Lattice - Based Cryptography - Security Foundations and Constructions

Adeline Roux-Langlois

THÈSE

en vue de l'obtention du grade de

Docteur de l'Université de Lyon, délivré par l'École Normale Supérieure de Lyon

Discipline : Informatique
Laboratoire de l'Informatique et du Parallélisme
École Doctorale Informatique et Mathématiques

présentée et soutenue publiquement le 17 octobre 2014
par Madame Adeline LANGLOIS

Lattice-Based Cryptography: Security Foundations and Constructions

---

Directeur de thèse : M. Damien STEHLÉ

Après avis de :
M. Eike KILTZ
M. Alon ROSEN
M. Gilles ZÉMOR

Devant la commission d'examen formée de :
M. Vadim LYUBASHEVSKY, INRIA, Examinateur
M. Alon ROSEN, IDC Herzliya, Rapporteur
M. Damien STEHLÉ, École Normale Supérieure de Lyon, Directeur
M. Stéphan THOMASSÉ, École Normale Supérieure de Lyon, Examinateur
M. Gilles ZÉMOR, Université de Bordeaux, Rapporteur

Acknowledgements

Après quelques jours de réflexion, je réalise qu'il n'est pas si simple de trouver les mots pour remercier, comme il se doit, les personnes qui m'ont aidée et soutenue pendant ces trois dernières années, ou depuis bien plus longtemps. Ainsi même si les mots sont simples, et se répètent parfois, ne vous méprenez pas, je mesure la chance que j'ai d'être si bien entourée depuis de nombreuses années et exprime ma profonde gratitude à toutes les personnes qui m'ont permis, de près ou de loin, de finir cette thèse aujourd'hui.

Je voudrais tout particulièrement et très sincèrement remercier mon directeur de thèse Damien Stehlé. Je le remercie de m'avoir donné l'opportunité de travailler dans ce laboratoire à ses côtés, de m'avoir consacré tout le temps nécessaire pour que je puisse découvrir et apprécier la recherche en cryptographie, et de m'avoir soutenue tout le long de ma thèse. Encore une fois, je mesure la chance que j'ai eue de recevoir un encadrement d'une telle qualité, et je lui en suis très reconnaissante.

I would like to thank the reviewers of my thesis Gilles Zémor, Alon Rosen and Eike Kiltz for their interest in my research. I also thank Alon Rosen for coming from abroad to be my committee. Je remercie également Gilles Zémor de s'être rendu disponible pour comprendre mon travail et de faire partie de mon jury de thèse. Je remercie Stéphan Thomassé d'avoir consacré du temps à mes travaux de thèse, qui sont pourtant éloignés de son domaine de recherche, pour faire partie de mon jury. I thank Vadim Lyubashevsky for all his advices along my PhD, and with whom I had very interesting conversations. I also thank him for having accepted to be in my committee on a very short notice.

Enfin, je remercie Brigitte Vallée qui n'a malheureusement pas pu venir à ma soutenance de thèse. Je la remercie tout particulièrement de m'avoir fait découvrir la recherche lors de mon premier stage en 2008, et de m'avoir très bien conseillée depuis.

Je remercie tous les membres de l'équipe Aric : Benoît, Bruno, Claude-Pierre, Clément, Fabien, Gilles, Guillaume, Jean-Michel, Marie, Nathalie, Nicolas B., Nicolas L., Valentina et Vincent, ainsi que ses anciens membres : Marc, Nicolas B., Stef, Rishi et Erik, qui m'ont très gentiment accueillie pendant ces trois ans et avec lesquels j'ai partagé de très bons moments. Je remercie aussi les doctorants/stagiaires du printemps 2014 : Vincent, Sébastien, François, Thomas, Catalin, Silviu, Philippe et Serge, pour les goûters Aric hebdomadaires ainsi que Eleonora et Ioana pour les "pauses filles" : un grand bol d'air et de détente.

Je remercie aussi et en particulier Philippe, qui a commencé sa thèse en même temps que moi dans l'équipe Aric, pour toutes nos "pauses thé, chocolat" (et parfois des chouquettes, merci pour la recette), de très bons moments pendant nos journées de thésards. Je remercie aussi tous mes co-bureaux pendant ces trois années : Xavier, Matei, Jingwei, Nicolas E., Silviu et Serge, dont

l'agréable compagnie m'a permis de venir au laboratoire tous les jours avec le sourire.

Un grand merci aussi à notre assistant d'équipe Damien Séon, qui m'a aidé à affronter toutes les procédures administratives durant mon séjour au LIP, ainsi qu'à toutes les assistances : Catherine, Chiraz, Évelyne, Laeticia, Marie, Séverine et Sylvie pour être toujours disponibles.

I deeply thank all my co-authors during those three years: Zvika, Oded, Chris, Fabien, Benoît, Khoa, San, Huaxiong and Ron. I felt really lucky to work with them, enjoyed it, and learned a lot discussing with them. I would like to thank in particular Ron Steinfeld, who welcomed me three times in Australia. These travels was an amazing experience and I really appreciated working with him. I thank him a lot for all the time he spent working with me, and I also learned a lot from all those discussions.

Enfin je remercie tous les autre doctorants et chercheurs, rencontrés au LIP : Jean-Marie (que je remercie aussi pour sa relecture de ces remerciements), Aurélie, Valentin, Sébastien, Guillaume, Wissam, ou rencontrés durant des visites ou des conférences : Julien, Léo, Anja, Alexandre ainsi que Loïck, Julien C. et Ali (pour leur accueil très chaleureux pendant mes séjours à Caen). Et je remercie aussi Tancrède, qui a commencé sa thèse en même temps que moi, et avec qui j'ai passé de très bons moments en conférence à découvrir Cambridge, Buenos Aires ou Copenhague.

Je voudrais enfin remercier ma famille et mes amis, et là encore, les mots me manquent. J'ai une profonde gratitude et beaucoup d'affection pour mes parents, Daniel et Brigitte, qui m'ont beaucoup apporté, toujours soutenue et à qui je dois beaucoup. Je remercie aussi ma sœur Marion et son compagnon Grégoire, toute ma famille ainsi que ma récente belle famille. Je remercie mes amis d'enfance : Raph, Anne-So, Elisa, d'études : Perrine, Gautier, Marie, Nicolas, Olivier et Clément M. et d'équitation : Caroline, Marie-Laure ...

Cela peut sembler étrange, mais une autre forme de soutien m'a aussi beaucoup aidée à finir cette thèse et j'ai quand même envie de les mentionner, il s'agit de Nazdac, mon cheval, et Miss Tigree, mon chat. Leur affection, et le bol d'air qu'ils m'apportent au quotidien me sont très précieux.

