leads to smaller spectral norms (which defines the quality of the elliptic sampler). We can also use spectral norm bounds that are satisfied only with constant probability instead of overwhelming, as long as the bound is enforced during key generation. It means that key generation might sample several secret keys until it finds a good one, and it only reduces the size of the secret key space by a constant factor. Then, at many occasions we need to bound the norm $\|\mathbf{Sx}\|_2$ for a ternary matrix $\mathbf{S}$ and a short integer vector $\mathbf{x}$. Although one could use the spectral norm of $\mathbf{S}$, it turns out to overshoot the bound we expect in practice. Instead, we use Johnson-Lindenstrauss-like bounds from Lemma 1.23 or Heuristic 1.3, as is done for example in [GHL22]. We obtain bounds of $O(\sqrt{N})\|\mathbf{x}\|_2$ instead

[6] More specifically, this strategy is more efficient than the one based on double trapdoors in the module case, i.e., whenever $d > 1$.

of $O(\sqrt{N}+\sqrt{M})\|\mathbf{x}\|_2$, where N is the number of rows of $\mathbf{S}$ and M the dimension of $\mathbf{x}$. Since N is usually much smaller than M, we get much tighter bounds leading to an improved parameter selection.

We give in Table 6.1 the resulting performance of our two SEPs compared to that of [LLM+16] to showcase the two waves of improvements. Note that it does not make sense to compare them with regular signatures like Phoenix in Chapter 5 or any of the signatures discussed in the latter chapter. This is because SEPs serve a totally different purpose which the aforementioned signatures cannot fulfil.

	Setting	|pk|	|sk|	|sig|
[LLM+16]	Standard	$3034 \cdot 10^6$	$1596 \cdot 10^4$	8617
Contribution 1	Module	3335	8879	289
Contribution 2	Module	47.5	10	6.8
		63876	159641	1265
			Improvement Factor	

Table 6.1: Comparison of efficiency estimates of the SEP from [LLM+16] and ours. The sizes are given in KB. They correspond to a security level of $\lambda = 128$. In the setting column, *Standard* stands for standard lattices (i.e., unstructured), in opposition to *Module* for module lattices.

6.1.3 Interfacing with Protocols

As mentioned above, a "signature scheme with efficient protocols" requires two kinds of protocols: (**P1**) get a signature on a committed message, and (**P2**) prove possession of a message-signature pair. Regarding the former, the problem is rather simple as the message $\mathbf{m}$ to sign is already embedded in a commitment $\mathbf{c} = \mathbf{A}\mathbf{r}_u + \mathbf{D}\mathbf{m} \bmod qR$. Then, the user needs to prove knowledge of $\mathbf{r}_u$ and $\mathbf{m}$ so as to rely on the EUF-CMA property of the signature scheme we introduced. In all cases, the user ends up with a signature $(\mathfrak{t}, \mathbf{v})$ on a binary $\mathbf{m}$ verifying an equation similar to Equation (6.1) and needs to prove it in a zero-knowledge way.

Proving knowledge of such a statement requires to prove that (1) $\mathfrak{t}$ is in the specified tag space, (2) $\mathbf{v}$ is short, (3) $\mathbf{m}$ is a vector of binary polynomials, and (4) that the quadratic equation is verified. Based on state-of-the-art proof systems, (1) constrains which tag space to choose so that we can efficiently prove membership, while ensuring that a difference of tags is in $R_q^\times$ as needed per the security proofs. Condition (2) requires to define a notion of shortness over the ring, which is usually defined based on the size of the polynomials' coefficients, i.e, ℓ_∞ norm. Up until recently, exact proofs performing the latter task [BLS19, ENS20] (also used for (3)) used NTT packing, i.e., interpreting the coefficients of $\mathbf{v}$ as the NTT (Number Theoretic Transform) of another vector $\mathbf{v}'$, which is most efficient when x^n+1 splits into low-degree irreducible factors modulo q. This splitting makes it harder to choose a proper tag space for which differences are always invertible. Finally, (4) requires a proof system able to deal with quadratic equations. Similar relations [dPLS18, LNPS21] were handled by transforming the relation quadratic in the witnesses into a linear relation in the commitment of the witnesses. Since efficient proofs of commitment opening rely on relaxed openings, this solution introduces a soundness gap in the proven statement, which we would like to avoid.

Instead, we use the very recent framework of Lyubashevsky et al. [LNP22] which provides a unified method to prove all our statements. It extends the previous works of [BLS19, ENS20] and enables proving quadratic relations exactly, as well as quadratic evaluations. The latter can be used to prove exact bounds directly in the ℓ_2 norm, which leads to more efficient proofs than proving ℓ_∞ bounds. The sizes are only discussed in Chapter 7 as they are only relevant when plugging the SEP into a concrete application.

Remark 6.1 (On a Unified Security Model for Privacy)

So far, we only discussed the unforgeability of signature with efficient protocols. The very purpose of an SEP is to be used as a building block for privacy-preserving primitives and is therefore not an autonomous construction. SEPs constitute an informal subclass of digital signatures whose relevance becomes clear only when plugged into a concrete privacy-preserving application. It is therefore tempting to try to define additional security properties

directly on the SEP, such as security requirements for the protocols (**P1**) and (**P2**). This is what was done in [LLM+16], and recently vastly abstracted by Bobolz et al. [BDK24] who attempted to formalize such a generic security model for what they called *Universal Anonymous Signatures*. Unfortunately, in order to cover the many use cases such as group signatures, blind signatures, anonymous credentials, etc, they have to introduce a very complex model. This is inevitable as these primitives have very different security requirements. Hence a unified security model is not particularly relevant. We give a detailed discussion on the security of protocols in Chapter 7.

6.2 Statistical Signature in the Standard Model

We present here our first signature with efficient protocols. It provides an alternative to the only such scheme based on lattices due to Libert et al. [LLM+16] that existed at that time.

The scheme in this section was initially introduced in [JRS23] which predates our study of the different gadget samplers of Part II. As such, our first construction only considers the MP sampler (Algorithm 4.1) and using a binary gadget ($b = 2$) as this is the one put forward in [MP12].

Nevertheless, our second signature of Section 6.4 and presented in [AGJ+24] embarks several optimizations including the use of the elliptic sampler MP* (Algorithm 4.5) with a higher base b (chosen to be the optimal base for the scheme).

6.2.1 The Signature Scheme

One of the main differences between the previous construction of [LLM+16] and ours is that we aim at optimizing the interactions between the commitment scheme implicitly used by such kind of protocols and the signature scheme itself. In [LLM+16], the public parameters of these two components were generated independently. We depart completely from this approach by generating these parameters conjointly and even by using a common matrix $\mathbf{A}$ for these two parts. Besides the natural gain in the public key size, this strategy allows one to merge different components of the signature itself. In particular, compared to [LLM+16], our signature no longer has to include the commitment opening, which significantly reduces its size.

Obviously, this has important consequences on the design of the scheme itself. One of them is that it forbids to re-use the approach of [LLM+16], inherited from Boyen signature [Boy10], where $\mathbf{A}$ was generated together with a trapdoor, because it would clearly break the hiding property of the commitment scheme. We instead rely on MP trapdoor functions of the form $[\mathbf{A}|\mathbf{tG} - \mathbf{AR}]$ where $\mathbf{t}$ is a tag from $R_q^\times$. We can therefore generate $\mathbf{A}$ as a random matrix[7] of size $d \times m_1$, where m_1 is the dimension of the commitment randomness. We then use it to construct the commitment $\mathbf{c}$ to a message $\mathbf{m} \in T_1^m$ as $\mathbf{c} = \mathbf{Ar} + \mathbf{Dm} \bmod qR$, where $\mathbf{D}$ is a random matrix of size $d \times m$ and m is the dimension of the message. Recall that $T_\eta = \tau^{-1}(\llbracket 0, \eta \rrbracket^n)$ is the set of polynomials with coefficients in $\llbracket 0, \eta \rrbracket$. The randomness $\mathbf{r}$ can then be merged with the short vector $\mathbf{v}$ generated thanks to the trapdoor, as mentioned above.

In [LLM+16], the authors had to first compute a binary decomposition $\mathbf{c}'$ of the commitment $\mathbf{c}$ to the message before generating a short preimage of $\mathbf{u} + \mathbf{Dc}'$ where $\mathbf{u}$ (resp. $\mathbf{D}$) was some public vector (resp. matrix). This might look harmless when we only consider the signature because it does not increase its size. However, when plugged in a zero-knowledge proof system, e.g. [YAZ+19] for proofs over $\mathbb{Z}_q$, this replaces one secret vector $\mathbf{c}$ by $\log_2 q$ ones and makes the overall statement to prove more complex. To remove this binary decomposition we revisit the security proof and show how to avoid it by using a noise flooding argument based on the Rényi Divergence, similar to that of Section 2.2. Additionally, this change seems necessary to extend our construction to polynomial rings. In this thesis we only describe the structured variant and refer the reader to the original paper [JRS23] for the unstructured one.

All the modifications we introduce have a second positive effect on complexity. In both our security proof and the one of [LLM+16], it is necessary to generate the public matrices with hidden relations, usually by multiplying one by some low-norm matrix $\mathbf{S}$ to generate the other. This impacts the norm of the extracted solutions, which grows with the number of such matrices

[7] In our protocol for signing hidden messages in Chapter 7, we will have to enforce this requirement but this can be done easily by setting $\mathbf{A}$ as some hash output.

and computational steps, and therefore impacts the system parameters. By reusing $\mathbf{A}$ for different purposes and by removing some computational steps (e.g., multiplication by $\mathbf{D}$), we significantly reduce the discrepancy between the adversary output and the resulting M-SIS solution, leading to much better parameters.

Working over algebraic rings requires changing a few design choices in the scheme. As we aim to be able to prove the verification of the signature in zero-knowledge, we need to identify a proof system and see the extent of the languages it covers. We employ the latest framework from Lyubashevsky et al. [LNP22], which we detail in Section 7.4. All we need to know for now is that it tackles quadratic relations and exact norm constraints (ℓ_2 in particular). This limits the choice for the tag space as we need an efficient membership proof. In our case, we choose tags in the set of binary polynomials T_1 with a fixed Hamming weight $\|\mathsf{t}\|_1$. It is similar to the identity space of the group signature construction of [LNP22]. We also use a message space that is similar to the latter but with no restriction on the number of non-zero coefficients.

We now describe the $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}$ algorithms of our signature scheme. It implicitly works over the cyclotomic ring of integers of conductor $2n$ where n is a power of two.

Algorithm 6.1: SEP.Setup

Input: Security parameter λ.

1. Select a positive integer d. ▷ M-SIS rank driving security
2. Select $\kappa \leq n$ to be a power of two. ▷ Number of splitting factors
3. Select a prime integer q such that $q = 2\kappa + 1 \bmod 4\kappa$ and $q > \sqrt{\kappa}^{\kappa}$.
4. Select a positive integer w such that $\binom{n}{w} \geq Q$. ▷ Hamming weight of tags
5. $k \leftarrow \lceil \log_2 q \rceil$.
6. $\mathcal{T}_w \leftarrow \{\mathsf{t} \in T_1 : \|\mathsf{t}\|_2 = \sqrt{w}\}$. ▷ Tag space
7. $m_1 \leftarrow \lceil d \log_3 q + f(\lambda) \rceil$ ▷ $f(\lambda) = \omega(\log_3 n)$
8. Choose a positive integer m. ▷ Maximum bit-size of $\mathbf{m}$ is $n \cdot m$
9. $\mathbf{G} = \mathbf{I}_d \otimes [1 | \cdots | 2^{k-1}] \in R_q^{d \times dk}$.
10. $r \leftarrow \eta_\varepsilon(\mathbb{Z})$. ▷ $r = 5.4$ leads to $\varepsilon \approx 2^{-131}$
11. Select $t > 0$. ▷ Spectral norm slack
12. $s \leftarrow r\sqrt{5}\sqrt{(\sqrt{nm_1} + \sqrt{ndk} + t)^2 + 1}$. ▷ Pre-image sampling width
13. $s_2 \leftarrow \sqrt{(\sqrt{nm_1} + \sqrt{nm} + t)^2 \cdot nm - s^2}$. ▷ Commitment randomness width
14. $s_1 \leftarrow \sqrt{s^2 + s_2^2}$.
15. $\mathbf{A} \hookleftarrow U(R_q^{d \times m_1})$.
16. $\mathbf{u} \hookleftarrow U(R_q^d)$.
17. $\mathbf{D} \hookleftarrow U(R_q^{d \times m})$. ▷ Message Commitment Key

Output: $\mathsf{pp} = (\mathbf{A}, \mathbf{u}, \mathbf{D}; \lambda, n, d, k, q, w, m_1, m, r, s, s_1, s_2)$. ▷ $\mathbf{A}, \mathbf{u}, \mathbf{D}$ can be stored as a 32-byte seed.

Algorithm 6.2: SEP.KeyGen

Input: Public parameters pp as in Algorithm 6.1.

1. $\mathbf{R} \hookleftarrow U(S_1^{m_1 \times dk})$ such that $\|\mathbf{R}\|_2 \leq \sqrt{nm_1} + \sqrt{ndk} + t$.
2. $\mathbf{B} \leftarrow \mathbf{AR} \bmod qR \in R_q^{d \times dk}$.

Output: $\mathsf{pk} = \mathbf{B}$, and $\mathsf{sk} = \mathbf{R}$. ▷ pp stored with pk for simplicity

Algorithm 6.3: SEP.Sign

Input: Signing key sk, Message $\mathbf{m} \in T_1^m$, Public key pk, State st.

1. $\mathbf{r} \hookleftarrow \mathcal{D}_{R^{m_1}, s_2}$.
2. $\mathbf{c} \leftarrow \mathbf{Ar} + \mathbf{Dm} \bmod qR$. ▷ Commitment to $\mathbf{m}$
3. $\mathsf{t} \leftarrow \mathsf{F}(\mathsf{st})$. ▷ $\mathsf{t} \in \mathcal{T}_w$
4. $\mathbf{v} \leftarrow \mathsf{MP\text{-}Sampler}(\mathbf{R}; \mathbf{A}, \mathbf{u} + \mathbf{c}, \mathsf{t}\mathbf{I}_d, \mathbf{u} + \mathbf{c}, s, r\sqrt{5}) - \begin{bmatrix} \mathbf{r} \\ \mathbf{0}_{dk} \end{bmatrix}$. ▷ Algorithm 4.1
5. $\mathsf{st} \leftarrow \mathsf{st} + 1$.

Output: $\mathsf{sig} = (\mathsf{t}, \mathbf{v})$.

Algorithm 6.4: SEP.Verify

Input: Public key pk, Message $\mathbf{m} \in T_1^m$, Signature sig.

1. $\mathbf{A}_\mathsf{t} \leftarrow [\mathbf{A} | \mathsf{t}\mathbf{G} - \mathbf{B}] \in R_q^{d \times (m_1 + dk)}$.
2. Parse $\mathbf{v}$ into $[\mathbf{v}_1^T | \mathbf{v}_2^T]^T$ with $\mathbf{v}_1 \in R^{m_1}$ and $\mathbf{v}_2 \in R^{dk}$.
3. $b_1 \leftarrow \|\mathbf{v}_1\|_2 \leq B_1$. ▷ $B_1 = s_1\sqrt{nm_1}$

4. $b_2 \leftarrow \|\mathbf{v}_2\|_2 \leq B_2$. ▷ $B_2 = s\sqrt{ndk}$
5. $b_3 \leftarrow \mathbf{A}_\mathsf{t}\mathbf{v} = \mathbf{u} + \mathbf{Dm} \bmod qR$.
6. $b_4 \leftarrow \mathsf{t} \in \mathcal{T}_w$).

Output: $b_1 \wedge b_2 \wedge b_3 \wedge b_4$. ▷ $b_1 = 1$ if valid, 0 otherwise

The condition on q allows us to limit the splitting so that $S_1 \bmod qR \subset R_q^\times$ per Lemma 1.4 (or rather Remark 1.2). Because tags have binary coefficients, not only do we have $\mathcal{T}_w \bmod qR \subset R_q^\times$ which is required by the MP sampler, but we also have that a difference of distinct tags is also a unit.

Then, the correctness of the signature scheme simply relies on the sum of discrete Gaussians (Lemma 1.12) and the Gaussian tail bound (Lemma 1.21). The former guarantees that $\mathbf{v}_1$ is statistically close to $\mathcal{D}_{\mathbb{Z}^{m_1},s_1}$ after substracting $\mathbf{r}$, and the latter ensures that the norm checks pass for an honest signature.

Lemma 6.1 (Correctness of the SEP)

The signature scheme of Algorithms 6.1, 6.2, 6.3, and 6.4 is correct.

Proof (Lemma 6.1). Let $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$, $(\mathsf{pk},\mathsf{sk}) = (\mathbf{B},\mathbf{R}) \leftarrow \mathsf{KeyGen}(\mathsf{pp})$. Let $\mathbf{m} \in T_1^m$ and $(\mathsf{t},\mathbf{v}) \leftarrow \mathsf{Sign}(\mathsf{sk},\mathbf{m},\mathsf{pk},\mathsf{st})$. Then, there exists a vector $\mathbf{r} \in R^{m_1}$ drawn from $\mathcal{D}_{R^{m_1},s_2}$ such that $\mathbf{v} = \mathbf{v}' - [\mathbf{r}^T|\mathbf{0}]^T$, where $\mathbf{v}'$ was obtained from $\mathsf{MP\text{-}Sampler}(\mathbf{R};\mathbf{A},\mathbf{u}+\mathbf{Ar}+\mathbf{Dm} \bmod qR, \mathsf{t}\mathbf{I}_d, s, r\sqrt{5})$. It thus holds that

$$\mathbf{A}_\mathsf{t}\mathbf{v} = \mathbf{A}_\mathsf{t}\mathbf{v}' - \mathbf{Ar} = \mathbf{u} + \mathbf{Ar} + \mathbf{Dm} - \mathbf{Ar} \bmod qR = \mathbf{u} + \mathbf{Dm} \bmod qR.$$

Then, also by [MP12, Thm. 5.5], it holds that $\mathbf{v}'$ is statistically close to $\mathcal{D}_{R^{m_1+dk},s}$ conditioned on $\mathbf{A}_\mathsf{t}\mathbf{v}' = \mathbf{u}+\mathbf{Ar}+\mathbf{Dm} \bmod qR$. Hence, by Lemma 1.12, $\mathbf{v}$ is statistically close to $\mathcal{D}_{R^{m_1+dk},\mathbf{s}}$ conditioned on $\mathbf{A}_\mathsf{t}\mathbf{v} = \mathbf{u}+\mathbf{Dm} \bmod qR$ and where $\mathbf{s} = [\sqrt{s^2 + s_2^2}\mathbf{1}_{m_1}|s\mathbf{1}_{dk}] = [s_1\mathbf{1}_{m_1}|s\mathbf{1}_{dk}]$. Finally, applying Lemma 1.21 with $c = 1$ yields the bounds on $\|\mathbf{v}_1\|_2$ and $\|\mathbf{v}_2\|_2$ and thus on $\|\mathbf{v}\|_2$. It gives that $b_1 = 1$ except with negligible probability as claimed.

Note that the randomness $\mathbf{r}$ used to commit to the message can be drawn from a Gaussian with any width $s_2 > 0$. However, the security proofs require s_1 to be at least $(\sqrt{nm_1} + \sqrt{nm} + t)\sqrt{nm}$ in order to hide the shifted center of the Gaussian vector, which in turns restricts the value of s_2. Additionally, the goal of this signature scheme being to allow signing on committed messages, s_2 must be chosen so that the commitment scheme is statistically hiding, which is why we take it sufficiently large anyway. We present our signature scheme for an arbitrary message length m to allow fine-tuning of the parameters depending on the specific application. Typically, an application requiring to sign only small messages of constant bit-size would be able to select a much smaller s_1 and would then yield smaller signatures.

Remark 6.2 (Stateful versus Stateless)

As discussed in Section 6.1, we choose to describe a stateful version of our construction that better suits our applications, hence the fact that our tags t are generated as $\mathsf{F}(\mathsf{st})$. The only requirements placed on F are that it must be injective with outputs in the tag space. This stems from the requirement, implicit in the security proof, that there shall be at most one signature for a given tag. It should easily be met in practice. For example, in the case of group signatures, one can proceed as in [dPLS18] and set the tags as the group members' identities. Nevertheless, if selecting such a function F proved to be difficult for some use case, we recall that a stateless version of our construction is provided in the original paper [JRS23, App. G]. In our implementation presented in Chapter 8, we instantiate F with Algorithm 8.1 sometimes referred to as the Fisher-Yates shuffle [Knu98].

6.2.2 Security Analysis

We distinguish two types of forgeries that an attacker can produce, which we treat separately for the sake of clarity. More precisely we distinguish between the cases depending on whether or not the tag $\mathsf{t}^\star$ of the forgery has been re-used from the signature queries. Combining the corresponding

theorems proves the EUF-CMA security of the signature under the M-SIS assumption. It consists in the M-SIS challenger tossing a coin and proceeding as in either Theorem 6.1 or 6.2 and aborting if the forgery does not match the coin toss. Because we only use statistical arguments, we present the proof without hybrid games and directly analyze the distributions.

Theorem 6.1 (Unforgeability Against Type ❶ Forgeries)

An adversary produces a *Type* ❶ forgery $(\mathsf{t}^\star, \mathbf{v}^\star)$ if the tag $\mathsf{t}^\star$ does not collide with the tags of the signing queries. The advantage of any PPT adversary $\mathcal{A}$ in producing a type ❶ forgery is at most

$$\mathrm{Adv}_{❶}[\mathcal{A}] \lesssim (|\mathcal{T}_w| - Q)\varepsilon_{\text{M-SIS}},$$

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound of $\text{M-SIS}_{n,d,m_1+1,q,\beta_{❶}}$ with

$$\beta_{❶} = \sqrt{1 + \left(B_1 + (\sqrt{nm_1} + \sqrt{ndk} + t)B_2 + (\sqrt{nm_1} + \sqrt{nm} + t)\sqrt{nm}\right)^2}$$

Proof (Theorem 6.1). Consider a PPT adversary $\mathcal{A}$ that produces Type ❶ forgeries for the signature scheme with advantage δ. We now construct an adversary $\mathcal{B}$ that solves the $\text{M-SIS}_{n,d,m_1+1,q,\beta_{❶}}$ problem. The adversary $\mathcal{B}$ is given $[\overline{\mathbf{A}}|\overline{\mathbf{u}}] \in R_q^{d\times m_1+1}$ as input and is asked to find $\mathbf{w} \in \mathcal{L}_q^\perp([\overline{\mathbf{A}}|\overline{\mathbf{u}}])$ such that $0 < \|\mathbf{w}\|_2 \le \beta_{❶}$.

Setup Stage: $\mathcal{B}$ first generates the cryptographic material to give to $\mathcal{A}$. We assume that the public parameters are already set, except for $\mathbf{A}, \mathbf{u}$ and $\mathbf{D}$. The adversary $\mathcal{B}$ first generates the tags $\mathsf{t}^{(1)}, \dots, \mathsf{t}^{(Q)}$ that will be used for the signing queries of $\mathcal{A}$ by calling F and incrementing the state st. It also makes a guess $\mathsf{t}^+ \hookleftarrow U(\mathcal{T}_w \setminus \{\mathsf{t}^{(i)}; i \in [\![Q]\!]\})$ on the tag that will be used in the adversary's forgery. In particular, we assume that $Q = \mathsf{poly}(\lambda)$ is the maximum number of signing queries that $\mathcal{A}$ is able to make.

Next, $\mathcal{B}$ samples $\mathbf{S}$ from $U(S_1^{m_1\times m})$ such that $\|\mathbf{S}\|_2 \le \sqrt{nm_1} + \sqrt{nm} + t$. By Heuristic 1.1, the distribution of $\mathbf{S}$ is statistically close to $U(S_1^{m_1\times m})$ and thus efficiently sampleable. It then defines $\mathbf{A} = \overline{\mathbf{A}}$, $\mathbf{u} = \overline{\mathbf{u}}$ and $\mathbf{D} = \mathbf{AS} \bmod qR$. This completes the public parameters pp. Then, $\mathcal{B}$ samples $\mathbf{R} \hookleftarrow U(S^{m_1\times dk})$ such that $\|\mathbf{R}\|_2 \le \sqrt{nm_1} + \sqrt{ndk} + t$ and defines $\mathbf{B} = \mathbf{AR} + \mathsf{t}^+\mathbf{G} \bmod qR$. The adversary $\mathcal{B}$ then forms $\mathsf{pk} = \mathbf{B}$. From these matrices, we can define $\mathbf{A}_\mathsf{t}$ for any tag $\mathsf{t} \in \mathcal{T}_w$ by

$$\mathbf{A}_\mathsf{t} = \left[\mathbf{A}|\mathsf{t}\mathbf{G} - \mathbf{B}\right] = \left[\mathbf{A}|(\mathsf{t} - \mathsf{t}^+)\mathbf{G} - \mathbf{AR}\right]. \tag{6.2}$$

Since t^+ does not collide with the tags $\mathsf{t}^{(1)}, \dots, \mathsf{t}^{(Q)}$ that will be used to answer the signing queries, we have $\mathsf{t}^{(i)} - \mathsf{t}^+ \in S_1$ and thus $\mathsf{t}^{(i)} - \mathsf{t}^+ \bmod qR \in R_q^\times$ by Lemma 1.4 or more precisely Remark 1.2. The matrices $\mathbf{A}_{\mathsf{t}^{(i)}}$ thus have the adequate form to sample preimages using Algorithm 4.1. Finally, $\mathcal{B}$ sends $(\mathsf{pk}, \mathsf{pp})$ to $\mathcal{A}$.

Query Stage: At the i-th signature query, $\mathcal{A}$ provides $\mathcal{B}$ with a message $\mathbf{m}^{(i)} \in T_1^m$. $\mathcal{B}$ can then faithfully run Algorithm 6.3 using the carefully crafted key material, and the tag $\mathsf{t}^{(i)}$. More precisely, it computes the message commitment $\mathbf{c} = \mathbf{Ar}^{(i)} + \mathbf{Dm}^{(i)} \bmod q$ for a fresh randomness $\mathbf{r}^{(i)} \hookleftarrow \mathcal{D}_{R^{m_1},s_2}$, and samples

$$\mathbf{v}^{(i)} = \mathsf{MP\text{-}Sampler}(\mathbf{R}; \mathbf{A}, \mathbf{u} + \mathbf{c}, (\mathsf{t}^{(i)} - \mathsf{t}^+)\mathbf{I}_d, s, r\sqrt{5}) - \begin{bmatrix}\mathbf{r}^{(i)} \\ \mathbf{0}_{dk}\end{bmatrix}.$$

Note that $\mathbf{v}^{(i)}$ is correctly distributed and passes verification just like regular signatures by Lemma 6.1. The signature given to $\mathcal{A}$ is $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}^{(i)})$.

Forgery Stage: After at most Q queries, the adversary returns a forgery $\mathsf{sig}^\star = (\mathsf{t}^\star, \mathbf{v}^\star)$ on a new message $\mathbf{m}^\star$ that passes verification. If $\mathcal{A}$ fails to produce such a forgery, $\mathcal{B}$ aborts. We call this event Abort_1. We now condition on $\neg\mathsf{Abort}_1$. At this point, $\mathcal{B}$ aborts if $\mathsf{t}^\star \neq \mathsf{t}^+$. We call this event Abort_2 and further condition on $\neg\mathsf{Abort}_2$. Then, the guess was correct and therefore the contribution of $\mathbf{G}$ in $\mathbf{A}_{\mathsf{t}^\star}$ vanishes. Since the forgery passes verification, we have $\mathbf{A}_{\mathsf{t}^\star}\mathbf{v}^\star = \mathbf{u} + \mathbf{Dm}^\star \bmod qR$. Using the definition of the cryptographic material from the setup stage, it can be written as

$$\left[\overline{\mathbf{A}}| - \overline{\mathbf{AR}}\right]\mathbf{v}^\star = \overline{\mathbf{u}} + \overline{\mathbf{A}}\mathbf{S}\mathbf{m}^\star \bmod qR.$$

This means that

$$\mathbf{w} = \begin{bmatrix} [\mathbf{I}_{m_1} | -\mathbf{R}]\mathbf{v}^\star - \mathbf{S}\mathbf{m}^\star \\ -1 \end{bmatrix} \in R^{m_1+1}$$

is in $\mathcal{L}_q^\perp([\overline{\mathbf{A}}|\overline{\mathbf{u}}])$. The adversary $\mathcal{B}$ thus returns $\mathbf{w}$ as a solution for M-SIS$_{n,d,m_1+1,q,\beta_{❶}}$.

Advantage: We now analyze the advantage of $\mathcal{B}$. We first look at the distribution of $(\mathsf{pk}, \mathsf{pp})$. Recall that $\mathbf{R}$ and $\mathbf{S}$ are negligibly close to the uniform over S_1 by Heuristic 1.1. Since $m_1 \geq d \log_3 q + \omega(\log_3 n)$, it holds by Lemma 1.16 that

$$\Delta((\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{R} \bmod qR), (\overline{\mathbf{A}}, U(R_q^{d\times dk}))) \leq \frac{dk}{2}\sqrt{\left(1 + q^d/3^{m_1}\right)^n - 1} + \mathsf{negl}(\lambda) \leq \mathsf{negl}(\lambda),$$
$$\Delta((\overline{\mathbf{A}}, \overline{\mathbf{A}}\mathbf{S} \bmod qR), (\overline{\mathbf{A}}, U(R_q^{d\times m}))) \leq \frac{m}{2}\sqrt{\left(1 + q^d/3^{m_1}\right)^n - 1} + \mathsf{negl}(\lambda) \leq \mathsf{negl}(\lambda),$$

Additionally, since $\overline{\mathbf{A}}$, $\mathbf{R}$ are independent of $\mathfrak{t}^+\mathbf{G}$, it holds that $\Delta(\mathbf{B}, \overline{\mathbf{A}}\mathbf{R} \bmod qR) \leq dk\sqrt{\left(1 + q^d/3^{m_1}\right)^n - 1} + 2\mathsf{negl}(\lambda) \leq \mathsf{negl}(\lambda)$ (by the triangle inequality). The signatures that are given to $\mathcal{A}$ in the query stage are distributed according to the legitimate distribution. This means that

$$\mathbb{P}[\neg\mathsf{Abort}_1] \geq \delta - \mathsf{negl}(\lambda). \tag{6.3}$$

As the guess $\mathfrak{t}^+$ is independent of $\mathcal{A}$'s view, we directly have

$$\mathbb{P}[\neg\mathsf{Abort}_2 | \neg\mathsf{Abort}_1] = \frac{1}{|\mathcal{T}_w| - Q}. \tag{6.4}$$

We now analyze the solution provided by $\mathcal{B}$. We have to show it is non-zero and have ℓ_2 norm at most $\beta_{❶}$. Since the last coefficient of $\mathbf{w}$ is -1, we directly get that $\mathbf{w} \neq \mathbf{0}$. Because the forgery passes verification, it holds that

$$\begin{aligned}
\|\mathbf{w}\|_2^2 &= 1 + \left\|[\mathbf{I}_{m_1} | -\mathbf{R}]\mathbf{v}^\star - \mathbf{S}\mathbf{m}^\star\right\|_2^2 \\
&\leq 1 + \left(B_1 + \|\mathbf{R}\|_2 B_2 + \|\mathbf{S}\|_2 \sqrt{nm}\right)^2 \\
&\leq 1 + \left(B_1 + (\sqrt{nm_1} + \sqrt{ndk} + t)B_2 + (\sqrt{nm_1} + \sqrt{nm} + t)\sqrt{nm}\right)^2 \\
&= \beta_{❶}^2.
\end{aligned}$$

where the last inequality follows from the bounds imposed on the spectral norms. Hence, $\mathbf{w}$ is a valid solution. Combining Equations (6.3) and (6.4) by the probability chain rule, we get

$$\varepsilon_{\text{M-SIS}} \geq \mathrm{Adv}_{\text{M-SIS}}[\mathcal{B}] \geq (\delta - \mathsf{negl}(\lambda)) \cdot \frac{1}{|\mathcal{T}_w| - Q} \approx \frac{\delta}{|\mathcal{T}_w| - Q},$$

as claimed.

Theorem 6.2 (Unforgeability Against Type ❷ Forgeries)

An adversary produces a *Type* ❷ forgery $(\mathfrak{t}^\star, \mathbf{v}^\star)$ if the tag $\mathfrak{t}^\star$ is re-used from some $i^\star$-th signing query $(\mathfrak{t}^{(i^\star)}, \mathbf{v}^{(i^\star)})$. The advantage of any PPT adversary $\mathcal{A}$ in producing a type ❷ forgery is at most

$$\mathrm{Adv}_{❷}[\mathcal{A}] \lesssim \left(\varepsilon_{\text{M-SIS}} \cdot Q \cdot e^{\alpha^\star \pi}\right)^{\frac{\alpha^\star - 1}{\alpha^\star}}$$

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound of M-SIS$_{n,d,m_1,q,\beta_{❷}}$ with

$$\beta_{❷} = 2B_1 + 2B_2(\sqrt{nm_1} + \sqrt{ndk} + t) + (\sqrt{nm_1} + \sqrt{nm} + t)\sqrt{nm},$$

and where $\alpha^\star$ is defined by $\alpha^\star = 1 + \sqrt{-\log_2(\mathrm{Adv}_{❷}[\mathcal{A}])/(\pi \log_2 e)}$.

Proof (Theorem 6.2). Consider a PPT adversary $\mathcal{A}$ that can produce a Type ❷ forgery for the signature scheme with advantage δ. We now construct an adversary $\mathcal{B}$ that solves the

M-SIS$_{n,d,m_1,q,\beta_{❷}}$ problem. The adversary $\mathcal{B}$ is given $\overline{\mathbf{A}} \in R_q^{d\times m_1}$ as input and is asked to find $\mathbf{w} \in \mathcal{L}_q^{\perp}(\overline{\mathbf{A}})$ such that $0 < \|\mathbf{w}\|_2 \leq \beta_{❷}$.

Setup Stage: The adversary $\mathcal{B}$ first generates the tags $\mathfrak{t}^{(1)}, \ldots, \mathfrak{t}^{(Q)}$ that will be used for the signing queries of $\mathcal{A}$ by calling F and incrementing the state st. Note that since F is injective, as mentioned in Remark 6.2, there is no collision among the tags. The adversary $\mathcal{B}$ makes a guess $i^+ \hookleftarrow U(\llbracket Q \rrbracket)$ on the index of the tag that will be re-used by $\mathcal{A}$ in the forgery stage. Then, $\mathcal{B}$ samples $\mathbf{R} \hookleftarrow U(S_1^{m_1\times dk})$, and $\mathbf{S}$ from $U(S_1^{m_1\times m})$ such that the spectral bounds from Heuristic 1.1 are verified as in the proof of Theorem 6.1. It then defines $\mathbf{A} = \overline{\mathbf{A}}$ and

$$\mathbf{B} = \mathbf{AR} + \mathfrak{t}^{(i^+)}\mathbf{G} \bmod qR, \quad \text{and} \quad \mathbf{D} = \mathbf{AS} \bmod qR.$$

The adversary $\mathcal{B}$ samples $\mathbf{v}$ from $\mathcal{D}_{R^{m_1+dk},s}$, $\mathbf{r}_0$ from $\mathcal{D}_{R^{m_1},s_2}$, and defines

$$\mathbf{u} = \mathbf{A}_{\mathfrak{t}^{(i^+)}}\left(\mathbf{v} - \begin{bmatrix}\mathbf{r}_0\\ \mathbf{0}_{dk}\end{bmatrix}\right) \bmod qR.$$

Note that for all $i \in \llbracket Q \rrbracket$, we have

$$\mathbf{A}_{\mathfrak{t}^{(i)}} = [\mathbf{A}|(\mathfrak{t}^{(i)} - \mathfrak{t}^{(i^+)})\mathbf{G} - \mathbf{AR}],$$

where the contribution in $\mathbf{G}$ vanishes only for $i = i^+$, as there is no collision. The adversary $\mathcal{B}$ thus defines the public parameters pp along with $\mathbf{A}, \mathbf{u}, \mathbf{D}$, and $\mathsf{pk} = \mathbf{B}$, before sending both to $\mathcal{A}$.

Query Stage: We distinguish the queries for $i \neq i^+$ from the i^+-th query. For $i \neq i^+$, the effective tag is $\mathfrak{t}^{(i)} - \mathfrak{t}^{(i^+)} \in R_q^{\times}$, which means we can handle these queries as in the proof of Theorem 6.1 by running the legitimate signing with the updated tag.

Now consider the i^+-th query. In this case, $\mathcal{B}$ simply computes $\mathbf{v}^{(i^+)} = \mathbf{v} - \begin{bmatrix}\mathbf{r}_0 - \mathbf{Sm}^{(i^+)}\\ \mathbf{0}_{dk}\end{bmatrix}$ and gives $\mathsf{sig}^{(i^+)} = (\mathfrak{t}^{(i^+)}, \mathbf{v}^{(i^+)})$ to $\mathcal{A}$. We analyze later the distribution of $\mathbf{v}^{(i^+)}$, but notice that the verification equation is verified because of the definition of $\mathbf{u}$.

$$\begin{aligned}\mathbf{A}_{\mathfrak{t}^{(i^+)}}\mathbf{v}^{(i^+)} &= \mathbf{u} + \mathbf{A}_{\mathfrak{t}^{(i^+)}}\begin{bmatrix}\mathbf{Sm}^{(i^+)}\\ \mathbf{0}_{dk}\end{bmatrix} \bmod qR\\ &= \mathbf{u} + \mathbf{ASm}^{(i^+)} \bmod qR\\ &= \mathbf{u} + \mathbf{Dm}^{(i^+)} \bmod qR.\end{aligned}$$

Forgery Stage: After at most Q queries, $\mathcal{A}$ outputs a Type ❷ forgery $(\mathfrak{t}^\star, \mathbf{v}^\star)$ on a new message $\mathbf{m}^\star$. If $\mathcal{A}$ fails to output a valid forgery, event that we denote by Abort_1, then $\mathcal{B}$ aborts. We now condition on $\neg\mathsf{Abort}_1$. At this point, $\mathcal{B}$ checks its guess on i^+ and aborts if $\mathfrak{t}^\star \neq \mathfrak{t}^{(i^+)}$. We denote this event Abort_2, and further condition on $\neg\mathsf{Abort}_2$. It holds that

$$\mathbf{A}_{\mathfrak{t}^{(i^+)}}\mathbf{v}^{(i^+)} - \mathbf{Dm}^{(i^+)} = \mathbf{u} \bmod qR = \mathbf{A}_{\mathfrak{t}^\star}\mathbf{v}^\star - \mathbf{Dm}^\star \bmod qR.$$

Since $\mathbf{A}_{\mathfrak{t}^\star} = \mathbf{A}_{\mathfrak{t}^{(i^+)}} = \overline{\mathbf{A}}[\mathbf{I}_{m_1}| - \mathbf{R}]$, it holds that

$$\overline{\mathbf{A}}\left([\mathbf{I}_{m_1}| - \mathbf{R}](\mathbf{v}^{(i^+)} - \mathbf{v}^\star) - \mathbf{S}(\mathbf{m}^{(i^+)} - \mathbf{m}^\star)\right) = \mathbf{0} \bmod qR.$$

As a result, $\mathcal{B}$ forms the vector

$$\mathbf{w} = [\mathbf{I}_{m_1}| - \mathbf{R}](\mathbf{v}^{(i^+)} - \mathbf{v}^\star) - \mathbf{S}(\mathbf{m}^{(i^+)} - \mathbf{m}^\star) \in R^{m_1},$$

which is in $\mathcal{L}_q^{\perp}(\overline{\mathbf{A}})$, and returns it as a solution for M-SIS.

Advantage: We now analyze the advantage of $\mathcal{B}$. We first focus on the distribution of $(\mathsf{pk}, \mathsf{pp})$. Using the same argument based on Lemma 1.16 and Heuristic 1.1, it holds that $\mathbf{B}, \mathbf{D}$ are with negligible statistical distance of the uniform. Then, let us analyze the distribution of $\mathbf{u}$. Define $\mathbf{A}' = [\mathbf{A}| - \mathbf{AR}] \bmod qR$. By construction, we have $\mathbf{u} = \mathbf{A}'\mathbf{v}' \bmod q$, where

$\mathbf{v}_1'$ is within statistical distance $3\varepsilon/2$ of $\mathcal{D}_{R^{m_1},s_1}$ by Lemma 1.12, and $\mathbf{v}_2'$ is distributed as $\mathcal{D}_{R^{dk},s}$. Using the regularity lemma from [LPR13b, Thm. 7.4] (adapted to Gaussians in the coefficient embedding τ, which for power-of-two cyclotomics only differs by $\sqrt{n}$ in the width), it holds that $\mathbf{u}$ is statistically close to uniform if $s, s_1 \geq 2\sqrt{n}q^{\frac{d+2/n}{m_1+dk}}$. Because $m_1 + dk \geq d(\log_2 q/\log_2 3 + k) + f(\lambda)$, the result holds whenever $s, s_1 \geq 3^{1+2/n} \cdot 2\sqrt{n}$ which is subsumed by the setting in $\mathsf{SEP.Setup}$. We thus have $\Delta(\mathbf{u}, U(R_q^d)) \leq \mathsf{negl}(\lambda)$ as desired.

We now analyze the distribution of the signature that are produced by $\mathcal{B}$. For the i-th query with $i \neq i^+$, the signature is distributed exactly as in the legitimate algorithm. At the i^+-th signing query, the vector $\mathbf{v}_1^{(i^+)}$ is within statistical distance $3\varepsilon/2$ of $\mathcal{D}_{R^{m_1},s_1,\mathbf{z}^+}$, where $\mathbf{z}^+ = \mathbf{S}\mathbf{m}^{(i^+)}$. It also holds that

$$\left\|\mathbf{z}^+\right\|_2 = \left\|\mathbf{S}\mathbf{m}^{(i^+)}\right\|_2 \leq (\sqrt{nm_1} + \sqrt{nm} + t)\sqrt{nm}.$$

We now measure the closeness of $\mathbf{v}^{(i^+)}$ to the real distribution by using the Rényi divergence of order α for a free parameter $\alpha > 1$. By Lemma 1.11 it holds that

$$\mathrm{RD}_\alpha(\mathcal{D}_{R^{m_1},s_1} \| \mathcal{D}_{R^{m_1},s_1,\mathbf{z}^+}) \leq \exp\left(\frac{\alpha\pi}{s_1^2}\left\|\mathbf{z}^+\right\|_2^2\right) \leq e^{\alpha\pi},$$

as $s_1 \geq (\sqrt{nm_1} + \sqrt{nm} + t)\sqrt{nm}$. Combining the probabilities for the distribution of the keys and the signatures, and by the probability preservation properties of the statistical distance and Rényi divergence of Lemma 1.7 and 1.8, we have

$$\mathbb{P}[\neg\mathsf{Abort}_1] \geq e^{-\alpha\pi}(\delta - \mathsf{negl}(\lambda))^{\alpha/(\alpha-1)} \geq e^{-\alpha\pi}\delta^{\alpha/(\alpha-1)} - \mathsf{negl}(\lambda). \tag{6.5}$$

We then optimize over α. The maximum value of the right-hand side is attained for $\alpha^\star = 1 + \sqrt{-\log_2(\delta)/(\pi \log_2 e)}$. Further, since the guess i^+ is independent of $\mathcal{A}$'s view it holds that

$$\mathbb{P}[\neg\mathsf{Abort}_2 | \neg\mathsf{Abort}_1] = \frac{1}{Q}. \tag{6.6}$$

We now analyze the solution constructed by $\mathcal{B}$. We have to show it is non-zero and has norm at most $\beta_{\mathbf{②}}$. We first focus on the former. We essentially show that for $\mathcal{A}$ to ensure $\mathbf{w} = \mathbf{0}$, it must predict at least one column $\mathbf{s}$ of $\mathbf{S}$ as $\mathbf{m}^\star \neq \mathbf{m}^{(i^+)}$. So we have

$$\begin{aligned}
\mathbb{P}[\mathbf{w} = \mathbf{0}] &\leq \mathbb{P}_{\mathbf{s}}[\mathbf{s} = \mathbf{s}^\star : \mathbf{s}^\star \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s} \bmod qR, \mathbf{v}_1^{(i^+)})] \\
&\leq \sqrt{\mathbb{P}_{\mathbf{s}}[\mathbf{s} = \mathbf{s}^\star : \mathbf{s}^\star \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s} \bmod qR)] \cdot \mathrm{RD}_2(\mathcal{D}_{R^{m_1},s_1,\mathbf{S}\mathbf{m}^{(i^+)}} \| \mathcal{D}_{R^{m_1},s_1})} + 3\varepsilon/2 \\
&\leq \frac{1+\varepsilon}{1-\varepsilon} e^\pi \sqrt{\mathbb{P}_{\mathbf{s}}[\mathbf{s} = \mathbf{s}^\star : \mathbf{s}^\star \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s} \bmod qR)]} + 3\varepsilon/2
\end{aligned}$$

where the last inequality stems from Lemma 1.11 as $s_1 \geq s \geq \eta_\varepsilon(R^{m_1})$[a]. Then, since $nm_1 \log_2 3 \geq nd \log_2 q + \omega(n \log_3 n)$ and that $\mathbf{A}\mathbf{s} \bmod qR$ can take $2^{nd \log_2 q}$ values, we get that $\mathbf{s}$ given $\mathbf{A}\mathbf{s} \bmod qR$ contains at least $nm_1 \log_2 3 - nd \log_2 q \geq \omega(n \log_3 n)$ bits of entropy (e.g., [DORS08, Lem. 2.2]). Thence, $\mathbf{w} \neq \mathbf{0}$ except with negligible probability. Finally, it holds that

$$\begin{aligned}
\|\mathbf{w}\|_2 &\leq 2B_1 + \|\mathbf{R}\|_2 \cdot 2B_2 + \|\mathbf{S}\|_2 \sqrt{nm} \\
&\leq 2B_1 + 2B_2(\sqrt{nm_1} + \sqrt{ndk} + t) + (\sqrt{nm_1} + \sqrt{nm} + t)\sqrt{nm} \\
&= \beta_{\mathbf{②}}.
\end{aligned}$$

where the first inequality uses Lemma 1.21 for $\mathbf{v}^{(i^+)}$ which is verified except with probability $3\varepsilon/2 + 2^{-2nm_1} + 2^{-2ndk} = \mathsf{negl}(\lambda)$.

$$\mathbb{P}[\mathbf{w} \text{ valid solution} | \neg\mathsf{Abort}_1 \wedge \neg\mathsf{Abort}_2] = 1 - \mathsf{negl}(\lambda). \tag{6.7}$$

Combining Equations (6.5), (6.6) and (6.7) by the probability chain rule, we get

$$\varepsilon_{\text{M-SIS}} \geq \mathrm{Adv}_{\text{M-SIS}}[\mathcal{B}] \geq (\delta^{\alpha^\star/(\alpha^\star-1)} e^{-\alpha^\star\pi} - \mathsf{negl}(\lambda) \cdot \frac{1}{Q} \cdot (1 - \mathsf{negl}(\lambda)) \approx \frac{\delta^{\alpha^\star/(\alpha^\star-1)} e^{-\alpha^\star\pi}}{Q},$$

as desired. Note that the parameters and the behavior of $\mathcal{B}$ do not depend on the order α that is used to compute the advantage bound. As such, $\alpha^\star$ can indeed depend on the forger's advantage δ.

[a] Note that the Rényi divergence is taken in the opposite direction than before, hence the presence of the factor $(1+\varepsilon)/(1-\varepsilon)$.

6.2.3 Interface with Commitments and Zero-Knowledge Proofs

Recalling the protocols (**P1**) and (**P2**) from Section 6.1, the signature designed in this section should fit the requirements of the protocols. Protocol (**P1**) requires the ability to sign hidden messages. At a high level, this is ensured by the implicit commitment $\mathbf{c} = \mathbf{Ar} + \mathbf{Dm} \bmod qR$. The randomness $\mathbf{r}$ can be selected so as to hide $\mathbf{m}$. Observe that this step intervenes before even needing the secret key sk of the signer. As a result, it can be delegated to the user who wants to have $\mathbf{m}$ signed. The one thing to note here is that in the proof of Theorem 6.1, the challenger (or signer) needs to control part of $\mathbf{r}$ ($\mathbf{r}_0$ in the proof) to answer the i^+-th query. This can be done by splitting the randomness into $\mathbf{r} = \mathbf{r}_u + \mathbf{r}_s$ where $\mathbf{r}_u$ would be selected by the user to hide $\mathbf{m}$ in $\mathbf{c}_u = \mathbf{Ar}_u + \mathbf{Dm}$, and the signer would re-randomize the commitment into $\mathbf{c} = \mathbf{Ar}_s + \mathbf{c}_u$. This adjustment would still allow the security argument to go through, while giving the user the possibility of hiding the message.

Then, Protocol (**P2**) requires the ability to prove knowledge of a signature and the corresponding message in zero-knowledge. This means that we need to prove the statement

$$\mathsf{SEP.Verify}(\mathsf{pk}, \mathbf{m}, (\mathsf{t}, \mathbf{v})) = 1, \quad \text{and} \quad \mathbf{m} \in T_1^m.$$

It then consists in a quadratic relation (because of the term in $\mathsf{t}\mathbf{G}\mathbf{v}_2$) with norm checks. Our relation is then part of the language encompassed by the zero-knowledge framework of Lyubashevsky et al. [LNP22]. We elaborate further on these protocols and zero-knowledge arguments in Chapter 7.