Enfin, et plus que tout, je voudrais remercier celui qui partage ma vie et sans qui rien n'aurait été possible. Mon ami, mon amour, mon mari maintenant, Clément. Merci pour ce soutien inconditionnel qui m'apporte tellement. Merci pour tout.

Contents

Introduction

Cryptologie

La cryptologie est la science de la sécurité de l'information. Elle permet, entre autres, de protéger les données en les « chiffrant », pour qu'il ne soit plus possible de trouver la moindre information les concernant. On distingue ensuite la cryptographie, relative à la construction des primitives, de la cryptanalyse, qui étudie les attaques possibles contre les schémas existants. Il existe deux types de cryptographies, la cryptographie symétrique, et la cryptographie asymétrique (ou à clé publique, introduite par Diffie et Hellman en 1976 [DH76]). Prenons l'exemple d'un schéma de chiffrement, l'objectif de ce schéma est d'envoyer un message de façon confidentielle. En cryptographie symétrique, deux utilisateurs partagent la même clé secrète, qui leur permet, grâce à des algorithmes de chiffrement et de déchiffrement, de chiffrer le message et de le déchiffrer (à partir du chiffré). Le problème ici est de trouver un moyen, lui-même sûr, de partager cette clé secrète entre les utilisateurs. Pour régler ce problème la cryptographie asymétrique propose à chaque utilisateur d'avoir un couple de clés : l'une secrète et l'autre publique, reliées bien sûr, mais de telle façon qu'il ne soit pas possible de retrouver la clé secrète étant donné la clé publique. Puis grâce à la clé publique d'un utilisateur A, un utilisateur B va pouvoir chiffrer un message et l'envoyer à A, qui sera le seul à pouvoir extraire le message de ce chiffré grâce à sa clé secrète. Ainsi les deux utilisateurs n'ont plus besoin de partager une clé secrète pour communiquer. Ils doivent cependant s'assurer que l'utilisateur avec lequel ils communiquent est bien celui qu'ils pensent.

Il existe de nombreuses autres primitives cryptographiques qui sont utilisées au quotidien, par exemple : les signatures numériques (qui permettent justement d'authentifier des messages), les fonctions de hachage (qui permettent par exemple de vérifier que deux données sont les mêmes, sans les dévoiler) ou les schémas de chiffrement basé sur l'identité (la clé publique sera alors l'identité de l'utilisateur). Il existe aussi des primitives prometteuses, qu'on espère pouvoir utiliser bientôt, comme le chiffrement complètement homomorphe (qui permet de faire des opérations, qu'on voudrait faire sur des données, directement sur les chiffrés de ces donnés, puis de déchiffrer pour obtenir le résultat voulu) ou les applications multilinéaires cryptographiques que nous étudierons en détail plus loin. Pour toutes ces primitives, différents modèles de sécurité peuvent être définis. Ils garantissent une sécurité contre différents types d'attaques où les adversaires sont plus ou moins puissants. Dans le cas du chiffrement par exemple, l'objectif minimal de sécurité est de ne pas pouvoir distinguer un message chiffré d'un autre : si un attaquant envoie deux messages et reçoit un chiffré de l'un des deux, il ne doit pas être capable de décider lequel a été chiffré. Les autres niveaux de sécurité vont ensuite permettre à l'attaquant de faire des requêtes de chiffrement, ou de déchiffrement, à différents moments de l'attaque.

Il existe aussi différentes familles de problèmes algorithmiques servant de fondement à la

construction de protocoles cryptographiques. Dans chacun d'entre eux, la sécurité des primitives va reposer de manière prouvée, ou heuristique, sur des problèmes mathématiques bien particuliers et différents. Un exemple parmi les plus utilisés est la cryptographie de type RSA, introduite en 1978 [RSA78]. Dans cet exemple, la difficulté de retrouver la clé secrète en fonction de la clé publique repose sur la difficulté algorithmique présumée de factoriser un produit de deux grands nombres premiers. La clé publique est un grand nombre N, produit de deux nombres premiers p et q, et la clé secrète est liée à ces deux nombres inconnus et est difficile à retrouver. Mais il n'existe pas de preuve de sécurité à proprement parler pour ces schémas. D'autres hypothèses algorithmique couramment utilisées sont la difficulté du problème du logarithme discret dans des groupes bien choisis [DH76, Gam85], le problème de décodage des codes correcteurs d'erreurs [McE78], ou celui de résoudre des systèmes polynomiaux [Pat96].

Cryptographie reposant sur les réseaux euclidiens

La cryptographie reposant sur les réseaux euclidiens est une autre branche de la cryptographie, qui a débuté avec les travaux d'Ajtai [Ajt96]. Le nom de cette cryptographie fait référence aux preuves de sécurité d'une partie de ses primitives, qui reposent sur la difficulté présumée de problèmes algorithmiques sur les réseaux euclidiens. Elle a de nombreux avantages : son fonctionnement est simple, son efficacité potentielle est importante et elle semble résister aux attaques quantiques. Les attaques quantiques sont une préoccupation importante en cryptographie depuis que Shor [Sho97] a réussit à résoudre, grâce à des algorithmes quantiques, des problèmes, comme la factorisation du produit deux grand nombres premier, que l'on ne sait pas résoudre efficacement en algorithmique classique.

De plus, comme nous allons le voir plus en détails, la cryptographie reposant sur les réseaux euclidiens permet de construire de très nombreux types de primitives, des plus basiques (par exemple : chiffrement [HPS98, Reg05, GPV08, PVW08, LP11], signature [GGH97, GPV08, LM08, Lyu09, Lyu12, GLP12, DDLL13], fonction de hachage [Ajt96, GGH96, Mic02a, PR06, LM06, LM08, Lyu08] ...) aux plus avancées (chiffrement reposant sur l'identité [GPV08, CHKP10, ABB10a, ABB10b], chiffrement par attribut [GVW13], chiffrement complètement homomorphe [Gen09, BV11, BGV12, Bra12], applications multilinéaires cryptographiques [GGH13a, CLT13] et offuscation de programmes [GGH+13b]).

Un réseau euclidien est un sous groupe additif discret de $\mathbb{R}^n$, et est représenté par une base $\mathbf{B}$. Le réseau engendré par la base $\mathbf{B}$ est l'ensemble des combinaisons linéaires entières des éléments de cette base. Un exemple est donné dans la Figure 1 pour $n = 2$.

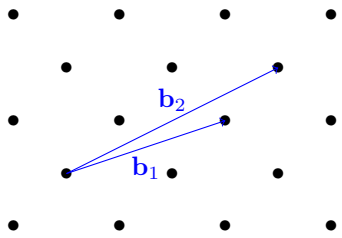

Figure 1: Exemple de réseau euclidien en dimension 2.

Il existe de nombreux problèmes algorithmiques concernant les réseaux. Les plus connus sont le problème du plus court vecteur, ou *Shortest Vector Problem* (SVP), et le problème du plus proche vecteur, ou *Closest Vector Problem* (CVP). Etant donné une base d'un réseau, le problème SVP

consiste à trouver un vecteur non nul le plus court (par exemple pour la norme euclidienne) de ce réseau. Le problème CVP prend de plus en entrée un vecteur quelconque $\mathbf{t} \in \mathbb{R}^n$, et demande de trouver un vecteur du réseau le plus proche de ce vecteur cible. Si on revient à l'exemple de la Figure 1, ces problèmes sont faciles à résoudre (car la dimension du réseau est petite), mais en grande dimension ils semblent très difficiles. Ils sont NP-difficile (classiquement pour CVP, sous des réductions probabilistes pour SVP) [vEB81, Mic98, BS99]. Ils restent difficiles à résoudre même si on leur autorise un facteur d'approximation γ, tant que γ reste petit : par exemple polynomial en la dimension. Pour SVP, au lieu de chercher un vecteur non-nul le plus court, on cherche un vecteur de norme au plus γ multipliée par la norme du plus petit vecteur du réseau. Les problèmes SVP et CVP ont de plus de nombreuses variantes, et c'est souvent sur ces variantes que repose la sécurité des primitives cryptographiques. La conjecture sur laquelle repose la cryptographie reposant sur les réseaux Euclidiens est la suivante : ces problèmes et leurs variantes sont difficiles à résoudre même pour des facteurs d'approximation polynomiaux en la dimension du réseau [MR09]. Cette conjecture est soutenue par de nombreuses années d'étude des algorithmes pouvant résoudre ces problèmes, qui gardent une complexité au mieux exponentielle en la dimension n du réseau pour un facteur d'approximation polynomial ou une complexité polynomiale mais pour un facteur d'approximation quasiment exponentiel [LLL82, Sch87, AKS01, GN08].

À partir de ces problèmes, le principe de la preuve de sécurité d'une primitive est le suivant. Lorsqu'on construit une primitive cryptographique, on modélise les attaques possibles contre cette primitive par un problème à résoudre. Si l'attaquant peut résoudre ce problème alors il est capable de retrouver des informations sur les données secrètes et la primitive n'est pas sûre, mais si on montre que ce problème n'est pas soluble en un temps raisonnable, alors on montre que la primitive construite est sûre. Pour chaque type de primitive, il existe différents niveaux de sécurité, qu'on peut alors modéliser par différents problèmes. Pour garantir la sécurité d'une primitive, on montre alors que le problème sur lequel repose l'attaque est au moins aussi difficile à résoudre qu'un problème réputé difficile. On dit qu'on effectue une réduction d'un problème difficile à cette attaque. Cette réduction garantit que l'attaque n'est pas possible en un temps raisonnable : la primitive est sûre. Dans la cryptographie reposant sur les réseaux Euclidiens, ce problème est un problème portant sur les réseaux Euclidiens, typiquement une variante de SVP. Notons qu'il existe aussi des constructions dont la sécurité ne repose pas sur une réductions aux problèmes dans les réseaux. C'est le cas du schéma de chiffrement NTRU [HPS98], mais dont la sécurité d'une variante a été récemment montrée par [SS13]. C'est aussi le cas du protocole d'échange de clé non-interactif de Diffie-Hellmann pour un grand nombre d'utilisateur, construit à partir applications multilinéaires cryptographiques [GGH13a]. Dans ces deux cas, la sécurité est présumée suite à l'étude des meilleures attaques connues, qui reviennent elles aussi à résoudre une variante de SVP.

Pour effectuer les réductions utilisées dans les preuves de sécurité, on utilise souvent des problèmes intermédiaires, appelés le problème *Small Integer Solution* (SIS), introduit par Ajtai en 1996 [Ajt96], et le problème *Learning With Errors* (LWE), introduit par Regev en 2005 [Reg05]. Ces deux problèmes sont à la base de cette cryptographie. Le problème SIS revient à trouver une solution « petite » d'un système d'équations linéaires sous-déterminé modulo un entier q à n inconnues, la dimension du problème. Il permet par exemple de construire des schémas de signature numérique [GPV08, CHKP10, Boy10] et des fonctions de hachage [Ajt96]. Le problème LWE consiste à trouver une solution d'un système d'équations linéaires sur-déterminé mais bruité, avec n inconnues et modulo un entier q. Il permet par exemple de construire des schémas de chiffrement avancés [Reg05, GPV08, BV11, GVW13]. Pour tous ces schémas on prouve la sécurité en montrant que réussir une attaque est au moins aussi difficile que de résoudre SIS ou LWE. La difficulté de ces deux problèmes est donc très importante pour assurer la sécurité des primitives construites à partir d'eux. Il reste ensuite à montrer que ces deux problèmes sont en

effet difficiles à résoudre. On utilise pour cela des réductions particulières, appelées réduction « pire-cas moyen-cas ». On montre que SIS ou LWE (qui est donné pour des instances aléatoires, donc un cas « moyen »), est au moins aussi difficile à résoudre que toutes les instances d'une variante de SVP (même les plus difficiles d'entre elles, le « pire » des cas). En 1996, Ajtai [Ajt96] a proposé la première réduction d'un problème difficile sur les réseaux Euclidiens au problème SIS, montrant ainsi que ce problème était difficile à résoudre. En 2005, Regev [Reg05, Reg09] a proposé une réduction quantique (utilisant de l'algorithmique quantique en plus de l'algorithmique classique) pour prouver la difficulté du problème LWE.

On peut ainsi construire des primitives sûres reposant sur ces deux problèmes. Mais, pour assurer leur sécurité, les paramètres de ces primitives doivent être suffisamment grands pour que les variantes de SVP soient difficiles à résoudre (en pratique, on veut un paramètre n au minimum de l'ordre de quelques centaines). Ceci rend les premières constructions reposant sur SIS et LWE difficiles à utiliser en pratique. Pour résoudre ce problème, des versions structurées de SIS et LWE, permettant un gain en complexité important, ont été proposées. Elles sont appelées R-SIS et R-LWE où le R correspond à « Ring » pour un anneau, typiquement ici un anneau de polynôme. Ces variantes structurées ont été prouvées difficiles, sous certaines restrictions, par des réductions à une variante de SVP [Mic02a, LM06, PR06, SSTX09, LPR10]. Il faut notamment restreindre le problème aux réseaux idéaux, qui sont un type bien particulier de réseaux définis à partir des idéaux de l'anneau concerné.