6.2.4 Performance Gains

We now describe the parameter selection. We aim at finding concrete parameters reaching $\lambda = 128$ bits of security and for $Q = 2^{32}$, representing the number of issued sigantures. Looking ahead to Chapter 7, we consider signatures of 10 attibutes to which is added a secret key of size $2d$ as part of the message. We thus set $m = 10 + 2d$.

In our case, we only have to estimate the hardness of the two M-SIS assumptions as the parameters set in Setup are chosen so that all the statistical arguments go through. The issue is that Theorems 6.1 and 6.2 feature a security loss between the unforgeability and the underlying assumption. The factors $|\mathcal{T}_w| - Q$ and Q of type ❶ and type ❷ forgeries respectively come from having to guess the tag $\mathsf{t}^\star$ in the security reduction to obtain a fully adaptive security. We note however that in the case of type ❷, the reduction loss also involves an order $\alpha^\star$ which corresponds to the Rényi divergence argument. Aiming for $\mathrm{Adv}_{❷}[\mathcal{A}] \leq 2^{-\lambda}$ gives $\alpha^\star \approx 6.31$. As a result, one should ensure that $\varepsilon_{\text{M-SIS}} \leq 2^{-213}$ to indeed obtain $\mathrm{Adv}_{❷}[\mathcal{A}] \leq 2^{-\lambda}$. As a result, there is a security loss of 85 bits between the computational assumption and the actual signature security. With these security targets for the M-SIS assumptions, we follow the methodology from Chapter 9 to select appropriate parameters. Table 6.2 gives an example parameter set meeting the security requirement, while optimizing the sizes.

We already showcase drastic performance improvements over the signature with efficient protocols of [LLM+16]. The sizes we obtain are yet still far from being practical for real-world applications. We thus now focus on finding optimizations of the scheme we presented with the hope of reaching much better performance.

6.3 Bypassing Double Trapdoors: Partial Trapdoor Switching

The main inefficiency in the construction of Section 6.2 is the use of statistical security arguments that requires to increase the number of columns of $\mathbf{A}$ to roughly $d \log_3 q + \omega(\log_3 n)$ (to use Lemma 1.16). To get rid of this complexity, one could leverage a computational assumption, such as M-LWE, to argue the same regularity. It however comes a few subtleties which invalidate the security proof, at least in the current state of the scheme.

6.3.1 The Double Trapdoor Problem

The core idea of the security proofs from Section 6.2 is to change the public key so as to have a valid trapdoor for all tags but one, which we denoted by t^+. This is concretely done by replacing $\mathbf{B} = \mathbf{AR}$ in the public key by $\mathbf{B} = \mathbf{AR} + \mathsf{t}^+\mathbf{G}$. As a result, for this new public key, we have $\mathbf{A}_\mathsf{t} = [\mathbf{A}|(\mathsf{t} - \mathsf{t}^+)\mathbf{G} - \mathbf{AR}]$ where the gadget vanishes for $\mathsf{t} = \mathsf{t}^+$. We have seen that statistical arguments can handle this change simply by arguing that $\mathbf{AR}$ is close to a uniform distribution, and so is $\mathbf{AR} + \mathsf{t}^+\mathbf{G}$. Let us now investigate how this argument would be transposed in the computational setting.

Symbol	Description	Value
	Signature Parameters	
λ	Security parameter	128
n	Ring degree	128
d	M-SIS Module rank	11
(k, b)	Gadget length and base	$(42, 2)$
m_1	First trapdoor rank	615
m	Message length	$10 + 2d = 32$
q	Modulus	$2^{42} - 215$
κ	Number of prime ideal factors of qR	4
s	Preimage sampling width	5346
s_1	Top width $(\sqrt{s^2 + s_2^2})$	25058
s_2	Commitment randomness width	24481
w	Hamming weight of tags	6
Q	Maximal number of signature queries	2^{32}
B_1	First verification bound	6321972
B_2	Second verification bound	1299939
	Efficiency Estimates	
$\lvert\mathsf{sk}\rvert$	Secret key size (KB)	8879 KB
$\lvert\mathsf{pk}\rvert$	Public key size (KB)$^{(*)}$	3335 KB
$\lvert\mathsf{sig}\rvert$	Signature size (KB)	289 KB
	Security Estimates	
$\lambda_{❶}/\lambda_{❷}$	Security targets for $\widetilde{\text{M-SIS}}$ (type ❶/❷, Theorems 6.1/6.2)	159/213
$\text{BKZ}_{❶}/\text{BKZ}_{❷}$	BKZ blocksize for $\widetilde{\text{M-SIS}}$ (type ❶/❷)	752/686
$\lambda^\star_{❶}/\lambda^\star_{❷}$	Reached $\widetilde{\text{M-SIS}}$ classical security (type ❶/❷)	235/216

Table 6.2: Suggested parameter set for the signature of Section 6.2.
$^{(*)}$ The public key size only contains $\mathbf{B}$. The other public elements $\mathbf{A}, \mathbf{D}, \mathbf{u}$ are stored via a 32-byte seed in the public parameters and shared between signers.

For the sake of presentation, assume that $\mathbf{A} = [\mathbf{I}_d | \mathbf{A}']$ so that $(\mathbf{A}', \mathbf{AR} \bmod qR)$ is an M-LWE instance according to Definition 1.15. We then show that this distribution is indistinguishable from that of $(\mathbf{A}', \mathbf{AR} + \mathfrak{t}^+\mathbf{G} \bmod qR)$. For that we first change $\mathbf{AR} \bmod qR$ into a uniform matrix $\mathbf{B}$, which is valid under the decisional version of M-LWE. Then, we change $\mathbf{B}$ into $\mathbf{B}' + \mathfrak{t}^+\mathbf{G} \bmod qR$ for another uniform $\mathbf{B}'$. Since $\mathbf{B}'$ is independent of $\mathfrak{t}^+\mathbf{G}$, the distributions are identical. Finally, we switch back from $\mathbf{B}'$ uniform to some $\mathbf{AR}$, resulting in $\mathbf{AR} + \mathfrak{t}^+\mathbf{G} \bmod qR$ which is again argued by the M-LWE assumption. We then obtain that the advantage of distinguishing the two distributions is bounded by $2\varepsilon_{\text{M-LWE}}$.

Although this is perfectly satisfactory when looking at the distribution of the public key, the security proof considers the distribution of $(\mathsf{pp}, \mathsf{pk}, (\mathsf{sig}_i)_{i \in [\![Q]\!]})$. In particular, the challenger must answer signing queries in each of the hybrid distributions of the previous paragraph, including the ones where $\mathbf{B}$ is uniform. In the latter, the challenger no longer owns a trapdoor enabling polynomial time signature generation, making them unable to answer signing queries and invalidating this security argument on the public key.

Remark 6.3 (Unplayable Games)

The unforgeability game in this hybrid distribution is called "unplayable" as it cannot be "played" in polynomial time. Such games do not cause issues if the comparisons with adjacent games are *statistical*. They do when a computational assumption is involved. Indeed, assume

that games G_1 and G_2 are indistinguishable under the decisional assumption P and are such that one of G_1 and G_2 is unplayable. The indistinguishability argument would take an instance X of P and transform it so as to match G_1 or G_2 depending on the hidden distribution of X. The solution to instance X would then use a solver to a game that can only be played in super-polynomial time, which would then not yield a polynomial time adversary for P.

Hash-and-sign signatures typically bypass this problem by simulating all of the signatures by reprogramming the random oracle queries (by the programmability propery in Definition 1.19), as depicted in Part II. As advanced signatures like the one from Section 6.2 are proven secure in the standard model, we do not have the freedom of the random oracle to simulate signatures. Actually, we are only able to simulate *one* signature by reprogramming the public syndrome $\mathbf{u}$.

This issue has been identified in other standard model signatures, e.g., [dPLS18, LNPS21], where they proposed a solution based on double trapdoors. More precisely, the problem was solved by artificially extending the public key so as to introduce a *second* trapdoor. In the case of MP trapdoors, this concretely means using matrices of the form $\mathbf{A}_\mathfrak{t} = [\mathbf{A}|\mathfrak{t}\mathbf{G} - \mathbf{A}\mathbf{R}|\mathbf{A}_3] \in R_q^{2d+2kd}$, where $\mathbf{A}_3$ is only used to embed a second trapdoor $\mathbf{R}'$ in the proof to sample preimages in the intermediate games. Before changing $\mathbf{AR}$ into a uniform $\mathbf{B}$, one would first change the uniform $\mathbf{A}_3$ into $\mathbf{G} + \mathbf{A}'_3$ and then $\mathbf{G} - \mathbf{AR}'$. At this point, the challenger is able to answer queries using either $\mathbf{R}$ or $\mathbf{R}'$. Then, changing $\mathbf{AR}$ by a uniform $\mathbf{B}$ does not cause the above problem as $\mathbf{R}'$ can still be used to generate valid signatures. One can therefore safely hide the tag guess $\mathfrak{t}^+$ and do the changes in reverse order to arrive to a matrix $\mathbf{A}_\mathfrak{t} = [\mathbf{A}|(\mathfrak{t} - \mathfrak{t}^+)\mathbf{G} - \mathbf{AR}|\mathbf{A}_3]$.

Although this solution has the merit of making the security proof work, one must almost double the dimension of the signatures because of it, which is quite frustrating. This peculiarity in the security proof was already questioned in prior works such as [dPLS18]. In the recent paper of Bootle et al. [BLNS23b] the authors propose to discard the second trapdoor and give a heuristic instantiation of their construction, which is in turn not supported by a security proof.

6.3.2 Trapdoor Switching Lemma

We significantly improve over this double trapdoors approach by presenting a new strategy that does not need a full extra trapdoor, but only part of one. Using this extra partial trapdoor slot in a hybrid argument allows us to carry the same security proof (namely moving from $\mathbf{AR}$ to $\mathbf{AR} + \mathfrak{t}^+\mathbf{G}$) to go through in a much more compact way. We present it for the elliptic sampler from Algorithm 4.5 because it is the one we use in Section 6.4, but it also applies to the original MP-Sampler [MP12].

The idea is as follows. The gadget matrix $\mathbf{G} = \mathbf{I}_d \otimes \mathbf{g}^T$ can be written as $[\mathbf{e}_1 \otimes \mathbf{g}^T | \dots | \mathbf{e}_d \otimes \mathbf{g}^T]$, where $\mathbf{e}_j$ is the j-th canonical vector of $\mathbb{R}^d$. For clarity, we define $\mathbf{G}_j = \mathbf{e}_j \otimes \mathbf{g}^T \in R^{d\times k}$. Assume we have a matrix of the form $\overline{\mathbf{A}} = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}|\mathfrak{t}_{d+1}\mathbf{G}_j - \mathbf{AR}_{d+1}]$ with $\mathbf{T} = \mathrm{diag}(\mathfrak{t}_1, \dots, \mathfrak{t}_d)$. To sample a preimage of $\mathbf{u}$ by $\overline{\mathbf{A}}$, we could proceed in two ways that we show are statistically close. The first way is to sample $\mathbf{v}_3$ from $\mathcal{D}_{R^k, s_2}$ and then use EllipticSampler (Algorithm 4.5) with the trapdoor $\mathbf{R}$ to find a preimage of $\mathbf{u} - (\mathfrak{t}_{d+1}\mathbf{G}_j - \mathbf{AR}_{d+1})\mathbf{v}_3$. The second way is to essentially exchange the j-th block of $\mathbf{TG} - \mathbf{AR}$, that is, the columns $jk+1, \dots, (j+1)k$, by the final block $\mathfrak{t}_{d+1}\mathbf{G}_j - \mathbf{AR}_{d+1}$. Concretely, one samples $\mathbf{v}_{2,j}$ from $\mathcal{D}_{R^k, s_2}$ and then samples $[\mathbf{v}_1, \mathbf{v}_{2,1}, \dots, \mathbf{v}_{2,j-1}, \mathbf{v}_3, \mathbf{v}_{2,j+1}, \dots, \mathbf{v}_{2,d}]$ from EllipticSampler with the trapdoor $[\mathbf{R}_1| \dots |\mathbf{R}_{j-1}|\mathbf{R}_{d+1}|\mathbf{R}_{j+1}| \dots |\mathbf{R}_d]$ on the syndrome $\mathbf{u} - (\mathfrak{t}_j\mathbf{G}_j - \mathbf{AR}_j)\mathbf{v}_{2,j}$, and with tag $\mathrm{diag}(\mathfrak{t}_1, \dots, \mathfrak{t}_{j-1}, \mathfrak{t}_{d+1}, \mathfrak{t}_{j+1}, \dots, \mathfrak{t}_d)$. The point of this second case is that $\mathbf{R}_j$ is no longer needed for preimage sampling, so the unused block $\mathbf{AR}_j$ can be replaced in the public key by k random vectors without impacting the ability to answer signature queries. Those random vectors can then be replaced by $\mathfrak{t}'_j\mathbf{G}_j + \mathbf{AR}_j$ for arbitrary $\mathfrak{t}'_j$ in a second game hop. In both cases, indistinguishability between those games relies on the M-LWE assumption.

We now formalize in Lemma 6.2 the (partial) trapdoor switching for the elliptic sampler with the detailed loss it incurs.

Lemma 6.2 (Trapdoor Switching Lemma)

Let R be the ring of integers of a number field. Let d, q, b be positive integers, and $k = \lceil \log_b(\lceil (q-1)/2 \rceil + 1) \rceil$. Let $\varepsilon \in (0, 1/4)$ and $s_\mathbf{G} \geq \eta_\varepsilon(\mathbb{Z}^{ndk})\sqrt{b^2+1}$. Then let $\mathbf{A}' \in R_q^{d\times d}$, $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$, $(\mathbf{R}_j)_{j\in[\![d+1]\!]} \in (R^{2d\times k})^{d+1}$, and the partial gadget matrices $(\mathbf{G}_j)_{j\in[\![d]\!]} = (\mathbf{e}_j \otimes [1|b| \dots |b^{k-1}])_j \in (R^{d\times k})^d$. Let $(\mathfrak{t}_j)_{j\in[\![d+1]\!]} \in (R_q^\times)^{d+1}$. Let $i \in [\![d]\!]$. We define $\mathbf{G} =$

$[\mathbf{G}_1|\dots|\mathbf{G}_d]$, $\mathbf{R} = [\mathbf{R}_1|\dots|\mathbf{R}_d]$ and $\mathbf{R}_{-i}$ the matrix where the block $\mathbf{R}_i$ in $\mathbf{R}$ has been replaced by $\mathbf{R}_{d+1}$. We also call $\mathbf{T} = \mathrm{diag}(\mathfrak{t}_1, \dots, \mathfrak{t}_d)$ and $\mathbf{T}_{-i}$ the matrix $\mathbf{T}$ where the i-th diagonal entry is replaced by $\mathfrak{t}_{d+1}$. Let s_1, s_2 be two positive reals such that $s_1 \geq \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2 - 1)} \cdot \max(\|\mathbf{R}\|_2, \|\mathbf{R}_{-i}\|_2)$ and $s_2 \geq \sqrt{2s_{\mathbf{G}}^2 + \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})^2}$. Finally, fix $\mathbf{u} \in R_q^d$. We call $\overline{\mathbf{A}}$ the matrix $[\mathbf{A}|\mathbf{TG} - \mathbf{AR}|\mathfrak{t}_{d+1}\mathbf{G}_i - \mathbf{AR}_{d+1}]$ mod qR for clarity, and then define the following distributions.

$\mathcal{P}_1$: Sample $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$, $(\mathbf{v}_1, \mathbf{v}_2) \leftarrow \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u} - (\mathfrak{t}_{d+1}\mathbf{G}_i - \mathbf{AR}_{d+1})\mathbf{v}_3 \bmod qR, \mathbf{T}, s_1, s_2, s_{\mathbf{G}})$ and output $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$.

$\mathcal{P}_2$: Sample $\mathbf{v}_{2,i} \hookleftarrow \mathcal{D}_{R^k, s_2}$, $(\mathbf{v}_1, (\mathbf{v}_{2,1}, \dots, \mathbf{v}_{2,i-1}, \mathbf{v}_3, \mathbf{v}_{2,i+1}, \dots, \mathbf{v}_{2,d})) \leftarrow \mathsf{EllipticSampler}(\mathbf{R}_{-i}; \mathbf{A}', \mathbf{u} - (\mathfrak{t}_i\mathbf{G}_i - \mathbf{AR}_i)\mathbf{v}_{2,i} \bmod qR, \mathbf{T}_{-i}, s_1, s_2, s_{\mathbf{G}})$, define and output $(\mathbf{v}_1, (\mathbf{v}_{2,j})_{j\in[d]}, \mathbf{v}_3)$.

It then holds that $\mathcal{P}_1 \approx_{\delta^{-1},\delta} \mathcal{P}_2$ and $\mathcal{P}_2 \approx_{\delta^{-1},\delta} \mathcal{P}_1$ where

$$\delta = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{12d(n-1)+5} \left(\frac{1+\varepsilon/ndk}{1-\varepsilon/ndk}\right)^{2ndk} \underset{\varepsilon\to 0}{\sim} 1 + 2(12d(n-1)+7)\varepsilon$$

Proof (Lemma 6.2). We define $\mathbf{s} = [s_1\mathbf{1}_{2nd}|s_2\mathbf{1}_{ndk}]$ and $\mathbf{s}' = [\mathbf{s}|s_2\mathbf{1}_{nk}]$. We additionally define $\overline{\mathbf{A}}' = [\mathbf{A}|\mathbf{TG} - \mathbf{AR}]$ and refer to $\mathbf{v}$ as the random variable $[\mathbf{v}_1^T|\mathbf{v}_2^T|\mathbf{v}_3^T]^T$. First starting from $\mathcal{P}_1$, conditioned on $\mathbf{v}_3 = \overline{\mathbf{v}_3}$, Lemma 4.3 yields that the distribution of $(\mathbf{v}_1, \mathbf{v}_2)$ is $[\delta_1, \delta_2]$-close to $\mathcal{D}_{\mathcal{L}_q^{\mathbf{u}_3}(\overline{\mathbf{A}}'),\mathbf{s}}$ where $\mathbf{u}_3 = \mathbf{u} - (\mathfrak{t}_{d+1}\mathbf{G}_i - \mathbf{AR}_{d+1})\overline{\mathbf{v}_3}$ mod qR. However, this distribution corresponds exactly to the distribution $\mathcal{D}_{R^{d(2+k)},\mathbf{s}}$ conditioned to $\overline{\mathbf{A}}'[\mathbf{v}_1^T|\mathbf{v}_2^T]^T = \mathbf{u}_3$ mod qR. Hence, we have

$$\begin{aligned}
\mathcal{P}_1(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}) &\in [\delta_1, \delta_2] \cdot \mathcal{D}_{R^k,s_2}(\overline{\mathbf{v}_3}) \cdot \mathcal{D}_{\mathcal{L}_q^{\mathbf{u}_3}(\overline{\mathbf{A}}'),\mathbf{s}}(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}) \\
&= [\delta_1, \delta_2] \cdot \left(\mathcal{D}_{R^{d(2+k)+k},\mathbf{s}'} \middle| \overline{\mathbf{A}}\mathbf{v} = \mathbf{u} \bmod qR\right)(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}) \\
&= [\delta_1, \delta_2] \cdot \mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}(\overline{\mathbf{A}}),\mathbf{s}'}(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}).
\end{aligned}$$

So denoting $\mathcal{P} = \mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}(\overline{\mathbf{A}}),\mathbf{s}'}$, it holds that $\mathcal{P}_1 \approx_{\delta_1,\delta_2} \mathcal{P}$. Similarly, starting from $\mathcal{P}_2$ and conditioning on $\mathbf{v}_{2,i} = \overline{\mathbf{v}_{2,i}}$ yields

$$\mathcal{P}_2(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}) \in [\delta_1, \delta_2] \cdot \mathcal{D}_{\mathcal{L}_q^{\mathbf{u}}(\overline{\mathbf{A}}),\mathbf{s}'}(\overline{\mathbf{v}_1}, \overline{\mathbf{v}_2}, \overline{\mathbf{v}_3}),$$

yielding $\mathcal{P}_2 \approx_{\delta_1,\delta_2} \mathcal{P}$. Combining both gives the result with a loss $\delta = \delta_2/\delta_1$. The expression of δ_1, δ_2 and their asymptotic equivalent for small ε are obtained from Lemma 4.3 and yield the expression and asymptotic equivalent for δ.

We note that our partial trapdoor technique can be used in other constructions as well. For example, the group signature [LNPS21] would benefit from our technique allowing to reduce the group signature size as well as the user secret key size.

These k columns $\mathfrak{t}_{d+1}\mathbf{G}_i - \mathbf{AR}_{d+1}$ are sufficient to carry a standard hybrid argument to hide a tag guess in the public key while keeping the ability to sample preimages. More concretely, the reduction would set $\mathfrak{t}_{d+1} = 1$. At the beginning of the proof, any signature with tag $\mathfrak{t}$ is answered normally with $\mathfrak{t}_i = \mathfrak{t}$ for $i \leq d$. Then, at the j-th hybrid, $j-1$ applications of the strategy above lead to a situation where the first $(j-1)$ blocks $\mathbf{AR}_i$ of the public key have been replaced by $\mathfrak{t}^+\mathbf{G}_i + \mathbf{AR}_i$, which means that signatures with tag $\mathfrak{t}$ are actually generated using $\mathfrak{t}_1 = \dots = \mathfrak{t}_{j-1} = \mathfrak{t} - \mathfrak{t}^+$ and $\mathfrak{t}_i = \dots = \mathfrak{t}_d = \mathfrak{t}$. Note that this is transparent to the adversary as the signatures do not leak any information on the actual tag: this is actually the core argument of the security proofs in Section 6.2 where the adversary obliviously use tags $\mathfrak{t} - \mathfrak{t}^+$. Moreover, generating a signature for tag $\mathfrak{t}^+$ is still possible using the very classical approach consisting in programming the public syndrome $\mathbf{u}$ accordingly. Our reduction can thus answer all signing queries at any stage using only this extra block $\mathfrak{t}_{d+1}\mathbf{G}_j - \mathbf{AR}_{d+1}$. As a consequence, the dimension of $\mathbf{A}_{\mathfrak{t}}$ and hence the one of our signatures will be $2d + k(d+1)$ instead of $2(d+kd)$, which leads to smaller signatures but also smaller zero-knowledge proofs (see Chapter 7).

6.4 Optimized Signature with Efficient Protocols

We here provide the intuition behind the most noticeable modifications, besides the trapdoor switching of Section 6.3, we are making compared to the construction and techniques of Section 6.2. Other modifications presented in Section 6.1, such as finer precision analysis or bound optimizations are not discussed here because they do not intrinsically change our previous construction.

Changing the Signature Distribution. After moving to a computational instantiation of MP trapdoors using our trapdoor switching technique, we then have a shorter signature $\mathbf{v}$ but it still follows a spherical Gaussian distribution. The main step in the procedure to generate $\mathbf{v}$ is the preimage sampling algorithm used on syndromes of the form $\mathbf{u} + \mathbf{Ar} + \mathbf{Dm} \bmod qR$. As discussed throughout Chapter 4, the MP sampler used in our previous construction is far from optimal and we can hope for a much lower Gaussian width for $\mathbf{v}_2$ (and $\mathbf{v}_3$).

From the study of Chapter 4, we have four potential replacements: the elliptic sampler (Algorithm 4.5), the rejection sampler (LW*, Algorithm 4.6), the approximate elliptic sampler (presented in [JHT22]), or the approximate rejection sampler (Algorithm 4.7). In any case, the preimage must be generated so as to hide information on the trapdoors. The subtlety here is that the inverted syndromes are not necessarily *statistically uniform*. We thus need a sampler whose security is guaranteed in the worst case. This unfortunately discards all of the choices above except for the elliptic sampler presented in Chapter 4. Regardless, the latter offers very nice improvements on the signature size and also on the security through smaller M-SIS bounds.

Removing Signer's Randomness. As explained above, the goal of the security proofs of signature schemes based on the MP sampler is to end up with a situation where the reduction can normally answer all signing queries but one, for which it has no trapdoor. For this special query, the reduction leverages some information hidden in the public parameters but, as the latter are defined at the beginning of the game, they do not necessarily compensate the $\mathbf{Dm}$ component of the syndrome which is adaptively chosen by the adversary. As a consequence, the distribution of the signature in this case may not be correct, leading the reduction to fail. In Section 6.2, we solve this problem using a rather conventional approach where the signer contributes to the syndrome so as to drown the uncontrolled term. Concretely, instead of computing a preimage of $\mathbf{u} + \mathbf{Dm} \bmod qR$, it selects a random vector $\mathbf{r}$ and then computes a preimage of $\mathbf{u} + \mathbf{Dm} + \mathbf{Ar} \bmod qR$. This additional randomness $\mathbf{r}$, chosen with the knowledge of $\mathbf{m}$, is sufficient to prove security using a standard noise drowning argument with the Rényi divergence.

Besides making the signing procedure more complex, the downside of this approach is that it adds an element $\mathbf{r}$ to the signature. Although it can be merged with the signature, it still has a cost as it increases the norm of the first part of the signature. To remove $\mathbf{r}$, we follow in our proof a different approach based on rejection sampling, as in [CKLL19]. The core idea is to abort the reduction if the message $\mathbf{m}$ leads to an invalid distribution of the signature while tolerating a small amount of leakage using Lemma 1.26 (contrarily to [CKLL19]) so as to improve performance. As this leakage only occurs once in the reduction, it does not significantly impact security. In all cases, this approach only entails a small constant reduction loss factor compared to the previous one based on the Rényi divergence. More precisely, we achieve a constant loss factor, but without having to increase the Gaussian width by a $\Theta(\sqrt{\lambda})$ factor. Decreasing the reduction loss allows us to use much smaller parameters as we need to aim for around 165 bits of M-SIS core-SVP hardness instead of 213 in Section 6.2.

We nevertheless stress that this only allows to remove the signer's randomness. Some situations (e.g. obtaining a signature in a privacy-preserving protocol) may indeed require the user to hide his message by adding $\mathbf{Ar}_u$ to the commitment $\mathbf{Dm}$ and this remains true in our case. This will be the case in Chapter 7. We will therefore need to consider two variants of our scheme, one for the standalone version of our signature and one for usage in the situations mentioned above. Actually, the only difference will be located in the verification bound on the first part of the signature ($\mathbf{v}_1$). For the signature itself, we have $\|\mathbf{v}_1\|_2 \leq B_1$ where B_1 is determined by Lemma 1.21, while in Chapter 7, we have $\|\mathbf{v}_1\|_2 \leq B_1 + \|\mathbf{r}_u\|_2$. At this point, changing to a rejection-based analysis also improves upon the prior construction. The choice of Gaussian randomness was initiallt motivated by the use of the Rényi divergence in the noise drowning step. Using a rejection-based method allows us to rely on a computationally hiding commitment and use $\mathbf{r}_u$ to be composed of binary polynomials, which results in only a $\sqrt{2nd}$ additive term in the verification bound.

6.4.1 Description

In the end, our signature $\mathbf{v}$ on a message $\mathbf{m}$ is now a vector of dimension $2d + k(d+1)$ following an elliptical distribution such that

$$\mathbf{A}_{\mathsf{t}}\mathbf{v} = [\mathbf{A}|\mathsf{t}\mathbf{G} - \mathbf{A}\mathbf{R}|\mathbf{A}_3]\mathbf{v} = \mathbf{u} + \mathbf{D}\mathbf{m} \bmod qR,$$

where $\mathbf{A}_3$ is a $d \times k$ random matrix. We also consider $\mathbf{A}$ of the form $[\mathbf{I}_d|\mathbf{A}']$ for $\mathbf{A}' \in R_q^{d\times d}$ so that the first d entries of $\mathbf{v}$ do not need to be transmitted. The formal description of the Setup, KeyGen, Sign and Verify algorithms that constitute our optimized signature with efficient protocols is provided below. To differentiate it with the one from Section 6.2, we prefix each algorithm with SEP*. We also present a stateful version of the signature, but it can be turned into a stateless signature using the method we presented in the first paper [JRS23].

Algorithm 6.5: SEP*.Setup

Input: Security parameter λ.

1. Choose a positive integer d.
2. Choose $\kappa \leq n$ to be a power of two.
3. Choose a prime q such that $q = 2\kappa + 1 \bmod 4\kappa$ and $q > \sqrt{\kappa}^{\kappa}$.
4. Choose positive integer w such that $\binom{n}{w} \geq Q$. ▷ Hamming weight of tags
5. Choose positive integer b. ▷ Gadget base
6. $\mathcal{T}_w \leftarrow \{\mathsf{t} \in T_1 : \|\mathsf{t}\|_1 = w\}$. ▷ Tag space
7. $k \leftarrow \lceil \log_b(\lceil (q-1)/2 \rceil + 1) \rceil$.
8. Choose a positive integer m. ▷ Maximum bit-size of $\mathbf{m}$ is nm
9. $\mathbf{G} = \mathbf{I}_d \otimes [1|b|\cdots|b^{k-1}] \in R_q^{d\times dk}$. ▷ Gadget matrix
10. $r \leftarrow \sqrt{\ln(2nd(2+k)(1+\varepsilon^{-1}))/\pi}$. ▷ $r \gtrsim \eta_\varepsilon(\mathbb{Z}^{nd(2+k)})$
11. $s_{\mathbf{G}} \leftarrow r\sqrt{b^2+1}$. ▷ Gadget sampling width
12. $s_1 \leftarrow \max\left(\sqrt{\frac{\pi}{\ln(2)}}n\sqrt{dm}, \sqrt{\frac{2s_{\mathbf{G}}^4}{s_{\mathbf{G}}^2 - 1}} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)\right)$. ▷ Top preimage width
13. $s_2 \leftarrow r\sqrt{2b^2+3}$. ▷ Bottom preimage width
14. $\gamma \leftarrow \frac{s_1}{n\sqrt{dm}}$ ▷ Rejection sampling slack
15. $M \leftarrow \exp(\pi/\gamma^2)$. ▷ Repetition rate
16. $\mathbf{A}' \hookleftarrow U(R_q^{d\times d})$.
17. $\mathbf{A}_3 \hookleftarrow U(R_q^{d\times k})$.
18. $\mathbf{u} \hookleftarrow U(R_q^{d})$.
19. $\mathbf{A} \leftarrow [\mathbf{I}_d|\mathbf{A}'] \in R_q^{d\times 2d}$.
20. $\mathbf{D} \hookleftarrow U(R_q^{d\times m})$. ▷ Message Commitment Key

Output: $\mathsf{pp} = (\mathbf{A}', \mathbf{A}_3, \mathbf{u}, \mathbf{D}; \lambda, n, d, b, k, q, w, m, s_1, s_2, s_{\mathbf{G}})$. ▷ $\mathbf{A}', \mathbf{A}_3, \mathbf{u}, \mathbf{D}$ can be stored as a 32-byte seed.

Algorithm 6.6: SEP*.KeyGen

Input: Public parameters pp as in Algorithm 6.5.

1. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d\times dk}$ conditioned on $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$.
2. $\mathbf{B} \leftarrow \mathbf{A}\mathbf{R} \bmod qR \in R_q^{d\times dk}$.

Output: $\mathsf{pk} = \mathbf{B}$, and $\mathsf{sk} = \mathbf{R}$. ▷ pp stored with pk for simplicity

Algorithm 6.7: SEP*.Sign

Input: Signing key sk, Message $\mathbf{m} \in T_1^m$, Public key pk, State st.

1. $\mathbf{c} \leftarrow \mathbf{D}\mathbf{m} \bmod qR$. ▷ Biding commitment to $\mathbf{m}$
2. $\mathsf{t} \leftarrow \mathsf{F}(\mathsf{st})$. ▷ $\mathsf{t} \in \mathcal{T}_w$
3. $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$.
4. $(\mathbf{v}_1, \mathbf{v}_2) \leftarrow \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u} + \mathbf{c} - \mathbf{A}_3\mathbf{v}_3 \bmod qR, \mathsf{t}, s_1, s_2, s_{\mathbf{G}})$ ▷ Algorithm 4.5
5. **if** $\|\mathbf{v}_1\|_2 > B_1 \ \vee\ \|\mathbf{v}_2\|_2 > B_2 \ \vee\ \|\mathbf{v}_3\|_2 > B_3$ **goto** 3).
6. $\mathsf{st} \leftarrow \mathsf{st} + 1$.
7. Parse $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T|\mathbf{v}_{1,2}^T]^T$ with $\mathbf{v}_{1,1}, \mathbf{v}_{1,2} \in R^d$.

Output: $\mathsf{sig} = (\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$.

Algorithm 6.8: SEP*.Verify

Input: Public key pk, Message $\mathbf{m} \in T_1^m$, Signature sig.

1. $\mathbf{v}_{1,1} \leftarrow \mathbf{u} + \mathbf{D}\mathbf{m} - \mathbf{A}'\mathbf{v}_{1,2} - (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 - \mathbf{A}_3\mathbf{v}_3 \bmod qR \in R^d$.
2. $\mathbf{v}_1 \leftarrow [\mathbf{v}_{1,1}^T|\mathbf{v}_{1,2}^T]^T$
3. $b_1 \leftarrow \|\mathbf{v}_1\|_2 \leq B_1$. ▷ $B_1 = c_{2nd}s_1\sqrt{2nd}$
4. $b_2 \leftarrow \|\mathbf{v}_2\|_2 \leq B_2$. ▷ $B_2 = c_{ndk}s_2\sqrt{ndk}$

5. $b_3 \leftarrow \|\mathbf{v}_3\|_2 \leq B_3$. ▷ $B_3 = c_{nk} s_2 \sqrt{nk}$
6. $b_4 \leftarrow \mathsf{t} \in \mathcal{T}_w$.
7. $b_5 \leftarrow \mathbf{m} \in T_1^m$.

Output: $b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge b_5$. ▷ 1 if valid, 0 otherwise

The verification bounds B_1, B_2 and B_3 are set using Lemma 1.21. Typically, for $c = 1$, the probability is at most 2^{-2N} where N is the dimension of the vector. This is a little too conservative as we usually have $2N \gg \lambda$. Instead, we use a slack c to tweak the tailcut probability. To be more precise, c now denotes a function that takes the dimension N as input and a parameter λ, and outputs the smallest $c > 1/\sqrt{2\pi}$ such that $(c\sqrt{2\pi e}e^{-\pi c^2})^N \leq 2^{-\lambda}$. For example, it holds for any dimension N that $c(N, N) \approx 0.767$. As an other example, we have $c(512, 128) \approx 0.5751$. For clarity, we simply use c_N to denote $c(N, \lambda + O(1))$ where λ is the security parameter. Heuristically, we could even choose $c_N = 1/\sqrt{2\pi} \approx 0.4$ and have the bound verified with high probability.

Lemma 6.3 (Correctness of the SEP*)

The signature scheme of Algorithms 6.5, 6.6, 6.7, and 6.8 is correct.

Proof (Lemma 6.3). Let pp, and $(\mathsf{pk}, \mathsf{sk}) = (\mathbf{B}, \mathbf{R})$ be obtained by running $\mathsf{Setup}(1^\lambda)$, and $\mathsf{KeyGen}(\mathsf{pp})$ respectively. Let $\mathbf{m} \in T_1^m$ be an arbitrary message and $(\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3) \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathbf{m}, \mathsf{pk}, \mathsf{st})$ a signature. We define

$$\mathbf{v}_{1,1} = \mathbf{u} + \mathbf{Dm} - \mathbf{A}'\mathbf{v}_{1,2} - (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 - \mathbf{A}_3\mathbf{v}_3 \bmod qR \in R^d.$$

It holds that $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T | \mathbf{v}_{1,2}^T]^T$ and $\mathbf{v}_2$ were obtained from $\mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}, \mathsf{t}\mathbf{I}_d, \mathbf{u} + \mathbf{Dm} - \mathbf{A}_3\mathbf{v}_3 \bmod qR, s_1, s_2, s_{\mathbf{G}})$. Using the same argument as the one from the proof of Lemma 6.2, we get that the distribution of $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is $[\delta_1, \delta_2]$-close from the elliptical distribution $\mathcal{D}_{R^{2d(1+k)}, \mathbf{s}'}$ conditioned on $\mathbf{A}\mathbf{v}_1 + (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 = \mathbf{u} + \mathbf{Dm} \bmod qR$, where δ_1, δ_2 are defined in Lemma 4.3 and $\mathbf{s}' = [s_1\mathbf{1}_{2nd} | s_2\mathbf{1}_{n(d+1)k}]$.

Applying Lemma 1.21 yields the bounds $B_1 = c_{2nd} s_1 \sqrt{2nd}$, $B_2 = c_{ndk} s_2 \sqrt{ndk}$ and $B_3 = c_{nk} s_2 \sqrt{nk}$ on $\|\mathbf{v}_1\|_2, \|\mathbf{v}_2\|_2, \|\mathbf{v}_3\|_2$. It gives that $b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge b_5 = 1$ except with probability $\delta_2 2^{-(\lambda + O(1))}$ by definition of c_{2nd}, c_{nkd}, c_{nk}. Since we set ε so that $\delta_2 = 1 + O(1)$, we indeed obtain the correctness with overwhelming probability as claimed. Note that since we reject signatures that exceed the bounds during the signing process, the correctness of outputted signatures is actually guaranteed. Nevertheless, the correctness error we just derived is helpful to establish that generated signatures are never rejected except with negligible probability, thus bounding the number of rejections during the signing procedure.

Remark 6.4

We can have smaller tailcuts by aiming for a probability bound of say 2^{-12} so that all three bounds are verified except with probability at most 2^{-10}. This would slightly improve the signature sizes and the M-SIS bounds used in the security assessment, but at the expense of rejecting signatures more often. It then provides a trade-off between size performance and computational performance. We decide not to feature this optimization as it has almost no limited impact on the zero-knowledge proof sizes of Chapter 7, which is the main metric we want to optimize over.

6.4.2 Security Analysis

We now give the formal security statement of our signature scheme. Although the high-level idea of hiding a tag guess in the public key is very similar to that of the proofs of Theorems 6.1 and 6.2, moving to a computational setting requires care. We therefore present the proof with sequences of hybrid games for clarity. Also, all of the optimizations we provide feed through to the security proofs, making the treatment of every argument fairly different from the previous proofs. We again distinguish between two different types a forgeries (❶ and ❷) and treat them separately. Combining both Theorem 6.3 and 6.4 proves the EUF-CMA security of the optimized signature with efficient protocols.

Theorem 6.3 (Unforgeability Against Type ❶ Forgeries - SEP*)

An adversary produces a forgery $(\mathfrak{t}^\star, \mathbf{v}_{1,2}^\star, \mathbf{v}_2^\star, \mathbf{v}_3^\star)$ of *Type* ❶ if the tag $\mathfrak{t}^\star$ does not collide with the tags of the signing queries. The advantage of any PPT adversary $\mathcal{A}$ in producing a type ❶ forgery is at most

$$\mathrm{Adv}_{❶}[\mathcal{A}] \leq h^{\circ d}\left(C(|\mathcal{T}_w| - Q)\varepsilon_{\text{M-SIS}}\right)$$

where C is a small constant from Heuristic 1.3, $\varepsilon_{\text{M-SIS}}$ is the hardness bound of $\text{M-SIS}_{n,d,2d+k+m+1,q,\beta_{❶}}$ for

$$\beta_{❶} = \sqrt{(B_1 + \sqrt{nd}B_2)^2 + B_3^2 + nm + 1}.$$

The function $h^{\circ d}$ is the function h composed d times, where h is defined by

$$h(x) = k\varepsilon_{\text{M-LWE}} + \delta'\left(2k\varepsilon_{\text{M-LWE}} + \delta'\left(k\varepsilon_{\text{M-LWE}} + x\right)^{1-1/2\lambda}\right)^{1-1/2\lambda},$$

with $\varepsilon_{\text{M-LWE}}$ the hardness bound of $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1,\mathcal{B}_1}$, and

$$\delta' = \left(1 + \frac{\lambda(2\lambda-1)(\delta-1)^2}{(2-\delta)^{2\lambda+1}}\right)^{Q/2\lambda} \underset{\varepsilon\to 0}{\sim} 1 + Q\cdot(\lambda - 1/2)\cdot(2(12d(n-1)+7)\varepsilon)^2,$$

and δ is defined in Lemma 6.2.

When setting parameters, choosing $\varepsilon = 1/\Omega(nd\sqrt{Q\lambda})$ leads to $\delta' = O(1)$. For example, for $\lambda = 128, n = 256, d = 4, k = 5, Q = 2^{32}$, choosing $\varepsilon = 2^{-36}$ gives $\delta' \leq 1.07206$, meaning it only incurs a loss of a tenth of a bit. In our parameter selection, we later choose $\varepsilon = 2^{-40}$ giving $\delta' \leq 1.000272$.

Proof (Theorem 6.3). Throughout the proof, we consider a PPT adversary $\mathcal{A}$ interacting with the challenger $\mathcal{B}$, and which aims at producing a valid Type ❶ forgery. We proceed by a game hop to modify the distribution of the view of $\mathcal{A}$ in a way that is indistinguishable from the real distribution. In the last game, the constructed elements that compose the distribution given to $\mathcal{A}$ allow to easily exploit the forgery to obtain a solution to M-SIS. Under the assumption that M-SIS is hard, it should thus be infeasible for $\mathcal{A}$ to produce a valid type ❶ forgery. We proceed using a game-based proof which follows the sequence summarized here.

Overview of the unforgeability reduction (type ❶)

- G_0
- G_1 ▷ Sampling tags at the start
- **For** $j \in [\![d]\!]$
  - $G_{j,0}$
  - $G_{j,1}$ ▷ Hiding a partial gadget in $\mathbf{A}_3$
  - $G_{j,2}$ ▷ Hiding a partial trapdoor in $\mathbf{A}_3$
  - $G_{j,3}$ ▷ Trapdoor switching
  - $G_{j,4}$ ▷ Partial key simulation
  - $G_{j,5}$ ▷ Hiding a tag guess
  - $G_{j,6}$ ▷ Reinstating partial key
  - $G_{j,7}$ ▷ Trapdoor switching
  - $G_{j,8}$ ▷ Removing the partial trapdoor from $\mathbf{A}_3$
  - $G_{j,9}$ ▷ Removing the partial gadget from $\mathbf{A}_3$
- Solve M-SIS using $\mathcal{A}$ against $G_{d,9}$.

Games Hops. We define the following games which are composed of three stages: setup, queries, forgery. Past the queries stage, the view of the adversary does not change so we only describe the first two stages. The matrix $\mathbf{A}'$ (and in turn $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$), the matrix $\mathbf{D}$, and the syndrome $\mathbf{u}$ are always generated the same way, i.e., $\mathbf{A}' \hookleftarrow U(R_q^{d\times d})$, $\mathbf{D} \hookleftarrow U(R_q^{d\times m})$ and $\mathbf{u} \hookleftarrow U(R_q^d)$, and we thus do not specify them in the games below. In each game, the view of $\mathcal{A}$ is $(\mathbf{A}, \mathbf{D}, \mathbf{B}, \mathbf{u}, \mathbf{A}_3, (\mathsf{sig}^{(i)})_{i\in[\![Q]\!]})$.

Game G_0. This corresponds to the original unforgeability game where the key material gen-

eration and signing queries are handled honestly. More precisely, we have

G_0

Setup

1. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d\times dk}$ such that $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)$
2. $\mathbf{B} = \mathbf{AR} \bmod qR$
3. $\mathbf{A}_3 \hookleftarrow U(R_q^{d\times k})$

Queries

Given $\mathbf{m}^{(i)} \in T_1^m$, compute $\mathsf{t}^{(i)} = \mathsf{F}(\mathsf{st})$ and increment st. Then

1. $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k,s_2}$
2. $(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}) \leftarrow \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u}+\mathbf{Dm}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathsf{t}^{(i)}, s_1, s_2, s_{\mathbf{G}})$
3. Send $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$.

Game G_1. In G_1, we simply change the way tags are generated. Instead of computing $\mathsf{t}^{(i)}$ at each signing query, we first generate and store all the Q tags during the setup stage. In the query stage, we simply look-up the corresponding tag. It also samples a tag guess $\mathsf{t}^+ \hookleftarrow U(\mathcal{T}_w \setminus \{\mathsf{t}^{(i)}; i \in [\![Q]\!]\})$, but it is so far not used. The view is exactly the same in G_1 because we only changed the moment when the tags are generated. Since they are generated deterministically from the state, both views are identically distributed.

We are now aiming to hide the tag guess within the public key, that is replace the public key $\mathbf{B} = \mathbf{AR}$ by $\mathbf{B} = \mathbf{AR} + \mathsf{t}^+\mathbf{G}$, while keeping the ability to answer signing queries. For that, we proceed with a hybrid argument defined by a sequence of games $G_{j,\ell}$ for $j \in [\![d]\!]$ and $\ell \in [\![0,9]\!]$. Recall the notation $\mathbf{G}_i = \mathbf{e}_i \otimes \mathbf{g}^T$ from Section 6.3, which corresponds to having the gadget only on the i-th row, thus allowing to invert only to i-th entry of a syndrome. In game $G_{j,9}$, the public key has been transformed to $\mathbf{B} = \mathbf{AR} + [\mathsf{t}^+\mathbf{G}_1|\dots|\mathsf{t}^+\mathbf{G}_j|\mathbf{0}|\dots|\mathbf{0}]$. We construct the games so that $G_{1,0} = G_1$, that for all $j \in [\![d-1]\!]$, $G_{j,9} = G_{j+1,0}$ and we give detailed arguments to go from $G_{j,0}$ to $G_{j,9}$. Let $j \in [\![d]\!]$.

Game $G_{j,0}$. In this game, the challenger performs the setup phase as follows. It computes all the $\mathsf{t}^{(i)}$ at the outset and samples a tag guess $\mathsf{t}^+ \hookleftarrow U(\mathcal{T}_w \setminus \{\mathsf{t}^{(i)}; i \in [\![Q]\!]\})$. It then samples $(\mathbf{R}_i)_{i\in[\![d]\!]}$ from $\mathcal{B}_1^{2d\times k}$ such that $\mathbf{R} = [\mathbf{R}_1|\dots|\mathbf{R}_d]$ satisfies $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)$. Then, for $i \in [\![j-1]\!]$ it defines $\mathbf{B}_i = \mathbf{AR}_i + \mathsf{t}^+\mathbf{G}_i \bmod qR$, and for $i \in [\![j,d]\!]$ it defines $\mathbf{B}_i = \mathbf{AR}_i \bmod qR$. It then constructs $\mathbf{B} = [\mathbf{B}_1|\dots|\mathbf{B}_d]$ as the public key. Note that when $j = 1$ we simply have $\mathbf{B} = \mathbf{AR} \bmod qR$. It then samples $\mathbf{A}_3$ from $U(R_q^{d\times k})$, and sends the public key and public parameters to $\mathcal{A}$.

When receiving a signing query on $\mathbf{m}^{(i)}$, the challenger first looks-up the tag $\mathsf{t}^{(i)}$. It then samples $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k,s_2}$, and

$$(\mathbf{v}_1^{(i)}, \mathbf{v}_2^{(i)}) \leftarrow \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u}+\mathbf{Dm}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathbf{T}_j, s_1, s_2, s_{\mathbf{G}}),$$

where

$$\mathbf{T}_j = \mathrm{diag}(\underbrace{\mathsf{t}^{(i)}-\mathsf{t}^+, \dots, \mathsf{t}^{(i)}-\mathsf{t}^+}_{j-1 \text{ times}}, \underbrace{\mathsf{t}^{(i)}, \dots, \mathsf{t}^{(i)}}_{d-(j-1) \text{ times}}).$$

It then returns the signature $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$. Note that although the signature tag is $\mathsf{t}^{(i)}$, the effective tag in the preimage sampling is $\mathbf{T}_j$. Since t^+ is different from all the $\mathsf{t}^{(i)}$, and since $\mathsf{t}^{(i)} - \mathsf{t}^+ \in S_1$, we can use Lemma 1.4 (or Remark 1.2) to argue that $\mathsf{t}^{(i)} - \mathsf{t}^+ \bmod qR$ is in $R_q^\times$ as desired. Hence, $\mathbf{T}_j \in GL_d(R_q)$. Also, notice that when $j = 1$, we can directly see that $G_{1,0}$ is exactly the game G_1 from before.

Game $G_{j,1}$. This game is the same as $G_{j,0}$ except in the way $\mathbf{A}_3$ is generated. Instead of sampling $\mathbf{A}_3$ uniformly, we hide the gadget $\mathbf{G}_j$ by first sampling $\mathbf{A}_3'$ from $U(R_q^{d\times k})$ and defining $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}_3 \bmod qR$. In $G_{j,1}$, $\mathbf{A}_3'$ is sampled uniformly and independently of $\mathbf{G}_j$. Thence, $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}_3' \bmod qR$ is also uniformly distributed, as in $G_{j,0}$. So the views are identically distributed.

Game $G_{j,2}$. We now hide a short relation in $\mathbf{A}_3'$. That is, we sample $\mathbf{R}_j'$ from $\mathcal{B}_1^{2d\times k}$ such that $\mathbf{R}_{-j} = [\mathbf{R}_1|\dots|\mathbf{R}_{j-1}|\mathbf{R}_j'|\mathbf{R}_{j+1}|\dots|\mathbf{R}_d]$ satisfies $\|\mathbf{R}_{-j}\|_2 \leq \frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)$. It then defines $\mathbf{A}_3' = \mathbf{AR}_j' \bmod qR$. At this point, the matrix $\mathbf{A}_3$ is now equal to $\mathbf{G}_j - \mathbf{AR}_j' \bmod qR$.