Contributions

Mon travail durant ces trois années de doctorat a porté sur deux aspects de la cryptographie reposant sur les réseaux Euclidiens : les fondements de la sécurité, en étudiant la difficulté du problème LWE et de variantes des problèmes SIS et LWE, et les constructions de primitives, en travaillant sur les signatures de groupe et sur les applications multilinéaires cryptographiques.

Fondements de sécurité

Comme nous l'avons vu, la difficulté des problèmes LWE et SIS, ainsi que de leurs variantes structurées est à la base de la sécurité des primitives cryptographiques. La difficulté du problème LWE en particulier a été prouvée en 2005 par Regev grâce à une réduction quantique. Ce type de réduction utilise de l'algorithmique quantique en plus de l'algorithmique classique. Ceci signifie par exemple, que si on découvre un algorithme efficace pour résoudre LWE, on ne saura pas pour autant résoudre le problème « difficile » dans les réseaux si on ne dispose pas d'un ordinateur quantique. D'autre part, si on découvre que les problèmes dans les réseaux sont possibles à résoudre avec des ordinateurs quantiques, cette réduction reviendrait à réduire LWE à un problème facile, et n'aurait donc plus d'intérêt (en particulier si LWE s'avère quantiquement facile mais classiquement difficile). Il était donc important de savoir si il existait une réduction classique des problèmes dans les réseaux à LWE. En 2009, Peikert [Pei09] a partiellement répondu à la question en proposant la première réduction classique, mais celle-ci ne fonctionne que si le paramètre du modulo pour LWE est exponentiel. Mais le modulo q utilisé dans LWE détermine la taille des données (on se place dans $\{0, \ldots, q-1\}$) et pour des applications cryptographiques un modulo trop grand ne permet pas d'avoir des schémas efficaces.

Dans un travail commun avec Zvika Brakerski, Chris Peikert, Oded Regev et Damien Stehlé, nous donnons la première réduction classique de GapSVP (la version décisionnelle de SVP) à LWE pour un modulo polynomial. Nous obtenons aussi des résultats intermédiaires sur des variantes de LWE, et notamment que la difficulté de LWE est une fonction de $n \log q$, où n est la dimension du problème et q le modulo.

- [BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev et Damien Stehlé. *Classical Hardness of Learning with Errors. STOC*, pages 575-584. ACM, 2013.

J'ai par ailleurs travaillé sur les variantes structurées de SIS et LWE. Dans un article commun avec Damien Stehlé, nous étudions deux problèmes : un nouveau problème appelé M-SIS et un problème appelé M-LWE, introduit par Brakerski, Gentry et Vaikuntanathan [BGV11]. Le problème M-SIS est un problème intermédiaire qui généralise les problèmes SIS et R-SIS, où le « M » correspond à « Module » qui est la structure algébrique généralisant l'espace vectoriel et l'anneau. De même, le problème M-LWE généralise les problèmes LWE et R-LWE. Nous faisons le lien entre les deux réductions existantes de SIS à R-SIS pour M-SIS [GPV08, LM06] et de LWE à R-LWE pour M-LWE [Reg05, LPR10] et nous construisons les réductions correspondantes pour montrer la difficulté de ces deux problèmes. Nous montrons aussi que la difficulté de M-LWE (et donc de R-LWE) ne dépend pas de la forme arithmétique de q, alors que la réduction existante [LPR10] pour R-LWE nécessitait un modulo q premier et congru à 1 modulo $2n$.

- [LS] Adeline Langlois et Damien Stehlé. *Worst-case to Average-case Reductions for Module Lattices*. Accepté à *Designs, Codes and Cryptography*. Springer.

Construction de primitives

Je me suis intéressée à deux primitives cryptographiques, les signatures de groupe et les applications multilinéaires cryptographiques.

Les signatures de groupe permettent à tous les membres d'un groupe d'authentifier un message, de façon anonyme, au nom du groupe. Une autorité, qui génère les clés, peut aussi retrouver le membre du groupe qui a signé le message en cas de conflit ou d'utilisation malhonnête. Chaque membre du groupe possède une clé privée, et il y a une seule clé publique qui est reliée au groupe. L'autorité possède de plus une clé secrète maîtresse qui permet de tracer les utilisateurs. Les deux signatures de groupe reposant sur les réseaux Euclidiens qui pré-existaient à mes travaux [GKV10, CNR12] ont des tailles de clés et de signature linéaires en le nombre de membres du groupe. Contrairement à des constructions reposant sur les couplages dans les courbes elliptiques, où l'on trouve des signatures de groupe de taille logarithmique en le nombre de membres du groupe [BW07, Gro07]. Dans un travail commun avec Fabien Laguillaumie, Benoît Libert et Damien Stehlé, nous construisons la première signature de groupe reposant sur les réseaux Euclidiens qui admet une taille de signature logarithmique en le nombre d'utilisateurs du groupe. À la suite de ce travail, et en commun avec San Ling, Khoa Nguyen et Huaxiong Wang, nous construisons une signature de groupe de même complexité mais avec une autre fonctionnalité qui est la révocation. La révocation permet d'exclure un membre du groupe sans avoir à réinitialiser tout le système. La sécurité de la première construction repose sur la difficulté des problèmes SIS et LWE, et celle de la seconde sur la difficulté du problème SIS uniquement.

- [LLLS13] Fabien Laguillaumie, Adeline Langlois, Benoît Libert et Damien Stehlé. *Lattice-based Group Signature with Logarithmic Signature Size. ASIACRYPT (2)*, volume 8270 de LNCS, pages 41-61. Springer, 2013.

- [LLNW14] Adeline Langlois, San Ling, Khoa Nguyen et Huaxiong Wang. *Lattice-based Group Signature with Verifier Local Revocation. Public Key Cryptography*, volume 8383 de LNCS, pages 345-361. Springer, 2014.

Construire des applications multilinéaires cryptographiques a été un problème ouvert pendant de nombreuses années, en particulier depuis le résultat de Boneh et Silverberg [BS03] qui donne

des applications à ces primitives. Ils décrivent par exemple un protocole non-interactif d'échange de clés entre N utilsateurs (où $N > 3$) qui généralise celui de Diffie-Hellman [DH76]. En 2013, Garg, Gentry et Halevi [GGH13a] ont proposé la première approximation d'une application multilinéaire cryptographique qu'ils ont appelée un schéma de codage à niveaux (*graded encoding scheme*). À partir de là, le protocole d'échange de clés devient possible pour un grand nombre d'utilisateurs, mais sa sécurité n'est pas prouvée, elle repose sur l'analyse de la meilleure attaque connue (qui revient à résoudre une variante de SVP dans des réseaux idéaux). Ma contribution pendant cette thèse, en commun avec Damien Stehlé et Ron Steinfeld, porte sur l'amélioration du schéma de codage à niveaux proposé par [GGH13a]. Nous avons, d'une part, analysé la sécurité du processus de *re-randomisation* de ce schéma. Ce processus est utilisé à chaque étape d'encodage, pour s'assurer qu'il n'y a pas de corrélation entre l'élément encodé et celui qui a permis de l'encoder. Cette analyse nous a permis d'améliorer la taille des paramètres, en utilisant notamment la divergence de Rényi, au lieu de la distance statistique utilisée classiquement pour étudier la distance entre deux distributions. D'autre part, nous avons démontré un nouveau *Leftover Hash Lemma*. Cet outil classique en cryptographie permet par exemple d'extraire de l'aléa uniformément distribué à partir d'une mauvaise source d'aléa. Ici nous l'utilisons sur des distributions Gaussiennes discrètes et l'aléa uniforme est lui-même une Gaussienne discrète. Notre résultat adapte celui de [AGHS13] à certains anneaux de polynômes (les mêmes que ceux utilisés dans les variantes structurées R-SIS et R-LWE). Ces deux contributions nous permettent de rendre la construction d'origine plus efficace.

- [LSS14] Adeline Langlois, Damien Stehlé et Ron Steinfeld. *GGHLite: More Efficient Multilinear Maps from Ideal Lattices. EUROCRYPT 2014*, volume 8441 de LNCS, pages 239-256. Springer, 2014.

Autre contribution

Durant ma thèse, j'ai aussi participé à l'écriture d'un chapitre de livre, en commun avec Fabien Laguillaumie et Damien Stehlé, intitulé « Chiffrement avancé à partir du problème *Learning with Errors* ». Ce chapitre présente le problème LWE, et propose une introduction aux schémas de chiffrements avancés en s'appuyant sur ceux construits en cryptographie reposant sur les réseaux euclidiens, et dont la sécurité repose sur LWE. En particulier nous décrivons les schémas de Regev et Dual-Regev [Reg05, GPV08], le schéma de chiffrement reposant sur l'identité de [CHKP10], et le schéma de chiffrement par attributs de [GVW13].

- [LLS14] Fabien Laguillaumie, Adeline Langlois et Damien Stehlé. Chiffrement avancé à partir du problème Learning With Errors. Chapitre du livre *Informatique Mathématique une photographie en 2014*, éditeur Sylvain Peyronnet, pages 179-225. Presses Universitaires de Perpignan, 2014.

Presentation of my contributions

Lattice-based cryptography

Lattice-based cryptography is a branch of cryptography exploiting the presumed hardness of lattice problems. Its main advantages are its simplicity, efficiency, and apparent security against quantum computers. Quantum computers are an important concern in cryptography since Shor [Sho97] solved problems serving as security foundation for many primitives, as the factorization of two large prime, that we can not solve efficiently with a classical computer. But perhaps the most appealing aspect is that lattice-based cryptographic protocols often enjoy very strong security proofs based on the hardness of worst-case problems. Moreover, lattice-based cryptography allows to construct a wide range of primitives, from the basic ones (for example: encryption schemes [HPS98, Reg05, GPV08, PVW08, LP11], signatures [GGH97, GPV08, LM08, Lyu09, Lyu12, GLP12, DDLL13], hash functions [Ajt96, GGH96, Mic02a, PR06, LM06, LM08, Lyu08] ...) to the most advanced ones (identity-based encryption [GPV08, CHKP10, ABB10a, ABB10b], attribute-based encryption [GVW13], fully homomorphic encryption [Gen09, BV11, BGV12, Bra12], cryptographic multilinear maps [GGH13a, CLT13] and programs obfuscation [GGH+13b]).

An Euclidean lattice is the set of all integer linear combinations of some n linearly independent vectors belonging to a euclidean space, that we call a basis $\mathbf{B}$. An example is given Figure 2 for $n = 2$.

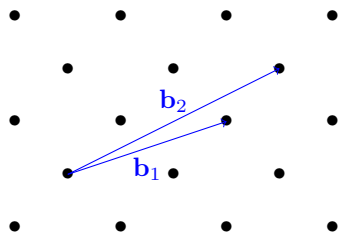

Figure 2: Example of a lattice in dimension 2.

There are many algorithmic problems related to lattices. The most classical ones are the *Shortest Vector Problem* (SVP), and the *Closest Vector Problem* (CVP). Given a basis of a lattice, SVP asks to find a shortest non-zero vector of the lattice. The CVP problem takes also as input a target vector $\mathbf{t} \in \mathbb{R}^n$, and asks to find one of the closest vectors of the target. Those two problems are easy to solve in dimension 2, but very hard in high dimension. They are NP-hard (classically

for CVP, under randomized reduction for SVP [vEB81, Mic98, BS99]) And they seem still hard to solve even with an approximation factor γ, at least as long as γ remains polynomial in the dimension of the lattice. For example for SVP, instead of looking for one of the shortest non-zero vectors of the lattice, the goal is to find a vector of norm which is at most γ times the norm of the shortest vector of the lattice. SVP and CVP also have many variants, and the security of the cryptographic primitives is often based on one of these variants. In particular we will consider the *Shortest Independent Vectors Problem* (SIVP), where the goal is to find n linearly independent vectors in an n-dimensional lattice, which have the shortest norm possible. Finally, a standard and well accepted conjecture is to assume that there is no polynomial time algorithm that achieves an approximation factor that is polynomial in n for any of these problems, even using quantum computing [MR09]. This conjecture is supported by years of study of algorithms to solve those problems, which keep a complexity exponential in the dimension for a polynomial approximation factor [LLL82, Sch87, AKS01, GN08].

The security of the cryptographic primitive is proven from the hardness of those problems. Typically the possible attacks against the primitive will be modelled by a problem to solve. If an attacker is able to solve this problem, then it will be able to break the security of the primitive, on the other hand if one can show that this problem is not possible to solve in a reasonable time, then the primitive is secure. For each sort of primitive, there exist different levels of security modelled by different problems. To guaranty the security, we show that the problem, on which is based the attack, is at least as hard to solve as a hard problem, typically a variant of SVP. This is called a reduction. Note that there also exist constructions for which the security does not rely on reductions from hard lattice problems. For example, this is the case of the NTRU encryption scheme [HPS98] (the security of one of its variant has recently be proven by [SS13]), and the N-party one round Diffie-Hellman key exchange constructed from cryptographic multilinear maps [GGH13a]. In both cases, the security is conjectured, based on the study of all the best known attacks (which themselves consist of solving a variant of SVP).