We now argue that if one distinguishes $G_{j,2}$ from $G_{j,1}$, then it can solve M-LWE. Let $\mathcal{D}$ be a distinguisher between the views from $G_{j,1}$ and $G_{j,2}$. We construct a distinguisher $\mathcal{D}'$ for $\text{M-LWE}^k_{n,d,d,q,\mathcal{B}_1,\mathcal{B}_1}$. Given a multiple-secret M-LWE challenge $(\mathbf{A}', \mathbf{A}'_3) \in R_q^{d\times d} \times R_q^{d\times k}$, $\mathcal{D}'$ assumes the role of the challenger in the games and uses $\mathbf{A}', \mathbf{A}'_3$ to perfectly simulate the interaction with $\mathcal{A}$. It then sends the resulting view to $\mathcal{D}$. If $\mathcal{D}$ responded $G_{j,1}$, then $\mathcal{D}'$ respond 0 (uniform), and 1 (LWE) if $\mathcal{D}$ responded $G_{j,2}$. Indeed, if $\mathbf{A}'_3$ is uniform, then the view exactly simulate that of $G_{j,1}$, and if $\mathbf{A}'_3 = [\mathbf{I}_d|\mathbf{A}']\mathbf{R}'_j$ for some $\mathbf{R}'_j \sim \mathcal{B}_1^{2d\times k}$, then it correctly simulates $G_{j,2}$. As a result, it holds that

$$\forall \mathcal{D}\ \mathsf{PPT}\ \text{distinguisher}, \mathrm{Adv}_{G_{j,1},G_{j,2}}[\mathcal{D}] \leq k\varepsilon_{\text{M-LWE}},$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bound for $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1,\mathcal{B}_1}$ defined as $\varepsilon_{\text{M-LWE}} = \sup_{\mathcal{D}''\ \mathsf{PPT}} \mathrm{Adv}_{\text{M-LWE}}[\mathcal{D}'']$. Note that here, we implicitly use a standard hybrid argument (e.g., Lemma 2.4) showing that $\text{M-LWE}^k_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ is at least as hard as $\text{M-LWE}^1_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ at the expense of a loss factor k in the reduction.

Game $G_{j,3}$. In game $G_{j,3}$, we modify the way signing queries are answered by switching the partial trapdoor $\mathbf{R}_j$ for $\mathbf{R}'_j$. Concretely, upon reception of a message $\mathbf{m}^{(i)} \in T_1^m$, the signer gets the tag $\mathfrak{t}^{(i)}$, samples $\mathbf{v}_{2,j}^{(i)} \hookleftarrow \mathcal{D}_{R^k,s_2}$ and then computes

$$\begin{aligned}&(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, (\mathbf{v}_{2,1}^{(i)}, \ldots, \mathbf{v}_{2,j-1}^{(i)}, \mathbf{v}_3^{(i)}, \mathbf{v}_{2,j+1}^{(i)}, \ldots, \mathbf{v}_{2,d}^{(i)}))\\ &= \mathsf{EllipticSampler}(\mathbf{R}_{-j}; \mathbf{A}', \mathbf{u} + \mathbf{D}\mathbf{m}^{(i)} - (\mathfrak{t}^{(i)}\mathbf{G}_j - \mathbf{B}_j)\mathbf{v}_{2,j}^{(i)}, \mathbf{T}_{-j}, s_1, s_2, s_{\mathbf{G}}),\end{aligned}$$

where

$$\mathbf{T}_{-j} = \mathrm{diag}(\underbrace{\mathfrak{t}^{(i)} - \mathfrak{t}^+, \ldots, \mathfrak{t}^{(i)} - \mathfrak{t}^+}_{j-1\text{ times}}, 1, \underbrace{\mathfrak{t}^{(i)}, \ldots, \mathfrak{t}^{(i)}}_{d-j\text{ times}}).$$

It then sends the signature $\mathsf{sig}^{(i)} = (\mathfrak{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$. Using the trapdoor switching result from Lemma 6.2 on a single query gives a relative error between $\mathcal{P}_1$ and $\mathcal{P}_2$ of $\delta - 1$, where

$$\delta = \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^{12d(n-1)+5} \left(\frac{1+\varepsilon/ndk}{1-\varepsilon/ndk}\right)^{2ndk},$$

and $\mathcal{P}_1$, $\mathcal{P}_2$ are the distributions from the lemma statement. Indeed $\mathcal{P}_1/\mathcal{P}_2 - 1 \in [\delta^{-1} - 1, \delta - 1] \subseteq [-(\delta-1), (\delta-1)]$. We then use the relative error lemma of Lemma 1.9 and the multiplicativity of the Rényi divergence (of order 2λ) from Lemma 1.8 to get

$$\mathrm{Adv}_{G_{j,2}}[\mathcal{A}] \leq \delta' \cdot \mathrm{Adv}_{G_{j,3}}[\mathcal{A}]^{1-1/2\lambda}$$

where

$$\delta' = \left(1 + \frac{\lambda(2\lambda-1)(\delta-1)^2}{(2-\delta)^{2\lambda+1}}\right)^{Q/2\lambda} \underset{\varepsilon\to 0}{\sim} 1 + Q\cdot(\lambda - 1/2)\cdot(2(12d(n-1)+7)\varepsilon)^2.$$

As mentioned before the proof, a typical parameter selection with $\varepsilon = O(1/nd\sqrt{Q\lambda})$ gives δ' extremely close to 1, incurring almost no security loss.

Game $G_{j,4}$. By noticing that the partial trapdoor $\mathbf{R}_j$ is no longer used in $G_{j,3}$, we can now simulate the public key $\mathbf{B}_j$. More precisely, we sample $\mathbf{B}_j$ directly from $U(R_q^{d\times k})$. Using the same argument as for $G_{j,1}$-$G_{j,2}$ on the M-LWE instance $(\mathbf{A}', \mathbf{B}_j)$ this time, we obtain

$$\forall \mathcal{D}\ \mathsf{PPT}\ \text{distinguisher}, \mathrm{Adv}_{G_{j,3},G_{j,4}}[\mathcal{D}] \leq k\varepsilon_{\text{M-LWE}}.$$

Game $G_{j,5}$. We now hide the guess on the forgery tag within the public key $\mathbf{B}_j$. For that, we sample $\mathbf{B}'_j \hookleftarrow U(R_q^{d\times k})$ and define $\mathbf{B}_j = \mathbf{B}'_j + \mathfrak{t}^+\mathbf{G}_j$. Since $\mathbf{B}'_j$ is uniform and independent of $\mathfrak{t}^+\mathbf{G}_j$, then $\mathbf{B}_j = \mathbf{B}'_j + \mathfrak{t}^+\mathbf{G}_j$ is also uniform, as in $G_{j,4}$. So the views are identically distributed.

Game $G_{j,6}$. We then re-hide a short trapdoor in the matrix $\mathbf{B}'_j$. We thus sample $\mathbf{R}_j$ from $\mathcal{B}_1^{2d\times k}$ conditioned on $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$, and then define $\mathbf{B}'_j = \mathbf{A}\mathbf{R}_j \bmod qR$. At

this point, the matrix $\mathbf{B}_j$ is now equal to $\mathbf{A}\mathbf{R}_j + \mathfrak{t}^+\mathbf{G}_j \bmod qR$. The same argument as for $G_{j,1}$-$G_{j,2}$ on the M-LWE instance $(\mathbf{A}', \mathbf{B}'_j)$ yields

$$\forall \mathcal{D}\ \mathsf{PPT}\ \text{distinguisher}, \mathrm{Adv}_{G_{j,5},G_{j,6}}[\mathcal{D}] \leq k\varepsilon_{\text{M-LWE}}.$$

Game $G_{j,7}$. In game $G_{j,7}$, we again modify the way signing queries are answered to use the partial trapdoor $\mathbf{R}_j$ instead of $\mathbf{R}'_j$. This means that when receiving $\mathbf{m}^{(i)} \in T_1^m$, the signer gets the tag $\mathfrak{t}^{(i)}$, sample $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k,s_2}$ and then compute

$$(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}) = \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u} + \mathbf{D}\mathbf{m}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathbf{T}_{j+1}, s_1, s_2),$$

where

$$\mathbf{T}_{j+1} = \mathrm{diag}(\underbrace{\mathfrak{t}^{(i)} - \mathfrak{t}^+, \ldots, \mathfrak{t}^{(i)} - \mathfrak{t}^+}_{j \text{ times}}, \underbrace{\mathfrak{t}^{(i)}, \ldots, \mathfrak{t}^{(i)}}_{d-j \text{ times}}).$$

and sends the signature $\mathsf{sig}^{(i)} = (\mathfrak{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$. As for $G_{j,2}$-$G_{j,3}$, combining Lemma 6.2, 1.8 and 1.9 yields

$$\mathrm{Adv}_{G_{j,6}}[\mathcal{A}] \leq \delta' \mathrm{Adv}_{G_{j,7}}[\mathcal{A}]^{1-1/2\lambda}.$$

Game $G_{j,8}$. We then remove the short relation in $\mathbf{A}'_3$. That is, instead of sampling $\mathbf{R}'_j$ and defining $\mathbf{A}'_3 = \mathbf{A}\mathbf{R}'_j$, we simply sample $\mathbf{A}'_3 \hookleftarrow U(R_q^{d\times k})$. The same argument as for $G_{j,1}$-$G_{j,2}$ on the M-LWE instance $(\mathbf{A}', \mathbf{A}'_3)$ yields

$$\forall \mathcal{D}\ \mathsf{PPT}\ \text{distinguisher}, \mathrm{Adv}_{G_{j,7},G_{j,8}}[\mathcal{D}] \leq k\varepsilon_{\text{M-LWE}}.$$

Game $G_{j,9}$. We finally remove the gadget from $\mathbf{A}_3$. Instead of sampling $\mathbf{A}'_3$ uniformly and defining $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}'_3$, we directly sample $\mathbf{A}_3 \hookleftarrow U(R_q^{d\times k})$. Since in $G_{j,8}$, $\mathbf{A}'_3$ is uniform and independent of $\mathbf{G}_j$, then $\mathbf{A}_3 = \mathbf{G}_j - \mathbf{A}'_3$ is also uniform, as in $G_{j,9}$. So the views are identically distributed.

We can clearly see that $G_{j,9} = G_{j+1,0}$ for $j \in [\![d-1]\!]$, meaning we can indeed chain these games in a hybrid argument. Additionally, hopping from $G_{j,0}$ to $G_{j,9}$ results in a loss characterized by the following inequality.

$$\mathrm{Adv}_{G_{j,0}}[\mathcal{A}] \leq k\varepsilon_{\text{M-LWE}} + \delta' \left(2k\varepsilon_{\text{M-LWE}} + \delta' \left(k\varepsilon_{\text{M-LWE}} + \mathrm{Adv}_{G_{j,9}}[\mathcal{A}]\right)^{\frac{2\lambda-1}{2\lambda}}\right)^{\frac{2\lambda-1}{2\lambda}},$$

that is $\mathrm{Adv}_{G_{j,0}}[\mathcal{A}] \leq h(\mathrm{Adv}_{G_{j,9}}[\mathcal{A}])$, where

$$h(x) = k\varepsilon_{\text{M-LWE}} + \delta' \left(2k\varepsilon_{\text{M-LWE}} + \delta' \left(k\varepsilon_{\text{M-LWE}} + x\right)^{\frac{2\lambda-1}{2\lambda}}\right)^{\frac{2\lambda-1}{2\lambda}}.$$

Because h is non-decreasing, looping over all $j \in [\![d]\!]$ thus gives

$$\mathrm{Adv}_{G_{1,0}}[\mathcal{A}] \leq h^{\circ d}(\mathrm{Adv}_{G_{d,9}}[\mathcal{A}]). \tag{6.8}$$

Although the powers $\frac{2\lambda-1}{2\lambda}$ will stack up with composing the function h d times due to the hybrid argument, the exponent is sufficiently close to 1 and d is a very small integer (typically $d = 4$) so that it only incurs a loss of a few bits, typically around d bits. We give more details on how to bound $h^{\circ d}$ in Lemma 6.4. We thus end up with the following game.

$G_{d,9}$

Setup

1. $\forall i \in [\![Q]\!], \mathfrak{t}^{(i)} = \mathsf{F}(\mathsf{st} + i - 1)$
2. $\mathfrak{t}^+ \hookleftarrow U(\mathcal{T}_w \setminus \{\mathfrak{t}^{(i)}; i \in [\![Q]\!]\})$
3. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d\times dk}$ such that $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$
4. $\mathbf{B} \leftarrow \mathbf{A}\mathbf{R} + \mathfrak{t}^+\mathbf{G} \bmod qR$
5. $\mathbf{A}_3 \hookleftarrow U(R_q^{d\times k})$

Queries

Given $\mathbf{m}^{(i)} \in T_1^m$, get $\mathfrak{t}^{(i)}$. Then

1. $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k,s_2}$
2. $(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_3^{(i)}) \leftarrow \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u} + \mathbf{D}\mathbf{m}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathfrak{t}^{(i)} - \mathfrak{t}^+, s_1, s_2)$
3. Send $\mathsf{sig}^{(i)} = (\mathfrak{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$.

Bounding the advantage. We now need to bound $\mathrm{Adv}_{G_{d,9}}[\mathcal{A}]$. For that we use an adversary in $G_{d,9}$ can be used to construct an adversary $\mathcal{B}$ to solve M-SIS$_{n,d,2d+k+m+1,q,\beta_{❶}}$. Upon reception of the M-SIS instance, $\mathcal{B}$ parses it into $[\mathbf{I}_d|\mathbf{A}'|\mathbf{A}_3|\mathbf{D}|\mathbf{u}]$ and uses these elements to simulate the challenger in $G_{d,9}$. After the queries stage, it receives a type ❶ forgery from $\mathcal{A}$, i.e., it receives a forgery $\mathsf{sig}^\star = (\mathfrak{t}^\star, \mathbf{v}_{1,2}^\star, \mathbf{v}_2^\star, \mathbf{v}_3^\star)$ on $\mathbf{m}^\star$ such that $\mathsf{SEP}^\star.\mathsf{Verify}(\mathsf{pk}, \mathbf{m}^\star, \mathsf{sig}^\star, \mathbf{m}^\star) = 1$. At this point, if $\mathfrak{t}^\star \neq \mathfrak{t}^+$ then $\mathcal{B}$ aborts which happens with probability $1 - 1/(|\mathcal{T}_w| - Q)$. Then, it also aborts if $\|\mathbf{R}\mathbf{v}_2\|_2 > \frac{1}{\sqrt{2}}\sqrt{2nd}\|\mathbf{v}_2^\star\|_2$. By Heuristic 1.3, this happens with probability at most $1 - 1/C$ for a small constant C (typically $C \approx 2$ in our parameter setting), because $\mathbf{R}$ is hidden in $\mathbf{B}$ under M-LWE. If it did not abort, it computes

$$\mathbf{v}_{1,1}^\star = \mathbf{u} + \mathbf{D}\mathbf{m}^\star - (\mathbf{A}'\mathbf{v}_{1,2}^\star + (\mathfrak{t}^\star\mathbf{G} - \mathbf{B})\mathbf{v}_2^\star + \mathbf{A}_3\mathbf{v}_3^\star) \bmod qR,$$

and defines $\mathbf{v}_1^\star = [{\mathbf{v}_{1,1}^\star}^T|{\mathbf{v}_{1,2}^\star}^T]^T$. Since $\mathfrak{t}^\star = \mathfrak{t}^+$, we have $\mathfrak{t}^\star\mathbf{G} - \mathbf{B} = [\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR$. Also, as verification passes, we know that $\|\mathbf{v}_1^\star\|_2, \|\mathbf{v}_3^\star\|_2$ are bounded by B_1, B_3 respectively. We can re-write the definition of $\mathbf{v}_{1,1}^\star$ as

$$[\mathbf{I}_d|\mathbf{A}'|\mathbf{A}_3|\mathbf{D}|\mathbf{u}]\mathbf{x}^\star = \mathbf{0} \bmod qR, \text{ where } \mathbf{x}^\star = \begin{bmatrix} \mathbf{v}_1^\star - \mathbf{R}\mathbf{v}_2^\star \\ \mathbf{v}_3^\star \\ \mathbf{m}^\star \\ -1 \end{bmatrix}.$$

It directly holds that $\mathbf{x}^\star \neq \mathbf{0}$ and we have

$$\|\mathbf{x}^\star\|_2^2 \leq \left(B_1 + \sqrt{nd}B_2\right)^2 + B_3^2 + nm + 1 = \beta_{❶}^2.$$

It thus means that $\mathbf{x}^\star$ is a solution to M-SIS$_{n,d,2d+k+m+1,q,\beta_{❶}}$ and the advantage of $\mathcal{B}$ is $\mathrm{Adv}_{G_{d,9}}[\mathcal{A}] \cdot (C(|\mathcal{T}_w| - Q))^{-1}$. It in turn gives

$$\mathrm{Adv}_{G_{d,9}}[\mathcal{A}] \leq C(|\mathcal{T}_w| - Q)\varepsilon_{\text{M-SIS}}, \tag{6.9}$$

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound of M-SIS. Combining Equations (6.8) and (6.9) and the fact that h is non-decreasing and that $\mathrm{Adv}_{❶}[\mathcal{A}] = \mathrm{Adv}_{G_{1,0}}[\mathcal{A}]$ yields the result.

Theorem 6.4 (Unforgeability Against Type ❷ Forgeries - SEP*)

An adversary produces a forgery $(\mathfrak{t}^\star, \mathbf{v}_{1,2}^\star, \mathbf{v}_2^\star, \mathbf{v}_3^\star)$ of *Type* ❷ if the tag $\mathfrak{t}^\star$ is re-used from some $i^\star$-th signing query $(\mathfrak{t}^{(i^\star)}, \mathbf{v}_{1,2}^{(i^\star)}, \mathbf{v}_2^{(i^\star)}, \mathbf{v}_3^{(i^\star)})$. The advantage of any PPT adversary $\mathcal{A}$ in producing a type ❷ forgery is at most

$$\mathrm{Adv}_{❷}[\mathcal{A}] \leq m\varepsilon_{\text{M-LWE}} + 2MC\frac{1+\varepsilon}{1-\varepsilon}h^{\circ d}(QC^2\varepsilon_{\text{M-SIS}}) + \mathsf{negl}(\lambda).$$

where C is a small constant from Heuristic 1.3, $\varepsilon_{\text{M-SIS}}$ is the hardness bound of M-SIS$_{n,d,d(2+k),q,\beta_{❷}}$ for

$$\beta_{❷} = \sqrt{(2B_1 + 2\sqrt{nd}B_2 + n\sqrt{dm})^2 + 4B_2^2}.$$

The function h, which depends on the loss δ' and the hardness bound $\varepsilon_{\text{M-LWE}}$ of M-LWE$_{n,d,d,q,\mathcal{B}_1,\mathcal{B}_1}$, is the same as that of Theorem 6.4.

Proof (Theorem 6.4). Throughout the proof, we consider a PPT adversary $\mathcal{A}$ interacting with the challenger $\mathcal{B}$, and which aims at producing a valid Type ❷ forgery. We proceed by a game hop to modify the distribution of the view of $\mathcal{A}$ in a way that is indistinguishable from the real distribution. In the last game, the constructed elements that compose the distribution given to $\mathcal{A}$ allow to easily exploit the forgery to obtain a solution to M-SIS. Under the assumption that M-SIS is hard, it should thus be infeasible for $\mathcal{A}$ to produce a valid type ❷ forgery. We again proceed using a game-based proof which follows the sequence summarized here.

Overview of the unforgeability reduction (type ❷)

- G_0
- G_1 ▷ Sampling tags at the start
- G_2 ▷ Hiding a short relation in $\mathbf{D}$
- G_3 ▷ Simulating $\mathbf{u}$
- G_4 ▷ Enforcing norm bounds
- G_5 ▷ Adding rejection
- G_6 ▷ Simulating i^+-th query
- **For** $j \in [\![d]\!]$
  - **For** $i \in [\![0, 9]\!]$ ▷ Hiding the tag guess in partial key j
    - $G_{j,i}$
- Solve M-SIS using $\mathcal{A}$ against $G_{d,9}$.

Games Hops. We define the following games which are composed of three stages: setup, queries, forgery. Past the queries stage, the view of the adversary does not change so we only describe the first two stages. The matrix $\mathbf{A}'$ (and in turn $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$) is always generated the same way, i.e., $\mathbf{A}' \hookleftarrow U(R_q^{d\times d})$, and we thus do not specify them in the games below. In each game, the view of $\mathcal{A}$ is $(\mathbf{A}, \mathbf{D}, \mathbf{B}, \mathbf{u}, \mathbf{A}_3, (\mathsf{sig}^{(i)})_{i\in[\![Q]\!]})$.

Game G_0. This corresponds to the original unforgeability game where the key material generation and signing queries are handled honestly. More precisely, we have

G_0

Setup

1. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d\times dk}$ such that $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$
2. $\mathbf{B} = \mathbf{AR} \bmod qR$
3. $\mathbf{A}_3 \hookleftarrow U(R_q^{d\times k})$
4. $\mathbf{D} \hookleftarrow U(R_q^{d\times m})$
5. $\mathbf{u} \hookleftarrow U(R_q^{d})$

Queries

Given $\mathbf{m}^{(i)} \in T_1^m$, compute $\mathsf{t}^{(i)} = \mathsf{F}(\mathsf{st})$ and increment st. Then

1. $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k,s_2}$
2. $(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}) \leftarrow \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u} + \mathbf{Dm}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathsf{t}^{(i)}, s_1, s_2, s_{\mathbf{G}})$
3. Send $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$.

Game G_1. In G_1, we simply change the way tags are generated. Instead of computing $\mathsf{t}^{(i)}$ at each signing query, we first generate and store all the Q tags during the setup stage. In the query stage, we simply look-up the corresponding tag. In addition, we make a guess on the tag that will be used in the forgery (although it is not used at this point). More precisely, we sample $i^+ \hookleftarrow U([\![Q]\!])$ and define $\mathsf{t}^+ = \mathsf{t}^{(i^+)}$. The view is exactly the same in G_1 because we only changed the moment when the tags are generated. Since they are generated deterministically from the state, and since the tag guess t^+ does not intervene, both views are identically distributed.

Game G_2. We now hide a short relation in $\mathbf{D}$. More precisely, we sample $\mathbf{S}$ from $\mathcal{B}_1^{2d\times m}$ and define $\mathbf{D} = \mathbf{AS} \bmod qR$. We now argue that if one distinguishes G_2 from G_1, then it can solve M-LWE. Let $\mathcal{D}$ be a distinguisher between the views from G_1 and G_2. We construct a distinguisher $\mathcal{D}'$ for $\text{M-LWE}^m_{n,d,d,q,\mathcal{B}_1,\mathcal{B}_1}$. Given a multiple-secret M-LWE challenge $(\mathbf{A}', \mathbf{D}) \in R_q^{d\times d} \times R_q^{d\times m}$, $\mathcal{D}'$ assumes the role of the challenger in the games and uses $\mathbf{A}', \mathbf{D}$ to perfectly simulate the interaction with $\mathcal{A}$. It then sends the resulting view to $\mathcal{D}$. If $\mathcal{D}$ responded G_1, then $\mathcal{D}'$ respond 0 (uniform), and 1 (LWE) if $\mathcal{D}$ responded G_2. Indeed, if $\mathbf{D}$ is uniform, then the view exactly simulate that of G_1, and if $\mathbf{D} = [\mathbf{I}_d|\mathbf{A}']\mathbf{S}$ for some $\mathbf{S} \sim \mathcal{B}_1^{2d\times m}$, then it perfectly simulates G_2. As a result, it holds that

$$\forall \mathcal{D}\ \mathsf{PPT}\ \text{distinguisher}, \mathrm{Adv}_{G_1,G_2}[\mathcal{D}] \leq m\varepsilon_{\text{M-LWE}},$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bound for $\text{M-LWE}_{n,d,d,q,\mathcal{B}_1,\mathcal{B}_1}$ defined as $\varepsilon_{\text{M-LWE}} = \sup_{\mathcal{D}''\ \mathsf{PPT}} \mathrm{Adv}_{\text{M-LWE}}[\mathcal{D}'']$.

Game G_3. We then change the way $\mathbf{u}$ is generated by hiding a short relation within it. Concretely, we sample $\mathbf{v}_1 \hookleftarrow \mathcal{D}_{R^{2d},s_1}$, $\mathbf{v}_2 \hookleftarrow \mathcal{D}_{R^{dk},s_2}$, and $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k,s_2}$ before defining

$$\mathbf{u} = \mathbf{Av}_1 + (\mathsf{t}^+\mathbf{G} - \mathbf{B})\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 \bmod qR.$$

To argue that it is well distributed, we use the regularity lemma from Lemma 1.19. We indeed define $\overline{\mathbf{A}} = [\mathbf{A}|\mathbf{t}^+\mathbf{G} - \mathbf{B}|\mathbf{A}_3]$ and $\mathbf{v} = [\mathbf{v}_1^T|\mathbf{v}_2^T|\mathbf{v}_3^T]^T$. The covariance matrix of $\mathbf{v}$ is $\mathrm{diag}(s_1^2\mathbf{I}_{2nd}, s_2^2\mathbf{I}_{nk(d+1)})$. By our conditions on s_1, s_2 obtained for the security of preimage sampling, we have $s_1 > s_2 \geq \eta_\varepsilon(\mathcal{L}_q^\perp(\overline{\mathbf{A}}))$, where ε is the same used to set $r = \eta_\varepsilon(\mathbb{Z}^{ndk})$. For the range given by Lemma 1.19, we can obtain the inverse and thus get

$$\mathrm{Adv}_{G_2}[\mathcal{A}] \in \left[\frac{1}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] \mathrm{Adv}_{G_3}[\mathcal{A}].$$

Game G_4. In this step, we enforce a bound on the i^+-th query and aborting if this bound is not verified. Concretely, for $i = i^+$, when receiving $\mathbf{m}^{(i^+)}$ the reduction aborts if $\left\|\mathbf{Sm}^{(i^+)}\right\|_2 > \sqrt{nd}\left\|\mathbf{m}^{(i^+)}\right\|_2$. If it did not abort, it handles the rest of the query as before. As $\mathbf{S}$ is unknown to $\mathcal{A}$ because hidden within $\mathbf{D}$ under M-LWE, Heuristic 1.3 yields that the norm constraint is verified with a probability negligibly close to $1/C$ for a small constant C (typically $C = 2$ in our parameter setting). We thus get

$$\mathrm{Adv}_{G_4}[\mathcal{A}] = \left(\frac{1}{C} - \mathsf{negl}(\lambda)\right) \mathrm{Adv}_{G_3}[\mathcal{A}].$$

Game G_5. Now, we add the main rejection in the i^+-th query only to anticipate the next game. For $i \neq i^+$, the queries are handled honestly, while for $i = i^+$ we proceed as follows after the norm check introduced in G_4. The signer samples $\mathbf{v}_3^{(i^+)} \hookleftarrow \mathcal{D}_{R^k,s_2}$ and then computes

$$(\mathbf{v}_{1,1}^{(i^+)}, \mathbf{v}_{1,2}^{(i^+)}, \mathbf{v}_2^{(i^+)}) = \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u} + \mathbf{Dm}^{(i^+)} - \mathbf{A}_3\mathbf{v}_3^{(i^+)}, \mathbf{t}^+, s_1, s_2, s_{\mathbf{G}}),$$

which so far is as usual. Then, it samples a continuous $\rho \hookleftarrow U([0,1))$. Now, the reduction continues only if $\rho \leq 1/M$ and if $\langle \mathbf{v}_1, \mathbf{Sm}^{(i^+)} \rangle \geq 0$. We insist on the fact that at this point $\mathbf{v}_1$ is the one used to define $\mathbf{u}$ which is different from $\mathbf{v}_1^{(i^+)}$.

First, since ρ is independent from the rest, the first condition is verified with probability $1/M$. Then, since the distribution of $\mathbf{S}$ is centered and because $\mathbf{v}_1$ is hidden in $\mathbf{u}$, the probability that $\langle \mathbf{v}_1, \mathbf{Sm}^{(i^+)} \rangle$ is non-negative is negligibly close to $1/2$ as $\mathcal{A}$ cannot predict the sign of $\mathbf{v}_1$ from $\mathbf{u}$. All in all, it means that

$$\mathrm{Adv}_{G_5}[\mathcal{A}] = \left(\frac{1}{2M} - \mathsf{negl}(\lambda)\right) \mathrm{Adv}_{G_4}[\mathcal{A}].$$

Game G_6. We now change how the i^+-th signing query is answered. Upon receiving $\mathbf{m}^{(i^+)}$, the challenger samples $\rho \hookleftarrow U([0,1))$ and computes $A = \langle \mathbf{v}_1 + \mathbf{Sm}^{(i^+)}, \mathbf{Sm}^{(i^+)} \rangle$. Then, it aborts the reduction if

$$A < 0 \text{ or } \rho > \frac{1}{M} \exp\left(\frac{\pi}{s_1^2}\left(\left\|\mathbf{Sm}^{(i^+)}\right\|_2^2 - 2A\right)\right).$$

If it did not abort, it sets

$$\begin{bmatrix} \mathbf{v}_{1,1}^{(i^+)} \\ \mathbf{v}_{1,2}^{(i^+)} \end{bmatrix} = \mathbf{v}_1 + \mathbf{Sm}^{(i^+)}, \mathbf{v}_2^{(i^+)} = \mathbf{v}_2, \text{ and } \mathbf{v}_3^{(i^+)} = \mathbf{v}_3,$$

and sends the signature $\mathsf{sig}^{(i^+)} = (\mathbf{t}^+, \mathbf{v}_{1,2}^{(i^+)}, \mathbf{v}_2^{(i^+)}, \mathbf{v}_3^{(i^+)})$.

We now use the rejection sampling result of Lemma 1.26 to argue on the views of G_5 and G_6. For that we simply need to ensure that $s_1 \geq \gamma \left\|\mathbf{Sm}^{(i^+)}\right\|_2$ for $M = \exp(\pi/\gamma^2)$. This is subsumed by the condition

$$s_1 \geq \gamma \cdot \sqrt{nd} \cdot \sqrt{nm},$$

as we enforce the bound on $\mathbf{Sm}^{(i^+)}$. For the correctness and security of sampling, we also need $s_1 \geq \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2 - 1)} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$. Depending on the value of m, we choose

s_1 and γ as follows. If $\sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)}\cdot\frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6) > \sqrt{\pi/\ln(2)}\cdot n\sqrt{dm}$, we set $s_1 = \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)}\cdot\frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)$, and

$$\gamma = \frac{s_1}{n\sqrt{dm}} = \frac{\sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)}\cdot\frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)}{n\sqrt{dm}}.$$

On the other hand, if the inequality is not verified we set $\gamma = \sqrt{\pi/\ln(2)}$, and

$$s_1 = \gamma n\sqrt{dm},$$

which indeed satisfies the sampler's requirements as we have

$$s_1 \geq \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)}\cdot\frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6).$$

In both cases, this ensures that $s_1 \geq \gamma\left\|\mathbf{Sm}^{(i^+)}\right\|_2$ for some $\gamma \geq \sqrt{\pi/\ln(2)}$. Note however that in the first case, it can lead to γ much larger than $\sqrt{\pi/\ln(2)}$ if m is small, which in turn yields a smaller repetition rate M. Both conditions can be expressed as

$$s_1 = \max\left(\sqrt{\frac{\pi}{\ln(2)}}n\sqrt{dm}, \sqrt{2s_{\mathbf{G}}^4/(s_{\mathbf{G}}^2-1)}\cdot\frac{7}{10}(\sqrt{2nd}+\sqrt{ndk}+6)\right),$$
$$\gamma = \frac{s_1}{n\sqrt{dm}}.$$

Based on these parameter constraints, we use Lemma 1.26 to argue that conditioned on not aborting, the distributions are identical. Hence, the view of $\mathcal{A}$ in G_5 and G_6 are identical.

From the previous game hops, we already have

$$\mathrm{Adv}_{❷}[\mathcal{A}] \leq m\varepsilon_{\text{M-LWE}} + 2MC\frac{1+\varepsilon}{1-\varepsilon}\mathrm{Adv}_{G_6}[\mathcal{A}] + \mathsf{negl}(\lambda). \tag{6.10}$$

At this point, we use the same hybrid argument that of the proof of Theorem 6.3. That is we are aiming to replace the public key $\mathbf{B} = \mathbf{AR} \bmod qR$ by $\mathbf{B} = \mathbf{AR} + \mathfrak{t}^+\mathbf{G} \bmod qR$. In order to do so while keeping the ability answer signing queries for $i \neq i^+$, we use the exact same sequence of games $G_{j,0}$ to $G_{j,9}$ for $j \in [\![d]\!]$ but by keeping the modifications we made up to G_6. Since the trapdoor is not used in the i^+-th query, we are able to perform these modifications.

Using the exact same reasoning, we have $G_{1,0} = G_6$, $G_{j,9} = G_{j+1,0}$ for all $j \in [\![d-1]\!]$, and it holds that $\mathrm{Adv}_{G_{j,0}}[\mathcal{A}] \leq h(\mathrm{Adv}_{G_{j,9}}[\mathcal{A}])$ where h is the same function as that of Theorem 6.3. As a result, we get

$$\mathrm{Adv}_{G_6}[\mathcal{A}] \leq h^{\circ d}(\mathrm{Adv}_{G_{j,9}}[\mathcal{A}]). \tag{6.11}$$

We end up with the following game.

$G_{d,9}$

Setup

1. $\forall i \in [\![Q]\!], \mathsf{t}^{(i)} = \mathsf{F}(\mathsf{st} + i - 1)$
2. $i^+ \hookleftarrow U([\![Q]\!]), \mathsf{t}^+ = \mathsf{t}^{(i^+)}$
3. $\mathbf{R} \hookleftarrow \mathcal{B}_1^{2d \times dk}$ such that $\|\mathbf{R}\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)$
4. $\mathbf{B} \leftarrow \mathbf{AR} + \mathsf{t}^+\mathbf{G} \bmod qR$
5. $\mathbf{A}_3 \hookleftarrow U(R_q^{d \times k})$
6. $\mathbf{S} \hookleftarrow \mathcal{B}_1^{2d \times m}$
7. $\mathbf{D} \leftarrow \mathbf{AS} \bmod qR$
8. $\mathbf{v}_1 \hookleftarrow \mathcal{D}_{R^{2d}, s_1}, \mathbf{v}_2 \hookleftarrow \mathcal{D}_{R^{dk}, s_2}, \mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$
9. $\mathbf{u} \leftarrow \mathbf{Av}_1 + (\mathsf{t}^+\mathbf{G} - \mathbf{B})\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 \bmod qR$

Queries

Given $\mathbf{m}^{(i)} \in T_1^m$, get $\mathsf{t}^{(i)}$. Then

If $i \neq i^+$:

1. $\mathbf{v}_3^{(i)} \hookleftarrow \mathcal{D}_{R^k, s_2}$
2. $(\mathbf{v}_{1,1}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_3^{(i)}) \leftarrow \mathsf{EllipticSampler}(\mathbf{R}, \mathbf{A}', \mathbf{u} + \mathbf{Dm}^{(i)} - \mathbf{A}_3\mathbf{v}_3^{(i)}, \mathsf{t}^{(i)} - \mathsf{t}^+, s_1, s_2)$
3. Send $\mathsf{sig}^{(i)} = (\mathsf{t}^{(i)}, \mathbf{v}_{1,2}^{(i)}, \mathbf{v}_2^{(i)}, \mathbf{v}_3^{(i)})$.

If $i = i^+$:

1. If $\left\|\mathbf{Sm}^{(i^+)}\right\|_2 > \sqrt{nd}\left\|\mathbf{m}^{(i^+)}\right\|_2$, then **abort**
2. $\rho \hookleftarrow U((0,1))$
3. $A \leftarrow \langle \mathbf{v}_1 + \mathbf{Sm}^{(i^+)}, \mathbf{Sm}^{(i^+)} \rangle$
4. If $A < 0$ or if $\rho > \frac{1}{M} \exp\left(\frac{\pi}{s_1^2}\left(\left\|\mathbf{Sm}^{(i^+)}\right\|_2^2 - 2A\right)\right)$, then **abort**.
5. Otherwise, set $\begin{bmatrix} \mathbf{v}_{1,1}^{(i+)} \\ \mathbf{v}_{1,2}^{(i^+)} \end{bmatrix} = \mathbf{v}_1 + \mathbf{Sm}^{(i^+)}$, and $\mathbf{v}_2^{(i^+)} = \mathbf{v}_2$, $\mathbf{v}_3^{(i^+)} = \mathbf{v}_3$.
6. Send $\mathsf{sig}^{(i^+)} = (\mathsf{t}^+, \mathbf{v}_{1,2}^{(i^+)}, \mathbf{v}_2^{(i^+)}, \mathbf{v}_3^{(i^+)})$.

Bounding the advantage. We now need to bound $\mathrm{Adv}_{G_{d,9}}[\mathcal{A}]$. For that we use an adversary in $G_{d,9}$ can be used to construct an adversary $\mathcal{B}$ to solve M-SIS$_{n,d,2d+k,q,\beta_{❷}}$. Given the M-SIS instance, $\mathcal{B}$ parses it into $[\mathbf{I}_d|\mathbf{A}'|\mathbf{A}_3]$ and uses these elements to simulate the challenger in $G_{d,9}$. After the queries stage, it receives a type ❷ forgery from $\mathcal{A}$, i.e., it receives a forgery $\mathsf{sig}^\star = (\mathsf{t}^\star, \mathbf{v}_{1,2}^\star, \mathbf{v}_2^\star, \mathbf{v}_3^\star)$ on $\mathbf{m}^\star$ such that $\mathsf{SEP}^\star.\mathsf{Verify}(\mathsf{pk}, \mathsf{sig}^\star, \mathbf{m}^\star) = 1$. At this point, if $\mathsf{t}^\star \neq \mathsf{t}^+$ then $\mathcal{B}$ aborts which happens with probability $1 - 1/Q$. Then, it also aborts if $\|\mathbf{R} \cdot \mathbf{\Delta v}_2\|_2 > \sqrt{nd}\|\mathbf{\Delta v}_2\|_2$ or $\|\mathbf{S} \cdot \mathbf{\Delta m}\|_2 > \sqrt{nd}\|\mathbf{\Delta m}\|_2$, where $\mathbf{\Delta v}_2 = \mathbf{v}_2^{(i^+)} - \mathbf{v}_2^\star$ and $\mathbf{\Delta m} = \mathbf{m}^{(i^+)} - \mathbf{m}^\star$. Because $\mathbf{R}, \mathbf{S}$ are independent and hidden in $\mathbf{B}$ and $\mathbf{D}$ respectively under M-LWE, Heuristic 1.3 gives that the bounds are verified with probability at least $1/C^2$ for a small constant C (typically $C = 2$ in our parameter setting). Hence this step aborts with probability at most $1 - 1/C^2$. If it did not abort, it computes

$$\mathbf{v}_{1,1}^\star = \mathbf{u} + \mathbf{Dm}^\star - (\mathbf{A}'\mathbf{v}_{1,2}^\star + (\mathsf{t}^\star\mathbf{G} - \mathbf{B})\mathbf{v}_2^\star + \mathbf{A}_3\mathbf{v}_3^\star) \bmod qR,$$

and defines $\mathbf{v}_1^\star = [\mathbf{v}_{1,1}^{\star T}|\mathbf{v}_{1,2}^{\star T}]^T$. Since $\mathsf{t}^\star = \mathsf{t}^+$, we have that $\mathsf{t}^\star\mathbf{G} - \mathbf{B} = -[\mathbf{I}_d|\mathbf{A}']\mathbf{R} \bmod qR$. Also, because verification passes, we know that $\|\mathbf{v}_1^\star\|_2, \|\mathbf{v}_3^\star\|_2$ are bounded by B_1, B_3 respectively. Then, by definition of $\mathbf{u}$ and the i^+-th query seen by the attacker, we can re-write this equation as

$$\mathbf{Av}_1^\star - \mathbf{ARv}_2^\star + \mathbf{A}_3\mathbf{v}_3^\star = \mathbf{A}(\mathbf{v}_1^{(i^+)} - \mathbf{Sm}^{(i^+)}) - \mathbf{ARv}_2^{(i^+)} + \mathbf{A}_3\mathbf{v}_3^{(i^+)} + \mathbf{ASm}^\star \bmod qR,$$

which leads to

$$[\mathbf{I}_d|\mathbf{A}'|\mathbf{A}_3]\mathbf{x}^\star = \mathbf{0} \bmod qR,$$

$$\text{where } \mathbf{x}^\star = \begin{bmatrix} (\mathbf{v}_1^{(i^+)} - \mathbf{v}_1^\star) - \mathbf{R}(\mathbf{v}_2^{(i^+)} - \mathbf{v}_2^\star) - \mathbf{S}(\mathbf{m}^{(i^+)} - \mathbf{m}^\star) \\ \mathbf{v}_3^{(i^+)} - \mathbf{v}_3^\star \end{bmatrix}.$$

There, we use the same argument as in Theorem 6.2 to argue that $\mathbf{x}^\star \neq \mathbf{0}$ with overwhelming probability. More precisely, since $\mathbf{m}^{(i^+)} \neq \mathbf{m}^\star$, at least one column $\mathbf{s}^\star$ of $\mathbf{S}$ appears in $\mathbf{x}^\star$. Yet,

$\mathbf{S}$ is hidden in $\mathbf{D}$ at the exception of at most one bit due to the rejection sampling leak of the sign of A. This only incurs a one bit loss on the conditional entropy of $\mathbf{s}^\star$, which is thence still unpredictable resulting in $\mathbf{x}^\star \neq \mathbf{0}$ with overwhelming probability. Finally, it holds that

$$\|\mathbf{x}^\star\|_2^2 \leq \left(2B_1 + \sqrt{nd} \cdot 2B_2 + \sqrt{nd} \cdot \sqrt{nm}\right)^2 + (2B_3)^2 = \beta_❷^2,$$

where the inequality holds based on the Gaussian tail bound from Lemma 1.21 and the Johnson-Lindenstrauss bound from Heuristic 1.3 we enforced.

It thus means that $\mathbf{x}^\star$ is a solution to M-SIS$_{n,d,2d+k,q,\beta_❷}$ and the advantage of $\mathcal{B}$ is linked to $\mathrm{Adv}_{G_{d,9}}[\mathcal{A}]$ by

$$\mathrm{Adv}_{G_{d,9}}[\mathcal{A}] \leq QC^2\mathrm{Adv}_{\text{M-SIS}}[\mathcal{B}] + \mathsf{negl}(\lambda) \leq QC^2\varepsilon_{\text{M-SIS}} + \mathsf{negl}(\lambda), \tag{6.12}$$

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound. Combining Equations (6.10), (6.11), (6.12), and the fact that h is non-decreasing, yields the result.

6.4.3 Performance Gains

We follow the same methodology as that of Section 6.2.4 to select parameters. We still aim for $\lambda = 128$ and $Q = 2^{32}$ with $m = 10 + 2d$. As for the statistical construction, we need to evaluate the security loss between the M-SIS (and M-LWE) assumption and the unforgeability. This loss still features the factors $|\mathcal{T}_w| - Q$ and Q of type ❶ and type ❷ forgeries respectively due to the tag guess.

The forgery reduction loss from Theorems 6.3 and 6.4 however involves the d-th functional power of the function h, which stacks up the exponents $1 - 1/2\lambda$. It makes it slightly less intuitive to see why these d compositions do not deteriorate the reduction loss too much. For the sole sake of simplifying this intuition, we give the following bound on $h^{\circ d}$. We insist that this bound is used only to justify that the reduction is controlled despite the hybrid argument, but in the parameter selection we compute $h^{\circ d}$ exactly.

Lemma 6.4 (Bounding the Reduction Loss Function)

Let a, b, c, μ be positive reals, and let α be in $(0, 1)$. We define the function h over $\mathbb{R}^+$ as

$$h : x \in \mathbb{R}^+ \mapsto a + \mu\left(b + \mu(c + x)^\alpha\right)^\alpha.$$

Then, for all positive integer d, it holds that for all $x \geq 0$

$$h^{\circ d}(x) \leq \mu^{\frac{1}{1-\alpha}} \sum_{j\in[d]} \left(\left(\mu^{\frac{-1}{1-\alpha}}a\right)^{\alpha^{2j-2}} + \left(\mu^{\frac{-1}{1-\alpha}}b\right)^{\alpha^{2j-1}} + \left(\mu^{\frac{-1}{1-\alpha}}c\right)^{\alpha^{2j}}\right) + \mu^{\frac{1-\alpha^{2d}}{1-\alpha}} x^{\alpha^{2d}}$$

Proof (Lemma 6.4). We proceed by induction on d. For $d = 1$, we need to prove that $h(x) \leq \mu^{1/(1-\alpha)} \cdot ((\mu^{-1/(1-\alpha)}a) + (\mu^{-1/(1-\alpha)}b)^\alpha + (\mu^{-1/(1-\alpha)}c)^{\alpha^2}) + \mu^{(1-\alpha^2)/(1-\alpha)}x^{\alpha^2}$ which can be re-written as $h(x) \leq a + \mu b^\alpha + \mu^{1+\alpha}(c^{\alpha^2} + x^{\alpha^2})$. The inequality follows by the non-increasing property of p-norms for $p > 0$, that is $0 < p \leq q$ implies $\|\cdot\|_q \leq \|\cdot\|_p$. Here, we thus have $\|\cdot\|_1 \leq \|\cdot\|_\alpha$ as $\alpha < 1$, and thus $(\sum x_i)^\alpha \leq \sum x_i^\alpha$ for non-negative x_i. Hence, we get that for all $x \geq 0$

$$h(x) \leq a + \mu(b^\alpha + (\mu(c + x)^\alpha)^\alpha) \leq a + \mu(b^\alpha + \mu^\alpha(c + x)^{\alpha^2}) \leq a + \mu b^\alpha + \mu^{1+\alpha}(c^{\alpha^2} + x^{\alpha^2}).$$

Now let us look at the induction step. Assume the inequality is verified at rank $d \geq 1$. Let $x \geq 0$. We have $h^{\circ(d+1)}(x) = h(h^{\circ d}(x))$. From the above, we get

$$h^{\circ(d+1)}(x) \leq a + \mu b^\alpha + \mu^{1+\alpha}c^{\alpha^2} + \mu^{1+\alpha}(h^{\circ d}(x))^{\alpha^2}.$$

Then, the induction hypothesis and the inequality $\|\cdot\|_1^{\alpha^2} \leq \|\cdot\|_{\alpha^2}^{\alpha^2}$ yields

$$h^{\circ d}(x)^{\alpha^2} \leq \mu^{\frac{\alpha^2}{1-\alpha}} \sum_{j\in[d]} \left(\left(\mu^{\frac{-1}{1-\alpha}}a\right)^{\alpha^{2j}} + \left(\mu^{\frac{-1}{1-\alpha}}b\right)^{\alpha^{2j+1}} + \left(\mu^{\frac{-1}{1-\alpha}}c\right)^{\alpha^{2j+2}}\right) + \mu^{\frac{\alpha^2-\alpha^{2d+2}}{1-\alpha}} x^{\alpha^{2d+2}}.$$

Then, because $1+\alpha+\alpha^2/(1-\alpha) = 1/(1-\alpha)$, and by reindexing the sum, we obtain

$$\mu^{1+\alpha} h^{\circ d}(x)^{\alpha^2} \leq \mu^{\frac{1}{1-\alpha}} \sum_{j\in[2,d+1]} \left(\left(\mu^{\frac{-1}{1-\alpha}} a\right)^{\alpha^{2j-2}} + \left(\mu^{\frac{-1}{1-\alpha}} b\right)^{\alpha^{2j-1}} + \left(\mu^{\frac{-1}{1-\alpha}} c\right)^{\alpha^{2j}} \right) + \mu^{\frac{1-\alpha^{2d+2}}{1-\alpha}} x^{\alpha^{2d+2}}.$$

Finally, we observe that $a+\mu b^{\alpha}+\mu^{1+\alpha}c^{\alpha^2}$ is equal to the missing term $\mu^{1/(1-\alpha)}\cdot((\mu^{-1/(1-\alpha)}a)+(\mu^{-1/(1-\alpha)}b)^{\alpha}+(\mu^{-1/(1-\alpha)}c)^{\alpha^2})$ which concludes the proof.

Applying the above lemma for $\alpha = 1-1/2\lambda$, $a = c = k\varepsilon_{\text{M-LWE}}$, $b = 2a = 2k\varepsilon_{\text{M-LWE}}$ and $\mu = \delta'$, we can bound the corresponding loss terms from Theorems 6.3 and 6.4. The additive term depending on a, b, c can be bounded by $\varepsilon_+ = d\mu^{1/(1-\alpha)}((a/\mu^{1/(1-\alpha)})^{\alpha^{2d-1}} + (b/\mu^{1/(1-\alpha)})^{\alpha^{2d-2}} + (c/\mu^{1/(1-\alpha)})^{\alpha^{2d}})$ which in our case yields about a 10.6 bit loss compared to $\varepsilon_{\text{M-LWE}} = 2^{-158}$. We then have that $h^{\circ d}(C\lfloor\mathcal{T}_w\rfloor - Q)\varepsilon_{\text{M-SIS}}) \leq \varepsilon_+ + \delta'^{(1-\alpha^{2d})/(1-\alpha)}(C\lfloor\mathcal{T}_w\rfloor - Q)\varepsilon_{\text{M-SIS}})^{\alpha^{2d}}$. We can then plug this bound and obtain the required M-SIS hardness to achieve an advantage of $2^{-\lambda}$. In particular, for the parameters given in Table 6.3, we get that $\varepsilon_{\text{M-SIS}}$ should be smaller than $2^{-165.146377}$ to ensure an advantage of at most 2^{-128} against type ❶ forgeries using the bounds we provide in this section. This is not far from the thorough parameter selection which gives a value of $2^{-165.146376}$. Doing the same for type ❷ forgeries would give $2^{-168.809055}$ instead of $2^{-168.809049}$. Lemma 6.4 thus tightly approximate the actual reduction loss. As opposed to the previous construction, the discrepancy between the security losses in type ❶ and ❷ is much smaller. This discrepancy previously led us to choose parameters that highly overshoots the requirement for type ❶ security (see Table 6.2). Beyond the relative difference between the two types, our new security reduction only incurs a loss of at most 41 bits between M-SIS and the actual security. This is much more acceptable than 85 bits in Section 6.2.