To construct the security reductions, two main problems serve as the foundation of numerous lattice-based cryptographic protocols. The first one, introduced by Ajtai in 1996 [Ajt96], is the *Small Integer Solution problem* (SIS): For parameters n, m and q positive integers, the problem is to find a short non-zero solution $\mathbf{z} \in \mathbb{Z}^m$ to the homogeneous linear system $\mathbf{z}^T\mathbf{A} = \mathbf{0} \bmod q$ for uniformly random $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$. The SIS problem is used for example to construct signature schemes [GPV08, CHKP10, Boy10] and hash functions [Ajt96]. The second one, introduced by Regev in 2005 [Reg05], is the *Learning With Errors problem* (LWE). The search version of LWE is as follows: For parameters n and q positive integers and χ a probability density function on $\mathbb{Z}^n$, the problem is to find $\mathbf{s}$, given arbitrarily many independent pairs $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$ for a vector $\mathbf{a} \in \mathbb{Z}_q^n$ chosen uniformly at random, and e sampled from χ. The decision counterpart of LWE consists in distinguishing between arbitrarily many independent pairs $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$ sampled as in the search version and the same number of uniformly random and independent pairs. The LWE problem is used to construct encryption schemes and variants with advanced functionalities [Reg05, GPV08, BV11, GVW13]. For all those schemes, the security is proven by providing a reduction from SIS or LWE to the possible attacks. The hardness of those two problems is then essential in lattice-based cryptography. To show this hardness, we use particular reductions called "worst-case to average-case" reductions. An average-case problem (as SIS or LWE) is shown to be at least as hard as the arbitrary instances of a variant of SVP (the worst-case problem) which is presumed difficult. Note here that a worst-case problem needs every instance to be solved (e.g., with non-negligible probability over the internal randomness of the algorithm), whereas an average-case problem only requires some instances (a non-negligible proportion) to be solved. Ajtai [Ajt96] proposed the first worst-case to average-case reduction for a lattice problem, by providing a reduction from SIVP_γ to SIS. Later, Regev [Reg05, Reg09] showed the hardness

of the LWE problem by describing a (quantum) reduction from SIVP_γ to LWE. Cryptographic protocols relying on SIS or LWE therefore enjoy the property of being provably as secure as a worst-case problem which is strongly suspected of being extremely hard. However, on the other hand, the cryptographic applications of SIS and LWE are inherently inefficient due to the size of the associated key (or public data), which typically consists of the matrix $\mathbf{A}$.

To circumvent this inherent inefficiency, Micciancio [Mic02a, Mic07] — inspired from the efficient NTRU encryption scheme [HPS98] that can itself be interpreted in terms of lattices — initiated an approach that consists in changing the SIS and LWE problems to variants involving structured matrices. In these variants, the random matrix $\mathbf{A}$ is replaced by one with a specific block-Toeplitz structure, thus allowing for more compact keys and more efficient algorithms. The problem considered by Micciancio in [Mic07] was later replaced by a more powerful variant [LM06, PR06], now commonly referred to as Small Integer Solution problem over Rings, or R-SIS (it was initially called Ideal-SIS). A similar adaptation for LWE, called R-LWE, was introduced by Lyubashevsky et al. [LPR10] (see also [SSTX09]). Similarly to SIS and LWE, these problems admit reductions from worst-case lattice problems [LM06, PR06, LPR10], but, however, the corresponding worst-case problem is now SIVP_γ restricted to ideal lattices (which correspond to ideals of the ring of integers of a number field corresponding to the specific matrix structure). The latter problem is denoted Id-SIVP_γ.

Contributions

During my three years of PhD, I worked on two main aspects of cryptography: the security foundations, by studying the hardness of LWE and variants of SIS and LWE, and the primitive constructions, in particular group signatures and cryptographic multilinear maps.

Security foundations

The hardness of the SIS and LWE problems is fundamental in lattice-based cryptography as most of the recent schemes are based on them. In 2005, Regev provided a quantum worst-case to average-case reduction from a standard lattice problem to prove the hardness of LWE. This reduction uses quantum computations in addition to classical ones. In the unfortunate event that we find in the future an efficient quantum algorithm to solve lattice problems, this reduction will reduce LWE to an easy problem, but it could still be classically hard. In 2009, Peikert [Pei09] provided the first classical reduction to show the hardness of LWE, but his reduction only worked for an exponential modulus q, and most of the cryptographic applications are instantiated with a polynomial modulus. In a joint work with Zvika Brakerski, Chris Peikert, Oded Regev and Damien Stehlé, I showed that LWE (in dimension n) is at least as hard as standard worst-case lattice problems (in dimension $\simeq \sqrt{n}$), even with polynomial modulus q. We also showed that the hardness of LWE is a function of $n \log q$.

- [BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev and Damien Stehlé. Classical Hardness of Learning with Errors. *STOC*, pages 575-584. ACM, 2013.

The efficiencies of the cryptographic schemes can be drastically improved by switching the hardness assumptions to the more compact R-SIS [PR06, LM06] and R-LWE [LPR10] problems (over rings). However, this change of hardness assumptions comes along with a possible security weakening: SIS and LWE are known to be at least as hard as standard (worst-case) problems on euclidean lattices, whereas R-SIS and R-LWE are only known to be at least as as hard as their restrictions to special classes of ideal lattices. In an other work with Damien Stehlé, we studied

the hardness of two variants: the M-SIS and M-LWE problems (over modules), which bridge SIS with R-SIS, and LWE with R-LWE, respectively. We proved that these average-case problems are at least as hard as standard lattice problems restricted to module lattices (which themselves bridge arbitrary and ideal lattices). We also generalized one of the reductions of the work on LWE to show that the R-LWE problem is hard independently of the arithmetic shape of the modulus q. Previously, this problem was only shown to be hard for some specific moduli.

- [LS] Adeline Langlois and Damien Stehlé. Worst-case to Average-case Reductions for Module Lattices. Accepted to *Designs, Codes and Cryptography*. Springer, 2014.