We give in Table 6.3 a parameter set meeting these security constraints. As before, we choose the parameters to be later used in an anonymous credentials system in Chapter 7, and the choice is also driven by our implementation of Chapter 8 in some aspects. One could choose slightly tighter parameters for the standalone signature. But we again insist that the signature is meaningless if not plugged into privacy-preserving protocols and applications.

6.5 Conclusion

Practical signatures with efficient protocols [CL04, BB08, PS16] have been successfully implemented in the classical setting, and led to extremely efficient privacy-driven constructions that have even been standardized [ISO13a, ISO13b]. Up until a few years ago, only one theoretical alternative existed in the post-quantum setting [LLM+16]. Although recent works improved the applications themselves [dPLS18, BEF19, CKLL19, LNPS21, dPK22, BLNS23a, BLNS23b], it was through dedicated constructions. The work presented in this chapter proposed two waves of improvements for post-quantum signatures with efficient protocols which are much more versatile than the above.

At this stage, we only presented the signature itself. Even though the signature size is fairly reasonable (only 2.9 times larger than the selected standard Dilithium [DKL+18]), the main size metric to consider is that of the applications the SEP is plugged into. To demonstrate the implications of our work, we decide to look at the general yet concrete application of anonymous credentials. They indeed encompass the constraints of many different use cases, such as signing numerous attributes, that are possibly secret, that can be arbitrary (e.g., low entropy, as opposed to DAA [CKLL19]), etc. An important metric of such systems is the size of a zero-knowledge proof of signature verification. Optimizing the SEP thus plays a crucial role in optimizing the subsequent anonymous credentials. In particular, the drastic performance improvements over the construction from Section 6.2, and in turn even more impressive over [LLM+16], leads to a practical post-quantum alternative for classical SEPs [CL04, BB08, PS16]. We showcase it through anonymous credentials in Chapter 7 and their implementation in Chapter 8.

Symbol	Description	Value
	Signature Parameters	
λ	Security parameter	128
n	Ring degree	256
d	M-SIS Module rank	4
(k, b)	Gadget length and base	$(5, 14)$
m	Message length	$10 + 2d = 18$
q	Modulus	$425801 \approx 2^{18.7}$
κ	Number of prime ideal factors of qR	4
s_1	Top preimage sampling width	5854.109
s_2	Bottom preimage sampling width	68.170
w	Hamming weight of tags	5
Q	Maximal number of signature queries	2^{32}
B_1	First verification bound ($\|\mathbf{v}_1\|_2$)	128673.751
B_1'	First verification bound, hiding case ($\|\mathbf{v}_1 - \mathbf{r}_u\|_2$)	128719.006
B_2	Second verification bound ($\|\mathbf{v}_2\|_2$)	2210.639
B_3	Third verification bound ($\|\mathbf{v}_3\|_2$)	1242.684
	Efficiency Estimates	
$\|\mathsf{sk}\|$	Secret key size (KB)	10 KB
$\|\mathsf{pk}\|$	Public key size (KB)$^{(*)}$	47.53 KB
$\|\mathsf{sig}\|$	Signature size (KB)	6.81 KB
	Security Estimates	
$\lambda_{❶}/\lambda_{❷}$	Security targets for M-SIS (type ❶/❷, Theorems 6.3/6.4)	166/169
$\mathrm{BKZ}_{❶}/\mathrm{BKZ}_{❷}$	BKZ blocksize for M-SIS (type ❶/❷)	653/560
$\lambda^\star_{❶}/\lambda^\star_{❷}$	Reached M-SIS classical security (type ❶/❷)	207/179
$\lambda^\star_{\text{M-LWE}}$	Reached M-LWE classical security (Key Rec.)	158

Table 6.3: Suggested parameter set for the optimized signature of Section 6.4.
$^{(*)}$ The public key size only contains $\mathbf{B}$. The other public elements $\mathbf{A}, \mathbf{A}_3, \mathbf{D}, \mathbf{u}$ are stored via a 32-byte seed in the public parameters and shared between signers.

7

Anonymous Credentials from Lattices

Building upon our *signature with efficient protocols* of Chapter 6, we propose generic protocols that can be declined for several privacy primitives such as group signatures, blind signatures, etc. We then give a construction of anonymous credentials which represents a general yet representative use case of this type of signatures. Our anonymous credentials system is competitive with existing ones while still relying on standard assumptions.

The work presented in this chapter is based on two papers with my co-authors Sven Argo, Tim Güneysu, Georg Land, Adeline Roux-Langlois and Olivier Sanders.

[JRS23] **Lattice Signature With Efficient Protocols, Application to Anonymous Credentials**. Published at Crypto 2023. Co-authored only with Adeline Roux-Langlois, and Olivier Sanders.

[AGJ+24] **Practical Post-Quantum Signatures for Privacy**. Published at ACM CCS 2024.

Contents

7.1 Introduction

Balancing security and privacy has become a growing concern within the cryptographic community but also on a larger scale, as mentioned in Section 6.1. Protecting the privacy of users through controlling the circulation of personal data while retaining the ability to guarantee their authenticity calls for advanced mechanisms. This has led to the constructions of many privacy-oriented primitives such as blind signatures [Cha82], group signatures [CvH91, BSZ05], or anonymous credentials [Cha85, CL01, CL04, FHS19]. Each of these serve different security purposes

and use-cases, but the common goal is to provide the means for somewhat anonymous authentication. Such systems have already been successfully implemented in industrial applications[1] and standards [TCG15].

Anonymous credentials, sometimes called attribute-based credential, provide a rather generic framework encompassing a wide spectrum of such privacy-preserving systems. They usually describe a system where users can obtain a certificate on multiple attributes from an issuer, and later authenticate to verifiers with this certificate in an anonymous manner.

The design and requirements of anonymous credentials, namely unforgeability and anonymity, strongly match the paradigm of signature with efficient protocols [CL02] presented in Chapter 6. And for good reason, the latter was mostly introduced to design the former. Since then, optimizing SEPs [ASM06, PS16] has become a way to produce more efficient anonymous credentials for concrete use. These designs, as detailed in Chapter 6, rely on a sufficiently algebraic signature scheme which can be used to sign commitments and whose verification circuit is provable in zero-knowledge. The latter intuitively helps in meeting the anonymity and unlinkability properties, while unforgeability is (mostly) inherited from that of the underlying signature. Other works have also proposed anonymous credentials through cryptographic accumulators [FHS19] or unlinkable redactable signatures [CDHK15, San20]. Unfortunately, all the designs cited above are based on mathematical assumptions that do not withstand quantum algorithms.

Before the beginning of this thesis, no explicit post-quantum anonymous credentials systems were known, besides the implicit one that could be obtained from the SEP of Libert et al. [LLM+16]. The latter however suffers from an extremely high complexity as the credential proof size, which corresponds to the size of a zero-knowledge proof of signature verification, approaches 700 MB. Our first work on the subject [JRS23], quickly followed by concurrent works [BLNS23b, LLLW23, BCR+23], kick-started a line of research towards practical post-quantum anonymous credentials. The SEP we introduced in Section 6.2 led to the first explicit anonymous credential system relying on standard post-quantum assumptions, featuring relatively short zero-knowledge proofs. When plugged in an anonymous credential framework, it results in a presentation transcript of about 660 KB which is a considerable improvement over [LLM+16]. Soon after, Bootle et al. [BLNS23b] managed to reduce this size to around 240 KB (or 60 KB when relying on an interactive assumption) but at the cost of relying on new ad-hoc computational assumptions. Similarly, [LLLW23] considers different security models to achieve different sizes ranging from 200 KB to 25 MB. Finally, [BCR+23] builds upon the groupe signature of [dPLS18] to design anonymous credentials on standard assumptions, but suffers from large proofs of around 2 MB. These approaches are then complementary as they share the same goal, but with a different trade-off between security and efficiency.

7.1.1 Our Contributions

In this chapter, we build upon our optimized signature with efficient protocols of Section 6.4 and construct an anonymous credentials system that benefits from both standard and non-interactive security assumptions and competitive compactness. More precisely, our system is adaptively secure, relies on the M-SIS, M-ISIS and M-LWE assumptions and achieve presentation transcripts of slightly under 80 KB. Also, we make design and parameter choices so that it can be efficiently implemented and yield practical timings for most use-cases. The implementation is the object of Chapter 8.

Our framework can be instantiated using either of the SEPs introduced in Chapter 6. The difference mainly comes from the signature scheme we plug into the framework (and thus the relations proven in zero-knowledge). We therefore present the system obtained from the optimized signature of Section 6.4 and only discuss the one gotten from the statistical signature in Section 7.5. A full comparison with other post-quantum anonymous credentials [BLNS23b, LLLW23, BCR+23] is also deferred to the latter section. Let us now give a few details on how to obtain anonymous credentials from our SEP scheme.

Generic Protocols: Oblivious Signing and Prove

The starting point in building our anonymous credentials from the signatures with efficient protocols of Chapter 6 is to design the generic protocols (**P1**) and (**P2**) mentioned in Section 6.1.

Protocol (**P1**) is an interaction between a user and a signer in which a signature is produced on a message the user has committed to in a hiding way. Our signatures were designed with an Ajtai commitment [Ajt96] in mind so as to straightforwardly allow for such oblivious signing. Indeed,

[1] Microsoft U-Prove Cryptographic Token

we observe that the commitment step $\mathbf{c} = \mathbf{A}\mathbf{r}_u + \mathbf{D}\mathbf{m} \bmod qR$ is performed at the very beginning of the signing procedure, before the secret key is even needed. As a result, one can delegate this commitment phase to an external user. Signing is concluded by merging the randomness $\mathbf{r}_u$ to the sampled preimage $\mathbf{v}'$, which can also be performed by the user. This then depicts the exact protocol where the user would first commit to $\mathbf{m}$, send the commitment $\mathbf{c}$, get back a signature $(\mathfrak{t}, \mathbf{v}')$, and complete the signature by merging $\mathbf{v}'$ and $\mathbf{r}_u$. In order to rely on the unforgeability of the signature scheme, the security proof needs to extract $\mathbf{r}_u + \mathbf{S}\mathbf{m}$ for the tag $\mathfrak{t}^+$ the signer does not have a trapdoor for. We thus include a zero-knowledge proof of the commitment opening $(\mathbf{r}_u, \mathbf{m})$ ensuring that $\mathbf{c}$ is well-formed. The knowledge extractor resulting from the soundness of the proof system would then conclude the security proof.

The second protocol (**P2**) simply requires the ability to prove knowledge of a signature $(\mathfrak{t}, \mathbf{v})$ on a message $\mathbf{m}$. Again, the verification of our signatures was thought to be sufficiently algebraic to be proven in zero-knowledge using the latest framework from Lyubashevsky et al. [LNP22].

On the Security of the Protocols

At this stage, coherently with previous works, we do not identify any properties expected from the protocols (**P1**) and (**P2**) above nor prove any results regarding their security. As this might look unconventional, we need to recall a few facts about SEPs and their use in privacy-preserving applications.

The use of signature schemes in the latter applications can be done based on formal generic frameworks, e.g., [BSZ05] for group signature or [BPS19] for e-cash, or on some rather common heuristics, e.g., for anonymous credentials [CL04]. In all cases, the point is that, in theory, no specific property is expected from the signatures beyond EUF-CMA security. Typically, [BSZ05] and [BPS19] consider standard digital signature schemes for their framework. However, in practice, the use of any digital signature is likely to lead to a totally impractical construction because of the difficult interactions between general-purpose signatures and the other building blocks such as zero-knowledge proofs. This is where SEPs prove handy. They are specifically designed to smoothly interact with the other building blocks so as to optimize the efficiency of the resulting construction.

In this context, defining security notions that such protocols should achieve would be meaningless as no such formal properties are expected by the constructions using them. Worse, this is likely to lead to unnecessary complications as it is difficult to define a relevant security model for SEPs. Typically, an SEP allows one to get a signature on hidden messages and then to prove knowledge of the message-signature pair. How to define a relevant security model in this context? Unforgeability indeed means the inability to produce a signature on new messages but here we do not know the messages requested by the adversary to the signing oracle and we do not know which message-signature pairs it is proving knowledge of. In other words, we cannot decide if the adversary won.

The work of Libert et al. [LLM+16] circumvents this issue by forcing the user to provide an encryption of the messages in the blind issuance process. This does not address the problem of formalizing the properties expected from the protocols (**P1**) and (**P2**) of SEPs (and indeed [LLM+16] does not define such properties) but this enables to provide some results regarding security as a reduction can recover all the messages it has signed (by decrypting the ciphertexts) and thus decide when a forgery occurs. Besides being unconventional (this led [LLM+16] to prove "security" of the protocols without defining what "security" means), this approach complicates the protocols by adding this encryption step that is not necessary in most applications using such signatures. Indeed, in concrete applications, this problem is usually solved by other means. For example, in e-cash systems, "forgeries" can be detected by comparing the amount of withdrawn coins with the one of spent coins. In group signatures, there is an opening procedure that allows to trace back a group signature to a group member. This enables to detect forgeries as the latter will be involved in group signatures that cannot be opened to anyone.

To sum up, defining specific security properties for the protocols associated with SEPs is not necessary for privacy-preserving applications and artificially increases complexity. Instead, and in accordance with previous works, we do not consider such generic security properties, but only specific ones for anonymous credentials in Section 7.3.

Anonymous Credentials

Our anonymous credentials construction almost directly follows from the SEPs of Chapter 6 and the generic protocols we discussed above. Although, there is no unified security model for the generic

protocols, we need to consider one for the credential system. We follow the model introduced in [FHS19] recalled in Section 1.5.2. We then need to provide a small modification to the generic protocol we mentioned. In the model from [FHS19], each user holds a key pair $(\mathsf{upk}, \mathsf{usk})$ generated from an algorithm $\mathsf{UKeyGen}$. The user secret key must be part of the attributes that are signed by the issuer in (**P1**). To avoid impersonations however, the user must also provide a proof of registration, meaning that they know the secret key corresponding to their public key.

We therefore modify the issuance process, or rather the commit-and-prove phase, as follows. The $\mathsf{UKeyGen}$ algorithm generates a key pair $(\mathbf{t} = \mathbf{D}_s\mathbf{s} \bmod qR, \mathbf{s})$ where $\mathbf{s}$ is uniform in T_1^{2d}, and $\mathbf{D}_s$ is uniform in $R_q^{d\times 2d}$. The user then re-uses $\mathbf{D}_s$ as part of the message commitment matrix and compute $\mathbf{c} = \mathbf{A}\mathbf{r}_u + \mathbf{D}_s\mathbf{s} + \mathbf{D}\mathbf{m} \bmod qR$. This corresponds to the generic version where the effective message is $[\mathbf{s}^T|\mathbf{m}^T]^T$ and the commitment matrix is $[\mathbf{D}_s|\mathbf{D}]$. Because $\mathbf{D}_s\mathbf{s} \bmod qR = \mathbf{t} = \mathsf{upk}$, an adversary could possibly impersonate the user from the sole knowledge of its public key. As a countermeasure, we also prove knowledge of $\mathbf{s}$ such that $\mathbf{D}_s\mathbf{s} = \mathbf{t} \bmod qR$. The security proof will then be able to detect impersonation attempts.

7.2 Generic Protocols: Oblivious Signing and Prove

We start by presenting the generic protocols (**P1**) and (**P2**) associated to our signatures with efficient protocols of Chapter 6. We call them $\mathsf{OblSign}$ and Prove respectively, and specify them for our anonymous credentials in Section 7.3. The different zero-knowledge arguments are dealt with the framework from [LNP22], and are detailed in Section 7.4.

7.2.1 Oblivious Signing Protocol

We first present the oblivious signing protocol between a signer S and a user U. The user U is interacting with S in order to obtain a signature $(\mathsf{t}, \mathbf{v})$ on a message $\mathbf{m}$, by only providing S with a commitment $\mathbf{c}$ to the message $\mathbf{m}$ and a proof of commitment opening. We assume that $\mathsf{SEP}^\star.\mathsf{Setup}$ and $\mathsf{SEP}^\star.\mathsf{KeyGen}$ (Algorithms 6.5 and 6.6) have been run prior to entering the protocol but with some slight modifications that we detail below. As explained in Section 6.4, the signature scheme is presented with a non-hiding commitment $\mathbf{D}\mathbf{m}$ but the hiding part $\mathbf{A}\mathbf{r}_u$ can be added at almost no cost. We aim at a computationally hiding commitment and choose $\mathbf{r}_u$ to be uniform over T_1^{2d}. Under the M-LWE assumption, $\mathbf{A}\mathbf{r}_u \bmod qR$ is indeed indistinguishable from uniform.

The user obtains a partial signature $(\mathsf{t}, \mathbf{v}'_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$ and can reconstruct the corresponding $\mathbf{v}'_{1,1} = \mathbf{u} + [\mathbf{I}_d|\mathbf{A}']\mathbf{r}_u + \mathbf{D}\mathbf{m} - \mathbf{A}'\mathbf{v}'_{1,2} - (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 - \mathbf{A}_3\mathbf{v}_3$. To obtain the full signature in the sense of $\mathsf{SEP}^\star.\mathsf{Verify}$, the user can parse $\mathbf{r}_u = [\mathbf{r}_{u,1,1}^T|\mathbf{r}_{u,1,2}^T]^T$ and re-write the equation as

$$(\mathbf{v}'_{1,1} - \mathbf{r}_{u,1,1}) = \mathbf{u} + \mathbf{D}\mathbf{m} - \mathbf{A}'(\mathbf{v}'_{1,2} - \mathbf{r}_{u,1,2}) - (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 - \mathbf{A}_3\mathbf{v}_3.$$

By defining $\mathbf{v}_{1,i} = \mathbf{v}'_{1,i} - \mathbf{r}_{u,1,i}$ for $i \in [\![2]\!]$, the verification then recomputes $\mathbf{v}_1 = [\mathbf{v}_{1,1}^T|\mathbf{v}_{1,2}^T]^T = \mathbf{v}'_1 - \mathbf{r}_u$. As a result, we need to adjust the verification bound B_1 on $\mathbf{v}_1$ in $\mathsf{SEP}^\star.\mathsf{Verify}$. In addition, we also need to slightly adjust the rejection sampling condition on s_1 for the reduction of Theorem 6.4 to go through because the randomness from the user is now part of the vector we perform rejection sampling on in the i^+-th query. As such we change $\mathsf{SEP}^\star.\mathsf{Setup}$ (Algorithm 6.5) with

$$s_1 = \max\left(\sqrt{\frac{\pi}{\ln 2}}\left(n\sqrt{dm} + \sqrt{2nd}\right), \sqrt{\frac{2s_{\mathbf{G}}^4}{s_{\mathbf{G}}^2 - 1}} \cdot \frac{7}{10}(\sqrt{2nd} + \sqrt{ndk} + 6)\right),$$
$$\gamma = \frac{s_1}{n\sqrt{dm} + \sqrt{2nd}},$$

and the verification bound becomes $B'_1 = B_1 + \sqrt{2nd} = c_{2nd}s_1\sqrt{2nd} + \sqrt{2nd}$. To avoid confusion, we call $\mathsf{SEP}^\star.\mathsf{Verify}'$ the modified verification where the bound B_1 is replaced by B'_1.

Also, as we are now considering a hiding commitment, the randomness commitment matrix $\mathbf{A}$, which is shared with the signature scheme, must be perfectly uniform without being tampered with. In particular, we need to ensure that no one has embedded a trapdoor within it. To do so, we generate $\mathbf{A}'$ (in $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$) as the hash of a public string. In the random oracle model, the matrix can be assumed to follow the prescribed uniform distribution over $R_q^{d\times d}$. We note that this is usually done in practice to compact the storage of $\mathbf{A}'$ to a public seed. This is in particular what we do in our implementation of Chapter 8.

Algorithm 7.1: OblSign

Input: Signer S with $\mathsf{sk}, \mathsf{pk}, \mathsf{st}$, and a user U with $\mathbf{m} \in T_1^m$ and pk.

User U.
1. $\mathbf{r}_u \hookleftarrow U(T_1^{2d})$.
2. $\mathbf{c} \leftarrow \mathbf{A}\mathbf{r}_u + \mathbf{D}\mathbf{m} \bmod qR$.
3. Send $\mathbf{c}$ to S.

User $U \longleftrightarrow$ Signer S.
4. Interactive zero-knowledge argument between U and S, where U proves that $\mathbf{c}$ is commitment to $\mathbf{m}$ with randomness $\mathbf{r}_u$. If S is not convinced, the protocol aborts.

Signer S.
5. $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$.
6. $\mathsf{t} \leftarrow \mathsf{F}(\mathsf{st})$.
7. $\mathbf{v}' \leftarrow \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u} + \mathbf{c} - \mathbf{A}_3\mathbf{v}_3, \mathsf{t}, s_1, s_2)$. ▷ Algorithm 4.5
8. Parse $\mathbf{v}' = [{\mathbf{v}'_{1,1}}^T | {\mathbf{v}'_{1,2}}^T | \mathbf{v}_2^T]^T$.
9. Send $(\mathsf{t}, \mathbf{v}'_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$ to U.
10. $\mathsf{st} \leftarrow \mathsf{st} + 1$.

User U.
11. Parse $\mathbf{r}_u$ as $[\mathbf{r}_{u,1,1}^T | \mathbf{r}_{u,1,2}^T]^T$ with $\mathbf{r}_{u,1,i} \in R^d$.
12. $\mathbf{v}_{1,2} \leftarrow \mathbf{v}'_{1,2} - \mathbf{r}_{u,1,2}$.
13. **if** $\mathsf{SEP}^\star.\mathsf{Verify}'(\mathsf{pk}, \mathbf{m}, (\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3)) = 1$, **then return** $(\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$. ▷ Modified Algorithm 6.8
14. **else return** $\perp$

7.2.2 Signature Presentation Protocol

The second protocol provides a user U, who obtained a certificate $\mathsf{sig} = (\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$ on a message $\mathbf{m}$, with the ability to prove possession of this valid message-signature pair. For that, they only have to prove that $\mathsf{SEP}^\star.\mathsf{Verify}'(\mathsf{pk}, \mathbf{m}, \mathsf{sig}) = 1$ without revealing neither $\mathbf{m}$ nor sig. The protocol of Algorithm 7.2 thus simply consists in using the zero-knowledge argument from [LNP22] which we detail in Section 7.4.3. The proof can be made non-interactive in the random oracle model using the Fiat-Shamir transform. This also allows one to turn it into a signature of knowledge by including another message in the challenges of the proof. The latter is leveraged in the design of group signatures for example.

Algorithm 7.2: Prove

Input: User U with $\mathsf{pk}, \mathbf{m}, \mathsf{sig}$, and a verifier V with pk.

User $U \longleftrightarrow$ Verifier V.
1. Interactive zero-knowledge argument between U and V, where U proves knowledge of $(\mathbf{m}; \mathsf{sig})$ such that $\mathsf{SEP}^\star.\mathsf{Verify}'(\mathsf{pk}, \mathbf{m}, \mathsf{sig}) = 1$.

7.3 Our Anonymous Credentials System

Following the syntax and model of [FHS19] recalled in Section 1.5.2, we consider an issuer S in charge of emmitting credentials with a key pair generated using $\mathsf{OKeyGen}$, and a user U owning a key pair generated by $\mathsf{UKeyGen}$ and attributes they want signed. Both interact in a protocol Issue so that U can obtain a credential cred on their secret key and attributes. U can then interact with a verifier V to show their credential through a protocol Show giving U the ability to hide the credential and attributes of their choice. The two protocols Issue and Show are essentially adaption of the generic protocols $\mathsf{OblSign}$ and Prove from Algorithms 7.1 and 7.2. From the security standpoint, two properties are expected: anonymity and unforgeability. The former informally requires that Show does not leak more information than necessary, i.e., the set of disclosed attributes (which also captures the the fact that different executions of Show for the same credential with the same revealed attributes are unlinkable). The second requires that no user can claim a credential on some attributes unless it has personally received a certificate from the organization. This in particular implies that nobody can present a credential that they do not own.

7.3.1 Description

We now describe these algorithms and protocols in Algorithms 7.3, 7.4, 7.5 and 7.6. We adjust Algorithm 6.5 so that the message commitment matrix is separated into two matrices $\mathbf{D}_s \in R_q^{d \times 2d}$ and $\mathbf{D} \in R_q^{d \times m'}$ where $m' = m - 2d$ is the number of attributes. Also, our scheme features *selective disclosure* of attributes. It means that the user can decide to reveal the attributes $(m_i)_{i \in \mathcal{I}}$ for a set of index $\mathcal{I} \subseteq [\![m']\!]$. The undisclosed attributes $(m_i)_{i \notin \mathcal{I}}$ must remain hidden. We present this

feature in the Show protocol, but it could also be done for the Issue one. We decide not to in order to match the model from [FHS19] where $\mathcal{I} = [\![m']\!]$ during the issuance. Nevertheless, our protocol allows for more flexibility and, in particular, our selected parameters and implementation of Chapter 8 use $\mathcal{I} = \varnothing$ which is the least favorable case in terms of performance.

Algorithm 7.3: OKeyGen

Input: Public parameters pp as in the modified Algorithm 6.5.
Output: $(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{SEP}^\star.\mathsf{KeyGen}(\mathsf{pp})$. ▷ Algorithm 6.6

Algorithm 7.4: UKeyGen

Input: Public parameters pp as in the modified Algorithm 6.5.

1. $\mathbf{s} \hookleftarrow U(T_1^{2d})$.
2. $\mathbf{t} \leftarrow \mathbf{D}_s\mathbf{s} \bmod qR$.

Output: $(\mathsf{upk}, \mathsf{usk}) = (\mathbf{t}, \mathbf{s})$. ▷ pp is stored in upk for simplicity.

Algorithm 7.5: Issue (Credential Issuance Protocol)

Input: Organization O with $\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{st}$, and a user U with $\mathbf{m} \in T_1^{m'}$ and $\mathsf{usk}, \mathsf{upk}, \mathsf{opk}$.

User U.
1. $\mathbf{r} \hookleftarrow U(T_1^{2d})$.
2. $\mathbf{c} \leftarrow \mathbf{A}\mathbf{r} + \mathbf{D}_s\mathsf{usk} + \mathbf{D}\mathbf{m} \bmod qR$.
3. Send $\mathbf{c}$ to O.

User $U \longleftrightarrow$ Organization O.
4. Interactive zero-knowledge argument between U and O. In this syntax, i.e., [FHS19], the organization knows $\mathbf{m}$ but not usk. Hence, in the zero-knowledge argument, U proves knowledge of short $(\mathbf{r}, \mathbf{s})$ such that $\mathbf{c} - \mathbf{D}\mathbf{m} = \mathbf{A}\mathbf{r} + \mathbf{D}_s\mathsf{usk} \bmod qR$, and additionally that $\mathbf{D}_s\mathsf{usk} = \mathsf{upk} \bmod qR$. If O is not convinced, the protocol aborts. The zero-knowledge argument is described in Section 7.4.2.[a]

Organization O.
5. $\mathsf{t} \leftarrow \mathsf{F}(\mathsf{st})$.
6. $\mathbf{v}_3 \hookleftarrow \mathcal{D}_{R^k, s_2}$.
7. $\mathbf{v}' \leftarrow \mathsf{EllipticSampler}(\mathbf{R}; \mathbf{A}', \mathbf{u} + \mathbf{c} - \mathbf{A}_3\mathbf{v}_3, \mathsf{t}, s_1, s_2)$. ▷ Algorithm 4.5
8. Parse $\mathbf{v}' = [{\mathbf{v}'_{1,1}}^T | {\mathbf{v}'_{1,2}}^T | \mathbf{v}_2^T]^T$.
9. **if** $\|\mathbf{v}'_1\|_2 > B_1$ **or** $\|\mathbf{v}_2\|_2 > B_2$ **or** $\|\mathbf{v}_3\|_2 > B_3$, **repeat from** 6.
10. Send $(\mathsf{t}, \mathbf{v}'_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$ to U.
11. $\mathsf{st} \leftarrow \mathsf{st} + 1$.

User U.
12. Parse $\mathbf{r}$ as $[\mathbf{r}_{1,1}^T | \mathbf{r}_{1,2}^T]^T$ with $\mathbf{r}_{1,i} \in R^d$.
13. $\mathbf{v}_{1,2} \leftarrow \mathbf{v}'_{1,2} - \mathbf{r}_{1,2}$.
14. **if** $\mathsf{SEP}^\star.\mathsf{Verify}'\left(\mathsf{opk}; \begin{bmatrix}\mathsf{usk}\\ \mathbf{m}\end{bmatrix}; (\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3)\right) = 1$, **then return** $\mathsf{cred} = (\mathsf{t}, \mathbf{v}_{1,2}, \mathbf{v}_2, \mathbf{v}_3)$. ▷ Modified Algorithm 6.8
15. **else return** $\perp$

[a] Our framework allows for hiding the message in this zero-knowledge proof as well. We decided to hide everything in our implementation of Chapter 8.

Algorithm 7.6: Show (Credential Showing Protocol)

Input: User U with $\mathsf{usk}, \mathsf{opk}, \mathbf{m}, \mathsf{cred}, \mathcal{I}$, and verifier V with $\mathsf{opk}, (m_i)_{i\in\mathcal{I}}$.

User $U \longleftrightarrow$ Verifier V.
1. Interactive zero-knowledge argument between U and V, where U proves knowledge of $(\mathsf{usk}, (m_i)_{i\notin\mathcal{I}}, \mathsf{cred})$ such that $\mathsf{SEP}^\star.\mathsf{Verify}'(\mathsf{opk}, [\mathsf{usk}^T | \mathbf{m}^T]^T, \mathsf{cred}) = 1$. The zero-knowledge argument is described in Section 7.4.3.

Before providing the security analysis, we first verify the correctness of the anonymous credentials, that is that honest executions of Issue do not fail, and that honestly obtained credentials can be shown successfully in Show.

Lemma 7.1 (Anonymous Credentials - Correctness)

The anonymous credentials system of Algorithms 7.3 to 7.6 is correct.

Proof (Lemma 7.1). Let $\mathsf{pp} \leftarrow \mathsf{SEP}^\star.\mathsf{Setup}(1^\lambda)$. Let $(\mathsf{opk}, \mathsf{osk}) \leftarrow \mathsf{OKeyGen}(\mathsf{pp})$ and $(\mathsf{upk}, \mathsf{usk}) \leftarrow \mathsf{UKeyGen}(\mathsf{pp})$. Then, let $\mathbf{m} \in T_1^{m'}$ and $\mathcal{I} \subseteq [\![m']\!]$. We consider an honest execution of the issuance protocol $\mathsf{Issue}_{O,U}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{st}, \mathbf{m}); (\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathbf{m}))$. From the completeness of the zero-knowledge argument of knowledge (Lemma 7.2), we only have to check the abort condition of step 14. This is essentially based on the correctness of Lemma 6.3 with the updated bound B_1'. We provide the full proof for completeness.

First, note that $\mathsf{t} \in \mathcal{T}_w$ and $\widetilde{\mathbf{m}} = [\mathsf{usk}^T | \mathbf{m}^T]^T \in T_1^{m+m_s}$. Then, we define $\mathbf{v}_{1,1} = \mathbf{u} + \mathbf{D}\mathbf{m} + \mathbf{D}_s\mathbf{s} - \mathbf{A}'\mathbf{v}_{1,2} - (\mathsf{t}\mathbf{G} - \mathbf{B})\mathbf{v}_2 - \mathbf{A}_3\mathbf{v}_3 \bmod qR$. As the signature was honestly generated, it holds that $\mathbf{v}_1 = \mathbf{v}_1' - \mathbf{r}$ and that $(\mathbf{v}_1', \mathbf{v}_2)$ were obtained by a call to $\mathsf{EllipticSampler}$. Carrying the same argument as in Lemma 6.3 (relying on Lemma 4.3 and 1.21), it holds that the bounds pass with overwhelming probability thus yielding a constant number of repetitions. In particular, the vector sent to U automatically verifies $\|\mathbf{v}_1'\|_2 \leq B_1$. As a result $\|\mathbf{v}_1\|_2 \leq B_1 + \sqrt{2nd} = B_1'$, meaning that $\mathsf{SEP}^\star.\mathsf{Verify}'(\mathsf{opk}, \widetilde{\mathbf{m}}, \mathsf{sig}) = 1$.

We now consider a successful execution of the credential issuance process, i.e., $(\perp ; \mathsf{cred}) \leftarrow \mathsf{Issue}_{O,U}((\mathsf{osk}, \mathsf{opk}, \mathsf{upk}, \mathsf{st}, \mathbf{m}); (\mathsf{usk}, \mathsf{upk}, \mathsf{opk}, \mathbf{m}))$. Because it did not abort, it means that the outputted credential passed verification, i.e., that $\mathsf{SEP}^\star.\mathsf{Verify}'(\mathsf{opk}, \widetilde{\mathbf{m}}, \mathsf{cred}) = 1$. The completeness of the zero-knowledge argument (Lemma 7.5) then yields that $\mathsf{Show}_{U,V}((\mathsf{usk}, \mathsf{opk}, \mathbf{m}, \mathsf{cred}, \mathcal{I}); (\mathsf{opk}, (m_i)_{i\in\mathcal{I}}))$ outputs $(\perp, 1)$, i.e., a successful showing.

Notice that the correctness is conditioned on non-aborting zero-knowledge arguments. As long as the completeness error are not overwhelming (i.e., protocol accepts with non-negligible probability), it is not an issue. In practice (Chapter 8), we consider non-interactive proofs using the Fiat-Shamir transform which then repeats until generating a non-aborting transcript.

7.3.2 Security Analysis

We now provide the security proofs of the anonymous credentials. Notice that as opposed to the constructions of [BLNS23b] and [LLLW23], we do not require straightline extractable proofs. We elaborate in Section 7.3.3 below.

Theorem 7.1 (Anonymous Credentials - Anonymity)

The anonymous credentials of Algorithms 7.3 to 7.6 is anonymous based on the zero-knowledge property of the proof system of Section 7.4.3. More precisely, the advantage of an adversary in breaking the anonymity of the anonymous credentials is upper-bounded by $\varepsilon_{\mathrm{zk}}^{(s)}$ defined in Lemma 7.7.

Proof (Theorem 7.1). The proof simply consists in simulating the zero-knowledge proof in the Show interaction in the anonymity game, relying on the zero-knowledge property of the proof system. More formally, we define the modified game to be exactly that of Figure 1.1 except that when interacting with $\mathcal{A}$ in $\mathsf{Show}_{\mathcal{C},\mathcal{A}}((\mathsf{usk}_{j_b}, \mathsf{opk}, \mathbf{m}^{(j_b')}, \mathsf{cred}^{(j_b')}, \mathcal{I}), \cdot)$, the challenger $\mathcal{C}$ simulates the zero-knowledge argument, i.e., without resorting to usk_{j_b}, $(m_i^{(j_b')})_{i\notin\mathcal{I}}$, $\mathsf{cred}^{(j_b')}$. By Lemma 7.7, the two games can be distinguished with advantage at most $\varepsilon_{\mathrm{zk}}^{(s)}$ defined in the latter lemma.

Now, the view of $\mathcal{A}$ only depends on $(m_i^{(j_b')})_{i\in\mathcal{I}}$, which does not depend on b as we require $(m_i^{(j_0')})_{i\in\mathcal{I}} = (m_i)_{i\in\mathcal{I}} = (m_i^{(j_1')})_{i\in\mathcal{I}}$. Thence, the view of $\mathcal{A}$ is independent of b and therefore its advantage is 0. It proves that the advantage of $\mathcal{A}$ in the original anonymity game is bounded by $\varepsilon_{\mathrm{zk}}^{(s)}$, which is defined in Lemma 7.7.

Theorem 7.2 (Anonymous Credentials - Unforgeability)

The anonymous credentials of Algorithms 7.3 to 7.6 is unforgeable based on the hardness of $\text{M-LWE}_{n,d,2d,q,U(R_q),U(T_1)}$, $\text{M-ISIS}_{n,d,2d,q,\sqrt{2nd}}$, the zero-knowledge and soundness properties of the proof systems of Section 7.4.2 and 7.4.3, and on the EUF-CMA security of the signature scheme of Section 6.4. More precisely, the advantage of a PPT adversary in breaking the

unforgeability of the anonymous credentials is upper-bounded by

$$3\left(\varepsilon_{\text{zk}}^{(i)}+\varepsilon_{\text{zk}}^{(s)}+K\varepsilon_{\text{sound}}^{(i)}+3\varepsilon_{\text{sound}}^{(s)}+2\delta_q(2d,d)+\frac{\varepsilon_{\text{M-LWE}}}{1-\delta_q(2d,d)}+|\mathcal{T}_w|\,\varepsilon_{\text{M-ISIS}}+\varepsilon^{❶}+\varepsilon^{❷}\right),$$

where K is a small constant, $\varepsilon_{\text{zk}}^{(i)}, \varepsilon_{\text{sound}}^{(i)}, \varepsilon_{\text{zk}}^{(s)}, \varepsilon_{\text{sound}}^{(s)}$ are defined in Lemma 7.3, 7.4, 7.6, and 7.7 respectively. Then, $\varepsilon_{\text{M-LWE}}$ is the hardness bound of M-LWE$_{n,d,2d,q,U(R_q),U(T_1)}$, $\varepsilon_{\text{M-ISIS}}$ that of M-ISIS$_{n,d,2d,q,\sqrt{2nd}}$, and $\varepsilon^{❶}$, $\varepsilon^{❷}$ are the loss defined in Theorems 6.3 and 6.4 (which depends on different M-LWE and M-SIS assumptions). The quantity $\delta_q(\mathsf{a},\mathsf{b})$ is the singularity probability[a] defined in Section 1.1.6 by $\mathbb{P}_{\mathbf{M}\sim U(R_q^{\mathsf{b}\times\mathsf{a}})}[\mathbf{M}R_q^{\mathsf{a}}=R_q^{\mathsf{b}}]$.

[a] We showed in Lemma 1.6 it can be bounded by $\mathsf{b}\kappa\cdot q^{-(\mathsf{a}-\mathsf{b}+1)n/\kappa}$, which here is negligible.

Proof (Theorem 7.2). We distinguish two types of forgeries: (1) impersonation forgeries, and (2) credential forgeries (either tampering with the proof or by forging a signature). The first case relies on the Lemma 7.4, 7.7 and Lemma 7.6 and the M-ISIS assumption on the matrix $\mathbf{D}_s$. The second relies on the Lemma 7.3, 7.6 and the EUF-CMA security of the signature captured by Theorems 6.3 and 6.4.

We consider a PPT adversary $\mathcal{A}$ against the unforgeability game. It receives opk and gives a set of disclosed attributes $\mathbf{m}_{\mathcal{I}}^{\star}=(m_i^{\star})_{i\in\mathcal{I}}$ while proving possession of a credential $\mathsf{cred}^{\star}$ on said attributes in a successful execution of Show with the honest organization. If $\mathbf{m}_{\mathcal{I}}^{\star}$ corresponds to an attribute vector $\mathbf{m}$ that was queried for issuance by a corrupt user, the forgery is not valid. We thus have two possible cases: (1) $\mathcal{A}$ tried to impersonate an honest user, or (2) they did not. As $\mathcal{A}$ must convince the challenger they know a secret $\mathbf{s}^{\star}$ satisfying $\mathbf{D}_s\mathbf{s}^{\star}=\mathbf{t}$, this means that (1) corresponds to the scenario where there exists $j\in\mathsf{HU}$ such that $\mathbf{s}^{\star}=\mathsf{usk}_j$, i.e., verifying $\mathbf{D}_s\mathbf{s}^{\star}=\mathsf{upk}_j \bmod qR$, and (2) where for every $j\in\mathsf{HU}$, $\mathbf{s}^{\star}\neq\mathsf{usk}_j$. We tackle these two types of forgeries separately.

(1) Impersonation Forgery. The challenger receives the M-ISIS instance $(\overline{\mathbf{D}_s},\bar{\mathbf{t}})$. It then runs $\mathsf{SEP}^{\star}.\mathsf{Setup}$ by setting $\mathbf{D}_s=\overline{\mathbf{D}_s}$ instead of sampling it themselves. It then makes a guess on which honest user will be targeted. For that it samples $j^{+}\hookleftarrow U(\llbracket\,|\mathcal{T}_w|\,\rrbracket)$. Indeed, the number of users requesting credentials to the organization is bounded by the number of possible tags, which is polynomial. It then runs $\mathsf{OKeyGen}(\mathsf{pp})$ to obtain $(\mathsf{opk},\mathsf{osk})=(\mathbf{B},\mathbf{R})$, and sends opk to $\mathcal{A}$. We now describe how the oracle queries are answered.

- $\mathcal{O}_{\mathsf{HU}}$: Given an index j, the challenger runs $(\mathsf{upk}_j,\mathsf{usk}_j)\leftarrow\mathsf{UKeyGen}(\mathsf{pp})$ and outputs upk_j if $j\neq j^{+}$, and outputs $\bar{\mathbf{t}}$ if $j=j^{+}$.
- $\mathcal{O}_{\mathsf{CU}}$: Given j, it gives usk_j to $\mathcal{A}$ if $j\neq j^{+}$. If $j=j^{+}$, the challenger aborts the reduction altogether as the guess was wrong.
- $\mathcal{O}_{\mathsf{ObtIss}}$: Given j and an attribute vector $\mathbf{m}\in T_1^{m'}$, it sends $\perp$ to $\mathcal{A}$ if $j\in\mathsf{CU}$. Otherwise, if $j\neq j^{+}$, the challenger can assume the role of the issuer and the user in the Issue protocol as it knows the issuer's key osk and the key usk_j of user j. If the execution fails, it sends $\perp$ to $\mathcal{A}$, and nothing if it succeeds. If $j=j^{+}$, it instead generates $\mathbf{c}$ as $\mathbf{A}\mathbf{r}+\bar{\mathbf{t}}+\sum_i\mathbf{D}_i\mathbf{m}_i \bmod qR$, and simulates the zero-knowledge argument when assuming the role of the user in Step 4 of Issue. By Lemma 7.4, this is unnoticeable by the adversary. Again, if this modified execution fails, it sends $\perp$ to $\mathcal{A}$, and nothing if it succeeds.
- $\mathcal{O}_{\mathsf{Issue}}$: Given j and an attribute vector $\mathbf{m}\in T_1^{m'}$, it returns $\perp$ to $\mathcal{A}$ and does not engage in the issuance protocol if $j\notin\mathsf{CU}$. Otherwise, since the challenger knows osk, it can run the Issue protocol where the adversary embodies the user j with public key upk_j, and the challenger embodies the signer. Then, either $\mathcal{A}$ gets $\perp$ if the execution failed, or obtained a credential cred on $\mathbf{m}$.
- $\mathcal{O}_{\mathsf{Show}}$: Given an issuance index j' corresponding to the j'-th credential issued on $\mathbf{m}^{(j')}$ for some user j, and also disclosed attributes $\mathbf{m}_{\mathcal{I}}^{(j')}$, the challenger outputs $\perp$ to $\mathcal{A}$ if $j\in\mathsf{CU}$. Otherwise, if $j\neq j^{+}$, it runs the legitimate protocol Show where $\mathcal{A}$ assumes the role of the verifier, which can be done as the challenger knows usk_j, the attributes

and the credential. If $j = j^+$ however, it cannot run Show. Instead, it simulates the zero-knowledge argument with the adversary as the verifier. By Lemma 7.7, this remains unnoticeable by $\mathcal{A}$.

The challenger thus perfectly simulate the oracle queries.

Then, if the guess j^+ is correct, which implies that j^+ is never queried to $\mathcal{O}_{\mathsf{CU}}$, then the game is correctly simulated up to a loss of $\varepsilon_{\text{zk}}^{(i)} + \varepsilon_{\text{zk}}^{(s)} + 2\delta_q(2d,d) + \varepsilon_{\text{M-LWE}}/(1-\delta_q(2d,d))$, where $\varepsilon_{\text{zk}}^{(i)}$ and $\varepsilon_{\text{zk}}^{(s)}$ are defined in Lemma 7.4 and 7.7 respectively. Indeed, the differences stem from the public key of user j^+ and the simulation of the zero-knowledge arguments. Since $\overline{\mathbf{t}}$ is uniform, it is indistinguishable from regular keys $\mathbf{D}_s\mathbf{s}$ under $\text{M-LWE}_{n,d,2d,q,U(R_q),U(T_1)}$ [a], whose hardness bound is denoted by $\varepsilon_{\text{M-LWE}}$. Hence, if $\mathcal{A}$ has advantage δ in performing a forgery attack satisfying (1), it can successfully prove knowledge of $(\mathbf{s}^\star, (\mathbf{m}_i^\star)_{i\notin\mathcal{I}^\star}, \mathsf{cred}^\star)$ with disclosed attributes $\mathbf{m}_{\mathcal{I}}^\star$ such that $\mathsf{SEP}^\star.\mathsf{Verify}'(\mathsf{opk}, \widetilde{\mathbf{m}}^\star, \mathsf{cred}^\star) = 1$ where $\widetilde{\mathbf{m}}^\star = [\mathbf{s}^{\star T}|\mathbf{m}^{\star T}]^T$. The challenger then extracts $\mathbf{s}^\star$ by Lemma 7.6. The probability that the extractor indeed extracts $\mathbf{s}^\star$ is then

$$\delta - \varepsilon_{\text{zk}}^{(i)} - \varepsilon_{\text{zk}}^{(s)} - 2\delta_q(2d,d) - \frac{\varepsilon_{\text{M-LWE}}}{1-\delta_q(2d,d)} - \varepsilon_{\text{sound}}^{(s)}$$

As it verifies the conditions of (1), there must exist $j^\star \in \mathsf{HU}$ such that $\mathbf{s}^\star = \mathsf{usk}_j$, thus implying $\mathbf{D}_s\mathbf{s}^\star = \mathsf{upk}_{j^\star}$. If $j^\star = j^+$, the challenger's guess is correct and this happens with probability at least $1/|\mathcal{T}_w|$ because j^+ was never queried to $\mathcal{O}_{\mathsf{CU}}$ and was therefore independent of the view of $\mathcal{A}$. In that case, we thus have $\overline{\mathbf{D}_s}\mathbf{s}^\star = \overline{\mathbf{t}} \bmod qR$, and $\mathbf{s}^\star \in T_1^{2d}$ yielding $\|\mathbf{s}^\star\|_2 \leq \sqrt{2nd}$. The challenger thus solves the M-ISIS instance. We then have

$$\delta \leq \varepsilon_{\text{zk}}^{(i)} + \varepsilon_{\text{zk}}^{(s)} + 2\delta_q(2d,d) + \frac{\varepsilon_{\text{M-LWE}}}{1-\delta_q(2d,d)} + \varepsilon_{\text{sound}}^{(s)} + |\mathcal{T}_w|\,\varepsilon_{\text{M-ISIS}},$$

as claimed.

(2) Credential Forgery. If the challenger expects this type of forgery, it expects a forgery on the signature scheme of Section 6.4. It therefore tosses a coin to guess which of type ❶ or type ❷ the forgery will be. Note that the M-SIS bounds underlying the security against those forgeries are updated to use B_1' instead of B_1.

If it expects a type ❶ forgery, it proceeds exactly as in the proof of Theorem 6.3, without having to extract the commitment randomness in the issuance. This is because signature queries are answered legitimately without having to tamper with the randomness. As a result, once the challenger has changed the setup, it can answer all the oracle queries $\mathcal{O}_{\mathsf{HU}}, \mathcal{O}_{\mathsf{CU}}, \mathcal{O}_{\mathsf{Obtlss}}, \mathcal{O}_{\mathsf{Issue}}, \mathcal{O}_{\mathsf{Show}}$ legitimately. When $\mathcal{A}$ eventually proves knowledge of $(\mathbf{s}^\star, (\mathbf{m}_i^\star)_{i\notin\mathcal{I}}, \mathsf{cred}^\star)$ with disclosed attributes $\mathbf{m}_{\mathcal{I}}^\star$ such that $\mathsf{Verify}'(\mathsf{opk}, \widetilde{\mathbf{m}}^\star, \mathsf{cred}^\star, \mathsf{pp}) = 1$, the challenger can extract $(\mathbf{s}^\star, (\mathbf{m}_i^\star)_{i\notin\mathcal{I}}, \mathsf{cred}^\star)$ by Lemma 7.6. Then, $\mathsf{cred}^\star$ is a valid type ❶ forgery for the signature as $\widetilde{\mathbf{m}}^\star$ is a fresh message. Indeed, by definition of type (2) forgeries, we have that $\mathbf{s}^\star \neq \mathsf{usk}_j$ for all $j \in \mathsf{HU}$. This first fact means that $\widetilde{\mathbf{m}}^\star$ differs from all the $\widetilde{\mathbf{m}}$ involved in calls to $\mathcal{O}_{\mathsf{Obtlss}}$. Secondly, by the definition of a forgery of the anonymous credentials, it must hold that for all $j \in \mathsf{CU}$, $(j, j', \mathbf{m}^\star) \notin \mathsf{A}$), which means that $\widetilde{\mathbf{m}}^\star$ must differ from all the $\widetilde{\mathbf{m}}$ involved in calls to $\mathcal{O}_{\mathsf{Issue}}$. As a result, we can invoke Theorem 6.3, thus relying on M-LWE and M-SIS, and get that the advantage is upper-bounded by

$$\varepsilon^{❶} + \varepsilon_{\text{sound}}^{(s)}$$

where $\varepsilon^{❶}$ is the maximal advantage against a type ❶ forgery given in Theorem 6.3.