Cryptographic constructions

My second research topic in lattice-based cryptography was the design of lattice-based group signatures. Group signatures are cryptographic primitives where users can anonymously sign messages in the name of a population they belong to. In 2010, Gordon et al. [GKV10] suggested the first realization of group signatures based on lattice assumptions in the random oracle model. A significant drawback of their scheme is its linear signature size in the cardinality N of the group. In a work with Fabien Laguillaumie, Benoît Libert and Damien Stehlé, we proposed the first lattice-based group signature schemes where the signature and public key sizes are essentially logarithmic in N (for any fixed security level). The security of our scheme is proved in the random oracle model under the SIS and LWE assumptions. In an other work with San Ling, Khoa Nguyen and Huaxion Wang, we introduced the first-lattice based group signature also with logarithmic signature size but enjoying another functionality, verifier local revocation (VLR). In the random oracle model, this scheme is proved to be secure based on the hardness of the SIS problem. Support of membership revocation is a desirable functionality for any group signature scheme, and prior to our work, all the VLR group signatures operated in the bilinear map setting.

- [LLLS13] Fabien Laguillaumie, Adeline Langlois, Benoît Libert and Damien Stehlé. Lattice-based Group Signature with Logarithmic Signature Size. *ASIACRYPT (2)*, volume 8270 of LNCS, pages 41-61. Springer, 2013.

- [LLNW14] Adeline Langlois, San Ling, Khoa Nguyen and Huaxiong Wang. Lattice-based Group Signature with Verifier Local Revocation. *Public Key Cryptography*, volume 8383 of LNCS, pages 345-361. Springer, 2014.

Finally, in a work with Damien Stehlé and Ron Steinfeld, we studied the GGH Graded Encoding Scheme introduced by Garg, Gentry and Halevi [GGH13a] in 2013. The GGH scheme, based on ideal lattices, is the first plausible approximation to a cryptographic multilinear map. Using the security analysis the authors provided, the scheme requires very large parameters to provide security for its underlying "encoding re-randomization" process. This process is important in the scheme as it is used each time an element is encoded, it avoids the correlation between the encoded element and the element used to construct it. The main contributions of our work were to formalize, simplify and improve the security analysis of the re-randomization process in the GGH construction. My co-authors and I applied these results in a new construction called GGHLite that enjoys improved efficiency. The first improvement is obtained by using the Rényi divergence instead of the conventional statistical distance as a measure of distance between distributions in the security reduction. The second improvement is to reduce the number of randomizers needed in the scheme. These two contributions allows to decrease the bit size of the public parameters from $O(\lambda^5 \log \lambda)$ for the GGH scheme to $O(\lambda \log^2 \lambda)$ in GGHLite, with respect to the security parameter λ (for a constant multilinearity parameter).

- [LSS14] Adeline Langlois, Damien Stehlé and Ron Steinfeld. GGHLite: More Efficient Multilinear Maps from Ideal Lattices. *EUROCRYPT 2014*, volume 8441 of LNCS, pages 239-256. Springer, 2014.

Other contribution

During my PhD, I also wrote a book chapter in French with Fabien Laguillaumie and Damien Stehlé, entitled *Chiffrement avancé à partir du problème Learning with Errors* (Advanced encryption from the Learning with Errors problem). This chapter introduces the LWE problem and several advances variants of encryption scheme. We describe in particular Regev and Dual-Regev encryption schemes [Reg05, GPV08], the identity-based encryption of [CHKP10] and the attribute-based encryption of [GVW13].

- [LLS14] Fabien Laguillaumie, Adeline Langlois et Damien Stehlé. Chiffrement avancé à partir du problème Learning With Errors. Chapitre du livre *Informatique Mathématique une photographie en 2014*, éditeur Sylvain Peyronnet, pages 179-225. Presses Universitaires de Perpignan, 2014.

Notations

$\mathbb{Z}$	the ring of integers
$\mathbb{R}$	the field of real numbers, $\mathbb{R}^+$ for the positive ones
$\mathbb{Z}_q$	the ring of integers modulo q, for an integer q
$[\cdot]_q$	(or mod q) corresponding operations are performed modulo q
K	a cyclotomic number field (for example the polynomial ring $\mathbb{Q}[X]/(x^n+1)$ for $n=2^k$)
R	the ring of integers of K (for example $\mathbb{Z}[X]/(x^n+1)$ for $n=2^k$)
R_q	the ring R/qR, for an integer q
(S)	the ideal of R generated by the set S
$\|g\|$	the norm of the coefficient vectors of $g \in K$
$[\cdot]_g$	all operations are performed modulo g, for $g \in R$
$\mathrm{MSB}_\ell(z)$	$\in \{0,1\}^{\ell \cdot n}$, the ℓ most significant bits of each of the n coefficients of $z \in R$, concatenated in a single bitstring
$\mathbb{T}$	the torus $\mathbb{R}/\mathbb{Z}$, i.e., the additive group of reals modulo 1
$\mathbb{T}_q$	its cyclic subgroup of order q, i.e., the subgroup given by $\{0, 1/q, \ldots, (q-1)/q\}$
$\mathbf{x}$	column vectors are denoted in bold
x_i	is the ith coordinate of $\mathbf{x}$
$(\mathbf{x}_1 \| \mathbf{x}_2)$	denotes the column concatenation of two vectors, i.e., $(\mathbf{x}_1 \| \mathbf{x}_2) = (\mathbf{x}_1^T \| \mathbf{x}_2^T)^T$
$\mathbf{e}_i$	the vector with 1 in its ith coordinate and 0 in all its other coordinates
$\langle \mathbf{x}, \mathbf{y} \rangle$	is the scalar product between two vectors $\mathbf{x}$ and $\mathbf{y}$
$\|\mathbf{x}\|_p$	the ℓ_p norm of the vector $\mathbf{x}$
$\|\mathbf{x}\|$	the Euclidean norm of the vector $\mathbf{x}$
$\mathbf{X}$	matrices are denoted in bold
$(\mathbf{X}_1 \| \mathbf{X}_2)$	denotes the concatenation of two matrices with the same number of rows
$U_{\mathbf{X}}$	$= \{\|\mathbf{X}\mathbf{u}\| : \mathbf{u} \in \mathbb{R}^n, \|\mathbf{u}\| = 1\}$ for a rank-n matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$
$\sigma_n(\mathbf{X})$	$= \inf(U_{\mathbf{X}})$ the smallest singular value of $\mathbf{X}$
$\sigma_1(\mathbf{X})$	$= \sup(U_{\mathbf{X}})$ the largest singular value of $\mathbf{X}$
$\|\mathbf{Y}\|$	$= \max_i \|\mathbf{y}_i\|$ for a tuple of vectors $\mathbf{Y} = (\mathbf{y}_i)_i$
$\widetilde{\mathbf{B}}$	the Gram-Schmidt orthogonalisation of a full column rank matrix $\mathbf{B}$

We use standard Landau notations $o(\cdot), O(\cdot), \omega(\cdot), \Omega(\cdot)$.