If it expects a type ❷ forgery of the signature, it proceeds as in the proof of Theorem 6.4 with the only difference that it needs to control the commitment randomness for the i^+-th signature query. In this context, in the issuance corresponding to the tag $\mathfrak{t}^\star = \mathfrak{t}^{(i^+)}$ that will be used in the forgery extracted from the showing, the challenger proceeds as follows. By Lemma 7.3, it extracts $(\mathbf{r}^{(i^+)}, \mathbf{s}^{(i^+)})$ such that $\mathbf{c}^{(i^+)} = \mathbf{A}\mathbf{r}^{(i^+)} + \mathbf{D}_s\mathbf{s}^{(i^+)} + \mathbf{D}\mathbf{m}^{(i^+)} \bmod qR$. As opposed to the proof of Theorem 6.4 where it performed rejection on $\mathbf{v}_1^{(i^+)} = \mathbf{v}_1 + \mathbf{S}\widetilde{\mathbf{m}}^{(i^+)}$, with $\widetilde{\mathbf{m}}^{(i^+)} = [{\mathbf{s}^{(i^+)}}^T|{\mathbf{m}^{(i^+)}}^T]^T$, here, it performs rejection on

$$\mathbf{v}_1^{(i^+)} = \mathbf{v}_1 + \mathbf{S}\widetilde{\mathbf{m}}^{(i^+)} + \mathbf{r}^{(i^+)}.$$

The rest of the proof remains the same. In the end, when $\mathcal{A}$ engages in Show to attack the unforgeability of the anonymous credentials, the challenger extracts $(\mathbf{s}^\star, (\mathbf{m}_i^\star)_{i\notin\mathcal{I}}, \mathsf{cred}^\star)$. It thus obtain a valid type ❷ forgery for the SEP on message $\widetilde{\mathbf{m}}^\star$ (which is fresh as explained above). We can thus invoke Theorem 6.4, thus relying on M-LWE and M-SIS, and get that the advantage is upper-bounded by

$$\varepsilon^{❷} + K\varepsilon^{(i)}_{\text{sound}} + \varepsilon^{(s)}_{\text{sound}},$$

with K a small constant and $\varepsilon^{❷}$ is the maximal advantage against a type ❷ forgery given in Theorem 6.4.

Combining all the above with a toss of coin whose result determines which type of forgery is expected, it holds that the advantage of a PPT adversary in breaking the unforgeability of the anonymous credentials is upper-bounded by

$$\begin{aligned}
&3\left(\varepsilon^{(i)}_{\text{zk}} + \varepsilon^{(s)}_{\text{zk}} + 2\delta_q(2d,d) + \frac{\varepsilon_{\text{M-LWE}}}{1-\delta_q(2d,d)} + \varepsilon^{(s)}_{\text{sound}} + |\mathcal{T}_w|\,\varepsilon_{\text{M-ISIS}} + \varepsilon^{❶} + \varepsilon^{(s)}_{\text{sound}} + \varepsilon^{❷}\right.\\
&\qquad \left. + K\varepsilon^{(i)}_{\text{sound}} + \varepsilon^{(s)}_{\text{sound}}\right)\\
&= 3\left(\varepsilon^{(i)}_{\text{zk}} + \varepsilon^{(s)}_{\text{zk}} + K\varepsilon^{(i)}_{\text{sound}} + 3\varepsilon^{(s)}_{\text{sound}} + 2\delta_q(2d,d) + \frac{\varepsilon_{\text{M-LWE}}}{1-\delta_q(2d,d)} + |\mathcal{T}_w|\,\varepsilon_{\text{M-ISIS}} + \varepsilon^{❶} + \varepsilon^{❷}\right)
\end{aligned}$$

as claimed.

[a] The indistinguishability is actually argued by the knapsack version (or decision M-ISIS) which is as hard as $\text{M-LWE}_{n,d,2d,q,U(R_q),U(T_1)}$ by Lemma 3.4.

7.3.3 On Straight-line Extractability

The Issue and Show protocols require two different zero-knowledge proof systems for (1) the proof of opening and proof of registration (for Algorithm 7.5) and (2) the proof of credential possession (for Algorithm 7.6).

This situation is typical of anonymous credentials (and related primitives) and sometimes leads to extractibility issues where the reduction would have to rewind several zero-knowledge proofs (potentially in parallel) to extract all the witnesses. This is specifically the case for (1) when one wants to prove unforgeability under the EUF-CMA security of the underlying signature scheme: one needs to "decapsulate" the committed messages so as to submit them to the EUF-CMA oracle and this is usually done through extraction of the corresponding witnesses. In such cases, one either needs to bound the number of parallel executions of the protocol (which is only possible in the interactive setting) or resort to straight-line extractable zero-knowledge proofs which are more complex. The latter strategy was chosen in [LLLW23] for example which actually presents it as an advantage over the state-of-the-art. The proof techniques from these other constructions [BLNS23b, LLLW23] require straightline extraction as they essentially need to extract every issuance proof to detect a forgery.

We however stress that our proof strategy (first used in our original paper [JRS23]) is *not* concerned by those extractibility issues. It indeed does not exactly rely on the EUF-CMA security of the signature scheme but directly on the underlying assumptions. Most importantly, it only requires to extract *one* commitment opening proof and is thus immune to the problems stemming from parallel rewindings. We therefore do not see any benefit in requiring straight-line extractability for step (1), and this remains true even if one considers using our SEP for a related privacy-preserving primitive (group signature, blind signature, etc).

The case of step (2) is harder to consider in general but we note that most models allow to clearly identify the zero-knowledge proof that needs to be extracted. This is exactly the situation in our case: we only need to extract the one zero-knowledge proof that is produced by the adversary when it "proves" authenticity of a set of attributes for which it never received a credential. As a consequence, we do not need straight-line extractable proof as our reduction only needs to perform two rewindings [LNP22].

7.4 Zero-Knowledge Arguments for the Protocols

We now discuss the zero-knowledge arguments used in the anonymous credentials (and can be declined in other versions with the generic protocols of Section 7.2). As opposed to the results of our first paper [JRS23], we present the arguments using subrings so as to improve the zero-knowledge proof size. As explained in [LNPS21] and recalled in Section 1.1.4, using a smaller ring reduces size of elements that are not dependent on the witness dimension, and thus reduces the overall proof size. To benefit from this improvement while keeping compact keys for the signature scheme, we consider a ring R of degree n for the signature, and a subring $\widehat{R}$ of degree $\widehat{n}|n$ for the zero-knowledge proof. We explain for each protocol exactly how to use the subring embedding θ and M_θ of Section 1.1.4 to map relations over R into relations over $\widehat{R}$.

As the two relations are fairly different to prove, we have one set of parameters for each of the following subsections. To avoid overloading the notations, we use the same notations for the presentation of the arguments, and only distinguish the notations in the associated losses. Typically, we use the superscript (i) for "issuance" in Section 7.4.2, and the superscript (s) for "show" in Section 7.4.3. For example, in Lemma 7.3, $\varepsilon^{(i)}_{\text{sound}}$ denotes the soundness loss for the issuance proof. It is expressed as a function of $\widehat{n}, \widehat{q}, \widehat{d}, m_1, m_2, \ell$, etc. which are specific to the issuance. In Lemma 7.6, $\varepsilon^{(s)}_{\text{sound}}$ would feature the same notations $\widehat{n}, \widehat{q}, \widehat{d}$, etc., but their value might be different. Also, we note that in this section the notations $\sigma_1, \sigma_2, \sigma_3$ refer to Gaussian widths and not field embeddings. Finally, everything is over power-of-two cyclotomic (sub)rings.

For simplicity, we also define the following rejection sampling routine.

Algorithm 7.7: $\mathsf{Rej}_1(\mathbf{z}, \mathbf{s}, s, M)$

1. $u \hookleftarrow U([0,1))$.
2. **return** 1 if $u \leq \frac{1}{M}\exp\left(\frac{\pi}{s^2}(\|\mathbf{s}\|_2^2 - 2\langle \mathbf{z}, \mathbf{s}\rangle)\right)$, and 0 otherwise.

7.4.1 Challenge Space

We use the same family of challenge spaces as [LNP22]. Recall that for any element $c = \sum_{i\in[\![0,\widehat{n}[\![} c_i \widehat{\zeta}^i$, its conjugate is defined as $c^* = c(\widehat{\zeta}^{-1}) = c_0 - \sum_{i\in[\![\widehat{n}-1]\!]} c_{\widehat{n}-i}\widehat{\zeta}^i$. For vectors and matrices, the superscript denotes the conjugate transpose. The conjugate operator corresponds to the automorphism σ_{-1} in [LNP22]. One can see that if $c^* = c$ then $c_i = -c_{\widehat{n}-i}$ for all $i \in [\![\widehat{n}-1]\!]$, thus implying $c_{\widehat{n}/2} = 0$. We define $\mathcal{C}' = \{c \in \widehat{S}_\rho : c^* = c\}$ where ρ is a positive integer. The challenge space is defined by

$$\mathcal{C} = \{c \in \mathcal{C}' : \sqrt[2k']{\left\|c^{2k'}\right\|_1} \leq \eta\},$$

where η is a positive integer, and k' is a power-of-two that we later choose to be $k' = 32$. From the observation above, we have $\left|\mathcal{C}'\right| = (2\rho+1)^{\widehat{n}/2}$. We thus choose ρ so that this size is at least $2^{\lambda+1}$, that is

$$\rho = \left\lceil \frac{1}{2}\left(2^{2(\lambda+1)/\widehat{n}} - 1\right)\right\rceil.$$

Then, we determine η heuristically so that $\mathbb{P}_{c\sim U(\mathcal{C}')}[\sqrt[2k']{\left\|c^{2k'}\right\|_1} \leq \eta] \geq 1/2$. As a result, we would end up with $|\mathcal{C}| \geq 2^\lambda$. The challenge space places the constraint on the proof modulus $\widehat{q}$. The proof modulus is a product of two primes $\widehat{q} = q \cdot q_1$, where q is the modulus of the SEP, and q_1 is specific to each proof system. In particular, we define $q_{\min} = \min(q, q_1)$. The choice of the challenge space then requires $q_{\min} > (2\rho\sqrt{\kappa})^\kappa$ which is almost always verified for typical parameters as κ is chosen to be either 2 or 4, and ρ is also small. Note that q_1 is chosen to have the same splitting behavior as q. For later, we also define $\ell = \lceil \lambda / \log_2 q_{\min}\rceil$ which is a parameter used for soundness amplification, i.e., so that $q_{\min}^{-\ell} \leq 2^{-\lambda}$.

7.4.2 Proof of Commitment Opening and User Registration

In Algorithm 7.5, the user needs to prove knowledge of a commitment opening as well as the secret key associated to its public key (which we call user registration). We present the argument so that the attributes remain hidden even though it differs from the presentation of Algorithm 7.5. Revealing the message will only make the proof simpler and smaller, so we deal with the worst case where everything must be concealed.

Relation

The relation entails proving knowledge of $\mathbf{r} \in R^{2d}$, $\mathbf{m} \in R^m$ and $\mathbf{s} \in R^{2d}$ such that

$$\mathbf{A}\mathbf{r} + \mathbf{D}\mathbf{m} = \mathbf{c} - \mathsf{upk} \bmod qR \text{ and } \mathbf{D}_s\mathbf{s} = \mathsf{upk} \bmod qR$$
$$\mathbf{r} \in T_1^{2d}, \mathbf{s} \in T_1^{2d}, \mathbf{m} \in T_1^{m'}$$

where $\mathbf{A} \in R_q^{d\times 2d}$, $\mathbf{D}_s \in R_q^{d\times 2d}$, $\mathbf{D} \in R_q^{d\times m'}$, $\mathbf{c} \in R_q^d$, and $\mathsf{upk} \in R_q^d$ are public elements part of the statement. To prove such a statement, we first lift the equation to $R_{\widehat{q}}$ where $\widehat{q} = q_1 q$ is the modulus for the proof system. Then, since all the vectors must be proven binary, we compact everything into a single equation. We also use the subring embedding θ and M_θ of Section 1.1.4 to map the relation to $\widehat{R}$. Recall that using M_θ, proving the linear relation $\mathbf{M}\mathbf{x} = \mathbf{y} \bmod \widehat{q}R$ is equivalent to proving $M_\theta(\mathbf{M})\theta(\mathbf{x}) = \theta(\mathbf{y}) \bmod \widehat{q}\widehat{R}$. In the end, we prove the following.

$$\mathbf{C}\mathbf{s}_1 = \mathbf{u} \bmod \widehat{q}\widehat{R}, \quad \text{and} \quad \mathbf{s}_1 \in \widehat{T}_1^{m_1},$$

where $\mathbf{s}_1 = [\theta(\mathbf{r})^T|\theta(\mathbf{s})^T|\theta(\mathbf{m})^T]^T$, $m_1 = \widehat{k}(2d + 2d + m') = \widehat{k}(2d + m)$, and

$$\mathbf{C} = q_1 M_\theta\left(\begin{bmatrix}\mathbf{A} & \mathbf{0}_{d\times 2d} & \mathbf{D}\\ \mathbf{0}_{d\times 2d} & \mathbf{D}_s & \mathbf{0}_{d\times m'}\end{bmatrix}\right), \quad \text{and} \quad \mathbf{u} = q_1\theta\left(\begin{bmatrix}\mathbf{c} - \mathsf{upk}\\ \mathsf{upk}\end{bmatrix}\right).$$

The Protocol

Let us now describe the full protocol. It is summarized in Figure 7.1.

First Round. We start by the main commitment phase. We sample $\mathbf{s}_2$ from χ^{m_2} where $\mathrm{Supp}(\chi) \subseteq \widehat{S}_1$ and compute an Ajtai commitment of $\mathbf{s}_1$ with randomness $\mathbf{s}_2$ as $\mathbf{t}_A = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \bmod \widehat{q}\widehat{R}$, where $\mathbf{A}_1 \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d}\times m_1})$ and $\mathbf{A}_2 \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d}\times m_2})$ are part of the common reference string crs. Then, we sample the Gaussian masks for what will later be $c\mathbf{s}_1$ and $c\mathbf{s}_2$. More precisely, we sample $\mathbf{y}_1$ from $\mathcal{D}_{\widehat{R}^{m_1},\sigma_1}$ and $\mathbf{y}_2$ from $\mathcal{D}_{\widehat{R}^{m_2},\sigma_2}$, and compute the commitment $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 \bmod \widehat{q}\widehat{R}$.

We then sample a mask $\mathbf{y}_3$ from $\mathcal{D}_{\widehat{R}^{256/\widehat{n}},\sigma_3}$ and a vector for soundness amplification by $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\}^\ell)$ where all the entries are polynomials with a constant coefficient equal to zero. We later use $\widehat{\mathbf{m}}$ to denote the vector $\widehat{\mathbf{m}} = [\mathbf{y}_3^T|\mathbf{g}^T]^T \in \widehat{R}^{256/\widehat{n}+\ell}$. We commit to it via $\mathbf{t}_B = \mathbf{B}_{y,g}\mathbf{s}_2 + \widehat{\mathbf{m}} \bmod \widehat{q}\widehat{R}$, where $\mathbf{B}_{y,g} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{(256/\widehat{n}+\ell)\times m_2})$ is part of crs.

The prover sends msg_1 as the first message and receives chal_1 as the first challenge, where they are both defined as

$$\mathsf{msg}_1 = (\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}) \in \widehat{R}_{\widehat{q}}^{2\widehat{d}+256/\widehat{n}+\ell}$$
$$\mathsf{chal}_1 = \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1) = (\mathsf{R}_0, \mathsf{R}_1) \in \left(\{0,1\}^{256\times m_1\widehat{n}}\right)^2$$

Second Round. Now, we conclude the approximate range proof part. We define $\mathsf{R} = \mathsf{R}_0 - \mathsf{R}_1$. In the second round, we respond to the challenge by masking $\mathsf{R}\tau(\mathbf{s}_1)$ with $\tau(\mathbf{y}_3)$, where τ is the coefficient embedding from Section 1.1.2. So we compute $\mathbf{z}_3^{\mathbb{Z}} = \tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1) \in \mathbb{Z}^{256}$. Then, we reject if $\mathsf{Rej}_1(\mathbf{z}_3^{\mathbb{Z}}, \mathsf{R}\tau(\mathbf{s}_1), \sigma_3, M_3) = 0$ (Algorithm 7.7). If the prover accepts, it sends msg_2 as the second message and receives chal_2 as the second challenge where they are defined by

$$\mathsf{msg}_2 = \mathbf{z}_3^{\mathbb{Z}} \in \mathbb{Z}^{256}$$
$$\mathsf{chal}_2 = \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2) = (\gamma_{i,j})_{\substack{i\in[\![\ell]\!]\\ j\in[\![257]\!]}} \in \mathbb{Z}_{\widehat{q}}^{\ell\times 257}.$$

Third Round. We now need to prove the following equations over $\mathbb{Z}_{\widehat{q}}$.

$$\tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1) = \mathbf{z}_3^{\mathbb{Z}}, \tag{3.1}$$
$$\langle\tau(\mathbf{s}_1), \tau(\mathbf{s}_1) - \mathbf{1}_{\widehat{n}m_1}\rangle = 0, \tag{3.2}$$

As observed in [LNP22], the equation $\langle\mathbf{x},\mathbf{y}\rangle = 0$ is equivalent to

$$\tau_0((\tau^{-1}(\mathbf{x}))^*\tau^{-1}(\mathbf{y})) = 0$$

which allows us to interpret $\mathbb{Z}_{\widehat{q}}$-equations as $\widehat{R}_{\widehat{q}}$-equations with automorphisms instead. Recall that τ_0 is the projection of the coefficient embedding (Section 1.1.2) which gives the constant coefficient.

We write $\mathbf{1}_{\widehat{R}^N} = \tau^{-1}(\mathbf{1}_{\widehat{n}N}) = [\sum_{i=0}^{\widehat{n}-1} \widehat{\zeta}^i]_{j\in[\![N]\!]}$. We also write $\mathbf{e}_j^{\mathbb{Z}}$ to be the j-th canonical vector of $\mathbb{Z}^{N\widehat{n}}$, where the dimension N is implicit, and let $\mathbf{e}_j = \tau^{-1}(\mathbf{e}_j^{\mathbb{Z}}) \in \widehat{R}^N$. As a contrast, we later write $\mathbf{e}_j^{\widehat{R}}$ to be the j-th canonical vector of $\widehat{R}^N$ for a rank N implicit, that is $\mathbf{e}_j^{\widehat{R}}$ has a 1 at position j and 0 elsewhere. The equations above are thus equivalent to

$$\forall j \in [\![256]\!], \tau_0\left(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}\right) = 0, \tag{3.1*}$$

$$\tau_0\left(\mathbf{s}_1^*(\mathbf{s}_1 - \mathbf{1}_{\widehat{R}^{m_1}})\right) = 0, \tag{3.2*}$$

where $\mathsf{r}_j = \tau^{-1}(\mathsf{R}^T\mathbf{e}_j^{\mathbb{Z}})$. We combine all of these quadratic equations with automorphisms by computing elements h_i for each $i \in [\![\ell]\!]$ as follows

$$h_i = g_i + \sum_{j\in[\![256]\!]} \gamma_{i,j}(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}) + \gamma_{i,257}(\mathbf{s}_1^*(\mathbf{s}_1 - \mathbf{1}_{\widehat{R}^{m_1}})) \tag{7.1}$$

The prover then sends msg_3 as the third message and receives chal_3 as the third challenge, where they are both defined as

$$\mathsf{msg}_3 = (h_1, \ldots, h_\ell) \in \widehat{R}_{\widehat{q}}^{\ell}$$
$$\mathsf{chal}_3 = \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3) = (\mu_i)_{i\in[\![\ell+2d\widehat{k}]\!]} \in \widehat{R}_{\widehat{q}}^{\ell+2d\widehat{k}}.$$

Fourth Round. We now need to prove all the equations over $\widehat{R}_{\widehat{q}}$. We need to prove that the h_i are well-formed and equal their expressions above, and we also need to prove the linear relation $\mathbf{C}\mathbf{s}_1 = \mathbf{u}$. The latter represent $2d\widehat{k}$ equations. We prove them all at once by combining them linearly with the challenges μ_i and prove that

$$\begin{aligned} 0 = &\sum_{i\in[\![\ell]\!]} \mu_i \left(g_i + \sum_{j\in[\![256]\!]} \gamma_{i,j}(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}) + \gamma_{i,257}(\mathbf{s}_1^*(\mathbf{s}_1 - \mathbf{1}_{\widehat{R}^{m_1}})) - h_i \right) \\ &+ \sum_{i\in[\![2d\widehat{k}]\!]} \mu_{\ell+i}\left(\mathbf{e}_i^{\widehat{R}^T}\mathbf{C}\mathbf{s}_1 - u_i\right). \end{aligned}$$

For that let us define $\widehat{\mathbf{s}} = [\mathbf{s}_1^T|\mathbf{s}_1^*|\widehat{\mathbf{m}}^T|\widehat{\mathbf{m}}^*]^T$. Then, the equation to be proven is equivalent to $\widehat{\mathbf{s}}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\widehat{\mathbf{s}} + f = 0 \bmod \widehat{q}\widehat{R}$, where

$$\begin{aligned} f &= -\sum_{i\in[\![\ell]\!]} \mu_i \left(\sum_{j\in[\![256]\!]} \gamma_{i,j} z_{3,j}^{\mathbb{Z}} + h_i \right) - \sum_{i\in[\![2d\widehat{k}]\!]} \mu_{\ell+i}u_i \\ \mathbf{f} &= \begin{bmatrix} \sum_{i\in[\![\ell]\!]}\sum_{j\in[\![256]\!]} \mu_i\gamma_{i,j}\mathsf{r}_j^{*T} + \sum_{i\in[\![2d\widehat{k}]\!]} \mu_{\ell+i}\mathbf{C}^T\mathbf{e}_i^{\widehat{R}} \\ -\sum_{i\in[\![\ell]\!]} \mu_i\gamma_{i,257}\mathbf{1}_{\widehat{R}^{m_1}} \\ \sum_{i\in[\![\ell]\!]} \mu_i \sum_{j\in[\![256]\!]} \gamma_{i,j}\mathbf{e}_j^{*T} \\ [\mu_1|\ldots|\mu_\ell]^T \\ \mathbf{0}_{256/\widehat{n}} \\ \mathbf{0}_\ell \end{bmatrix} \\ \mathbf{F} &= \begin{bmatrix} \mathbf{0}_{m_1\times m_1} & \sum_{i\in[\![\ell]\!]} \mu_i\gamma_{i,257}\mathbf{I}_{m_1} & \mathbf{0}_{m_1\times 2(256/\widehat{n}+\ell)} \\ \multicolumn{3}{c}{\mathbf{0}_{m_1+2(256/\widehat{n}+\ell)\times 2(m_1+256/\widehat{n}+\ell)}} \end{bmatrix}, \end{aligned} \tag{7.2}$$

Once we have defined these (public) elements, we can compute the garbage terms and commit to them. More precisely, we define

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_1^{*T} \\ -\mathbf{B}_{y,g}\mathbf{y}_2 \\ -(\mathbf{B}_{y,g}\mathbf{y}_2)^{*T} \end{bmatrix} \in \widehat{R}_{\widehat{q}}^{2(m_1+256/\widehat{n}+\ell)}, \tag{7.3}$$

and compute $e_0 = \mathbf{y}^T\mathbf{F}\mathbf{y} \bmod \widehat{q}\widehat{R}$, $e_1 = \widehat{\mathbf{s}}^T\mathbf{F}\mathbf{y} + \mathbf{y}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\mathbf{y}$, and the commitments $t_0 = \mathbf{b}^T\mathbf{y}_2 + e_0 \bmod \widehat{q}\widehat{R}$ and $t_1 = \mathbf{b}^T\mathbf{s}_2 + e_1 \bmod \widehat{q}\widehat{R}$, where $\mathbf{b} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{m_2})$ is part of crs. The prover then sends

msg_4 as the fourth message and receives chal_4 as the fourth challenge, where they are both defined as

$$\begin{aligned}\mathsf{msg}_4 &= (t_0, t_1) \in \widehat{R}_{\widehat{q}}^2\\ \mathsf{chal}_4 &= \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) = c \in \mathcal{C}.\end{aligned}$$

Fifth Round. In the final round, the prover responds to the challenge by masking $c\mathbf{s}_1$ and $c\mathbf{s}_2$ with $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively. So we compute $\mathbf{z}_1 = \mathbf{y}_1 + c\mathbf{s}_1$ and $\mathbf{z}_2 = \mathbf{y}_2 + c\mathbf{s}_2$. Then, we reject if $\mathsf{Rej}_1(\tau(\mathbf{z}_1), \tau(c\mathbf{s}_1), \sigma_1, M_1) = 0$ or if $\mathsf{Rej}_1(\tau(\mathbf{z}_2), \tau(c\mathbf{s}_2), \sigma_2, M_2) = 0$ (Algorithm 7.7). If the prover accepts, it sends msg_5 as the final message defined by

$$\mathsf{msg}_5 = (\mathbf{z}_1, \mathbf{z}_2) \in \widehat{R}^{m_1+m_2}.$$

Verification. Upon receiving msg_5, the verifier computes $\mathbf{F}, \mathbf{f}, f$, as well as

$$\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_1^{*T} \\ c\mathbf{t}_B - \mathbf{B}_{y,g}\mathbf{z}_2 \\ (c\mathbf{t}_B - \mathbf{B}_{y,g}\mathbf{z}_2)^{*T} \end{bmatrix}, \tag{7.4}$$

and then checks the following six conditions.

$$\|\mathbf{z}_1\|_2 \le c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1}, \|\mathbf{z}_2\|_2 \le c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2}, \|\mathbf{z}_3^{\mathbb{Z}}\|_2 \le c_{256}\sigma_3\sqrt{256} \tag{7.5}$$

$$\forall i \in [\![\ell]\!], \tau_0(h_i) = 0 \tag{7.6}$$

$$\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 = \mathbf{w} + c\mathbf{t}_A \bmod \widehat{q}\widehat{R} \tag{7.7}$$

$$\mathbf{z}^T\mathbf{F}\mathbf{z} + c\mathbf{f}^T\mathbf{z} + c^2 f - (ct_1 - \mathbf{b}^T\mathbf{z}_2) = t_0 \bmod \widehat{q}\widehat{R}. \tag{7.8}$$

Transcript and Communication Complexity.

The transcript is thus composed of the five messages and four challenges. Note that in the interactive setting, the challenges are selected uniformly in their respective space and not computed from $\mathcal{H}$. The hash function $\mathcal{H}$ is presented here if one desires to make the proof non-interactive.

The total size of the messages send by the prover to the verifier can be details as follows. The elements $\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}, h_1, \ldots, h_\ell, t_0, t_1$ cannot be compressed as they all[2] look uniformly random modulo $\widehat{q}$. To evaluate the size of the discrete Gaussian vectors $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3^{\mathbb{Z}}$, we use the entropy bound which can be achieved using the rANS encoding as discussed [ETWY22]. More precisely, for a discrete Gaussian over $\mathbb{Z}^N$ of width s, the estimated bit size is $N(1/2 + \log_2 s)$. It means the total bit-size of the message part can be estimated by

$$\left(2\widehat{d} + \frac{256}{\widehat{n}} + 2\ell + 2\right)\widehat{n}\lceil\log_2 \widehat{q}\rceil + \widehat{n}m_1\left(\frac{1}{2} + \log_2\sigma_1\right) + \widehat{n}m_2\left(\frac{1}{2} + \log_2\sigma_2\right) + 256\left(\frac{1}{2} + \log_2\sigma_3\right).$$

For the challenges, the maximal bit-size can be easily bounded by

$$2 \cdot 256 \cdot m_1\widehat{n} + (\ell(256+3) + (2d\widehat{k} + \ell)\widehat{n})\lceil\log_2 \widehat{q}\rceil + \widehat{n}\lceil\log_2(2\rho+1)\rceil.$$

As $\mathbf{w}$, t_0 and the challenges can be re-computed from the rest, the proof can be condensed to $\pi = (\mathbf{t}_A, \mathbf{t}_B, \mathbf{z}_3^{\mathbb{Z}}, h_1, \ldots, h_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_2)$ in the non-interactive case. In that case, the overall proof size can be bounded by

$$\begin{aligned}|\pi| \le &\left(\widehat{d} + \frac{256}{\widehat{n}} + 2\ell + 1\right)\widehat{n}\lceil\log_2 \widehat{q}\rceil + \widehat{n}m_1(1/2 + \log_2\sigma_1)\\ &+ \widehat{n}m_2(1/2 + \log_2\sigma_2) + 256(1/2 + \log_2\sigma_3) + \widehat{n}\lceil\log_2(2\rho+1)\rceil.\end{aligned}$$

[2]The elements $h_1, \ldots, h_\ell$ are uniform among those that have a constant coefficient equal to zero.

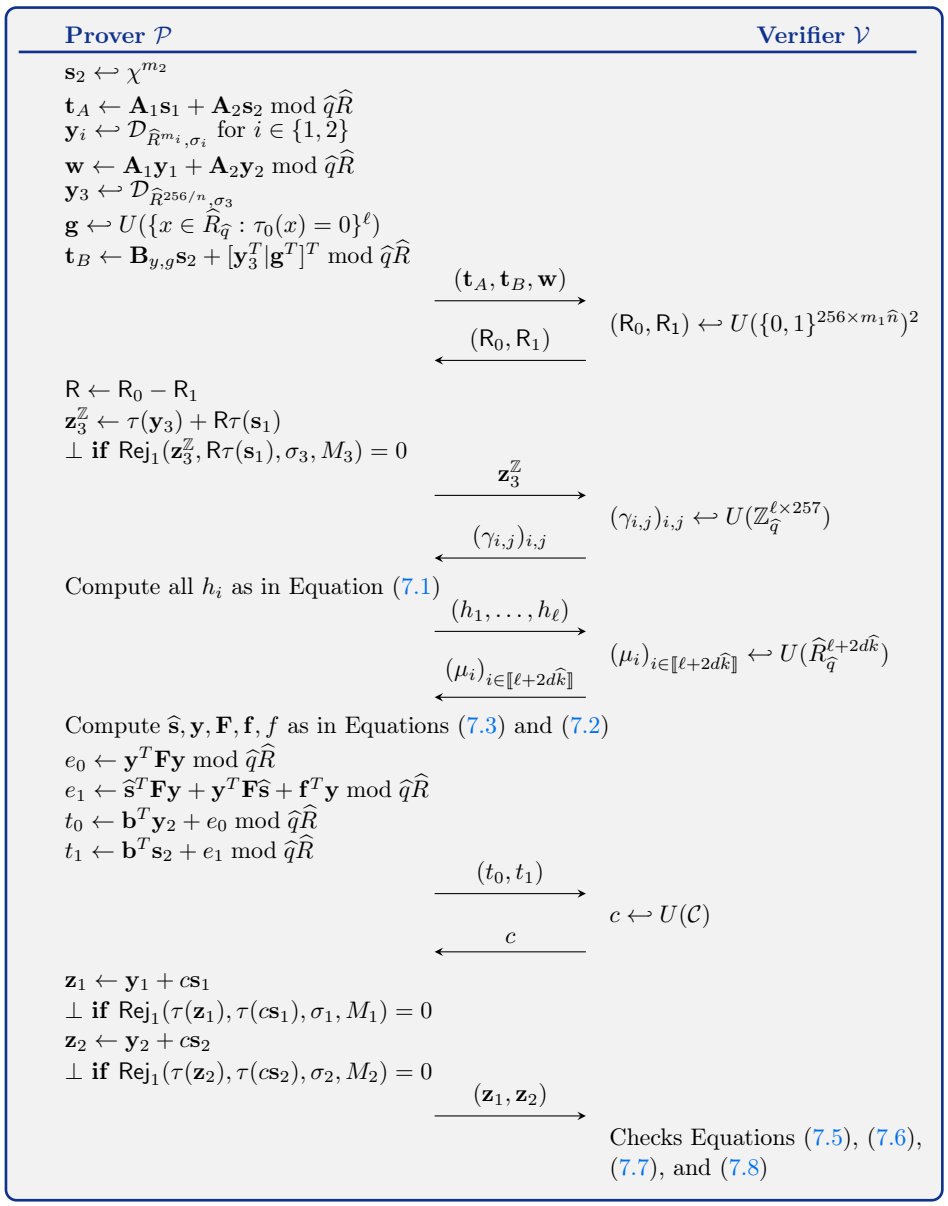


Figure 7.1: Interactive zero-knowledge argument for commitment opening and user registration

Security Analysis

The proof of completeness from Lemma 7.2 follows from that of [LNP22] by combining the rejection sampling result of Lemma 1.24, the tail bound of Lemma 1.21 and careful inspection of the verification equations with respect to the committed variables. The proof of knowledge soundness of Lemma 7.3 also follows the exact blueprint of that of [LNP22, Thm. B.7]. Finally, the zero-knowledge property follows from the M-LWE assumption (albeit in its knapsack form) and the rejection sampling result. Although it is generally interesting to use the Rényi divergence that is provided in Lemma 1.24, its use for distinguishing problems such as this one is more delicate as mentioned in Section 2.3. We are then bound to use the statistical distance. As such, one needs to choose ε_j that are negligible in the security parameter. All the proofs being slight adaption of that of [LNP22], we do not include them.

Lemma 7.2 (Issuance Proof - Completeness)

Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and let M_1, M_2, M_3 in $(1, \infty)$. For $j \in [\![3]\!]$, we define

$$\gamma_j = \frac{\sqrt{\pi}}{\ln(M_j)} \cdot \left(\sqrt{\ln(1/\varepsilon_j) + \ln(M_j)} + \sqrt{\ln(1/\varepsilon_j)}\right).$$

Let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \gamma_1\eta\sqrt{nm_1}$, $\sigma_2 = \gamma_2\eta\sqrt{nm_2}$ and $\sigma_3 = \gamma_3\sqrt{337}\sqrt{nm_1}$. Then, the (interactive) zero-knowledge argument in Figure 7.1 is complete.

Lemma 7.3 (Issuance Proof - Knowledge Soundness)

Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and let M_1, M_2, M_3 in $(1, \infty)$. For $j \in [\![3]\!]$, we define

$$\gamma_j = \frac{\sqrt{\pi}}{\ln(M_j)} \cdot \left(\sqrt{\ln(1/\varepsilon_j) + \ln(M_j)} + \sqrt{\ln(1/\varepsilon_j)}\right).$$

We let $B = \sqrt{nm_1}$ be a bound on $\|\mathbf{s}_1\|_2$. Then, let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \gamma_1\eta B$, $\sigma_2 = \gamma_2\eta\sqrt{nm_2}$, $\sigma_3 = \gamma_3\sqrt{337}B$, and define $B_{256} = c_{256}\sigma_3\sqrt{256}$. Assume that $q_\pi > \max(B^2, 82/\sqrt{26} \cdot \widehat{n}m_1B_{256}, 2B_{256}^2/13 - B_{256})$.
Then, the (interactive) zero-knowledge argument in Figure 7.1 is knowledge sound with an extractor running in expected polynomial time, and soundness error

$$\varepsilon_{\text{sound}}^{(i)} = \frac{2}{|\mathcal{C}|} + q_{\min}^{-\widehat{n}/\kappa} + q_{\min}^{-\ell} + 2^{-128} + \varepsilon_{\text{M-SIS}}$$

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound for $\text{M-SIS}_{\widehat{n},\widehat{d},m_1+m_2,\widehat{q},\beta}$ for

$$\beta = 8\eta\sqrt{(c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1})^2 + (c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2})^2}$$

Lemma 7.4 (Issuance Proof - Zero-Knowledge)

Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and let M_1, M_2, M_3 in $(1, \infty)$. For $j \in [\![3]\!]$, we define

$$\gamma_j = \frac{\sqrt{\pi}}{\ln(M_j)} \cdot \left(\sqrt{\ln(1/\varepsilon_j) + \ln(M_j)} + \sqrt{\ln(1/\varepsilon_j)}\right).$$

Let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \gamma_1\eta\sqrt{\widehat{n}m_1}$, $\sigma_2 = \gamma_2\eta\sqrt{\widehat{n}m_2}$ and $\sigma_3 = \gamma_3\sqrt{337}\sqrt{\widehat{n}m_1}$. We define $m_2' = \widehat{d} + 256/\widehat{n} + \ell + 1$ and assume that $m_2 > m_2'$. Then, the (interactive) zero-knowledge argument in Figure 7.1 is honest-verifier zero-knowledge. More precisely, there exists a simulator $\mathcal{S}$ that outputs a distribution that is $\varepsilon_{\text{zk}}^{(i)}$-indistinguishable from that of an honest transcript, where

$$\varepsilon_{\text{zk}}^{(i)} = \frac{\varepsilon_1}{M_1} + \frac{\varepsilon_2}{M_2} + \frac{\varepsilon_3}{M_3} + 2\delta_{q_{\min}}(m_2, m_2') + \frac{\varepsilon_{\text{M-LWE}}}{1 - \delta_{q_{\min}}(m_2, m_2 - m_2')}$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bound of $\text{M-LWE}_{\widehat{n},m_2-(\widehat{d}+256/\widehat{n}+\ell+1),m_2,\widehat{q},U(\widehat{R}_{\widehat{q}}),\chi}$, and $\delta_{q_{\min}}(\mathsf{a}, \mathsf{b}) = \mathbb{P}_{\mathbf{M}\sim U(\widehat{R}_{q_{\min}}^{\mathsf{b}\times\mathsf{a}})}[\mathbf{M} \cdot \widehat{R}_{q_{\min}}^{\mathsf{a}} \neq \widehat{R}_{q_{\min}}^{\mathsf{b}}]$ is the singularity probability.

7.4.3 Proof of Valid Credential

Algorithm 7.6 solely relies on a zero-knowledge argument for the signature verification of Algorithm 6.8. The user needs to hide its secret key, the desired attributes and the credential, while convincing the verifier that it holds such elements. We use the same techniques as in Section 7.4.2, although this relation is slightly more complex as it directly involves quadratic equations. Although we use the same notations, all the parameters of the proof system in this section (e.g. $m_1, m_2, q_1, \widehat{d}, \ell, \rho, \eta, \varepsilon_i, M_i$) are most likely different from those of the previous protocol unless specified otherwise.

Relation

The prover starts by reconstructing $\mathbf{v}_1$ as in Algorithm 6.8. For clarity, we denote by $\mathbf{m}_{\mathcal{I}}$ the sub-vector of attributes that are revealed and $\mathbf{m}_{sm}$ the sub-vector of concealed attributes concatenated with the secret key $\mathbf{s}$. We similarly define $\mathbf{D}_{\mathcal{I}}$ and $\mathbf{D}_{sm}$ such that $\mathbf{D}_s\mathbf{s} + \mathbf{D}\mathbf{m} = \mathbf{D}_{sm}\mathbf{m}_{sm} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}}$. In particular, we let m_{sm} be the dimension of $\mathbf{m}_{sm}$, namely $m_{sm} = 2d + (m' - |\mathcal{I}|) = m -$

$|\mathcal{I}|$. We use the same process to lift the relation modulo $\widehat{q} = q_1 q$ and to select the soundness amplification parameter ℓ, and the challenge space parameters k', ρ, η. The user proves knowledge of $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathfrak{t}, \mathbf{m}_{sm}) \in R^{2d+kd+k+1+m_{sm}}$ such that

$$q_1 \left(\mathbf{A}\mathbf{v}_1 - \mathbf{B}\mathbf{v}_2 + \mathbf{A}_3\mathbf{v}_3 + \mathbf{G}(\mathfrak{t}\mathbf{v}_2) - \mathbf{D}_{sm}\mathbf{m}_{sm}\right) = q_1\left(\mathbf{u} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}}\right) \bmod \widehat{q}R$$
$$\|\mathbf{v}_1\|_2 \leq B_1', \|\mathbf{v}_2\|_2 \leq B_2, \|\mathbf{v}_3\|_2 \leq B_3$$
$$\|\mathfrak{t}\|_2 = \sqrt{w}, \mathfrak{t} \in T_1, \mathbf{m}_{sm} \in T_1^{m_{sm}}$$

where $\mathbf{A} \in R_q^{d\times 2d}$, $\mathbf{B}, \mathbf{A}_3 \in R_q^{d\times k}$, $\mathbf{G} \in R_q^{d\times kd}$, $\mathbf{D}_{sm} \in R_q^{d\times m_{sm}}$, $\mathbf{D}_{\mathcal{I}} \in R_q^{d\times|\mathcal{I}|}$, $\mathbf{u} \in R_q^d$, $\mathbf{m}_{\mathcal{I}} \in T_1^{|\mathcal{I}|}$, w, B_1', B_2 and B_3 are public elements part of the statement. We then embed everything using θ and M_θ. For clarity, we define $\mathbf{A}' = q_1 M_\theta(\mathbf{A})$, $\mathbf{B}' = q_1 M_\theta(\mathbf{B})$, $\mathbf{G}' = q_1 M_\theta(\mathbf{G})$, $\mathbf{A}_3' = q_1 M_\theta(\mathbf{A}_3)$, $\mathbf{D}_{sm}' = q_1 M_\theta(\mathbf{D}_{sm})$, and $\mathbf{u}' = q_1\theta(\mathbf{u} + \mathbf{D}_{\mathcal{I}}\mathbf{m}_{\mathcal{I}})$.

As it is needed later in the protocol, we detail how to tackle the quadratic term $\mathbf{G}'\theta(\mathfrak{t}\mathbf{v}_2)$, in particular how to express its i-th coefficient in terms of $\theta(\mathfrak{t})$ and $\theta(\mathbf{v}_2)$. Let $i \in [0, d\widehat{k} - 1]$. We decompose it as $i = i_1\widehat{k} + i_2$ for $i_1 \in [\![0, d[\![$ and $i_2 \in [\![0, \widehat{k}[\![$. We call $\mathbf{e}_i$ the vector $\widehat{R}^{d\widehat{k}}$ that is 1 at position i and 0 elsewhere. We also call $\mathbf{e}_{i_2}$ the vector of $\widehat{R}^{\widehat{k}}$ that is 1 at position i_2 and 0 elsewhere. It holds that

$$[\theta(\mathfrak{t}\mathbf{G}\mathbf{v}_2)]_i = \mathbf{e}_i^T\theta((\mathbf{I}_d \otimes \mathfrak{t})\mathbf{G}\mathbf{v}_2) = \mathbf{e}_i^T(\mathbf{I}_d \otimes M_\theta(\mathfrak{t}))M_\theta(\mathbf{G})\theta(\mathbf{v}_2).$$

We have that $\mathbf{e}_i^T(\mathbf{I}_d \otimes M_\theta(\mathfrak{t})) = [\mathbf{0}_{1\times i_1\widehat{k}}|\mathbf{e}_{i_2}^T M_\theta(\mathfrak{t})|\mathbf{0}_{1\times(d-i_1-1)\widehat{k}}]$, where the non-zero block is at the block position i_1. We can now express

$$\mathbf{e}_{i_2}^T M_\theta(\mathfrak{t}) = \mathrm{Row}_{i_2}(M_\theta(\mathfrak{t})) = \theta(x^{\widehat{k}-1-i_2} \otimes_R \mathfrak{t})^T \cdot \mathbf{P} = \theta(\mathfrak{t})^T M_\theta(x^{\widehat{k}-1-i_2})^T\mathbf{P},$$

where $\mathbf{P}$ is the permutation of $[\![0, \widehat{k}[\![$ having 1 only on the anti-diagonal, i.e.,

$$\mathbf{P} = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix}.$$

As a result, we have that $[\theta(\mathfrak{t}\mathbf{G}\mathbf{v}_2)]_i$ is equal to

$$\theta(\mathfrak{t})^T \cdot [\mathbf{0}_{\widehat{k}\times i_1\widehat{k}}|M_\theta(x^{\widehat{k}-1-i_2})^T\mathbf{P}|\mathbf{0}_{\widehat{k}\times(d-i_1-1)\widehat{k}}]M_\theta(\mathbf{G}) \cdot \theta(\mathbf{v}_2),$$

which means the i-th coefficient of $\theta(q_1\mathfrak{t}\mathbf{G}\mathbf{v}_2)$ can be expressed as $\theta(\mathfrak{t})^T\mathbf{G}_i'\theta(\mathbf{v}_2)$, where

$$\mathbf{G}_i' = [\mathbf{0}_{\widehat{k}\times i_1\widehat{k}}|M_\theta(x^{\widehat{k}-1-i_2})^T\mathbf{P}|\mathbf{0}_{\widehat{k}\times(d-i_1-1)\widehat{k}}]\mathbf{G}',$$

where the non-zero block is at position i_1, where $i = i_1\widehat{k} + i_2$ for $i_1 \in [\![0, d[\![$ and $i_2 \in [\![0, \widehat{k}[\![$. In the remainder of the protocol description, we define $\mathbf{v}_j' = \theta(\mathbf{v}_j)$ for $j \in [\![3]\!]$, $\mathfrak{t}' = \theta(\mathfrak{t})$, $\mathbf{m}_{sm}' = \theta(\mathbf{m}_{sm})$.

The Protocol

We start by expressing $B_1'^2 - \|\mathbf{v}_1'\|_2^2$ as the sum of four square integer $a_{1,0}^2 + a_{1,1}^2 + a_{1,2}^2 + a_{1,3}^2$. Then, define $a_1 = a_{1,0} + a_{1,1}x + a_{1,2}x^2 + a_{1,3}x^3$ and $\mathbf{v}_1'' = [\mathbf{v}_1'^T|a_1]^T$ so that $\|\mathbf{v}_1''\|_2 = B_1'$. We perform the same decomposition and define $a_2, a_3, \mathbf{v}_2'', \mathbf{v}_3''$. We also define $\mathbf{A}'' = [\mathbf{A}'|\mathbf{0}_d]$, $\mathbf{B}'' = [\mathbf{B}'|\mathbf{0}_d]$, $\mathbf{G}_i'' = [\mathbf{G}_i'|\mathbf{0}_{\widehat{k}}]$ and $\mathbf{A}_3'' = [\mathbf{A}_3'|\mathbf{0}_d]$. Later, we also pack the witnesses into the vector $\mathbf{s}_1 = (\mathbf{v}_1'', \mathbf{v}_2'', \mathbf{v}_3'', \mathfrak{t}', \mathbf{m}_{sm}') \in \widehat{R}^{m_1}$ for $m_1 = (2d\widehat{k} + 1) + (kd\widehat{k} + 1) + (k\widehat{k} + 1) + \widehat{k} + m_{sm}\widehat{k}$.

First Round. We start by sampling $\mathbf{s}_2$ from χ^{m_2} where $\mathrm{Supp}(\chi) \subseteq \widehat{S}_1$ and compute an Ajtai commitment of $\mathbf{s}_1$ with randomness $\mathbf{s}_2$ as $\mathbf{t}_A = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \bmod \widehat{q}\widehat{R}$, where $\mathbf{A}_1 \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d}\times m_1})$ and $\mathbf{A}_2 \hookleftarrow U(\widehat{R}_{\widehat{q}}^{\widehat{d}\times m_2})$ are part of the common reference string crs. Then, we sample the Gaussian masks for what will later be $c\mathbf{s}_1$ and $c\mathbf{s}_2$. More precisely, we sample $\mathbf{y}_1$ from $\mathcal{D}_{\widehat{R}^{m_1},\sigma_1}$ and $\mathbf{y}_2$ from $\mathcal{D}_{\widehat{R}^{m_2},\sigma_2}$, and compute the commitment $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 \bmod \widehat{q}\widehat{R}$.

We then sample a mask $\mathbf{y}_3$ from $\mathcal{D}_{\widehat{R}^{256/\widehat{n}},\sigma_3}$ and a vector for soundness amplification by $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\}^\ell)$ where all the entries are polynomials with a constant coefficient equal to zero. We later use $\widehat{\mathbf{m}}$ to denote the vector $\widehat{\mathbf{m}} = [\mathbf{y}_3^T|\mathbf{g}^T]^T \in \widehat{R}^{256/\widehat{n}+\ell}$. We commit to it via $\mathbf{t}_B = \mathbf{B}_{y,g}\mathbf{s}_2 + \widehat{\mathbf{m}} \bmod \widehat{q}\widehat{R}$, where $\mathbf{B}_{y,g} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{(256/\widehat{n}+\ell)\times m_2})$ is part of crs.

The prover sends msg_1 as the first message and receives chal_1 as the first challenge, where they are both defined as

$$\mathsf{msg}_1 = (\mathbf{t}_A, \mathbf{t}_B, \mathbf{w}) \in \widehat{R}_{\widehat{q}}^{2\widehat{d}+256/\widehat{n}+\ell}$$
$$\mathsf{chal}_1 = \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1) = (\mathsf{R}_0, \mathsf{R}_1) \in \left(\{0,1\}^{256\times m_1\widehat{n}}\right)^2$$

Second Round. We define $\mathsf{R} = \mathsf{R}_0 - \mathsf{R}_1$. In the second round, we respond to the challenge by masking $\mathsf{R}\tau(\mathbf{s}_1)$ with $\tau(\mathbf{y}_3)$. So we compute $\mathbf{z}_3^{\mathbb{Z}} = \tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1) \in \mathbb{Z}^{256}$. Then, we reject if $\mathsf{Rej}_1(\mathbf{z}_3^{\mathbb{Z}}, \mathsf{R}\tau(\mathbf{s}_1), \sigma_3, M_3) = 0$ (Algorithm 7.7). If the prover accepts, itsends msg_2 as the second message and receives chal_2 as the second challenge where they are defined by

$$\mathsf{msg}_2 = \mathbf{z}_3^{\mathbb{Z}} \in \mathbb{Z}^{256}$$
$$\mathsf{chal}_2 = \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2) = (\gamma_{i,j})_{\substack{i\in[\![\ell]\!]\\ j\in[\![262]\!]}} \in \mathbb{Z}_{\widehat{q}}^{\ell\times 262}.$$

Third Round. We now need to prove the following equations over $\mathbb{Z}_{\widehat{q}}$.

$$\tau(\mathbf{y}_3) + \mathsf{R}\tau(\mathbf{s}_1) = \mathbf{z}_3^{\mathbb{Z}}, \tag{3.1b}$$
$$\langle\tau(\mathbf{v}_1''),\tau(\mathbf{v}_1'')\rangle = {B_1'}^2, \tag{3.2b}$$
$$\langle\tau(\mathbf{v}_2''),\tau(\mathbf{v}_2'')\rangle = B_2^2, \tag{3.3b}$$
$$\langle\tau(\mathbf{v}_3''),\tau(\mathbf{v}_3'')\rangle = B_3^2, \tag{3.4b}$$
$$\langle\tau(\mathbf{t}'),\tau(\mathbf{t}')\rangle = w, \tag{3.5b}$$
$$\langle\tau(\mathbf{t}'),\tau(\mathbf{t}') - \mathbf{1}_{\widehat{n}\widehat{k}}\rangle = 0. \tag{3.6b}$$
$$\langle\tau(\mathbf{m}'_{sm}),\tau(\mathbf{m}'_{sm}) - \mathbf{1}_{\widehat{n}\widehat{k}m_{sm}}\rangle = 0. \tag{3.7b}$$

Using the same method and notations as in Section 7.4.2, we combine the quadratic equations with automorphisms over $\widehat{R}_{\widehat{q}}$ and define for all $i \in [\![\ell]\!]$

$$\begin{aligned} h_i = g_i + \sum_{j\in[\![256]\!]} \gamma_{i,j}(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}) + \gamma_{i,257}(\mathbf{v}_1''^*\mathbf{v}_1'' - {B_1'}^2) \\ + \gamma_{i,258}(\mathbf{v}_2''^*\mathbf{v}_2'' - B_2^2) + \gamma_{i,259}(\mathbf{v}_3''^*\mathbf{v}_3'' - B_3^2) + \gamma_{i,260}(\mathbf{t}'^*\mathbf{t}' - w) \\ + \gamma_{i,261}(\mathbf{t}'^*(\mathbf{t}' - \mathbf{1}_{\widehat{R}^{\widehat{k}}})) + \gamma_{i,262}({\mathbf{m}'_{sm}}^*(\mathbf{m}'_{sm} - \mathbf{1}_{\widehat{R}^{\widehat{k}m_{sm}}})). \end{aligned} \tag{7.9}$$

The prover then sends msg_3 as the third message and receives chal_3 as the third challenge, where they are both defined as

$$\mathsf{msg}_3 = (h_1, \ldots, h_\ell) \in \widehat{R}_{\widehat{q}}^{\ell}$$
$$\mathsf{chal}_3 = \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3) = (\mu_i)_{i\in[\![\ell+d\widehat{k}]\!]} \in \widehat{R}_{\widehat{q}}^{\ell+d\widehat{k}}.$$

Fourth Round. We now need to prove all the quadratic equations over $\widehat{R}_{\widehat{q}}$. We need to prove that the h_i are well-formed and equal their expressions above, and we also need to prove the main quadratic relation $\mathbf{A}''\mathbf{v}_1'' - \mathbf{B}''\mathbf{v}_2'' + \mathbf{A}_3''\mathbf{v}_3'' - \mathbf{D}''_{sm}\mathbf{m}_{sm} + \mathbf{t}'\mathbf{G}''\mathbf{v}_2'' = \mathbf{u}'$. The latter represents $d\widehat{k}$ equations. We prove them all at once by combining them linearly with the challenges μ_i and prove that

$$\begin{aligned} 0 = \sum_{i\in[\![\ell]\!]} \mu_i \Bigg(g_i + \sum_{j\in[\![256]\!]} \gamma_{i,j}(\mathbf{e}_j^*\mathbf{y}_3 + \mathsf{r}_j^*\mathbf{s}_1 - z_{3,j}^{\mathbb{Z}}) + \gamma_{i,257}(\mathbf{v}_1''^*\mathbf{v}_1'' - {B_1'}^2) \\ + \gamma_{i,258}(\mathbf{v}_2''^*\mathbf{v}_2'' - B_2^2) + \gamma_{i,259}(\mathbf{v}_3''^*\mathbf{v}_3' - B_3^2) + \gamma_{i,260}(\mathbf{t}'^*\mathbf{t}' - w) \\ + \gamma_{i,261}(\mathbf{t}'^*(\mathbf{t}' - \mathbf{1}_{\widehat{R}^{\widehat{k}}})) + \gamma_{i,262}({\mathbf{m}'_{sm}}^*(\mathbf{m}'_{sm} - \mathbf{1}_{\widehat{R}^{\widehat{k}m_{sm}}})) - h_i \Bigg) \\ + \sum_{i\in[\![d\widehat{k}]\!]} \mu_{\ell+i}(\mathbf{t}'^T\mathbf{G}_i''\mathbf{v}_2'' + \mathbf{e}_i^{\widehat{R}^T}(\mathbf{A}''\mathbf{v}_1'' - \mathbf{B}''\mathbf{v}_2'' + \mathbf{A}_3''\mathbf{v}_3'' - \mathbf{D}'_{sm}\mathbf{m}'_{sm} - \mathbf{u}'). \end{aligned} \tag{7.10}$$

For that let us define $\widehat{\mathbf{s}} = [\mathbf{s}_1^T|\mathbf{s}_1^*|\widehat{\mathbf{m}}^T|\widehat{\mathbf{m}}^*]^T$. We also define $\mathsf{r}_{1,j}$, $\mathsf{r}_{2,j}$, $\mathsf{r}_{3,j}$, $\mathsf{r}_{t,j}$, and $\mathsf{r}_{sm,j}$ such that

$$\mathsf{r}_j^*\mathbf{s}_1 = \mathsf{r}_{1,j}^*\mathbf{v}_1'' + \mathsf{r}_{2,j}^*\mathbf{v}_2'' + \mathsf{r}_{3,j}^*\mathbf{v}_3'' + \mathsf{r}_{t,j}^*\mathbf{t}' + \mathsf{r}_{sm,j}^*\mathbf{m}'_{sm}.$$

Then, the equation to be proven is equivalent to $\widehat{\mathbf{s}}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\widehat{\mathbf{s}} + f = 0 \bmod \widehat{q}\widehat{R}$, where

$$
\begin{aligned}
f = &- \sum_{i\in[\![\ell]\!]} \mu_i \left(\sum_{j\in[\![256]\!]} \gamma_{i,j} z_{3,j}^{\mathbb{Z}} + \gamma_{i,257} {B'_1}^2 + \gamma_{i,258} B_2^2 + \gamma_{i,259} B_3^2 + \gamma_{i,260} w + h_i \right) \\
&- \sum_{i\in[\![d\widehat{k}]\!]} \mu_{\ell+i} u'_i \\
\mathbf{f} = &\begin{bmatrix}
\sum_{i\in[\![\ell]\!]} \sum_{j\in[\![256]\!]} \mu_i \gamma_{i,j} \mathsf{r}_{1,j}^{*T} + \sum_{i\in[\![d\widehat{k}]\!]} \mu_{\ell+i} \mathbf{A}''^T \mathbf{e}_i^{\widehat{R}} \\
\sum_{i\in[\![\ell]\!]} \sum_{j\in[\![256]\!]} \mu_i \gamma_{i,j} \mathsf{r}_{2,j}^{*T} - \sum_{i\in[\![d\widehat{k}]\!]} \mu_{\ell+i} \mathbf{B}''^T \mathbf{e}_i^{\widehat{R}} \\
\sum_{i\in[\![\ell]\!]} \sum_{j\in[\![256]\!]} \mu_i \gamma_{i,j} \mathsf{r}_{3,j}^{*T} + \sum_{i\in[\![d\widehat{k}]\!]} \mu_{\ell+i} \mathbf{A}''^T_3 \mathbf{e}_i^{\widehat{R}} \\
\sum_{i\in[\![\ell]\!]} \sum_{j\in[\![256]\!]} \mu_i \gamma_{i,j} \mathsf{r}_{t,j}^{*T} \\
\sum_{i\in[\![\ell]\!]} \sum_{j\in[\![256]\!]} \mu_i \gamma_{i,j} \mathsf{r}_{sm,j}^{*T} + \sum_{i\in[\![d\widehat{k}]\!]} \mu_{\ell+i} {\mathbf{D}'_{sm}}^T \mathbf{e}_i^{\widehat{R}} \\
\mathbf{0}_{2d\widehat{k}+1} \\
\mathbf{0}_{kd\widehat{k}+1} \\
\mathbf{0}_{k\widehat{k}+1} \\
-\sum_{i\in[\![\ell]\!]} \mu_i \gamma_{i,261} \mathbf{1}_{\widehat{R}^{\widehat{k}}} \\
-\sum_{i\in[\![\ell]\!]} \mu_i \gamma_{i,262} \mathbf{1}_{\widehat{R}^{\widehat{k}m_{sm}}} \\
\sum_{i\in[\![\ell]\!]} \mu_i \sum_{j\in[\![256]\!]} \gamma_{i,j} \mathbf{e}_j^{*T} \\
[\mu_1 | \dots | \mu_\ell]^T \\
\mathbf{0}_{256/\widehat{n}} \\
\mathbf{0}_\ell
\end{bmatrix} \\
\mathbf{F} = &\begin{bmatrix} \mathbf{F}' & \mathbf{F}'' & \mathbf{0}_{m_1\times 2(256/\widehat{n}+\ell)} \\ \mathbf{0}_{m_1+2(256/\widehat{n}+\ell)\times 2(m_1+256/\widehat{n}+\ell)} \end{bmatrix},
\end{aligned}
\tag{7.11}
$$

where

$$
\mathbf{F}' = \begin{bmatrix} & \mathbf{0}_{\widehat{k}(2d+kd+k)+3\times m_1} & \\ \mathbf{0}_{\widehat{k}\times 2d\widehat{k}+1} & \sum_{i\in[d]} \mu_{\ell+i} \mathbf{G}_i'' & \mathbf{0}_{\widehat{k}\times(m_1-d\widehat{k}(2+k)-2)} \\ & \mathbf{0}_{\widehat{k}m_{sm}\times m_1} & \end{bmatrix},
$$

and

$$
\begin{aligned}
\mathbf{F}'' = \sum_{i\in[\![\ell]\!]} \mu_i \cdot \mathrm{diag}(&\gamma_{i,257}\mathbf{I}_{2d\widehat{k}+1}, \gamma_{i,258}\mathbf{I}_{kd\widehat{k}+1}, \gamma_{i,259}\mathbf{I}_{k\widehat{k}+1}, \\
&(\gamma_{i,260}+\gamma_{i,261})\mathbf{I}_{\widehat{k}}, \gamma_{i,262}\mathbf{I}_{\widehat{k}m_{sm}}).
\end{aligned}
$$

Once we have defined these (public) matrices, we can compute the garbage terms and commit to them. More precisely, we define

$$
\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_1^{*T} \\ -\mathbf{B}_{y,g}\mathbf{y}_2 \\ -(\mathbf{B}_{y,g}\mathbf{y}_2)^{*T} \end{bmatrix} \in \widehat{R}_{\widehat{q}}^{\,2(m_1+256/\widehat{n}+\ell)},
\tag{7.12}
$$

and compute $e_0 = \mathbf{y}^T\mathbf{F}\mathbf{y} \bmod \widehat{q}\widehat{R}$, $e_1 = \widehat{\mathbf{s}}^T\mathbf{F}\mathbf{y} + \mathbf{y}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\mathbf{y} \bmod \widehat{q}\widehat{R}$, and the commitments $t_0 = \mathbf{b}^T\mathbf{y}_2 + e_0 \bmod \widehat{q}\widehat{R}$ and $t_1 = \mathbf{b}^T\mathbf{s}_2 + e_1 \bmod \widehat{q}\widehat{R}$, where $\mathbf{b} \hookleftarrow U(\widehat{R}_{\widehat{q}}^{m_2})$ is part of crs. The prover then sends msg_4 as the fourth message and receives chal_4 as the fourth challenge, where they are both defined as

$$
\begin{aligned}
\mathsf{msg}_4 &= (t_0, t_1) \in \widehat{R}_{\widehat{q}}^2 \\
\mathsf{chal}_4 &= \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) = c \in \mathcal{C}.
\end{aligned}
$$

Fifth Round. In the final round, the prover responds to the challenge by masking $c\mathbf{s}_1$ and $c\mathbf{s}_2$ with $\mathbf{y}_1$ and $\mathbf{y}_2$ respectively. So we compute $\mathbf{z}_1 = \mathbf{y}_1 + c\mathbf{s}_1$ and $\mathbf{z}_2 = \mathbf{y}_2 + c\mathbf{s}_2$. Then, we reject if $\mathsf{Rej}_1(\tau(\mathbf{z}_1), \tau(c\mathbf{s}_1), \sigma_1, M_1) = 0$ or if $\mathsf{Rej}_1(\tau(\mathbf{z}_2), \tau(c\mathbf{s}_2), \sigma_2, M_2) = 0$ (Algorithm 7.7). If the prover accepts, it sends msg_5 as the final message defined by

$$
\mathsf{msg}_5 = (\mathbf{z}_1, \mathbf{z}_2) \in \widehat{R}^{m_1+m_2}.
$$

Non-Interactive Proof

We summarize the proof and verification in Figure 7.2. The proof is $\pi = (\mathbf{t}_A, \mathbf{t}_B, \mathbf{z}_3^{\mathbb{Z}}, h_1, \ldots, h_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_2)$ as the elements $\mathbf{w}$ and t_0 and the challenges can be re-computed from the rest. The elements $\mathbf{t}_A, \mathbf{t}_B, h_1, \ldots, h_\ell, t_1$ cannot be compressed as they all look uniformly random modulo q_π. We again use the entropy bound to evaluate the bit-size of discrete Gaussian vectors. It means the total bit-size can be bounded by

$$|\pi| \leq \left(\widehat{d} + \frac{256}{\widehat{n}} + 2\ell + 1\right)\widehat{n}\lceil\log_2 \widehat{q}\rceil + \widehat{n}m_1(1/2 + \log_2 \sigma_1) + \widehat{n}m_2(1/2 + \log_2 \sigma_2) + 256(1/2 + \log_2 \sigma_3) + \widehat{n}\lceil\log_2(2\rho+1)\rceil.$$

Proof

$\mathsf{keep} \leftarrow 0$
while $\mathsf{keep} = 0$ **do**
  $\mathbf{s}_2 \hookleftarrow \chi^{m_2}$
  $\mathbf{t}_A \leftarrow \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \bmod \widehat{q}\widehat{R}$
  $\mathbf{y}_i \hookleftarrow \mathcal{D}_{\widehat{R}^{m_i},\sigma_i}$ for $i \in \{1,2\}$
  $\mathbf{w} \leftarrow \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 \bmod \widehat{q}\widehat{R}$
  $\mathbf{y}_3 \hookleftarrow \mathcal{D}_{\widehat{R}^{256/\widehat{n}},\sigma_3}$
  $\mathbf{g} \hookleftarrow U(\{x \in \widehat{R}_{\widehat{q}} : \tau_0(x) = 0\}^\ell)$
  $\mathbf{t}_B \leftarrow \mathbf{B}_{y,g}\mathbf{s}_2 + [\mathbf{y}_3^T|\mathbf{g}^T]^T \bmod \widehat{q}\widehat{R}$
  $\mathsf{msg}_1 \leftarrow (\mathbf{t}_A, \mathbf{t}_B, \mathbf{w})$
  $(\mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1) \in (\{0,1\}^{256\times m_1\widehat{n}})^2$
  $\mathsf{R} \leftarrow \mathsf{R}_0 - \mathsf{R}_1$
  $\mathbf{z}_3^{\mathbb{Z}} \leftarrow \tau(\mathbf{y}_3) - \mathsf{R}\tau(\mathbf{s}_1)$
  $\mathsf{keep}_3 \leftarrow \mathsf{Rej}_1(\mathbf{z}_3^{\mathbb{Z}}, \mathsf{R}\tau(\mathbf{s}_1), \sigma_3, M_3)$
  $\mathsf{msg}_2 \leftarrow \mathbf{z}_3^{\mathbb{Z}}$
  $(\gamma_{i,j})_{i,j} \leftarrow \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2) \in \mathbb{Z}_{\widehat{q}}^{\ell\times 262}$
  Compute all h_i as in Equation (7.10)
  $\mathsf{msg}_3 \leftarrow (h_1, \ldots, h_\ell)$
  $(\mu_i)_{i\in[\![\ell+d\widehat{k}]\!]} \leftarrow \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3) \in \widehat{R}_{\widehat{q}}^{\ell+d\widehat{k}}$
  Compute $\widehat{\mathbf{s}}, \mathbf{y}, \mathbf{F}, \mathbf{f}, f$ as in Equations (7.12) and (7.11)
  $e_0 \leftarrow \mathbf{y}^T\mathbf{F}\mathbf{y} \bmod \widehat{q}\widehat{R}$
  $e_1 \leftarrow \widehat{\mathbf{s}}^T\mathbf{F}\mathbf{y} + \mathbf{y}^T\mathbf{F}\widehat{\mathbf{s}} + \mathbf{f}^T\mathbf{y} \bmod \widehat{q}\widehat{R}$
  $t_0 \leftarrow \mathbf{b}^T\mathbf{y}_2 + e_0 \bmod \widehat{q}\widehat{R}$
  $t_1 \leftarrow \mathbf{b}^T\mathbf{s}_2 + e_1 \bmod \widehat{q}\widehat{R}$
  $\mathsf{msg}_4 \leftarrow (t_0, t_1)$
  $c \leftarrow \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) \in \mathcal{C}$
  $\mathbf{z}_1 \leftarrow \mathbf{y}_1 + c\mathbf{s}_1$
  $\mathsf{keep}_1 \leftarrow \mathsf{Rej}_1(\mathbf{z}_1, c\mathbf{s}_1, \sigma_1, M_1)$
  $\mathbf{z}_2 \leftarrow \mathbf{y}_2 + c\mathbf{s}_2$
  $\mathsf{keep}_2 \leftarrow \mathsf{Rej}_1(\mathbf{z}_2, c\mathbf{s}_2, \sigma_2, M_2)$
  $\mathsf{keep} \leftarrow \mathsf{keep}_1 \wedge \mathsf{keep}_2 \wedge \mathsf{keep}_3$
$\pi \leftarrow (\mathbf{t}_A, \mathbf{t}_B, \mathbf{z}_3^{\mathbb{Z}}, h_1, \ldots, h_\ell, t_1, c, \mathbf{z}_1, \mathbf{z}_2)$
return π

Verification

$\mathbf{w} \leftarrow \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A \bmod \widehat{q}\widehat{R}$
$\mathsf{msg}_1 \leftarrow (\mathbf{t}_A, \mathbf{t}_B, \mathbf{w})$
$(\mathsf{R}_0, \mathsf{R}_1) \leftarrow \mathcal{H}(1, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1)$
$\mathsf{msg}_2 \leftarrow \mathbf{z}_3^{\mathbb{Z}}$
$(\gamma_{i,j})_{i,j} \leftarrow \mathcal{H}(2, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2)$
$\mathsf{msg}_3 \leftarrow (h_1, \ldots, h_\ell)$
$(\mu_i)_{i\in[\![\ell+d\widehat{k}]\!]} \leftarrow \mathcal{H}(3, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3)$
Compute $\mathbf{F}, \mathbf{f}, f$ as in Equation (7.11)
$\mathbf{z} \leftarrow \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_1^{*T} \\ c\mathbf{t}_B - \mathbf{B}_{y,g}\mathbf{z}_2 \\ (c\mathbf{t}_B - \mathbf{B}_{y,g}\mathbf{z}_2)^{*T} \end{bmatrix}$
$t_0 \leftarrow \mathbf{z}^T\mathbf{F}\mathbf{z} + c\mathbf{f}^T\mathbf{z} + c^2 f - (ct_1 - \mathbf{b}^T\mathbf{z}_2) \bmod \widehat{q}\widehat{R}$
$\mathsf{msg}_4 = (t_0, t_1)$
$b_1 \leftarrow \|\mathbf{z}_1\|_2 \leq c_{\widehat{n}m_1}\sigma_1\sqrt{\widehat{n}m_1}$
$b_2 \leftarrow \|\mathbf{z}_2\|_2 \leq c_{\widehat{n}m_2}\sigma_2\sqrt{\widehat{n}m_2}$
$b_3 \leftarrow \|\mathbf{z}_3^{\mathbb{Z}}\|_2 \leq c_{256}\sigma_3\sqrt{256}$
$b_4 \leftarrow \forall i \in [\![\ell]\!], \tau_0(h_i) = 0$
$b_5 \leftarrow \mathcal{H}(4, \mathsf{crs}, \mathsf{x}, \mathsf{msg}_1, \mathsf{msg}_2, \mathsf{msg}_3, \mathsf{msg}_4) = c$
return $b_1 \wedge b_2 \wedge b_3 \wedge b_4 \wedge b_5$

Figure 7.2: Non-interactive zero-knowledge argument for credential showing

Security Analysis

The proofs of Lemmas 7.5, 7.6 and 7.7 follow the same reasoning as that of Section 7.4.2. As the proof is presented to be non-interactive, there are a few modifications. In the completeness, the equations that would need to be satisfied on $\mathbf{w}$ and t_0 are automatically verified as these elements are recovered from c in the verification. Instead, one simply need to check that c indeed corresponds to the correct hash output. For knowledge soundness and zero-knowledge, the proof in the non-interactive case follows the same arguments as e.g. [BLNS23b], which only slightly differs.

Lemma 7.5 (Show Proof - Completeness)

Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and let M_1, M_2, M_3 in $(1, \infty)$. For $j \in [\![3]\!]$, we define

$$\gamma_j = \frac{\sqrt{\pi}}{\ln(M_j)} \cdot \left(\sqrt{\ln(1/\varepsilon_j) + \ln(M_j)} + \sqrt{\ln(1/\varepsilon_j)} \right).$$

We let $B = \sqrt{B_1'^2 + B_2^2 + B_3^2 + w + nm_{sm}}$ be a bound on $\|\mathbf{s}_1\|_2$. Let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \gamma_1 \eta B$, $\sigma_2 = \gamma_2 \eta \sqrt{\widehat{n} m_2}$ and $\sigma_3 = \gamma_3 \sqrt{337} B$. Then, the zero-knowledge argument in Figure 7.2 is complete.

Lemma 7.6 (Show Proof - Knowledge Soundness)

Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and let M_1, M_2, M_3 in $(1, \infty)$. For $j \in [\![3]\!]$, we define

$$\gamma_j = \frac{\sqrt{\pi}}{\ln(M_j)} \cdot \left(\sqrt{\ln(1/\varepsilon_j) + \ln(M_j)} + \sqrt{\ln(1/\varepsilon_j)} \right).$$

We let $B = \sqrt{B_1'^2 + B_2^2 + B_3^2 + w + nm_{sm}}$ be a bound on $\|\mathbf{s}_1\|_2$. Then, let χ be a distribution over $\widehat{S}_1$, and let $\sigma_1 = \gamma_1 \eta B$, $\sigma_2 = \gamma_2 \eta \sqrt{\widehat{n} m_2}$, $\sigma_3 = \gamma_3 \sqrt{337} B$, and define $B_{256} = c_{256} \sigma_3 \sqrt{256}$. Assume that $\widehat{q} > \max(B^2, 82/\sqrt{26} \cdot \widehat{n} m_1 B_{256}, 2B_{256}^2/13 - B_{256})$.
Then, the zero-knowledge argument in Figure 7.2 is knowledge sound with an extractor running in expected polynomial time, and soundness error

$$\varepsilon_{\text{sound}}^{(s)} = \frac{2}{|\mathcal{C}|} + q_{\min}^{-\widehat{n}/\kappa} + q_{\min}^{-\ell} + 2^{-128} + \varepsilon_{\text{M-SIS}}$$

where $\varepsilon_{\text{M-SIS}}$ is the hardness bound for $\text{M-SIS}_{\widehat{n}, \widehat{d}, m_1 + m_2, \widehat{q}, \beta}$ for

$$\beta = 8\eta \sqrt{(c_{\widehat{n} m_1} \sigma_1 \sqrt{\widehat{n} m_1})^2 + (c_{\widehat{n} m_2} \sigma_2 \sqrt{\widehat{n} m_2})^2}$$

Lemma 7.7 (Show Proof - Zero-Knowledge)

Let $\varepsilon_1, \varepsilon_2, \varepsilon_3$ be in $(0, 1/2]$ and let M_1, M_2, M_3 in $(1, \infty)$. For $j \in [\![3]\!]$, we define

$$\gamma_j = \frac{\sqrt{\pi}}{\ln(M_j)} \cdot \left(\sqrt{\ln(1/\varepsilon_j) + \ln(M_j)} + \sqrt{\ln(1/\varepsilon_j)} \right).$$

We let $B = \sqrt{B_1'^2 + B_2^2 + B_3^2 + w + nm_{sm}}$ be a bound on $\|\mathbf{s}_1\|_2$. Let χ be a distribution over S_1, and let $\sigma_1 = \gamma_1 \eta B$, $\sigma_2 = \gamma_2 \eta \sqrt{\widehat{n} m_2}$ and $\sigma_3 = \gamma_3 \sqrt{337} B$. We define $m_2' = \widehat{d} + 256/\widehat{n} + \ell + 1$ and assume that $m_2 > m_2'$. Then, the zero-knowledge argument in Figure 7.1 is zero-knowledge. More precisely, there exists a simulator $\mathcal{S}$ that outputs a distribution that is $\varepsilon_{\text{zk}}^{(s)}$-indistinguishable from that of an honest proof, where

$$\varepsilon_{\text{zk}}^{(s)} = \frac{\varepsilon_1}{M_1} + \frac{\varepsilon_2}{M_2} + \frac{\varepsilon_3}{M_3} + 2\delta_{q_{\min}}(m_2, m_2') + \frac{\varepsilon_{\text{M-LWE}}}{1 - \delta_{q_{\min}}(m_2, m_2 - m_2')}$$

where $\varepsilon_{\text{M-LWE}}$ is the hardness bound of $\text{M-LWE}_{\widehat{n}, m_2 - (\widehat{d} + 256/\widehat{n} + \ell + 1), m_2, \widehat{q}, U(\widehat{R}_{\widehat{q}}), \chi}$, and $\delta_{q_{\min}}(\mathsf{a}, \mathsf{b}) = \mathbb{P}_{\mathbf{M} \sim U(\widehat{R}_{q_{\min}}^{\mathsf{b} \times \mathsf{a}})}[\mathbf{M} \cdot \widehat{R}_{q_{\min}}^{\mathsf{a}} \neq \widehat{R}_{q_{\min}}^{\mathsf{b}}]$ is the singularity probability.

7.5 Performance

The anonymous credentials we presented in this chapter strongly depends on the signature with efficient protocols of Section 6.4 and the zero-knowledge arguments of Section 7.4 following the framework from [LNP22]. There is, as a result, a multitude of parameters to consider. We note however that the parameters of the proof system can be adjusted once the parameters of the signature have been set. Indeed, the two main parameters driving the security of lattice systems are the dimension and the modulus. The dimension, which is mostly driven by the ring degree,

can be tweaked in the proof system by adjusting the subring embedding dimension. We then first set the degree n for the SEP, and then the subring dimension $\widehat{n} = n/\widehat{k}$ to adjust the security of the proof system. The same goes for the modulus as we can consider composite moduli $\widehat{q} = qq_1$ and adjust q_1 to achieve the appropriate security target. All in all, we take the parameters from Table 6.3 for the SEP and build upon them by adding the parameters for the proof systems. We choose most parameters to minimize the proof sizes, except for the rejection sampling rates M_j which we take to achieve fewer rejections for a better implementation performance in Chapter 8. Additionally, our suggested parameter set correspond to credentials over 10 hidden attributes. This is mainly for comparison with previous works, but also because it is the order of magnitude we expect for the typical use case of identification documents, e.g., passport, national ID, driver's license, etc.

Table 7.1 gives parameter sets for each of the proof systems we presented, one for the issuance proof (commitment opening and user registration) and one for the show proof. We report the corresponding proof sizes by considering the non-interactive versions of the protocols, and we also give the final classical security of our anonymous credentials following Theorems 7.1 and 7.2. The methodology used to assess the hardness of the plethora of M-SIS, M-ISIS, M-LWE assumptions is the one given in Chapter 9. The same estimation strategy with the construction from Section 6.2 gives a credential proof size of 660 KB for 10 attributes[3].

Symbol	Description	Issuance	Show
	Proof System Parameters		
λ	Security parameter	128	128
$\widehat{n}$	Proof system ring degree	64	64
$\widehat{k}$	Subring embedding dimension	4	4
$\widehat{d}$	Ajtai commitment module rank	20	23
q_1	Modulus factor	$524201 \approx 2^{19}$	$549755813881 \approx 2^{39}$
q_{min}	Smallest modulus factor	$425801 \approx 2^{18.7}$	$425801 \approx 2^{18.7}$
$\widehat{q}$	Proof system modulus (qq_1)	$\approx 2^{37.7}$	$\approx 2^{57.7}$
ℓ	Soundness amplification dimension	7	7
$\mathcal{I}$	Disclosed attributes index set	$\varnothing$	$\varnothing$
m_1	Witness dimension	104	211
m_2	ABDLOP commitment randomness dimension	58	74
χ	ABDLOP commitment randomness distribution	$\mathcal{B}_1$	$\mathcal{B}_1$
ρ	ℓ_∞ norm of challenges	8	8
η	ℓ_1-like norm of challenges	93	93
γ_j	Rejection sampling slacks ($j \in [\![3]\!]$)	48.64	48.64
M_j	Rejection sampling rates ($j \in [\![3]\!]$)	2	2
σ_1	First mask width	369051	582380223
σ_2	Second mask width	275603	311305
σ_3	Third mask width	72848	114957847
	Efficiency Estimates		
$\lvert\pi\rvert$	Proof size (KB)	35.99 KB	**79.58 KB**
	Security Estimates		
$\lambda^\star_{\text{anon}}$	Reached anonymity security	129	
$\lambda^\star_{\text{unf}}$	Reached unforgeability security	124	

Table 7.1: Suggested parameter set for the proof systems and security estimate for the anonymous credentials.

Let us now compare our scheme to the existing lattice-based anonymous credentials [BLNS23b, LLLW23, BCR+23] on their compromise between security and credential proof size, i.e., the size

[3]The parameter selection in this thesis is updated compared to the original paper [JRS23].

of a non-interactive proof in Algorithm 7.6, which represents the main metric we want to optimize over. We first note that the presented anonymous credentials directly stems from the optimized SEP of Section 6.4. Our first anonymous credentials construction was presented in the first paper [JRS23]. In comparison to the figures reported in the latter, which is based on the SEP from Section 6.2, our optimized construction drastically improves upon its performances on all metrics and with a tighter security proof. We also achieve more compact sizes than [LLLW23]. In the latter, the authors propose parameter sets for three different security reductions. The first achieves credential proofs or around 190 KB but for selective unforgeability. The second builds upon the selective parameter set and achieves adaptive security via complexity leveraging for a credential of 370 KB, but it results in a reduction loss of 2^{128}. The third achieves adaptive security directly but results in much larger credentials of around 24.7 MB. Then, the construction of [BCR+23] which is based on the group signature of [dPLS18] reaches credential proofs of around 1.83 MB (for a single hidden attribute) which is again much larger than our work. Finally, in [BLNS23b], the authors relaxed the hardness assumption by introducing the NTRU-ISIS$_f$ (and its interactive version) to reach smaller credentials. We reach credential proofs around 3 times smaller than their construction based on NTRU-ISIS$_f$, and get close to the performance of their construction based on the interactive assumption Int-NTRU-ISIS$_f$, but by relying on standard non-interactive assumptions (M-ISIS, M-SIS, M-LWE). One of the caveat of the latter two schemes [BLNS23b, BCR+23] is that they need to hash the attributes before signing them. From a theoretical perspective, this is not an issue because one can consider that an *effective* attribute is the hash of the *actual* attribute. This however places a limitation on the practical use cases the construction covers. In a variety of them, one may desire the ability to prove statements on their hidden attributes. In the case of age control for example, the user would want to hide their age, represented by the attribute m_i say, while proving that $m_i \geq 18$. Such a statement cannot be proven efficiently on $\mathcal{H}(m_i)$.

We summarize this comparison in Table 7.2. In particular, we are the only scheme achieving credentials smaller than 100 KB without relying on interactive and non-standard assumptions.

	Assumptions	Interactive Assumption	Security	Attribute Statement	Credential Proof Size
Sec. 6.2 [JRS23]	M-(I)SIS/M-LWE	No	Adaptive	Yes	660 KB
[BLNS23b]	NTRU-ISIS$_f$	No	Adaptive	No	243 KB
	Int-NTRU-ISIS$_f$	Yes	Adaptive	No	62 KB
[LLLW23]	M-SIS/M-LWE	No	Selective	Yes	193 KB
	M-SIS/M-LWE	No	Adaptive$^{(+)}$	Yes	372 KB
	M-SIS/M-LWE	No	Adaptive	Yes	25365 KB
[BCR+23]	M-SIS/M-LWE	No	Adaptive	No	1878 KB
Chap. 7 [AGJ+24]	M-(I)SIS/M-LWE	No	Adaptive	Yes	80 KB

Table 7.2: Comparison of existing post-quantum anonymous credentials. They all reach a security target of 128 bits. The "Attribute Statement" column reports the ability to prove statements on the attributes.
$^{(+)}$ The adaptive security proof incurs an exponential loss.

7.6 Conclusion

Building anonymous credentials from signature with efficient protocols has proven to be particularly relevant in the classical setting [CL04, ASM06, PS16]. Post-quantum secure constructions were unfortunately lagging behind as there were no explicit design before the beginning of this thesis. We showed in this chapter that our signatures with efficient protocols of Chapter 6 can be turned into anonymous credentials systems by only slightly adapting generic protocols to obtain signatures on hidden messages and proving knowledge of a valid signature. Several works [BLNS23b, LLLW23, BCR+23] have since then proposed interesting alternative constructions by stretching the security models or assumptions. Although proposing new lattice assumptions to improve the efficiency of constructions is a relevant and promising research direction, we showed that one can obtain the same level of compactness from standard security foundations. The remaining question is about the computational efficiency of our design, which we answer in Chapter 8 by presenting a practically efficient implementation.

8

Implementation of Anonymous Credentials

The chapter focuses on the implementation of the anonymous credentials of Chapter 7. We provide some of the implementation details, the samplers floating-point precision analysis, and implementation benchmarks.

The work presented in this chapter is based on a paper with my co-authors Sven Argo, Tim Güneysu, Georg Land, Adeline Roux-Langlois and Olivier Sanders.

[AGJ+24] **Practical Post-Quantum Signatures for Privacy**. Published at ACM CCS 2024.

Contents

8.1 Introduction

Signature with efficient protocols yields a variety of applications that provide us with an interesting security-privacy balance, like anonymous credentials as discussed in Chapter 7. The natural question is then whether there is a significant downside in using such mechanisms. Typically, one may wonder about the computational performance of advanced authentications compared to regular ones. Taking the example of digital identity, the issuance of a credential is generally done once while the presentation of the latter is done at every authentication. Following the syntax of anonymous credentials, it means the Show protocol must be relatively efficient so as not to hinder user experience. This includes both the generation of the presentation transcript and the verification.

Classical constructions have demonstrated impressive performance on a variety of platforms, answering positively to the above concerns. For example, a prototype implementation of anonymous credentials based on the unlinkable redactable signatures from [San20], with the optimization from [San21], was developped at Orange. It showed that the certificate generation and user computation on a smartphone took around 6 ms and verification around 4 ms, compared to 0.5 ms and 0.4 ms respectively for a regular ECDSA signature.

The state-of-affairs is rather different in the post-quantum setting. Many implementations have been proposed for regular signature schemes that are currently being standardized, showing they

can be as efficient as their classical counterparts, if not more. However, implementations of post-quantum advanced mechanisms are scarcer. Almost all the different constructions of lattice-based anonymous credentials have so far only considered the size metric which is not sufficient when we consider real-world deployments. It prevents us to assess their actual computational complexity. To the best of our knowledge, the only existing implementation of anonymous credentials over lattices is the one from Blazy et al. [BCR+23]. Although it yields promising timings for such a large credential proof size (see Table 7.2), the average duration of the Show protocol is around 2 seconds which is 200 times larger than for classical primitives. The implementation benchmarks are also given for a credential over a single attribute which only addresses very specific use cases.

Because regular lattice signatures are as efficient as classical ones, this discrepancy seems to come from the increasingly complex zero-knowledge proof system that advanced protocols resort to, and the relations that need to be proven. Simple relations, like proving knowledge of an LWE secret, have been shown to be rather efficient to prove [ENS20]. For more intricate frameworks that provide more compact proof sizes, e.g., [LNP22], there is currently no public implementation which makes it hard to assess the performance of the protocols using them solely based on their formal descriptions. This concretely means that, despite the relatively small sizes offered by prior anonymous credentials [JRS23, BLNS23b, LLLW23], it is still impossible to affirm that they provide a real-world solution for the post-quantum transition of privacy-preserving authentication mechanisms.

8.1.1 Our Contributions

In this chapter, we present our implementation of the anonymous credentials of Chapter 7. We implemented the scheme in C to evaluate its concrete performance when run on a laptop. The code is publicly available[1]. We discuss some implementation details and choices and also provide a precision analysis of the elliptic sampler (Algorithm 4.5) needed for our implementation.

Although our code is designed to be portable (it uses a generic arithmetic backend and does not use parallelization), we get timings that we deem reasonable for most use-cases on this type of hardware. In particular, issuance and showing (including verification) of a credential take respectively 400 ms and 500 ms on average, values that seemed beyond reach a few years ago. The full benchmarks for the parameters of Table 7.1 are given in Section 8.4, but we provide a quick comparison with the only such implementation [BCR+23] in Table 8.1.

	Proof Size (KB)	Show Proof Gen.		Show Proof Verif.		Full Show Protocol	
		Time (ms)	Cycles ($\times 10^6$)	Time (ms)	Cycles ($\times 10^6$)	Time (ms)	Cycles ($\times 10^6$)
[BCR+23]	1878 KB	1842.76	5887.53	171.87	549.23	2014.63	6436.76
Ours	80 KB	354.59	993.99	147.14	412.46	504.12	1413.12

Table 8.1: Timing comparison of existing lattice-based anonymous credentials.

Our code also provides a better understanding of the actual performance of the zero-knowledge proof system from [LNP22]. It is therefore likely to have applications outside the sole anonymous credentials area, by providing a way to judge the performance of related privacy-preserving primitives such as group signatures and blind signatures.

8.2 Implementation Details

We start by discussing a selection of the implementation details and techniques we used. Apart from the complexity of the protocols themselves, the first notable challenge we faced was implementing polynomial arithmetic in *five* different rings, each presenting unique characteristics. Among these rings, three operate with coefficients modulo single-precision primes or single-precision products of two primes, posing challenges for efficient multiplication as they inherently lack native support for the Number Theoretic Transform (NTT)[2]. Another ring operates over multi-precision integers in order to estimate the spectral norm during the key generation whose methodology is described in Section 8.2.1. The fifth ring is over $\mathbb{R}$ for the SEP perturbation sampling. We carried a precision

[1] https://github.com/Chair-for-Security-Engineering/lattice-anonymous-credentials

[2] The NTT represents a discrete version of the FFT tailored to operations in R_q. It leads to very efficient computations when q is fully split, i.e., qR factors into n distinct prime ideals of inertia degree 1. In our case, qR factors into only 4 prime ideals, each with inertia degree 64.

analysis of the different Gaussian samplers in order to determine the necessary floating-point precision needed in the implementation of our scheme. It can be found in Section 8.3. Overall, we show that the standard precision of 53 bits is sufficient and leads to no noticeable security loss.

Faced with the intricacies of polynomial arithmetic across multiple rings, and considering that the actual construction is highly complex already[3], we chose FLINT [FLI23] as our arithmetic backend. However, it is important to acknowledge several downsides of this choice: Firstly, FLINT implements arithmetic operations usually in a very generic way which may be non-optimal given that our parameters are static at compile time. Moreover, this generic arithmetic also includes the usage of branches for trivial cases, which breaks the constant-time paradigm for cryptographic implementations. Secondly, FLINT heavily relies on dynamic memory allocations, both internally and when handling passed data. In contrast to stack allocations, which are usually used in cryptographic implementations, these dynamic ones are significantly slower. To mitigate this performance drop to a certain extent, we employ static, pre-allocated variables within the wrapper.

For these reasons, our implementation prioritizes accessibility and clarity for future research. We have abstracted calls to FLINT functions with a wrapper which offers the flexibility to replace the FLINT-based arithmetic with custom constant-time, parameter-specific code without the necessity of touching the protocol layer. Importantly, it requires no other dependencies beyond FLINT, and the code is thoroughly documented to enhance comprehension.

We want to emphasize that, apart from a parameter-specific, optimized arithmetic backend, our code could be further optimized by leveraging vectorized computations for x86 processors supporting the AVX2 instruction set. For example, one could easily deploy AVX2-vectorized hashing in our implementation. Through profiling, however, we have confirmed that for our code hashing is not the main bottleneck for both proof generation procedures as well as the verifications.

8.2.1 Spectral Norm Estimation

We now explain how to sample the SEP signing key, which requires rejecting secret keys based on their spectral norm. More precisely, during the key generation of the signature, we need to enforce a bound on the secret key, i.e., $\left\|M_\tau(\mathbf{R})\right\|_2 \leq \frac{7}{10}(\sqrt{2nd} + \sqrt{nkd} + 6)$. Naively, it would require computing $\left\|M_\tau(\mathbf{R})\right\|_2$ and performing a singular value decomposition. To avoid performing such a decomposition, we only approximate the value of $\left\|M_\tau(\mathbf{R})\right\|_2$. For that, we use the iterated power method, which we tweak to our specific use case. The iterated power method estimates the largest eigenvalue of a matrix $\mathbf{M}$ over $\mathbb{C}$ by selecting a random $\mathbf{u}$ over $\mathbb{C}$ and iterating ℓ times the update $\mathbf{u} \leftarrow \mathbf{Mu}/\|\mathbf{Mu}\|_2$ before returning $\mathbf{u}^H\mathbf{Mu}$ as the estimate of $\lambda_{\max}(\mathbf{M})$. The method is rather simple, but usually converges faster when $\mathbf{M}$ has separated eigenvalues, which is not the case of $M_\tau(\mathbf{R})M_\tau(\mathbf{R})^T$ where each eigenvalue is doubled by conjugation symmetry.

We thus change the approach to optimize this computation. First, we observe that $\left\|M_\tau(\mathbf{R})\right\|_2 = \max_{i\in[\![n]\!]}\left\|\sigma_i(\mathbf{R})\right\|_2$ by Lemma 1.3, where the σ_i are the complex embeddings of the field. As we work in cyclotomic fields, the conjugation symmetry allows to only look at $n/2$ embeddings. Hence, we have

$$\left\|M_\tau(\mathbf{R})\right\|_2 = \max_{i\in[\![n/2]\!]}\left\|\sigma_i(\mathbf{R})\right\|_2 = \max_{i\in[\![n/2]\!]}\sqrt{\lambda_{\max}(\sigma_i(\mathbf{RR}^*))}.$$

We thus only have to estimate $n/2$ maximal eigenvalues of complex matrices with small dimensions ($\mathbb{C}^{2d\times 2d}$). For that we can update the iterated power method as follows. First, the updated vector $\mathbf{u}$ does not have to be re-normalized at each step, meaning that the estimate computes $\widetilde{\mathbf{u}} = \mathbf{M}^\ell\mathbf{u}$ and returns $\widetilde{\mathbf{u}}^H\mathbf{M}\widetilde{\mathbf{u}}/\|\widetilde{\mathbf{u}}\|_2^2 = \mathbf{u}^H\mathbf{M}^{2\ell+1}\mathbf{u}/\mathbf{u}^H\mathbf{M}^{2\ell}\mathbf{u}$. Second, the starting vector $\mathbf{u}$ does not need to be random. In our experiments, choosing $\mathbf{u}$ to be the first column of $\mathbf{M}$ actually converges faster. In this case, the output value is

$$\frac{\mathbf{e}_1^T\mathbf{M}^{2\ell+3}\mathbf{e}_1}{\mathbf{e}_1^T\mathbf{M}^{2\ell+2}\mathbf{e}_1} = \frac{[\mathbf{M}^{2\ell+3}]_{1,1}}{[\mathbf{M}^{2\ell+2}]_{1,1}}.$$

Since $\mathbf{M}$ is some $\sigma_i(\mathbf{RR}^*)$, we have that

$$\left\|M_\tau(\mathbf{R})\right\|_2^2 \approx \max_{i\in[\![n/2]\!]}\frac{\sigma_i([(\mathbf{RR}^*)^{2\ell+3}]_{1,1})}{\sigma_i([(\mathbf{RR}^*)^{2\ell+2}]_{1,1})}.$$

To minimize the number of matrix multiplications, we choose $\ell = 2^{\ell'} - 1$. As we need to compute $\mathbf{RR}^*$ to generate the perturbation sampling material, the spectral norm estimation thus requires

[3] Excluding any arithmetic, our implementation has about 4700 lines of code compared to, e.g., 890 lines for the official Kyber [BDK+18] code without arithmetic.

$\ell'+1$ matrix multiplication over $R^{2d\times 2d}$ to get $(\mathbf{R}\mathbf{R}^*)^{2\ell+2}$, 1 extra multiplication to get $(\mathbf{R}\mathbf{R}^*)^{2\ell+3}$ and then the computation of $2\cdot n/2$ complex embeddings, i.e., two half FFT. In our implementation, we choose $\ell'=4$ which gives the estimate

$$\left\|M_\tau(\mathbf{R})\right\|_2^2 \approx \max_{i\in[\![n/2]\!]} \frac{\sigma_i([(\mathbf{R}\mathbf{R}^*)^{33}]_{1,1})}{\sigma_i([(\mathbf{R}\mathbf{R}^*)^{32}]_{1,1})}.$$

It approximates the actual norm with at least 10^{-5} precision, which is more than sufficient for our purposes. One could even reduce the value of ℓ' if they can tolerate a slightly looser estimate. Additionally, as we only perform two (half) FFT, we do not need a large precision for the FFT computation.

We note that although this estimate is rather fast, it requires computing $(\mathbf{R}\mathbf{R}^*)^{33}$ in R and not R_q. As a result, the coefficients of $(\mathbf{R}\mathbf{R}^*)^{33}$ become extremely large (around 420 bits in our case) which calls for multi-precision integers. The renormalization in the iterated power method may mitigate this blow-up of coefficients but would require working over the complex embedded matrices directly. It in turn leads to more FFT computations (for all the matrix embeddings) and operations over floating-point complex numbers. This would also require using a larger precision for the FFT in the first place.

8.2.2 Choosing Tags

The specifications of our anonymous credentials features a function F that is used to derive tags deterministically from a state counter st. As discussed in Chapter 6, there is no requirement on F except that it should prevent collisions of tags. Although we discarded choosing the tags as hash outputs of the messages, it can be computed from the state instead. We thus generate them from an extendable output function (SHAKE256) with st as input. This procures a pseudorandom buffer which we need to convert to map to the proper tag space. For that we use the Fisher-Yates shuffle, e.g., [Knu98, Alg. P], which is used for example in Dilithium [DKL+18, Fig. 2]. We recall it here for completeness.

Algorithm 8.1: F

Input: State st, Weight w.

1. $\mathsf{t}\leftarrow 0$. ▷ and SHAKE256.Absorb(st)
2. **For** $i\in[\![n-w,n[\![$ **do**
3. $\quad j\hookleftarrow U([\![0,i]\!])$ ▷ using SHAKE256.Squeeze
4. $\quad \tau_i(\mathsf{t})\leftarrow\tau_j(\mathsf{t})$
5. $\quad \tau_j(\mathsf{t})\leftarrow 1$

Output: $\mathsf{t}\in\mathcal{T}_w$. ▷ $\mathcal{T}_w=\{\mathsf{t}\in T_1 : \|\mathsf{t}\|_1=w\}$

8.2.3 Simpler Gadget Sampling Description

In our implementation, we decide to slightly change our definition of the gadget matrix $\mathbf{G}$ by essentially permuting its columns. In this thesis, we presented the gadget matrix $\mathbf{G}=\mathbf{I}_d\otimes[1|b|\dots b^{k-1}]$ according to the original paper of Micciancio and Peikert [MP12]. We observe however that defining it as $\mathbf{G}=[1|b|\dots|b^{k-1}]\otimes\mathbf{I}_d$ gives a simpler expression, as it is now equal to $[\mathbf{I}_d|b\mathbf{I}_d|\dots|b^{k-1}\mathbf{I}_d]$. Notice that it preserves the expected properties of the gadget lattice because they are the same up to permutation. In particular, it does not deteriorate the quality of the gadget Gaussian sampling, nor does it hinder the arguments that were specific to $\mathbf{G}$, i.e., the trapdoor switching of Section 6.3. This simpler form allows us to parse vectors of size dk as k vectors of size d and perform the multiplication by $\mathbf{G}$ more efficiently. More precisely, given $\mathbf{v}=[\mathbf{v}_0^T|\dots|\mathbf{v}_{k-1}^T]^T$, $\mathbf{G}\mathbf{v}=\mathbf{v}_0+b\mathbf{v}_1+\ldots+b^{k-1}\mathbf{v}_{k-1}$, which represents k multiplications between a vector of R^d and a scalar integer, and $k-1$ additions over R^d.

This change also yield a more compact description of the gadget sampling step. The elliptic sampler of Algorithm 4.5 needs to first compute an arbitrary solution $\mathbf{c}$ of $\mathbf{G}\mathbf{c}=\mathbf{w}$ mod qR, before sampling $\mathbf{y}$ from $\mathcal{D}_{\mathcal{L}_q^\perp(\mathbf{G}),s_\mathbf{G},-\mathbf{c}}$ using KleinSampler (Algorithm 4.2). As there is no restriction on $\mathbf{c}$, the first step can simply use $\mathbf{c}=[\mathbf{w}^T|\mathbf{0}|\dots|\mathbf{0}]^T$ as $\mathbf{G}\mathbf{c}=\mathbf{I}_d\mathbf{w}+\sum_{i\in[\![k-1]\!]}b^i\mathbf{I}_d\mathbf{0}$. This avoids interlacing the coordinates of $\mathbf{w}$ to construct $\mathbf{c}$.

Now, let us explain how to specify KleinSampler to this specific gadget structure. Recall that the latter (Algorithm 4.2) takes as input a basis $\mathbf{B}_\mathbf{G}$ of $\mathcal{L}_q^\perp(\mathbf{G})$, its scaled Gram-Schmidt $\widetilde{\mathbf{B}_\mathbf{G}}'$, widths $(s_i)_{i\in[\![ndk]\!]}$ and the center $\mathbf{c}$ (or $-\mathbf{c}$ actually). The structure of $\mathbf{G}$ yields that $\mathbf{B}_\mathbf{G}=\mathbf{B}_\mathbf{g}\otimes\mathbf{I}_{nd}$ and

$\widetilde{\mathbf{B_G}}' = \widetilde{\mathbf{B_g}}' \otimes \mathbf{I}_{nd}$, where $\mathbf{B_g} \in \mathbb{Z}^{k \times k}$ is defined in Section 4.3.1. This means we only have to store k^2 integers and rationals. Similarly, the widths s_i can be precomputed and stored using only k double-precision floats.

KleinSampler then performs inner products with the columns of the scaled Gram-Schmidt. Because of its form, each column only has k non-zero coefficients which means that each inner product can be performed using k multiplications of an integer and a rational, and $k-1$ additions of rationals. The integers involved in the inner product can be obtained by an easy slicing of the corresponding vector. More precisely, let $\widetilde{\mathbf{B_g}}' = [b'_{i,j}]_{i,j \in [\![0,k[\![} \in \mathbb{Q}^{k \times k}$, and $\mathbf{y} = [\mathbf{y}_0^T | \dots | \mathbf{y}_{k-1}^T]^T \in \mathbb{Z}^{ndk}$. If $\widetilde{\mathbf{b}}'_i$ is the i-th column of $\widetilde{\mathbf{B_G}}'$ for $i \in [\![0, ndk[\![$, we then have

$$\langle \widetilde{\mathbf{b}}'_i, \mathbf{y} \rangle = \sum_{j \in [\![0,k[\![} b'_{i \ // \ nd, j} \cdot [\mathbf{y}_j]_{i \ \% \ nd}$$

where $i = (i \ // \ nd) \cdot nd + (i \ \% \ nd)$ is the Euclidean division of i by nd. Finally, the update stage $\mathbf{v}_{i-1} = \mathbf{v}_i + z_i \mathbf{b}_i$ in KleinSampler can be dealt with k integer multiplications and additions. We now give the updated description of the implemented algorithm. Instead of $\mathbf{c}$, it simply takes $\mathbf{w}$ because of the observation above. The negative sign is relocated to the scaled Gram-Schmidt directly.

Algorithm 8.2: GadgetKlein

Input: $\mathbf{w} \in R^d$.

1. $\mathbf{v} = [\mathbf{v}_0 | \dots | \mathbf{v}_{k-1}] = [\mathbf{0}_d | \dots | \mathbf{0}_d] \in R^{dk}$.
2. **For** $i = ndk, \dots, 1$ **do**
3. $\quad (i_1, i_2) = ((i-1) \ // \ nd, (i-1) \ \% \ nd)$.
4. $\quad (i_{2,1}, i_{2,2}) = (i_2 \ // \ n, i_2 \ \% \ n)$.
5. $\quad \texttt{wcoeff} = \tau_{i_{2,2}}(w_{i_{2,1}})$ ▷ Slicing
6. $\quad$ **If** $i = ndk - 1$ **then** $d_i = \texttt{neg\_scaled\_gso}_{0,i_1} \cdot \texttt{wcoeff}$.
7. $\quad$ **Else**
8. $\quad\quad \texttt{vcoeff} = \tau_{i_{2,2}}([\mathbf{v}_0]_{i_{2,1}})$. ▷ Slicing
9. $\quad\quad d_i = \texttt{neg\_scaled\_gso}_{0,i_1} \cdot (\texttt{wcoeff} + \texttt{vcoeff})$.
10. $\quad\quad$ **For** $j \in [\![k-1]\!]$ **do**
11. $\quad\quad\quad \texttt{vcoeff} = \tau_{i_{2,2}}([\mathbf{v}_j]_{i_{2,1}})$. ▷ Slicing
12. $\quad\quad\quad d_i = d_i + \texttt{neg\_scaled\_gso}_{j,i_1} \cdot \texttt{vcoeff}$.
13. $\quad z_i \hookleftarrow \mathcal{D}_{\mathbb{Z}, \texttt{widths}_{i_1}, d_i}$. ▷ Base Sampler
14. $\quad$ **For** $j \in [\![0,k[\![$ **do**
15. $\quad\quad \tau_{i_{2,2}}([\mathbf{v}_j]_{i_{2,1}}) = \tau_{i_{2,2}}([\mathbf{v}_j]_{i_{2,1}}) + z_i \cdot \texttt{basis}_{j,i_1}$. ▷ Update

Output: $\mathbf{v}$.

The variables `neg_scaled_gso`, `widths`, `basis` correspond to $-\widetilde{\mathbf{B_g}}' \in \mathbb{Q}^{k \times k}$, $(s_{\mathbf{G}} / \|\widetilde{\mathbf{b}}_i\|_2)_{i \in [\![0,k[\![}$ and $\mathbf{B_g} \in \mathbb{Z}^{k \times k}$ respectively, which are public and defined as static parameters in the implementation.

8.3 Samplers Precision Analysis

In this section, we detail the precision analysis of the different samplers that we require to determine the minimal floating-point precision for our implementation. The systematic analysis of floating-point arithmetic (FPA) precision in Gaussian samplers has been bootstrapped by Prest and Lyubashevsky [LP15, Pre15, Pre17]. In these works, they provide a detailed floating-point precision analysis of Klein's sampler [Kle00, GPV08] and Peikert's sampler [Pei10].

Our construction uses three kinds of Gaussian samplers. The first is a spherical Gaussian sampler over $\mathbb{Z}$ (or $\mathbb{Z}^N$ or R), and is used as a base sampler for the others as well as for the masks in the zero-knowledge arguments. The second is the one presented in Algorithm 4.3 which samples a non-spherical perturbation over R. Finally, the third is the gadget sampler to sample points on $\mathcal{L}_q^\perp(\mathbf{G})$ based on Klein's sampler in Algorithms 4.2 and 8.2. The base sampler is well understood and well studied which is why we only focus on the remaining two by assuming a perfect base sampler over $\mathbb{Z}$.

8.3.1 Klein's Sampler on the Gadget Lattice

The analysis of KLEIN's sampler has been thoroughly done in the general case by PREST [Pre15, Pre17]. As our version based on the scaled Gram-Schmidt is thoroughly equivalent, one can carry the same precision analysis but by assuming a precision error on the scaled Gram-Schmidt instead of the Gram-Schmidt itself. In our case, the scaled Gram-Schmidt is rational and could possibly be stored exactly if the denominators are not too large.

Also, as observed in Section 8.2.3, the gadget sampling material has a very specific structure which allows us to reduce the number of multiplications and additions compared to the general case, thus reducing the propagation of the errors. We therefore carry the precision analysis of Algorithm 8.2 by following the blueprint of [Pre17], and specify it when `neg_scaled_gso` is known exactly.

Lemma 8.1 (Adapted from [Pre15, Lem. 3.12])

Let $n, d, b, k, s_{\mathbf{G}}, \varepsilon$ be defined as in $\mathsf{SEP}^\star.\mathsf{Setup}$ (Algorithm 6.5), and $\mathbf{w} \in R^d$. We let $\mathcal{P}$ be the output of $\mathsf{KleinGadget}(\mathbf{w})$ of Algorithm 8.2 where the variables `neg_scaled_gso`, `widths` correspond to $-\widetilde{\mathbf{B}_{\mathbf{g}}}' \in \mathbb{Q}^{k\times k}$ and $(s_i)_i = (s_{\mathbf{G}}/\|\widetilde{\mathbf{b}}_i\|_2)_{i\in[\![0,k[\![}$ respectively precomputed with infinite precision. Similarly, let $\overline{\mathcal{P}}$ be the output distribution of $\mathsf{KleinGadget}(\mathbf{w})$ where the variables correspond to $\overline{-\widetilde{\mathbf{B}_{\mathbf{g}}}'}$ and $(\overline{s_i})_i$ which are precomputed with finite precision. Let $\delta \in [0, 1)$ be such that

- $\forall i \in [\![k]\!], \|\overline{\widetilde{\mathbf{b}}_i'} - \widetilde{\mathbf{b}}_i'\|_\infty \leq \delta$
- $\forall i \in [\![k]\!], |s_i - \overline{s}_i| \leq \delta s_i$

We then define

$$C = \delta\frac{\pi ndk}{(1-\delta)^2}\left(2kc_k\|\widetilde{\mathbf{B}_{\mathbf{g}}}\|\left(c_{ndk} + \frac{(1+\delta)^2\varepsilon}{1-\varepsilon}\right) + (2+\delta)\left(c_{ndk}^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right)\right)$$
$$+ \frac{\pi\delta^2(1+(1+\delta)^2)c_k^2 ndk^3\|\widetilde{\mathbf{B}_{\mathbf{g}}}\|^2}{(1-\delta)^2}$$
$$\underset{\delta,\varepsilon\to 0}{\sim} \delta\cdot ndk(2\pi kc_kc_{ndk}\sqrt{b^2+1} + 2\pi c_{ndk}^2 + 1).$$

Then, it holds that $\overline{\mathcal{P}} \approx_{e^{-C},e^C} \mathcal{P}$. When $\widetilde{\mathbf{B}_{\mathbf{g}}}'$ can be computed exactly, the expression of C is improved to

$$C = \delta\cdot ndk\frac{2\pi(1+\delta/2)}{(1-\delta)^2}\left(c_{ndk}^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right) \underset{\delta,\varepsilon\to 0}{\sim} \delta\cdot ndk(2\pi c_{ndk}^2 + 1).$$

Proof (Lemma 8.1). Let $\mathbf{v} \in \mathcal{L}_q^\perp(\mathbf{G})$ be a possible outcome of $\mathcal{P}$ and $\overline{\mathcal{P}}$. There exists a unique $\mathbf{z} \in \mathbb{Z}^N$ such that $\tau(\mathbf{v}) = \mathbf{B}_{\mathbf{G}}\mathbf{z}$ whose entries are the outputs of the base sampler in the loop. In particular, there exists a unique $(d_i)_i$ (resp. $(\overline{d}_i)_i$) such that the infinite (resp. finite) precision sampler computes those centers in the loop. We write $\mathbf{c} = [\mathbf{w}^T|\mathbf{0}|\dots|\mathbf{0}]^T$ and $\mathbf{y} = \tau(\mathbf{c}+\mathbf{v}) \in \mathbb{Z}^{ndk}$. Also for clarity, we omit the subscript $\mathbf{G}$ or $\mathbf{g}$ in the widths and (scaled) Gram-Schmidt.

We first bound the differences $\left|d_i - \overline{d}_i\right|$. Let $i \in [\![ndk]\!]$. We can rewrite d_i in terms of $\mathbf{y}$ rather than $\mathbf{v}_i$ as $d_i = \langle\mathbf{y}, -\widetilde{\mathbf{b}}_i'\rangle + z_i$. Hence, $\overline{d}_i = \langle\mathbf{y}, -\widetilde{\mathbf{b}}_i' + \delta_i\rangle + z_i = d_i + \langle\mathbf{y}, -\delta_i\rangle$, where by assumption $\|\delta_i\|_\infty \leq \delta$ and δ_i has at most k non-zero entries. This gives

$$\left|d_i - \overline{d}_i\right| \leq \left\|\mathsf{proj}_i(\mathbf{y})\right\|_2\|\delta_i\|_2 \leq c_k s\sqrt{k}\cdot\delta\sqrt{k} = c_k s\delta k,$$

where $\mathsf{proj}_i(\mathbf{y}) \in \mathbb{Z}^k$ is the subvector of $\mathbf{y}$ whose entries are $y_{i \,\%\, nd}, y_{nd+(i \,\%\, nd)}, \dots, y_{(k-1)nd+(i \,\%\, nd)}$. The last inequality comes from the (uncentered) Gaussian tail bound of Lemma 1.21. Indeed, $\mathbf{y}$ is close to a *centered* spherical discrete Gaussian on the coset $\mathcal{L}_q^{\mathbf{w}}(\mathbf{G})$. As such $\mathsf{proj}_i(\mathbf{y})$ follows a spherical discrete Gaussian on the

projected lattice coset $\mathsf{proj}_i(\mathcal{L}_q^{\mathbf{w}}(\mathbf{G}))$. Although it is not needed at this stage, we note that $s = s_{\mathbf{G}} \geq \eta_\varepsilon(\mathcal{L}_q^\perp(\mathbf{G})) \geq \eta_\varepsilon(\mathsf{proj}_i(\mathcal{L}_q^\perp(\mathbf{G})))$. Note that $d_i = \overline{d}_i$ if $\widetilde{\mathbf{B}_{\mathbf{g}}}'$ can be computed exactly.

Now, we bound the ratio $\mathcal{P}(\mathbf{v})/\overline{\mathcal{P}}(\mathbf{v})$. Since both sampler output $\mathbf{v}$ only if the base samplers output the z_i, we have

$$\frac{\mathcal{P}(\mathbf{v})}{\overline{\mathcal{P}}(\mathbf{v})} = \prod_{i\in[\![ndk]\!]} \frac{\rho_{s_i,d_i}(z_i)\rho_{\overline{s}_i,\overline{d}_i}(\mathbb{Z})}{\rho_{\overline{s}_i,\overline{d}_i}(z_i)\rho_{s_i,d_i}(\mathbb{Z})} = \prod_{i\in[\![ndk]\!]} e^{u_i(z_i)} \frac{\rho_{\overline{s}_i,\overline{d}_i}(\mathbb{Z})}{\rho_{s_i,d_i}(\mathbb{Z})},$$

where $u_i(z) = \pi(z-\overline{d}_i)^2/\overline{s}_i^2 - \pi(z-d_i)^2/s_i^2$. By [Pre15, Lem. 3.10], we can bound the ratio of Gaussian sums and get

$$A := \sum_{i\in[\![ndk]\!]} u_i(z_i) - \mathbb{E}_{z\sim\mathcal{D}_i}[u_i(z)] \leq \ln \frac{\mathcal{P}(\mathbf{v})}{\overline{\mathcal{P}}(\mathbf{v})} \leq \sum_{i\in[\![ndk]\!]} u_i(z_i) - \mathbb{E}_{z\sim\overline{\mathcal{D}}_i}[u_i(z)] =: B,$$

where $\mathcal{D}_i = \mathcal{D}_{\mathbb{Z},s_i,d_i}$ and $\overline{\mathcal{D}}_i = \mathcal{D}_{\mathbb{Z},\overline{s}_i,\overline{d}_i}$. We now have to show that $-C \leq A$ and $B \leq C$.

First, we rewrite $u_i(z)$ in two different ways as in [Pre15]. We have

$$u_i(z) = \frac{\pi}{\overline{s}_i^2}((d_i-\overline{d}_i)^2 + 2(d_i-\overline{d}_i)(z-d_i) - \delta_i(2+\delta_i)(z-d_i)^2)$$
$$u_i(z) = \frac{\pi}{\overline{s}_i^2}(-(1+\delta_i)^2(d_i-\overline{d}_i)^2 + 2(1+\delta_i)^2(d_i-\overline{d}_i)(z-\overline{d}_i) - \delta_i(2+\delta_i)(z-\overline{d}_i)^2),$$

where $\overline{s}_i = (1+\delta_i)s_i$ with $|\delta_i| \leq \delta$ by assumption. We use the first expression and have the following inequalities. We can upper-bound $|A|$ by

$$\begin{aligned}
&\sum_{i\in[\![ndk]\!]} \frac{\pi}{\overline{s}_i^2}\left(2\left|d_i-\overline{d}_i\right|\left(|z_i-d_i| + \left|\mathbb{E}_{z\sim\mathcal{D}_i}[z-d_i]\right|\right) + \delta_i(2+\delta_i)((z_i-d_i)^2 + \mathbb{E}_{z\sim\mathcal{D}_i}[(z-d_i)^2])\right)\\
&\leq \frac{\pi}{(1-\delta)^2 s^2} \sum_{i\in[\![ndk]\!]} 2c_k s\delta k\|\widetilde{\mathbf{b}}_i\|_2^2\left(|z_i-d_i| + \left|\mathbb{E}_{z\sim\mathcal{D}_i}[z-d_i]\right|\right)\\
&\qquad\qquad + \delta(2+\delta)\|\widetilde{\mathbf{b}}_i\|_2^2((z_i-d_i)^2 + \mathbb{E}_{z\sim\mathcal{D}_i}[(z-d_i)^2])\\
&\leq \frac{2\pi c_k sk\delta\|\widetilde{\mathbf{B}}\|}{(1-\delta)^2 s^2}\left(\|\mathbf{y}\|_1 + ndks\frac{\varepsilon}{1-\varepsilon}\right) + \frac{\pi\delta(2+\delta)}{(1-\delta)^2 s^2}\left(\|\mathbf{y}\|_2^2 + ndks^2\left(\frac{1}{2\pi}+\frac{\varepsilon}{1-\varepsilon}\right)\right)\\
&\leq \frac{2\pi c_{ndk} ndk^2\delta\|\widetilde{\mathbf{B}}\|}{(1-\delta)^2}\left(c_{ndk} + \frac{\varepsilon}{1-\varepsilon}\right) + \frac{\pi\delta ndk(2+\delta)}{(1-\delta)^2}\left(c_{ndk}^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right)\\
&\leq C,
\end{aligned}$$

where the second inequality comes from the bound on $\left|d_i-\overline{d}_i\right|$ and the fact that $\overline{s}_i = (1+\delta_i)s/\|\widetilde{\mathbf{b}}_i\|_2 \geq (1-\delta)s/\|\widetilde{\mathbf{b}}_i\|_2$. The third inequality comes by bounding $\|\widetilde{\mathbf{b}}_i\|_2$ by $\|\widetilde{\mathbf{B}}\|$, by the fact that $\sum_i \|\widetilde{\mathbf{b}}_i\|_2|z_i-d_i| = \|\mathbf{y}\|_1$, $\sum_i \|\widetilde{\mathbf{b}}_i\|_2^2(z_i-d_i)^2 = \|\mathbf{y}\|_2^2$ and by bounding the expectations using [MR07, Lem. 4.2] as we have $s_i \geq 2\eta_\varepsilon(\mathbb{Z})$. The fourth inequality comes from the Gaussian tail bound of Lemma 1.21 as above and $\|\mathbf{y}\|_1 \leq \sqrt{ndk}\|\mathbf{y}\|_2$.

Following the method of [Pre15], we use the first expression of u_i for the $u_i(z_i)$ and the second expression for the expectations. Using the same arguments as before, we obtain

$$\begin{aligned}
|B| &\leq \frac{\pi\delta ndk}{(1-\delta)^2}\left(2kc_k\|\widetilde{\mathbf{B}}\|\left(c_{ndk} + \frac{(1+\delta)^2\varepsilon}{1-\varepsilon}\right) + (2+\delta)\left(c_{ndk}^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right)\right)\\
&\quad + \frac{\pi\delta^2(1+(1+\delta)^2)c_k^2 ndk^3\|\widetilde{\mathbf{B}}\|^2}{(1-\delta)^2}\\
&= C
\end{aligned}$$

The equivalence is taken at the first order in δ and ε which indeed simplifies to $\delta ndk(2\pi kc_k c_{ndk}\|\widetilde{\mathbf{B}}\| + 2\pi c_{ndk}^2 + 1)$.

Finally, the expression of C when $\widetilde{\mathbf{B}}'$ can be represented exactly comes from the exact same process. The only difference is that $d_i = \overline{d}_i$ which simplifies the two expressions of $u_i(z)$ to $u_i(z) = -\pi\delta_i(2+\delta_i)(z-d_i)^2/\overline{s}_i^2$.

Since the gadget lattice has a specific structure, we can derive a closed-form expression of the scaled Gram-Schmidt which is why we may decide to store it exactly. For typical parameters as those from Table 6.3 where $n = 256$, $k = 5$, $d = 4$, $b = 14$, we have $c_{ndk} \approx 0.453$ and $c_k \approx 2.555$. It then holds that $C \approx 2^{21.33}\delta$ in the general case and $C \approx 2^{13.52}\delta$ in the exact scaled Gram-Schmidt case. Plugging this in our security proof gives a requirement of 42 bits and 34 bits of precision respectively aiming for $C = 1/2\sqrt{\lambda Q}$. The standard precision of 53 bits for floating points is therefore enough to incur almost no security loss.

8.3.2 Perturbation Sampler

The perturbation sampler is very similar to the Fast Fourier Sampler of [DP16] which is used in the Falcon signature scheme [PFH+20]. The algorithm is recursive in the subroutine that samples from $\mathcal{D}_{R,\sqrt{M_\tau(f_i)},d_i}$. In particular, it makes an overall number of $2d \cdot n$ calls to integer samplers $\mathcal{D}_{\mathbb{Z},s_j,e_j}$. We can, as is done for the Fast Fourier Sampler, analyze the precision needed for Algorithm 4.3 using an adapted version of the analysis of Klein's sampler. More precisely, we assume a relative error of at most δ_s on the s_j and an absolute error of at most δ_e on the centers e_j. We thus bound the quantities $\left|\overline{e}_j - e_j\right|$ by δ_e in the above proof, and the $\left|z_j - e_j\right|$ by $s_j t$ using Lemma 1.21 for a slack $t \approx 6$. Using those upper bounds, and the fact that $s_j \geq \eta_\varepsilon(\mathbb{Z})$, we obtain that the relative error between the infinite and finite precision versions of the sampler is of $e^C - 1$, for

$$C = \frac{2\pi N}{(1-\delta_s)^2}\left(\frac{\delta_e}{\eta_\varepsilon(\mathbb{Z})}\left(t + (1+\delta_s)^2\frac{\varepsilon}{1-\varepsilon}\right) + \delta_s\left(1+\frac{\delta_s}{2}\right)\left(t^2 + \frac{1}{2\pi} + \frac{\varepsilon}{1-\varepsilon}\right) + \delta_e^2\frac{1+(1+\delta_s)^2}{2\eta_\varepsilon(\mathbb{Z})^2}\right)$$
$$\leq N(2\pi t^2 + 2\pi\sqrt{\varepsilon} + 1)(\delta_s + \delta_e),$$

where $N = 2nd$, and where the inequality holds for all $\varepsilon, \delta_s, \delta_e \leq 2^{-10}$. In our context, this gives $C \leq 2^{18.83}(\delta_s + \delta_e)$, which, when plugged into our security proof, gives a precision requirement of $\delta_s + \delta_e \leq 2^{-39.4}$.

We use the same methodology than [PFH+20] to verify this bound. More precisely, we ran the signature process in both standard precision of 53 bits and high precision of 200 bits using the same random tape[4]. By comparing the values of the s_j and e_j between the two versions, we observe that we have $\delta_s + \delta_e \leq 2^{-36.9}$. Although this is slightly higher that $2^{-39.4}$, choosing the standard precision of 53 bits gives a sufficient margin so that it incurs no noticeable loss of security.

8.4 Implementation Benchmark

We now give a benchmark our implementation on a laptop featuring an Intel Core i7 12800H CPU running at 4.6 GHz and the scaling governor set to `performance`. Both our code and the FLINT library have been compiled with `gcc 11.4.0` with the options `-O3 -march=native`. For building FLINT, we explicitly enabled AVX2 and disabled the `pthread` option to ensure that no thread pools are used and the program runs on a single core.

We decided to benchmark each of the main steps of the signature scheme and protocols. More precisely, we first give the timings for the SEP as a standalone signature, namely running SEP*.KeyGen, SEP*.Sign and SEP*.Verify from Algorithms 6.6, 6.7, and 6.8. Then, we benchmark the Issue protocol from Algorithm 7.5 by breaking it down into several tasks. The ones prefixed by U are user tasks, while those prefixed by S are performed by the signer. The correspondance with the protocol is as follows.

- U. key gen.: Algorithm 7.4
- U. commit: Algorithm 7.5, steps 1 and 2.
- U. embed, U. prove, S. verify: Algorithm 7.5, step 4.
- S. sign cmt.: Algorithm 7.5, steps 5 to 10.
- U. verify: Algorithm 7.5, step 13.

The embed procedure corresponds to embedding the relation in the subring used in the zero-knowledge argument. We did not include the signature completion corresponding to Algorithm 7.5, steps 11 and 12. We finish by benchmarking the Show protocol from Algorithm 7.6, which includes embedding the relation, generating the proof, and the verification by the verifier V. The timing results are shown in Table 8.2 in milliseconds as well as cycle counts (in million cycles).

[4]Sampling can easily be made deterministic by generating the needed randomness via an extendable output function such as SHAKE256.

Protocol	Procedure	Time (ms)				Cycles ($\times 10^6$)			
		min	mean	med	max	min	mean	med	max
SEP	key gen.	241.01	414.21	270.33	1086.56	675.60	1161.12	757.79	3045.88
	sign	57.36	58.83	58.51	61.73	160.78	164.90	164.00	173.03
	verify	1.68	1.69	1.69	1.70	4.68	4.71	4.71	4.75
Credential Issuance	U. key gen.	0.46	0.47	0.47	0.53	1.27	1.30	1.29	1.48
	U. commit	0.79	0.81	0.81	0.88	2.20	2.25	2.25	2.44
	U. embed	0.74	0.78	0.78	0.86	2.07	2.17	2.17	2.40
	U. prove	126.57	221.33	167.20	644.58	354.80	620.44	468.68	1806.93
	S. verify	100.01	100.94	100.78	103.98	280.32	282.93	282.50	291.48
	S. sign cmt.	56.42	56.84	56.75	62.49	158.15	159.30	159.06	175.15
	U. verify	1.68	1.69	1.69	1.76	4.69	4.72	4.71	4.91
	Total	286.67	382.86	328.48	815.08	803.50	1073.11	920.66	2284.79
Credential Showing	U. embed	2.35	2.39	2.38	2.52	6.56	6.67	6.65	7.05
	U. prove	197.42	354.59	280.29	1019.18	553.41	993.99	785.72	2856.98
	V. verify	145.96	147.14	147.10	152.21	409.15	412.46	412.32	426.67
	Total	345.73	504.12	429.77	1173.91	969.12	1413.12	1204.69	3290.70

Table 8.2: Benchmark results. Statistics over 100 executions. Where applicable, the key and message were randomized (e.g., the SEP signing is benchmarked over random keys and random messages). High variance timings are due to rejection sampling. Note that we omitted the benchmark result for the oblivious signing user signature completion, which takes on average 1.2 µs (or 1611 cycles).

As expected, there are notable variations in the timings due to rejection sampling, but also for procedures that do not involve rejection steps, which stems from the use of FLINT. Note, however, that we clear all FLINT-internal caches after each iteration of the benchmarked function. The SEP key generation is also subject to rejections and is quite computationally intensive due to the spectral norm estimation which represents 87.7% of the key generation time. As the organization key generation is not run often and because it is still reasonably efficient despite this rejection step, we do not optimize it further.

The most important steps for anonymous credentials are *issuance* and *credential showing* as they directly impact the user experience. Regarding issuance, the full protocol takes about 400 ms (on average) which we deem very reasonable. Credential showing is slightly slower as it takes about 500 ms (including verification) on average, which should be imperceptible in most cases.

We also recall that the point of our implementation was to provide a better understanding of the performance of privacy-preserving solutions, not to provide the most optimized code for a specific setting. In particular, we did not implement our own arithmetic backend tailored to our moduli, nor did we leverage the multiple cores of modern CPUs (our timings were obtained without any parallelization) or precomputations. In other words, there are many ways one could improve performance without changing the cryptographic protocol itself and, given the already appealing benchmarks as shown in Table 8.2, we are confident that our solution should be sufficiently practical for most use-cases.

8.5 Conclusion

Our implementation, albeit not fully optimized, shows promising persepectives for the design of efficient privacy-oriented applications in the post-quantum setting. In particular, by implementing the complex zero-knowledge framework from [LNP22], we contribute to improving the understanding of its efficiency and its implications for deployment in real-world systems.

As explained in Chapter 7, anonymous credentials are representative of a wide class of such privacy applications. As a result, the proposed implementation could easily be specialized to other more specific constructions, hoping for similar or better performance depending on the use case. We stress that our signature with efficient protocols and its implementation are very versatile and well documented so that they could be adapted without too much effort.

The results of Part III lean towards similar conclusions than those drawn from advanced classical authentication mechanisms. They indeed give similar security assurances to regular signatures, while allowing a more fine-grained control on what information is disclosed. Although our scheme

is an order of magnitude larger than standard lattice signatures, if comparing the size of a credential proof to the size of a regular signature, it is still rather efficient for the flexibility if offers. Our work thus fosters practical post-quantum privacy and makes a significant step towards it.

Part IV

Appendix - Concrete Security

This part contains the description of the methodology we used throughout this thesis in order to select concretely secure parameter sets.

9

Concrete Security Analysis

In this chapter, we recall the methodology we use to estimate the bit security of the different schemes we designed in this thesis. In particular, we give the necessary notions and tools to estimate the concrete hardness of the underlying assumptions.

Contents

9.1 Lattice Reduction and Heuristics

As explained in Part I, provably secure parameters taken from reduction results usually lead to non-optimal, and sometimes impractical, instantiations. Concretely efficient schemes then resort to estimating the complexity of the best attacks to evaluate their security and in turn choose parameter sets that meet the security target. Most lattice-based primitives can be attacked using lattice reduction algorithms, which offer trade-offs between the quality of the attack (usually linked to an approximation factor of the targetted lattice problem) and the time complexity.

As cryptographers aim at a time complexity of the best attack exponential in the security parameter λ, it is infeasible to simply run and check the latter attack. We thus rely on an expected behavior of said lattice reduction algorithms which generally postulate further assumptions or heuristics to establish a relevant trend of their complexity.

9.1.1 Heuristics for BKZ

Lattice reduction essentially aims at finding a basis of shortest vectors of a lattice $\mathcal{L}$ represented by the given input basis $\mathbf{B}$. The quality of a basis is formalized by its *root Hermite factor* δ_0.

Definition 9.1 (Root Hermite Factor)

Let d be a positive integer, and $\mathcal{L}$ a lattice of rank d. Given a basis $\mathbf{B}$ of $\mathcal{L}$, the root Hermite factor δ_0 of $\mathbf{B}$ is defined by the relation

$$\delta_0 = \frac{\min_{i\in[\![d]\!]}\|\mathbf{B}\mathbf{e}_i\|_2^{1/d}}{\mathrm{Vol}(\mathcal{L})^{1/d^2}},$$

where the numerator essentially captures the (d-th root of the) length of the shortest column of $\mathbf{B}$.

In this thesis, we consider the celebrated BKZ algorithm [SE94] which somewhat generalizes the LLL algorithm [LLL82]. The latter essentially reduces pairs of vectors while the former extends it to blocks of b vectors defining lattices of rank b. It then requires a solver for the exact SVP problem in dimension b. Since its introduction in 1994, the BKZ algorithm has been improved in a number of ways. The optimized version is sometimes referred to as BKZ 2.0 [CN11].

A popular heursitic on the output of BKZ is the specific shape of its Gram-Schmidt vectors which are assumed to have a geometric decay.

Heuristic 9.1 (Geometric Series Assumption)

Given a basis of a lattice of rank d, the basis $\mathbf{B}$ obtained after lattice reduction verifies that there exists a constant $\gamma \in (0,1)$ such that for all i, $\|\widetilde{\mathbf{b}_i}\|_2 / \|\widetilde{\mathbf{b}_{i+1}}\|_2 = \gamma$.

Another heuristic widely used in the cryptanalysis of lattice systems is called the *Gaussian heuristic* which approximates the length of a shortest non-zero vector of a lattice. The idea is that the number of points of the intersection $\mathcal{L} \cap S$ for a set S is roughly $\mathrm{Vol}(S)/\mathrm{Vol}(\mathcal{L})$. Taking $S = \mathcal{B}_2^o(\mathbf{0}, \lambda_1(\mathcal{L}))$, it holds that $|S \cap \mathcal{L}| = 1$ and

$$\mathrm{Vol}(S) = \frac{(\pi \lambda_1(\mathcal{L}))^d}{\Gamma(d/2+1)}.$$

Under the approximation $|S \cap \mathcal{L}| \approx \mathrm{Vol}(S)/\mathrm{Vol}(\mathcal{L})$ and the Stirling approximation of the Gamma function, we get the following.

Heuristic 9.2 (Gaussian Heuristic)

Given a lattice $\mathcal{L}$ of dimension d, it holds that

$$\lambda_1(\mathcal{L}) \approx \sqrt{\frac{d}{2\pi e}} \mathrm{Vol}(\mathcal{L})^{1/d}.$$

Under the Geometric Series Assumption and the Gaussian Heuristic, Chen [Che13] proposed an expression of the root Hermite factor of a basis reduced by BKZ with blocksize b. Our estimations use this formula to link δ_0 to b.

Heuristic 9.3 ([Che13])

Under Heuristics 9.1 and 9.2, it holds that the root Hermite factor of a BKZ_b-reduced basis is given by

$$\delta_0 \approx \left(\frac{\mathsf{b}}{2\pi e} (\pi \mathsf{b})^{1/\mathsf{b}} \right)^{1/2(\mathsf{b}-1)}$$

We later specify the block size b with a subscript by using the notation δ_b.

9.1.2 Cost Models

The time complexity of BKZ_b can be estimated by the running time of solving a single SVP instance in dimension b as well as the number of calls made by the BKZ algorithm to the SVP oracle. The Core-SVP model considers that the cost of the attack is given by the cost of a single call to the SVP oracle. We are now left with estimating the time complexity of the latter.

The time needed to solve SVP in dimension b highly depends on the method. There are typically two main family of algorithms for this task, namely sieving and enumeration. Many works [CN11, BDGL16, Laa15, CL21] have then proposed estimation of the complexity as a function of b. This led to different BKZ cost models [CN11, APS15, ADPS16].

The cost of lattice sieving in dimension b is estimated by $2^{\mu_1 \mathsf{b} + \mu_2}$ with $\mu_2 = o(\mathsf{b})$, and μ_1 a small constant [BDGL16, Laa15, CL21]. Typically, classical sieves achieve $\mu_1 = \log_2 \sqrt{3/2} \approx 0.2924$ [BDGL16] while quantum sieves achieve $\mu_1 \approx 0.2570$ [CL21]. On the other hand, the cost

of enumeration is estimated by $2^{\mu'_1 \mathsf{b} \log \mathsf{b} + \mu'_2 \mathsf{b} + \mu'_3}$ or $2^{\mu'_1 \mathsf{b}^2 + \mu'_2 \mathsf{b} + \mu'_3}$ classically, and the square root of that quantumly.

The constructions presented in this thesis assumed two different cost models of lattice sieving. As most digital signature schemes submitted to standardization efforts use the Core-SVP model based on a sieving SVP oracle with cost $2^{\mu_1 \mathsf{b}}$, i.e., with $\mu_2 = 0$ term, we adopt the latter for our estimation of Phoenix and $\mathsf{Phoenix}_{\mathsf{U}}$. We note however that this model is rather conservative and several works use $\mu_2 = 16.4$. It essentially accounts for the different calls to the SVP oracle as well as processing. We decide to adopt the latter for our signatures with efficient protocols and anonymous credentials of Part III, as we think it provides a more realistic security assessment. We still note that for our parameters, this extra term 16.4 remains much smaller than $\mu_1 \mathsf{b}$. Our schemes would then only be a few bits below the actual security target if considering the Core-SVP model. We summarized the models we use in Table 9.1.

Part	Construction	Section	Classical Cost ($\log_2$)	Quantum Cost ($\log_2$)
Part II	Base study Phoenix $\mathsf{Phoenix}_{\mathsf{U}}$	Sec. 4.5.2 Sec. 5.3 Sec. 5.4	$0.292\mathsf{b}$	$0.257\mathsf{b}$
Part III	SEP SEP* ZKP	Sec. 6.2 Sec. 6.4 Sec. 7.4	$0.292\mathsf{b} + 16.4$	$0.257\mathsf{b} + 16.4$

Table 9.1: $\mathrm{BKZ}_{\mathsf{b}}$ cost models used in this thesis.

9.2 Estimating the LWE Hardness

Several cryptanalytic works target the LWE problem, with sometimes increased efficiency when the parameters are small, e.g. particularly small secret, or particularly small error. They leverage either lattice reduction [LP11, LN13], combinatorial [Wag02, BKW03, KF15] or algebraic [AG11, ACF+15] techniques. Estimating the hardness of LWE thus comes down assessing the complexity of these attacks. Despite some recent works on the cryptanalysis of Ideal-SVP [EHKS14, BS16, CDPR16, CDW17, DPW19, PHS19, BR20, BLNR22], leveraging the structure of ideals and modules remains an interesting cryptanalytic challenge. At the moment, the concrete hardness of structured problems such as M-LWE, M-SIS or M-ISIS, is estimated by looking at their unstructured versions once embedded into the integers.

9.2.1 Attacks on LWE

We start by giving a brief overview of attacks on LWE. We refer to [APS15] for a better survey of the different attack strategies.

Solving CVP [LP11]. In Section 1.4.2, we introduce the LWE problem as an instance of CVP on the q-ary lattice $\mathcal{L}_q(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \exists \mathbf{s} \in \mathbb{Z}_q^d, \mathbf{x} = \mathbf{A}\mathbf{s} \bmod q\mathbb{Z}\}$ defined by the LWE matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times d}$. Given the instance $(\mathbf{A}, \mathbf{b})$, one can try to directly solve this CVP instance on the lattice $\mathcal{L}_q(\mathbf{A})$ by finding a vector $\mathbf{s} \in \mathbb{Z}_q^d$ such that $\|\mathbf{b} - \mathbf{A}\mathbf{s}\|_2$ is small. Note that if $\mathbf{b} = \mathbf{A}\mathbf{s}^\star + \mathbf{e}^\star \bmod q\mathbb{Z}$, the distribution of the error $\mathbf{e}^\star$ may give information on the expected distance between $\mathbf{b}$ and the lattice $\mathcal{L}_q(\mathbf{A})$. In this case, we usually call this variant *Bounded Distance Decoding* (BDD). This attack is then called *decoding attack*.

Primal Attack [BG14]. Although we introduced LWE as a special CVP instance, the LWE problem can be interpreted as a specific SVP instance instead. Let us consider the search problem LWE with secret and error distributions $\mathscr{D}_s = \mathscr{D}_e = U(\{0,1\})$. Let $(\mathbf{A}, \mathbf{b})$ with $\mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e} \bmod q\mathbb{Z}$ be an instance of the problem. The definition of $\mathbf{b}$ can be re-written as

$$[\mathbf{I}_m | \mathbf{A} | \mathbf{b}] \cdot \begin{bmatrix} \mathbf{e} \\ \mathbf{s} \\ -1 \end{bmatrix} = \mathbf{0} \bmod q\mathbb{Z}.$$

Defining $\mathbf{x} = [\mathbf{e}^T | \mathbf{s}^T | -1]^T$, it holds that $\mathbf{x}$ is a non-zero vector of norm at most $\sqrt{m + d + 1}$ in the lattice $\mathcal{L}_q^\perp([\mathbf{I}_m | \mathbf{A} | \mathbf{b}])$. It can be shown that $\mathbf{x}$ actually verifies $\|\mathbf{x}\|_2 = \lambda_1(\mathcal{L}_q^\perp([\mathbf{I}_m | \mathbf{A} | \mathbf{b}]))$ and, as such, solving SVP on this lattice allows one to recover $\mathbf{s}$ (and $\mathbf{e}$). This solving method of

LWE is called the *primal attack*, which consists in interpreting the LWE instance as an instance of *Unique*-SVP. The *Unique*-SVP problems corresponds to SVP where there is the extra assurance that there are only two non-zero vectors having the shortest norm.

Decision via Dual Attack. Another way to attack the (decision) LWE problem is to consider its dual problem: SIS. Indeed, given an instance $(\mathbf{A}, \mathbf{b})$, and a short non-zero vector $\mathbf{x}$ such that $\mathbf{x}^T\mathbf{A} = \mathbf{0} \bmod q\mathbb{Z}$, one can use it to solve LWE. If $\mathbf{b} = \mathbf{As} + \mathbf{e} \bmod q\mathbb{Z}$, then $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T\mathbf{e} \bmod q\mathbb{Z}$ which can be bounded by $\|\mathbf{x}\|_2\|\mathbf{e}\|_2$. On the contrary, if $\mathbf{b}$ is uniform, then $\mathbf{x}^T\mathbf{b}$ is also uniform. Thence, if $\mathbf{x}$ is sufficiently short, one can decide on the distribution of $\mathbf{b}$ based on the size of $\mathbf{x}^T\mathbf{b} \bmod q\mathbb{Z}$. As a result, it suffices to efficiently solve SIS on the matrix $\mathbf{A}^T \in \mathbb{Z}_q^{d\times m}$.

9.2.2 The Lattice Estimator

The three previous attacks represent only a subset of the different methods to solve LWE. In order to efficiently estimate the LWE hardness through all known attacks, Albrecht et al. [APS15] designed the so-called *lattice estimator*. By simply providing the different parameters of the LWE instance, it assesses the complexity of the different attacks. In particular, it gives the necessary BKZ block size b or reachable root Hermite factor δ_0 for the attacks relying on lattice reduction. This provides a simple way to estimate the time complexity using the cost models discussed in Section 9.1.2.

In this thesis, we estimate the hardness of multiple secret M-LWE$^k_{n,d,m,q,\mathscr{D}_s,\mathscr{D}_e}$ instances of the form $(\mathbf{A}, \mathbf{B} = \mathbf{AS}+\mathbf{E} \bmod qR)$, with $\mathbf{A} \in R_q^{m\times d}$, and $\mathbf{S} \hookleftarrow \mathscr{D}_s^{d\times k}$ and $\mathbf{E} \hookleftarrow \mathscr{D}_e^{m\times k}$. In particular, in our case $\mathscr{D}_s \in \{U(R_q), U(S_1), \mathcal{B}_1\}$ and $\mathscr{D}_e \in \{U(T_1), U(S_1), \mathcal{B}_1\}$. We thus run the lattice estimator to estimate the hardness of M-LWE$^k_{1,nd,nm,q,\mathscr{D}'_s,\mathscr{D}'_e}$ (i.e., corresponding to the unstructured problem in dimension nd), with $\mathscr{D}'_s \in \{U(\mathbb{Z}_q), U(\llbracket -1, 1 \rrbracket), \psi_1\}$ and $\mathscr{D}_e \in \{U(\{0,1\}), U(\llbracket -1, 1 \rrbracket), \psi_1\}$. Note that the multiple secrets variant is handled by accounting for a loss factor of k based on Lemma 2.4.

We recall that in the case of Phoenix and Phoenix$_\mathsf{U}$ in Chapter 5, the M-LWE instance given by the public key is compressed. This means we are only given the high-order bits $\mathbf{B}_H$ of $\mathbf{B}$. Since $\mathbf{B}_H$ contains less information on $(\mathbf{S}, \mathbf{E})$ than the uncompressed matrix $\mathbf{B}$, we simply evaluate the hardness of the uncompressed version.

9.2.3 A Thought on M-LWE with Small Error

In Chapter 3, we show that M-LWE with a small uniform error is hard when restricting the number of samples m. This restriction actually makes sense when studying algebraic ways to attack the problem. For example, the attack by Arora and Ge [AG11] specifically targets LWE with small errors. It does not depend on the underlying structure, and therefore also applies to the more general case of M-LWE. The idea is to see the (search) LWE problem as solving a noisy system of equations, and transforming it into a noiseless polynomial system (where the degree of the polynomials depend on the size of the LWE error). Then, using root finding algorithms for multivariate polynomials, one can solve the new system.

More precisely in the case of LWE with η-bounded error ($\mathbf{e} \in \{-\eta, ..., \eta\}^m$), the Arora and Ge attack [AG11] solves the problem in polynomial time if $m \approx \binom{d+2\eta+1}{2\eta+1} = \Omega(d^{2\eta+1})$, where d is the LWE dimension. For $\eta = 1$, the attack becomes exponential for $m = O(d)$. It has been refined in [ACF+15] to obtain subexponential attacks whenever $m = \Omega(d \log_2 \log_2 d)$ in the uncentered binary case ($\{0,1\}$). As the attack ignores the structure, one can embed the m M-LWE equations with d unknowns over R_q into nm equations with nd unknowns over $\mathbb{Z}_q$ and apply the same attack. However, we now obtain a polynomial attack only for $nm = \Omega((nd)^{2\eta+1})$ and therefore $m = \Omega(n^{2\eta}d^{2\eta+1})$. In practical schemes relying on M-LWE with small errors like our schemes of Part II and III or the recent PQC standards [BDK+18, DKL+18], the rank d is a small constant and n drives the security level. Additionally, we saw in Section 3.4 that $m = m' + d$ is enough to establish the hardness of M-LWE with small secret *and* error with m' samples. For common parameters where $m' = d$ or $d+1$, we thus have $m = m' + d = O(d) \ll n^{2\eta}d^{2\eta+1}$. This is why we think that the hardness of M-LWE with both small secret and error is yet to be determined. The gap between what we proved in Chapter 3 and the applicable range of attacks can still be reduced in either direction: either by finding new attacks that require fewer samples, or by improving theoretical hardness results to allow for more samples.

9.3 Estimating the SIS Hardness

The forgery of our signature schemes of Parts II and III or the soundness of the zero-knowledge arguments in Section 7.4 all rely on the M-SIS or M-ISIS assumptions. We thus present the methodology we use to evaluate their hardness. Our methodology only accounts for bounds in the Euclidean norm. In particular, ℓ_∞ norm bounds are discarded in our estimation, except sometimes for ensuring that trivial q-vectors are not solutions. Again, we also make the assumption that the module structure does not yield better attacks and we simply evaluate the hardness of the problems once embedded into the integers.

9.3.1 Solving SIS

To estimate the security of $\text{M-SIS}_{n,d,m,q,\beta}$, we find the cost of finding $\mathbf{v} \in \mathcal{L}_q^\perp(\mathbf{A})$ such that $\|\mathbf{v}\|_2 \leq \beta$ and $\|\mathbf{v}\|_\infty \leq \beta_\infty$, given $\mathbf{A} \sim U(R_q^{d\times m})$ or $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$ with $\mathbf{A}' \sim U(R_q^{d\times m-d})$ equivalently. We thus look at the unstructured problem $\text{M-SIS}_{1,nd,nm,q,\beta}$. For that, we first check that our parameter setting does not allow for q-vectors, that is vectors of ℓ_∞ norm larger than q. Then, a standard optimization consists in finding a solution in a lattice of smaller dimension $nd \leq m^* \leq nm$ and completing the solution with zeros. It then comes down to using BKZ in block size b such that

$$\beta \geq \min_{nd\leq m^*\leq nm} \delta_{\mathsf{b}}^{m^*} q^{nd/m^*}.$$

More precisely, for a fixed β, we find m^* that maximizes $\delta_{\mathsf{b}} = \beta^{1/m^*} q^{-nd/m^{*2}}$ and then use Heuristic 9.3 to determine the corresponding block size b. We can then use the block size in the appropriate cost model to estimate the time complexity of solving the M-SIS instance.

9.3.2 Solving ISIS

The estimation of M-ISIS follows the same idea but with additional restrictions. We generally consider matrices of the form $\mathbf{A} = [\mathbf{I}_d|\mathbf{A}']$. Hence finding an M-ISIS solution $\mathbf{x} = [\mathbf{x}_1|\mathbf{x}_2]$ for the instance $(\mathbf{A}, \mathbf{u})$ comes down to solving $\mathbf{x}_1 + \mathbf{A}'\mathbf{x}_2 = \mathbf{u} \bmod qR$. To achieve more compactness, many schemes consider a bound β on $\|\mathbf{x}\|_2$ that is larger than q. Although this would be a problem for M-SIS as it would yield trivial q-vectors as solutions, the hardness of M-ISIS is more complicated to determine.

In the case where $\beta < q$, both M-SIS and M-ISIS are essentially equivalent. One could therefore use the estimation strategy from Section 9.3.1. On the other extreme, if $\beta \gtrsim q\sqrt{nd/12}$, $\mathbf{x} = [\mathbf{u}|\mathbf{0}]$ is a solution with noticeable probability. This comes from the fact that if $\mathbf{u}$ is uniformly random in R_q^d, then its Euclidean norm (in centered representation) is close to $q\sqrt{nd/12}$. Even if β is slightly below this bound, randomization techniques may still find vectors whose norm are a bit smaller than $q\sqrt{nd/12}$. Up until recently [DEP23], the estimation of M-ISIS when β was smaller than $q\sqrt{nd/12}$ was carried by estimating the Approximate CVP attack using the nearest-colattice algorithm of Espitau and Kirchner [EK20]. Given the embedded instance $(\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{N\times D} \times \mathbb{Z}_q^N$, where $N = nd$ and $D = nm$, the algorithm can compute a solution within Euclidean norm β with BKZ of block size b such that

$$\beta \geq \min_{k^*\leq D-N} \delta_{\mathsf{b}}^{D-k^*} q^{N/(D-k^*)}.$$

Again, for a fixed β, we find k^* which maximizes $\delta_{\mathsf{b}} = \beta^{1/(D-k^*)} q^{-N/(D-k^*)^2}$ and use Heuristic 9.3 to find the corresponding b.

A recent work by Ducas et al. [DEP23] refined the landscape of attacks on M-ISIS when β is larger than q. They exploit the geometry of the bases outputted by BKZ for q-ary lattices. By combining lattice reduction as well as randomization techniques on the vectors outputted by lattice sieves, they show that one can have an efficient attack even when $\beta < q\sqrt{nd/12}$. The only M-ISIS instances we consider in this thesis that feature $\beta > q$ are those related to the forgery of Phoenix and $\text{Phoenix}_\mathsf{U}$. As discussed extensively in Chapter 5, we place stronger restrictions on the shape of the expected M-ISIS solutions so as to mitigate these attacks. In particular, we impose ℓ_∞ norm bounds which are generally not considered in lattice reduction algorithms. Also, our signature has a strong asymmetry which may also thwart randomization techniques.

Conclusion

We presented several contributions ranging from the study of fundamental assumptions underlying the security of lattice schemes, to the optimization of trapdoor preimage sampling for signature designs, ending in the design of advanced post-quantum authentication mechanisms. We conclude this manuscript by going back on these different results and giving some open research perspectives on these subjects.

Hardness of M-LWE with Small Secret

In Chapter 2, we examined the hardness of the Module Learning With Errors (M-LWE) problem when changing the secret distribution from large uniform secrets modulo q, to small uniform secrets bounded by η. We showed, through two different reductions, that the latter is not easier than the former if one is willing to adjust some of the parameters of the problem. This leads us to the main drawback of our result which is that the module rank must be increased from k to at least $d \geq k \log_{2\eta+1} q$ throughout the reduction. This increase stems from the use of regularity results, e.g., the leftover hash lemma, which seems necessary in our proof method. The hardness of the problem for lower ranks d thus remains open, in particular for very small values $d \in \{1, \ldots, 4\}$. The case of $d = 1$ corresponding to the *Ring*-LWE problem is of particular interest as there is no known results on its hardness for small uniform secrets, even though it is being used in practical applications.

Several versions of the leftover hash lemma exist, giving sometimes tighter constraints. Investigating new regularity results or new proof methods altogether could allow one to break the $\log q$ barrier that all these results suffer from. This would narrow the everlasting gap between theoretically proven variants and the heuristically assessed ones that are used in practice.

Hardness of M-LWE with Small Error

Chapter 3 proposes one such proof method which gets slightly under $k \log_{2\eta+1} q$ but with other parameter constraints. More precisely, we showed that M-LWE with large secret modulo q remains hard if this time we change the error distribution from Gaussian to a uniform distribution over the cube of half-side η. This comes at the cost of limiting the number of M-LWE samples m depending on the rank d and the error bound η. We highlight that η is exponential in m, meaning that we cannot reach small values of η for a large m.

Nevertheless, we show that for constant values of d and reasonable bounds η (compared to q), we can use the above result to obtain the hardness of M-LWE with uniform η-bounded secret *and* error. Albeit reasonable compared to q, the reachable η are much larger than what is used in practice. Finding new reductions that somehow lighten the constraints on the parameters would be of high importance as it would get closer to theoretically proving the hardness of the variants of M-LWE that are used in countless cryptographic primitives.

More generally, the landscape of lattice-based assumptions has been vastly extended to produce even more efficient schemes. Through these variants, we further dive into uncharted territory by proposing parameters which are no longer proven secure but whose security guarantees are argued based on the best attacks and cryptanalytic efforts. Certain constructions even resorted to brand new variants [DKL+18, AKSY22, BLNS23b]. Making sure the theoretical assessment catch up to these practical considerations is paramount in order to trust the long-term guarantees of lattice cryptographic schemes.

Optimization of Gadget Samplers

We then focused on the tools that are prominent in the design of lattice-based signatures: trapdoor preimage samplers. In Chapter 4, we propose to revisit the gadget-based sampler first proposed in [MP12] with several contributions. We first give detailed (worst-case) security analysis of an elliptic version of the sampler, which can then be used as a drop-in replacement in all mechanisms using the sampler from [MP12]. Along those lines, we provide an improved analysis of the sampler of [LW15] (based on the trapdoors from [MP12]) by identifying that inverting uniform syndromes can help allay the main shortcoming of the original proposal. Building upon this average-case simulatability, we show how to employ the techniques of [CGM19] to obtain a very compact and versatile sampler which we call *approximate rejection sampler*.

From the different results and proof techniques present in the literature, one could conclude that there is a clear distinction between these worst-case and average-case analysis. We however think that studying the actual security gap between these two regimes is of particular interest. Indeed, several mechanisms cannot make the assumption that the syndromes to invert are uniform, which in turn require a worst-case sampler. To our knowledge, it is yet not clear if these constructions would become insecure if using an average-case sampler instead. A short-term perspective to better grasp the differences between worst-case and average-case in this context could be to provide a worst-case analysis of approximate gadget-based samplers such as the ones from [CGM19, GL20, YJW23] or ours.

Hash-and-Sign with Aborts: Phoenix

In Chapter 5, we demonstrate the benefits of our approximate rejection sampler by designing a family of signature schemes called Phoenix. In particular, Phoenix refers to the signature instantiated with discrete Gaussian distributions, while $\text{Phoenix}_{\mathsf{U}}$ uses uniform distributions over hypercubes. This family sort of bridges the Hash-and-Sign paradigm instantiated over lattices in [GPV08] with the Fiat-Shamir with Aborts paradigm proposed in [Lyu12]. It turns out to be rather competitive with previous lattice signatures, while enjoying some of the most interesting features of each paradigm, e.g., plethora of signature distributions, tight security proofs in the (quantum) random oracle model.

An interesting research direction would be to find specific distributions Phoenix could be instantiated with. As ease of implementation and side-channel protection are now of utmost importance in the transition and deployement of post-quantum cryptography, finding distributions for the approximate rejection sampler that would preserve compactness and whose implementation could be efficiently protected would be a positive differentiator for Phoenix.

Beyond the sole digital signature use-case, we hope that our work will incite the investigation of other practical applications of the Lyubashevsky-Wichs sampler [LW15] that could benefit from its unique characteristics.

Signature with Efficient Protocols from Lattices

We finally focused on the design of advanced authentication mechanisms. In Chapter 6, we constructed so-called *signatures with efficient protocols* (SEP) from lattices in a much more efficient way than the only existing construction [LLM+16] at this time. By leveraging the previous contributions, we show that we can obtain very compact signatures while preserving the versatility of this type of constructions. More precisely, SEPs are designed so as to smoothly plug into more advanced constructions such as blind signatures, group signatures, anonymous credentials, e-cash, EPID, DAA, etc. Our work thus provide a compact building block to be used in all these primitives in a rather straightforward way.

Improving the security and efficiency of SEPs will thus naturally improve that of the overall protocols and advanced primitives. For example, following our approach, it would be interesting to have more concrete elements to assess the necessity of a (partial) trapdoor slot to carry out the security proof. Removing the latter slot would improve our construction. Along those lines, finding a way to leverage average-case samplers such as the ones covered in Chapter 4 together with tags would likely improve our current signature. Finally, although our goal was to provide a better efficiency while relying on well-studied lattice assumptions such as M-LWE, a promising research direction is to identify new security assumptions to hopefully improve the compactness of such schemes. Albeit aimed at specific constructions like blind signatures or anonymous credentials, this has recently been done for example by Agrawal et al. [AKSY22] (one-more-ISIS) or Bootle et al. [BLNS23b] (NTRU-ISIS_f). Pursuing this effort is very important to consider the standardization and deployment of these advanced authentication schemes.

Practical Post-Quantum Privacy through Anonymous Credentials

Chapter 7 demonstrates the extent of our work on SEPs by devising an anonymous credentials system. Our construction retains all the expected security requirements while also providing the ability for selective disclosure of attributes, proving statements on the attributes, etc. It also outperforms almost all the existing post-quantum anonymous credentials [BLNS23b, LLLW23, BCR+23] that all appeared in the time period of this thesis. We indeed reach much more compact sizes than [LLLW23, BCR+23] and are competitive with [BLNS23b] even though we rely on well-studied assumptions as opposed to them. Beyond the consideration of sizes, we also give an implementation of our scheme which outperforms that of [BCR+23]. It is in particular the first implementation of (an instantiation of) the zero-knowledge proof framework of [LNP22]. The timings are extremely promising for concrete use cases of anonymous digital identity, bringing us one step closer to the concrete adoption of post-quantum privacy.

From a theoretical perspective, improving anonymous credentials mainly stems from improving the SEP scheme itself (see above), or the associated zero-knowledge proof systems. As the framework of [LNP22] is already optimized towards short proof sizes, it seems complicated to hope for much smaller proofs using their methods. Instead, we would need to find new and more compact ways of proving lattice relations in zero-knowledge, which would dramatically impact the performance of the resulting anonymous credentials. Albeit highly relevant, this remains a complex task if one wants to retain the strong properties of such proof systems as it sometimes precludes the use of zk-SNARKs which could be more compact already. From a practical perspective, on the other hand, many standard optimizations (e.g., parallelization) or more elaborate ones (e.g., parameter-specific backend) could be brought to our implementation in order to reduce the timings.

Lattice-based cryptography has proven very successful in the design of quantum resistant cryptographic algorithms. It established itself as one of the most promising alternatives to classical cryptography in many aspects and even provided us with new advanced functionalitites, e.g., FHE. Several facets unfortunately do not translate well to the lattice setting. In this thesis, we targeted the realm of *privacy-enhanced* primitives. The alarming scarcity of post-quantum designs, which in addition were either inefficient or targeting very specific use cases, was in sharp contrast with the rich lanscape of classical constructions. Alongside very recent efforts, e.g., [AKSY22, LNP22, dPK22, BLNS23b, BLNS23a], we contributed towards filling this gap by providing *efficient*, *versatile*, *privacy-preserving* primitives based on *solid security foundations*. Pursuing these efforts by developing the set of available techniques (e.g., preimage sampling, zero-knowledge proofs, randomization), functionalities (e.g. stronger privacy, revocation) and assumptions (e.g., one-more-ISIS, NTRU-ISIS$_f$) is paramount in the quest of designing, standardizing and deploying practical post-quantum cryptography for privacy.

Bibliography

[AA16] J. Alperin-Sheriff and D. Apon. Dimension-Preserving Reductions from LWE to LWR. In *IACR Cryptol. ePrint Arch.*, page 589, 2016.

[ABB+20] E. Alkim, P. S. L. M. Barreto, N. Bindel, J. Krämer, P. Longa and J. E. Ricardini. The Lattice-Based Digital Signature Scheme qTESLA. In *ACNS*, 2020. doi:10.1007/978-3-030-57808-4_22.

[ACF+15] M. R. Albrecht, C. Cid, J. Faugère, R. Fitzpatrick and L. Perret. Algebraic Algorithms for LWE Problems. In *ACM Commun. Comput. Algebra*, 2015. doi:10.1145/2815111.2815158.

[ACPS09] B. Applebaum, D. Cash, C. Peikert and A. Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In *CRYPTO*, 2009. doi:10.1007/978-3-642-03356-8_35.

[AD17] M. R. Albrecht and A. Deo. Large Modulus Ring-LWE $\geq$ Module-LWE. In *ASIACRYPT*, 2017. doi:10.1007/978-3-319-70694-8_10.

[ADPS16] E. Alkim, L. Ducas, T. Pöppelmann and P. Schwabe. Post-quantum Key Exchange - A New Hope. In *USENIX Security Symposium*, 2016.

[AG11] S. Arora and R. Ge. New Algorithms for Learning in Presence of Errors. In *ICALP*, 2011. doi:10.1007/978-3-642-22006-7_34.

[AGJ+24] S. Argo, T. Güneysu, C. Jeudy, G. Land, A. Roux-Langlois and O. Sanders. Practical Post-Quantum Signatures for Privacy. In *CCS*, 2024.

[Ajt96] M. Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *STOC*, 1996. doi:10.1145/237814.237838.

[AKSY22] S. Agrawal, E. Kirshanova, D. Stehlé and A. Yadav. Practical, Round-Optimal Lattice-Based Blind Signatures. In *CCS*, 2022. doi:10.1145/3548606.3560650.

[ALS20] T. Attema, V. Lyubashevsky and G. Seiler. Practical Product Proofs for Lattice Commitments. In *CRYPTO*, 2020. doi:10.1007/978-3-030-56880-1_17.

[AP09] J. Alwen and C. Peikert. Generating Shorter Bases for Hard Random Lattices. In *STACS*, 2009.

[APS15] M. R. Albrecht, R. Player and S. Scott. On the Concrete Hardness of Learning With Errors. In *J. Math. Cryptol.*, 2015.

[ASM06] M. H. Au, W. Susilo and Y. Mu. Constant-Size Dynamic k-TAA. In *SCN*, 2006. doi:10.1007/11832072_8.

[Bab85] L. Babai. On Lovász' Lattice Reduction and the Nearest Lattice Point Problem. In *STACS*, 1985. doi:10.1007/BFB0023990.

[Bab86] L. Babai. On Lovász' Lattice Reduction and the Nearest Lattice Point Problem. In *Combinatorica*, 1986. doi:10.1007/BF02579403.

[Ban93] W. Banaszczyk. New Bounds in Some Transference Theorems in the Geometry of Numbers. In *Math. Ann.*, 1993.

[BB08] D. Boneh and X. Boyen. Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups. In *J. Cryptol.*, 2008. doi:10.1007/s00145-007-9005-7.

[BBC+18] C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth and V. Lyubashevsky. Sub-linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits. In *CRYPTO*, 2018. doi:10.1007/978-3-319-96881-0_23.

[BCC04] E. F. Brickell, J. Camenisch and L. Chen. Direct Anonymous Attestation. In *CCS*, 2004. doi:10.1145/1030083.1030103.

[BCR+23] O. Blazy, C. Chevalier, G. Renaut, T. Ricosset, E. Sageloli and H. Senet. Efficient Implementation of a Post-Quantum Anonymous Credential Protocol. In *ARES*, 2023. doi:10.1145/3600160.3600188.

[BD20] Z. Brakerski and N. Döttling. Hardness of LWE on General Entropic Distributions. In *EUROCRYPT*, 2020. doi:10.1007/978-3-030-45724-2_19.

[BDF+11] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner and M. Zhandry. Random Oracles in a Quantum World. In *ASIACRYPT*, 2011. doi:10.1007/978-3-642-25385-0_3.

[BDGL16] A. Becker, L. Ducas, N. Gama and T. Laarhoven. New Directions in Nearest Neighbor Searching with Applications to Lattice Sieving. In *SODA*, 2016. doi:10.1137/1.9781611974331.ch2.

[BDK+18] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler and D. Stehlé. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM. In *EuroS&P*, 2018. doi:10.1109/EuroSP.2018.00032.

[BDK24] J. Bobolz, J. Diaz and M. Kohlweiss. Foundations of Anonymous Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *Financial Cryptography*, 2024.

[BEF19] D. Boneh, S. Eskandarian and B. Fisch. Post-quantum EPID Signatures from Symmetric Primitives. In *CT-RSA*, 2019. doi:10.1007/978-3-030-12612-4_13.

[BEK+21] J. Bobolz, F. Eidens, S. Krenn, S. Ramacher and K. Samelin. Issuer-Hiding Attribute-Based Credentials. In *CANS*, 2021. doi:10.1007/978-3-030-92548-2_9.

[BEP+21] P. Bert, G. Eberhart, L. Prabel, A. Roux-Langlois and M. Sabt. Implementation of Lattice Trapdoors on Modules and Applications. In *PQCrypto*, 2021. doi:10.1007/978-3-030-81293-5_11.

[BFGP22] D. Bosk, D. Frey, M. Gestin and G. Piolle. Hidden Issuer Anonymous Credential. In *Proc. Priv. Enhancing Technol.*, 2022(4):571–607, 2022. doi:10.56553/POPETS-2022-0123.

[BFRS18] P. Bert, P. Fouque, A. Roux-Langlois and M. Sabt. Practical Implementation of Ring-SIS/LWE Based Signature and IBE. In *PQCrypto*, 2018. doi:10.1007/978-3-319-79063-3_13.

[BG14] S. Bai and S. D. Galbraith. Lattice Decoding Attacks on Binary LWE. In *ACISP*, 2014. doi:10.1007/978-3-319-08344-5_21.

[BGG+22] F. Boudot, P. Gaudry, A. Guillevic, N. Heninger, E. Thomé and P. Zimmermann. The State of the Art in Integer Factoring and Breaking Public-Key Cryptography. In *IEEE Secur. Priv.*, 20(2):80–86, 2022. doi:10.1109/MSEC.2022.3141918.

[BGV12] Z. Brakerski, C. Gentry and V. Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping. In *ITCS*, 2012. doi:10.1145/2090236.2090262.

[BJRW20] K. Boudgoust, C. Jeudy, A. Roux-Langlois and W. Wen. Towards Classical Hardness of Module-LWE: The Linear Rank Case. In *ASIACRYPT*, 2020. doi:10.1007/978-3-030-64834-3_10.

[BJRW21] K. Boudgoust, C. Jeudy, A. Roux-Langlois and W. Wen. On the Hardness of Module-LWE with Binary Secret. In *CT-RSA*, 2021. doi:10.1007/978-3-030-75539-3_21.

[BJRW22] K. Boudgoust, C. Jeudy, A. Roux-Langlois and W. Wen. Entropic Hardness of Module-LWE from Module-NTRU. In *INDOCRYPT*, 2022. doi:10.1007/978-3-031-22912-1_4.

[BJRW23] K. Boudgoust, C. Jeudy, A. Roux-Langlois and W. Wen. On the Hardness of Module Learning with Errors with Short Distributions. In *J. Cryptol.*, 2023. doi:10.1007/s00145-022-09441-3.

[BKW03] A. Blum, A. Kalai and H. Wasserman. Noise-tolerant Learning, the Parity Problem, and the Statistical Query Model. In *J. ACM*, 2003. doi:10.1145/792538.792543.

[BL07] E. Brickell and J. Li. Enhanced Privacy ID: a Direct Anonymous Attestation Scheme with Enhanced Revocation Capabilities. In *WPES*, 2007. doi:10.1145/1314333.1314337.

[Bla22] I. Blanco-Chacón. On the RLWE/PLWE equivalence for cyclotomic number fields. In *Appl. Algebra Eng. Commun. Comput.*, 2022. doi:10.1007/S00200-020-00433-Z.

[BLL+15] S. Bai, A. Langlois, T. Lepoint, D. Stehlé and R. Steinfeld. Improved Security Proofs in Lattice-Based Cryptography: Using the Rényi Divergence Rather Than the Statistical Distance. In *ASIACRYPT*, 2015. doi:10.1007/978-3-662-48797-6_1.

[BLNR22] O. Bernard, A. Lesavourey, T.-H. Nguyen and A. Roux-Langlois. Log-S-unit Lattices Using Explicit Stickelberger Generators to Solve Approx Ideal-SVP. In *ASIACRYPT*, 2022. doi:10.1007/978-3-031-22969-5_23.

[BLNS23a] W. Beullens, V. Lyubashevsky, N. K. Nguyen and G. Seiler. Lattice-Based Blind Signatures: Short, Efficient, and Round-Optimal. In *CCS*, 2023. doi:10.1145/3576915.3616613.

[BLNS23b] J. Bootle, V. Lyubashevsky, N. K. Nguyen and A. Sorniotti. A Framework for Practical Anonymous Credentials from Lattices. In *CRYPTO*, 2023. doi:10.1007/978-3-031-38545-2_13.

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev and D. Stehlé. Classical Hardness of Learning With Errors. In *STOC*, 2013. doi:10.1145/2488608.2488680.

[BLR+18] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé and R. Steinfeld. Improved Security Proofs in Lattice-Based Cryptography: Using the Rényi Divergence Rather than the Statistical Distance. In *J. Cryptol.*, 2018. doi:10.1007/s00145-017-9265-9.

[BLS19] J. Bootle, V. Lyubashevsky and G. Seiler. Algebraic Techniques for Short(er) Exact Lattice-Based Zero-Knowledge Proofs. In *CRYPTO*, 2019. doi:10.1007/978-3-030-26948-7_7.

[Bou21] K. Boudgoust. *Theoretical Hardness of Algebraically Structured Learning With Errors.* Ph.D. thesis, Université de Rennes 1, Rennes, France, 2021.

[Boy10] X. Boyen. Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More. In *PKC*, 2010. doi:10.1007/978-3-642-13013-7_29.

[BPR12] A. Banerjee, C. Peikert and A. Rosen. Pseudorandom Functions and Lattices. In *EUROCRYPT*, 2012. doi:10.1007/978-3-642-29011-4_42.

[BPS19] F. Bourse, D. Pointcheval and O. Sanders. Divisible E-Cash from Constrained Pseudo-Random Functions. In *ASIACRYPT*, 2019. doi:10.1007/978-3-030-34578-5_24.

[BR20] O. Bernard and A. Roux-Langlois. Twisted-PHS: Using the Product Formula to Solve Approx-SVP in Ideal Lattices. In *ASIACRYPT*, 2020. doi:10.1007/978-3-030-64834-3_12.

[Bra00] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. MIT Press, 2000. ISBN 9780262526302.

[BS16] J.-F. Biasse and F. Song. Efficient Quantum Algorithms for Computing Class Groups and Solving the Principal Ideal Problem in Arbitrary Degree Number Fields. In *SODA*, 2016. doi:10.1137/1.9781611974331.CH64.

[BSZ05] M. Bellare, H. Shi and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In *CT-RSA*, 2005. doi:10.1007/978-3-540-30574-3_11.

[BV14] Z. Brakerski and V. Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In *SIAM J. Comput.*, 2014. doi:10.1137/120868669.

[CCD+23] J. H. Cheon, H. Choe, J. Devevey, T. Güneysu, D. Hong, M. Krausz, G. Land, M. Möller, D. Stehlé and M. Yi. HAETAE: Shorter Lattice-Based Fiat-Shamir Signatures. In *IACR Cryptol. ePrint Arch.*, page 624, 2023.

[CDHK15] J. Camenisch, M. Dubovitskaya, K. Haralambiev and M. Kohlweiss. Composable and Modular Anonymous Credentials: Definitions and Practical Constructions. In *ASIACRYPT*, 2015. doi:10.1007/978-3-662-48800-3_11.

[CDL16] J. Camenisch, M. Drijvers and A. Lehmann. Anonymous Attestation Using the Strong Diffie Hellman Assumption Revisited. In *TRUST*, 2016. doi:10.1007/978-3-319-45572-3_1.

[CDPR16] R. Cramer, L. Ducas, C. Peikert and O. Regev. Recovering Short Generators of Principal Ideals in Cyclotomic Rings. In *EUROCRYPT*, 2016. doi:10.1007/978-3-662-49896-5_20.

[CDW17] R. Cramer, L. Ducas and B. Wesolowski. Short Stickelberger Class Relations and Application to Ideal-SVP. In *EUROCRYPT*, 2017. doi:10.1007/978-3-319-56620-7_12.

[CGM19] Y. Chen, N. Genise and P. Mukherjee. Approximate Trapdoors for Lattices and Smaller Hash-and-Sign Signatures. In *ASIACRYPT*, 2019. doi:10.1007/978-3-030-34618-8_1.

[Cha82] D. Chaum. Blind Signatures for Untraceable Payments. In *CRYPTO*, 1982. doi:10.1007/978-1-4757-0602-4_18.

[Cha85] D. Chaum. Showing Credentials Without Identification: Signatures Transferred Between Unconditionally Unlinkable Pseudonyms. In *EUROCRYPT*, 1985. doi:10.1007/3-540-39805-8_28.

[Che13] Y. Chen. *Réduction de Réseau et Sécurité Concrète du Chiffrement Complètement Homomorphe*. Ph.D. thesis, Paris 7, 2013.

[CKLL19] L. Chen, N. E. Kassem, A. Lehmann and V. Lyubashevsky. A Framework for Efficient Lattice-Based DAA. In *CYSARM@CCS*, 2019. doi:10.1145/3338511.3357349.

[CL01] J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *EUROCRYPT*, 2001. doi:10.1007/3-540-44987-6_7.

[CL02] J. Camenisch and A. Lysyanskaya. A Signature Scheme with Efficient Protocols. In *SCN*, 2002. doi:10.1007/3-540-36413-7_20.

[CL04] J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *CRYPTO*, 2004. doi:10.1007/978-3-540-28628-8_4.

[CL21] A. Chailloux and J. Loyer. Lattice Sieving via Quantum Random Walks. In *ASIACRYPT*, 2021. doi:10.1007/978-3-030-92068-5_3.

[CLP22] A. Connolly, P. Lafourcade and O. Perez-Kempner. Improved Constructions of Anonymous Credentials from Structure-Preserving Signatures on Equivalence Classes. In *PKC*, 2022. doi:10.1007/978-3-030-97121-2_15.

[CN11] Y. Chen and P. Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In *ASIACRYPT*, 2011. doi:10.1007/978-3-642-25385-0_1.

[CvH91] D. Chaum and E. van Heyst. Group Signatures. In *EUROCRYPT*, 1991. doi:10.1007/3-540-46416-6_22.

[DDLL13] L. Ducas, A. Durmus, T. Lepoint and V. Lyubashevsky. Lattice Signatures and Bimodal Gaussians. In *CRYPTO*, 2013. doi:10.1007/978-3-642-40041-4_3.

[DEP23] L. Ducas, T. Espitau and E. W. Postlethwaite. Finding Short Integer Solutions When the Modulus Is Small. In *CRYPTO*, 2023. doi:10.1007/978-3-031-38548-3_6.

[DFPS22] J. Devevey, O. Fawzi, A. Passelègue and D. Stehlé. On Rejection Sampling in Lyubashevsky's Signature Scheme. In *ASIACRYPT*, 2022. doi:10.1007/978-3-031-22972-5_2.

[DH76] W. Diffie and M. E. Hellman. New Directions in Cryptography. In *IEEE Trans. Inf. Theory*, 1976. doi:10.1109/TIT.1976.1055638.

[DKL+18] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler and D. Stehlé. CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme. In *TCHES*, 2018. doi:10.13154/tches.v2018.i1.238-268.

[DLP14] L. Ducas, V. Lyubashevsky and T. Prest. Efficient Identity-Based Encryption over NTRU Lattices. In *ASIACRYPT*, 2014. doi:10.1007/978-3-662-45608-8_2.

[DM14] L. Ducas and D. Micciancio. Improved Short Lattice Signatures in the Standard Model. In *CRYPTO*, 2014. doi:10.1007/978-3-662-44371-2_19.

[DM15] L. Ducas and D. Micciancio. FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second. In *EUROCRYPT*, 2015. doi:10.1007/978-3-662-46800-5_24.

[DN12] L. Ducas and P. Q. Nguyen. Learning a Zonotope and More: Cryptanalysis of NTRUSign Countermeasures. In *ASIACRYPT*, 2012. doi:10.1007/978-3-642-34961-4_27.

[DORS08] Y. Dodis, R. Ostrovsky, L. Reyzin and A. D. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In *SIAM J. Comput.*, 2008. doi:10.1137/060651380.

[DP16] L. Ducas and T. Prest. Fast Fourier Orthogonalization. In *ISSAC*, 2016. doi:10.1145/2930889.2930923.

[dPEK+] R. del Pino, T. Espitau, S. Katsumata, M. Maller, F. Mouhartem, T. Prest, M. Rossi and M.-J. Saarinen. *Raccoon: A Side-Channel Secure Signature Scheme*. URL `https://github.com/masksign/raccoon/blob/main/doc/raccoon.pdf`.

[dPK22] R. del Pino and S. Katsumata. A New Framework for More Efficient Round-Optimal Lattice-Based (Partially) Blind Signature via Trapdoor Sampling. In *CRYPTO*, 2022. doi:10.1007/978-3-031-15979-4_11.

[dPLS18] R. del Pino, V. Lyubashevsky and G. Seiler. Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability. In *CCS*, 2018. doi:10.1145/3243734.3243852.

[DPS23] J. Devevey, A. Passelègue and D. Stehlé. G+G: A Fiat-Shamir Lattice Signature Based on Convolved Gaussians. In *ASIACRYPT*, 2023. doi:10.1007/978-981-99-8739-9_2.

[DPW19] L. Ducas, M. Plançon and B. Wesolowski. On the Shortness of Vectors to be Found by the Ideal-SVP Quantum Algorithm. In *CRYPTO*, 2019. doi:10.1007/978-3-030-26948-7_12.

[DSH21] A. L. Dévéhat, H. Shizuya and S. Hasegawa. On the Higher-Bit Version of Approximate Inhomogeneous Short Integer Solution Problem. In *CANS*, 2021. doi:10.1007/978-3-030-92548-2_14.

[EFG+22] T. Espitau, P. Fouque, F. Gérard, M. Rossi, A. Takahashi, M. Tibouchi, A. Wallet and Y. Yu. Mitaka: A Simpler, Parallelizable, Maskable Variant of Falcon. In *EUROCRYPT*, 2022. doi:10.1007/978-3-031-07082-2_9.

[EHKS14] K. Eisenträger, S. Hallgren, A. Y. Kitaev and F. Song. A Quantum Algorithm for Computing the Unit Group of an Arbitrary Degree Number Field. In *STOC*, 2014. doi:10.1145/2591796.2591860.

[EK20] T. Espitau and P. Kirchner. The Nearest-Colattice Algorithm: Time-Approxmation Tradeoff for Approx-CVP. In *ANTS XIV*, 2020.

[ENS20] M. F. Esgin, N. K. Nguyen and G. Seiler. Practical Exact Proofs from Lattices: New Techniques to Exploit Fully-Splitting Rings. In *ASIACRYPT*, 2020. doi:10.1007/978-3-030-64834-3_9.

[ETWY22] T. Espitau, M. Tibouchi, A. Wallet and Y. Yu. Shorter Hash-and-Sign Lattice-Based Signatures. In *CRYPTO*, 2022. doi:10.1007/978-3-031-15979-4_9.

[FHS19] G. Fuchsbauer, C. Hanser and D. Slamanig. Structure-Preserving Signatures on Equivalence Classes and Constant-Size Anonymous Credentials. In *J. Cryptol.*, 2019. doi:10.1007/s00145-018-9281-4.

[FHT03] A. A. Fedotov, P. Harremoës and F. Topsøe. Refinements of Pinsker's inequality. In *IEEE Trans. Inf. Theory*, 49(6):1491–1498, 2003. doi:10.1109/TIT.2003.811927.

[FLI23] FLINT Team. FLINT: Fast Library for Number Theory, 2023. Version 3.0.0, https://flintlib.org.

[Gen09] C. Gentry. Fully Homomorphic Encryption using Ideal Lattices. In *STOC*, 2009. doi:10.1145/1536414.1536440.

[GGH97] O. Goldreich, S. Goldwasser and S. Halevi. Public-Key Cryptosystems from Lattice Reduction Problems. In *CRYPTO*, 1997. doi:10.1007/BFB0052231.

[GHL22] C. Gentry, S. Halevi and V. Lyubashevsky. Practical Non-interactive Publicly Verifiable Secret Sharing with Thousands of Parties. In *EUROCRYPT*, 2022. doi:10.1007/978-3-031-06944-4_16.

[GKPV10] S. Goldwasser, Y. T. Kalai, C. Peikert and V. Vaikuntanathan. Robustness of the Learning with Errors Assumption. In *ICS*, 2010.

[GL20] N. Genise and B. Li. Gadget-Based iNTRU Lattice Trapdoors. In *INDOCRYPT*, 2020. doi:10.1007/978-3-030-65277-7_27.

[GM18] N. Genise and D. Micciancio. Faster Gaussian Sampling for Trapdoor Lattices with Arbitrary Modulus. In *EUROCRYPT*, 2018. doi:10.1007/978-3-319-78381-9_7.

[GMPW20] N. Genise, D. Micciancio, C. Peikert and M. Walter. Improved Discrete Gaussian and Subgaussian Analysis for Lattice Cryptography. In *PKC*, 2020. doi:10.1007/978-3-030-45374-9_21.

[GMR85] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). In *STOC*, 1985. doi:10.1145/22145.22178.

[GPV08] C. Gentry, C. Peikert and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, 2008. doi:10.1145/1374376.1374407.

[HHP+03] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman and W. Whyte. NTRUSIGN: Digital Signatures Using the NTRU Lattice. In *CT-RSA*, 2003. doi:10.1007/3-540-36563-X_9.

[HILL99] J. Håstad, R. Impagliazzo, L. A. Levin and M. Luby. A Pseudorandom Generator from any One-way Function. In *SIAM J. Comput.*, 1999. doi:10.1137/S0097539793244708.

[Int16] Intel. A Cost-Effective Foundation for End-to-End IoT Security, White Paper. https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/intel-epid-iot-security-white-paper.pdf, 2016.

[ISO13a] ISO/IEC. ISO/IEC 18370-2:2016 Information Technology — Security Techniques — Blind Digital Signatures — Part 2: Discrete Logarithm Based Mechanisms. https://www.iso.org/standard/62544.html, 2013.

[ISO13b] ISO/IEC. ISO/IEC 20008-2:2013 Information Technology — Security Techniques — Anonymous Digital Signatures — Part 2: Mechanisms using a Group Public Key. https://www.iso.org/standard/56916.html, 2013.

[JHT22] H. Jia, Y. Hu and C. Tang. Lattice-Based Hash-and-Sign Signatures using Approximate Trapdoor, Revisited. In *IET Inf. Secur.*, 2022. doi:10.1049/ise2.12039.

[JMW23] K. Jackson, C. Miller and D. Wang. Evaluating the Security of CRYSTALS-Dilithium in the Quantum Random Oracle Model. In *IACR Cryptol. ePrint Arch.*, page 1968, 2023.

[JR23] C. Jeudy and A. Roux-Langlois. Cryptographie Reposant sur les Réseaux Euclidiens. In *Techniques de l'Ingénieur Sécurité des Systèmes d'Information*, (Ref. Article : h5216), 2023. doi:10.51257/a-v1-h5216.

[JRS23] C. Jeudy, A. Roux-Langlois and O. Sanders. Lattice Signature with Efficient Protocols, Application to Anonymous Credentials. In *CRYPTO*, 2023. doi:10.1007/978-3-031-38545-2_12.

[JRS24] C. Jeudy, A. Roux-Langlois and O. Sanders. Phoenix: Hash-and-Sign with Aborts from Lattice Gadgets. In *PQCrypto*, 2024. doi:10.1007/978-3-031-62743-9_9.

[KF15] P. Kirchner and P. Fouque. An Improved BKW Algorithm for LWE with Applications to Cryptography and Lattices. In *CRYPTO*, 2015. doi:10.1007/978-3-662-47989-6_3.

[KL14] J. Katz and Y. Lindell. *Introduction to Modern Cryptography, Second Edition*. CRC Press, 2014.

[Kle00] P. N. Klein. Finding the Closest Lattice Vector when it's Unusually Close. In *SODA*, 2000.

[KLS18] E. Kiltz, V. Lyubashevsky and C. Schaffner. A Concrete Treatment of Fiat-Shamir Signatures in the Quantum Random-Oracle Model. In *EUROCRYPT*, 2018. doi:10.1007/978-3-319-78372-7_18.

[KLSS23] D. Kim, D. Lee, J. Seo and Y. Song. Toward Practical Lattice-Based Proof of Knowledge from Hint-MLWE. In *CRYPTO*, 2023. doi:10.1007/978-3-031-38554-4_18.

[Knu98] D. E. Knuth. *The art of computer programming, Volume II: Seminumerical Algorithms, 3rd Edition*. Addison-Wesley, 1998.

[KTW+22] K. Kim, M. Tibouchi, A. Wallet, T. Espitau, A. Takahashi, Y. Yu and S. Guilley. *SOLMAE Algorithm Specifications*, 2022. Available at http://solmae-sign.info/.

[Laa15] T. Laarhoven. Search Problems in Cryptography: From Fingerprinting to Lattice Sieving, 2015. http://www.thijs.com/docs/phd-final.pdf.

[LLL82] A. K. Lenstra, H. W. Lenstra and L. Lovász. Factoring Polynomials with Rational Coefficients. In *Math. Ann.*, 1982.

[LLLW23] Q. Lai, F.-H. Liu, A. Lysyanskaya and Z. Wang. Lattice-based Commit-Transferrable Signatures and Applications to Anonymous Credentials. In *IACR Cryptol. ePrint Arch.*, page 766, 2023.

[LLM+16] B. Libert, S. Ling, F. Mouhartem, K. Nguyen and H. Wang. Signature Schemes with Efficient Protocols and Dynamic Group Signatures from Lattice Assumptions. In *ASIACRYPT*, 2016. doi:10.1007/978-3-662-53890-6_13.

[LM06] V. Lyubashevsky and D. Micciancio. Generalized Compact Knapsacks Are Collision Resistant. In *ICALP*, 2006. doi:10.1007/11787006_13.

[LN13] M. Liu and P. Q. Nguyen. Solving BDD by Enumeration: An Update. In *CT-RSA*, 2013. doi:10.1007/978-3-642-36095-4_19.

[LNP22] V. Lyubashevsky, N. K. Nguyen and M. Plançon. Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General. In *CRYPTO*, 2022. doi:10.1007/978-3-031-15979-4_3.

[LNPS21] V. Lyubashevsky, N. K. Nguyen, M. Plançon and G. Seiler. Shorter Lattice-Based Group Signatures via "Almost Free" Encryption and Other Optimizations. In *ASIACRYPT*, 2021. doi:10.1007/978-3-030-92068-5_8.

[LNS20] V. Lyubashevsky, N. K. Nguyen and G. Seiler. Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations. In *CCS*, 2020. doi:10.1145/3372297.3417894.

[LNS21] V. Lyubashevsky, N. K. Nguyen and G. Seiler. Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments. In *PKC*, 2021. doi:10.1007/978-3-030-75245-3_9.

[LP11] R. Lindner and C. Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. In *CT-RSA*, 2011. doi:10.1007/978-3-642-19074-2_21.

[LP15] V. Lyubashevsky and T. Prest. Quadratic Time, Linear Space Algorithms for Gram-Schmidt Orthogonalization and Gaussian Sampling in Structured Lattices. In *EUROCRYPT*, 2015. doi:10.1007/978-3-662-46800-5_30.

[LPR10] V. Lyubashevsky, C. Peikert and O. Regev. On Ideal Lattices and Learning with Errors over Rings. In *EUROCRYPT*, 2010. doi:10.1007/978-3-642-13190-5_1.

[LPR13a] V. Lyubashevsky, C. Peikert and O. Regev. A Toolkit for Ring-LWE Cryptography. In *EUROCRYPT*, 2013. doi:10.1007/978-3-642-38348-9_3.

[LPR13b] V. Lyubashevsky, C. Peikert and O. Regev. On Ideal Lattices and Learning with Errors over Rings. In *J. ACM*, 2013. doi:10.1145/2535925.

[LS15] A. Langlois and D. Stehlé. Worst-case to Average-case Reductions for Module Lattices. In *DCC*, 2015. doi:10.1007/s10623-014-9938-4.

[LS18] V. Lyubashevsky and G. Seiler. Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs. In *EUROCRYPT*, 2018. doi:10.1007/978-3-319-78381-9_8.

[LSS14] A. Langlois, D. Stehlé and R. Steinfeld. GGHLite: More Efficient Multilinear Maps from Ideal Lattices. In *EUROCRYPT*, 2014. doi:10.1007/978-3-642-55220-5_14.

[LW15] V. Lyubashevsky and D. Wichs. Simple Lattice Trapdoor Sampling from a Broad Class of Distributions. In *PKC*, 2015. doi:10.1007/978-3-662-46447-2_32.

[LW20] F. Liu and Z. Wang. Rounding in the Rings. In *CRYPTO*, 2020. doi:10.1007/978-3-030-56880-1_11.

[Lyu12] V. Lyubashevsky. Lattice Signatures without Trapdoors. In *EUROCRYPT*, 2012. doi:10.1007/978-3-642-29011-4_43.

[Mic07] D. Micciancio. Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions. In *Comput. Complex.*, 2007. doi:10.1007/s00037-007-0234-9.

[Mic18] D. Micciancio. On the Hardness of Learning With Errors with Binary Secrets. In *Theory Comput.*, 2018. doi:10.4086/toc.2018.v014a013.

[MM11] D. Micciancio and P. Mol. Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In *CRYPTO*, 2011. doi:10.1007/978-3-642-22792-9_26.

[MP12] D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, 2012. doi:10.1007/978-3-642-29011-4_41.

[MP13] D. Micciancio and C. Peikert. Hardness of SIS and LWE with Small Parameters. In *CRYPTO*, 2013. doi:10.1007/978-3-642-40041-4_2.

[MR07] D. Micciancio and O. Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. In *SIAM J. Comput.*, 2007. doi:10.1137/S0097539705447360.

[NISa] NIST. Post-Quantum Cryptography Standardization. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization.

[NISb] NIST. Post-Quantum Cryptography: Standardization of Additional Digital Signature Schemes. https://csrc.nist.gov/Projects/pqc-dig-sig/standardization.

[NR06] P. Q. Nguyen and O. Regev. Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures. In *EUROCRYPT*, 2006. doi:10.1007/11761679_17.

[Pei08] C. Peikert. Limits on the Hardness of Lattice Problems in l_p Norms. In *Comput. Complex.*, 2008. doi:10.1007/s00037-008-0251-3.

[Pei10] C. Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In *CRYPTO*, 2010. doi:10.1007/978-3-642-14623-7_5.

[PFH+20] T. Prest, P. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte and Z. Zhang. *FALCON. Tech. rep.*, 2020. Available at https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022.

[PHS19] A. Pellet-Mary, G. Hanrot and D. Stehlé. Approx-SVP in Ideal Lattices with Pre-processing. In *EUROCRYPT*, 2019. doi:10.1007/978-3-030-17656-3_24.

[PP19] C. Peikert and Z. Pepin. Algebraically Structured LWE, Revisited. In *TCC*, 2019. doi:10.1007/978-3-030-36030-6_1.

[PR06] C. Peikert and A. Rosen. Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In *TCC*, 2006. doi:10.1007/11681878_8.

[Pre15] T. Prest. *Gaussian Sampling in Lattice-Based Cryptography.* Ph.D. thesis, École Normale Supérieure, Paris, France, 2015.

[Pre17] T. Prest. Sharper Bounds in Lattice-Based Cryptography Using the Rényi Divergence. In *ASIACRYPT*, 2017. doi:10.1007/978-3-319-70694-8_13.

[PRS17] C. Peikert, O. Regev and N. Stephens-Davidowitz. Pseudorandomness of Ring-LWE for Any Ring and Modulus. In *STOC*, 2017. doi:10.1145/3055399.3055489.

[PS16] D. Pointcheval and O. Sanders. Short Randomizable Signatures. In *CT-RSA*, 2016. doi:10.1007/978-3-319-29485-8_7.

[Rén61] A. Rényi. On Measures of Entropy and Information. In *Proc. 4th Berkeley Sympos. Math. Statist. and Prob., Vol. I*, 1961.

[Reg05] O. Regev. On Lattices, Learning With Errors, Random Linear Codes, and Cryptography. In *STOC*, 2005. doi:10.1145/1060590.1060603.

[Reg09] O. Regev. On Lattices, Learning With Errors, Random Linear Codes, and Cryptography. In *J. ACM*, 2009. doi:10.1145/1568318.1568324.

[Rja94] S. Rjasanow. Effective Algorithms with Circulant-Block Matrices. In *Linear Algebra and its Applications*, 1994.

[Ros20] M. Rossi. *Extended Security of Lattice-Based Cryptography. (Sécurité Étendue de la Cryptographie Fondée sur les Réseaux Euclidiens).* Ph.D. thesis, Paris Sciences et Lettres University, France, 2020.

[RSW18] M. Rosca, D. Stehlé and A. Wallet. On the Ring-LWE and Polynomial-LWE Problems. In *EUROCRYPT*, 2018. doi:10.1007/978-3-319-78381-9_6.

[San20] O. Sanders. Efficient Redactable Signature and Application to Anonymous Credentials. In *PKC*, 2020. doi:10.1007/978-3-030-45388-6_22.

[San21] O. Sanders. Improving Revocation for Group Signature with Redactable Signature. In *PKC*, 2021. doi:10.1007/978-3-030-75245-3_12.

[SE94] C. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. In *Math. Program.*, 66:181–199, 1994. doi:10.1007/BF01581144.

[Sho94] P. W. Shor. Polynominal Time Algorithms for Discrete Logarithms and Factoring on a Quantum Computer. In *ANTS*, 1994. doi:10.1007/3-540-58691-1_68.

[SSTX09] D. Stehlé, R. Steinfeld, K. Tanaka and K. Xagawa. Efficient Public Key Encryption Based on Ideal Lattices. In *ASIACRYPT*, 2009. doi:10.1007/978-3-642-10366-7_36.

[ST23] O. Sanders and J. Traoré. Efficient Issuer-Hiding Authentication, Application to Anonymous Credential. In *IACR Cryptol. ePrint Arch.*, page 1845, 2023.

[STA20] C. Sun, M. Tibouchi and M. Abe. Revisiting the Hardness of Binary Error LWE. In *ACISP*, 2020. doi:10.1007/978-3-030-55304-3_22.

[TCG15] TCG. https://trustedcomputinggroup.org/authentication/, 2015.

[vEH14] T. van Erven and P. Harremoës. Rényi Divergence and Kullback-Leibler Divergence. In *IEEE Trans. Inf. Theory*, 2014.

[Ver12] R. Vershynin. Introduction to the Non-Asymptotic Analysis of Random Matrices. In *Compressed Sensing*, 2012. doi:10.1017/cbo9780511794308.006.

[Wag02] D. A. Wagner. A Generalized Birthday Problem. In *CRYPTO*, 2002. doi:10.1007/3-540-45708-9_19.

[WW19] Y. Wang and M. Wang. Module-LWE versus Ring-LWE, Revisited. In *IACR Cryptol. ePrint Arch.*, page 930, 2019. Version dated from Aug. 18th 2019.

[YAZ+19] R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu and W. Whyte. Efficient Lattice-Based Zero-Knowledge Arguments with Standard Soundness: Construction and Applications. In *CRYPTO*, 2019. doi:10.1007/978-3-030-26948-7_6.

[YJW23] Y. Yu, H. Jia and X. Wang. Compact Lattice Gadget and Its Applications to Hash-and-Sign Signatures. In *CRYPTO*, 2023. doi:10.1007/978-3-031-38554-4_13.

COLLEGE DOCTORAL BRETAGNE

MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

Université de Rennes

Titre : Conception d'Algorithmes de Signatures Avancées Post-Quantiques

Mot clés : Cryptographie sur Réseaux Euclidiens, Apprentissage Avec Erreurs sur les Modules, Échantillonnage d'Antécédents, Signatures Numériques, Accréditations Anonymes, Vie Privée, Implémentation

Résumé : La transition vers la cryptographie post-quantique est une tâche considérable ayant suscité un nombre important de travaux ces dernières années. En parallèle, la cryptographie pour la protection de la vie privée, visant à pallier aux limitations inhérentes des mécanismes cryptographiques basiques dans ce domaine, a connu un véritable essor. Malgré le succès de chacune de ces branches prises individuellement, combiner les deux aspects de manière efficace s'avère extrêmement difficile.

Le but de cette thèse de doctorat consiste alors à proposer de nouvelles constructions visant à garantir une protection efficace et post-quantique de la vie privée, et plus généralement des mécanismes d'authentification avancés. Dans ce but, nous nous consacrons tout d'abord à l'étude de l'une des hypothèses mathématiques fondamentales utilisées en cryptographie sur les réseaux Euclidiens: *Module Learning With Errors*. Nous prouvons que le problème ne devient pas significativement plus facile même en choisissant des distributions de secret et d'erreur plus courtes. Ensuite, nous proposons des optimisations des échantillonneurs d'antécédents utilisés par de nombreuses signatures avancées. Loin d'être limitées à ce cas d'usage, nous montrons que ces optimisations mènent à la conception de signatures standards efficaces. Enfin, à partir de ces contributions, nous concevons des algorithmes de *signatures avec protocoles efficaces*, un outil polyvalent utile à la construction d'applications avancées. Nous en montrons les capacités en proposant le premier mécanisme d'accréditation anonyme post-quantique, que nous implémentons afin de mettre en exergue son efficacité aussi bien théorique que pratique.

Title: Design of Advanced Post-Quantum Signature Schemes

Keywords: Lattice-Based Cryptography, Module Learning With Errors, Preimage Sampling, Digital Signatures, Anonymous Credentials, Privacy, Implementation

Abstract: The transition to post-quantum cryptography has been an enormous effort for cryptographers over the last decade. In the meantime, cryptography for the protection of privacy, aiming at addressing the limitations inherent to basic cryptographic mechanisms in this domain, has also attracted a lot of attention. Nevertheless, despite the success of both individual branches, combining both aspects along with practicality turns out to be very challenging.

The goal of this thesis then lies in proposing new constructions for practical post-quantum privacy, and more generally advanced authentication mechanisms. To this end, we first focus on the lower level by studying one of the fundamental mathematical assumptions used in lattice-based cryptography: Module Learning With Errors. We show that it does not get significantly easier when stretching the secret and error distributions. We then turn to optimizing preimage samplers which are used in advanced signature designs. Far from being limited to this use case, we show that it also leads to efficient designs of regular signatures. Finally, we use some of the previous contributions to construct so-called signatures with efficient protocols, a versatile building block in countless advanced applications. We showcase it by giving the first post-quantum anonymous credentials, which we implement to demonstrate a theoretical and practical efficiency.