$\widetilde{O}(\cdot)$	is used to hide the poly-logarithmic factors, i.e., $f(n) = \widetilde{O}(g(n)) = O(g(n)\log^c(n))$ for some constant c
$\text{poly}(n)$	a function is $\text{poly}(n)$ if it is bounded by a polynomial in n
$\omega(f(n))$	the set of functions growing faster than $c \cdot f(n)$ for any constant c
$\text{negl}(n)$	a function f is negligible if $f(n) = n^{-\omega(1)}$, i.e., it decreases faster than the inverse of any polynomial function
$1 - \text{negl}(n)$	a function is overwhelming if it is $1 - \text{negl}(n)$
$2^{-\Omega(n)}$	such a function is said exponentially small in n
$f(E)$	$= \sum_{x \in E} f(x)$ for a function f over a countable domain E
$\Delta(X, Y)$	the statistical distance between the two statistical distributions X and Y
$R(X\|Y)$	the Rényi divergence between the two statistical distributions X and Y
$U(E)$	the uniform distribution over a finite set E
$D_{s,\mathbf{c}}$	the continuous Gaussian distribution of parameter s and center $\mathbf{c}$
$D_{\Lambda,s,\mathbf{c}}$	the discrete Gaussian distribution of support Λ, parameter s and center $\mathbf{c}$
$D_s, D_{\Lambda,s}$	we omit the center if $\mathbf{c} = 0$
$\hookleftarrow$	$x \hookleftarrow D$ means that the element x is sampled from the distribution D
log	the decimal logarithm function
ln	the binary logarithm function
1^λ	to give a parameter λ as input in unary of an algorithm
PPT	Probabilistic Polynomial Time
ROM	Random Oracle Model

Part One

An Introduction to Lattice-based Cryptography

In this first part of the thesis, we introduce lattice-based cryptography more in details. We first give preliminaries in Chapter 1: we recall statistical notions used in cryptography as the statistical distance, the Rényi divergence and the leftover hash lemma. We give some reminders about algebraic number theory that we use in the Ring/Module versions of SIS and LWE. And we introduce lattices, computational problems on lattices and Gaussian distributions (also on lattices). In Chapter 2, we define formally the SIS and LWE problems and recall the existing hardness results concerning those problems. Finally in Chapter 3, we describe cryptographic primitives based on SIS and LWE. In particular we describe two encryption schemes and three signature schemes that we will use further during the description of our group signatures. We also recall in this chapter the notion of trapdoor for lattices.

Chapter 1

Preliminaries

In this chapter, we recall the preliminaries needed for this thesis. We start with some statistical notions and with the Leftover Hash Lemma. In Section 1.2, we recall a few facts on algebraic number theory in the special case we use in Chapters 5 and 10. In Section 1.3, we introduce the notion of lattice, and the computational problems used in lattice-based cryptography. Finally, in Section 1.4, we define continuous and discrete Gaussian distributions, we give a a result of an exact Gaussian sample published in [BLP+13] and then, we recall a selection of properties on Gaussian.

1.1 Statistical notions

We recall several notions of closeness between two distributions.

1.1.1 Entropy

The Shannon Entropy is a measure of the quantity of information contained in a random variable.

Definition 1.1 (Shannon Entropy)**.** Let P be a distribution over a common countable set D. The Shannon Entropy H of P is defined as:

$$H(P) = -\sum_{d \in D} P(d) \log P(d).$$

The min-entropy measures the entropy of the best possible value of the random variable.

Definition 1.2 (Min-entropy)**.** Let P be a distribution over a common countable set D. The min-entropy H_∞ of P is defined as:

$$H_\infty(P) = -\max_{d \in D} \log P(d).$$

1.1.2 Statistical distance

For two probability distributions P, Q over some discrete domain, we define their statistical distance as:

$$\Delta(P, Q) = \frac{1}{2} \sum_{d \in D} |P(d) - Q(d)|,$$

and extend this to continuous distributions in the obvious way. We say that two sequences $(P_n)_n$, $(Q_n)_n$ of distributions indexed by a variable n are *negligibly close* if $\Delta(P_n, Q_n)$ is negligible in n.

We recall the following fact (see, e.g., [AD87, Eq. (2.3)] for a proof).

Lemma 1.3. *If P and Q are two probability distributions such that $P(d) \geq (1-\varepsilon)Q(d)$ holds for all d, then the statistical distance between P and Q is at most ε.*

1.1.3 Indistinguishability

A *distinguishing problem* $\mathcal{P}$ is defined by two distributions P_0 and P_1, and a solution to the problem is the ability to distinguish between these distributions. The *advantage* on $\mathcal{P}$ of an algorithm $\mathcal{A}$ with binary output is defined as

$$Adv[\mathcal{A}] = |\Pr[\mathcal{A}(P_0) = 1] - \Pr[\mathcal{A}(P_1) = 1]| \ .$$

If there is no algorithm $\mathcal{A}$ such that $Adv[\mathcal{A}]$ is non-negligible in the dimension of the problem, then we say that the two distributions P_0 and P_1 are *computationally indistinguishable*. If the statistical distance $\Delta(P_0, P_1)$ is a function negligible in the dimension, then we say that the two distributions P_0 and P_1 are *statistically indistinguishable*. Note that the statistical indistinguishability is stronger than the computational one.

A reduction from a problem $\mathcal{P}$ to a problem $\mathcal{Q}$ is an efficient (i.e., polynomial-time) algorithm $\mathcal{A}^{\mathcal{O}}$ that solves $\mathcal{P}$ given access to an oracle $\mathcal{O}$ that solves $\mathcal{Q}$. Most of the reductions that we will consider are what we call "transformation reductions:" these reductions perform some transformation to the input and then apply the oracle to the output of the transformation.

1.1.4 Rényi divergence

Another way to measure closeness between distributions is the Rényi divergence. For convenience, our definition of the Rényi divergence [Rén61, EH12] is the exponential of the usual definition used in information theory [EH12], and coincides with a discrete version of the quantity R defined for continuous density functions in [LPR13, Claim 5.11].

For any two continuous probability density functions $P, Q : \mathbb{R}^n \to \mathbb{R}^{\geq 0}$ such that $\mathrm{Supp}(P) \subset \mathrm{Supp}(Q)$, we define the RD of order $\alpha \neq 1$ by:

$$R(P\|Q) = \int_{\mathbb{R}^n} \frac{P(\mathbf{x})^2}{Q(\mathbf{x})} d\mathbf{x}.$$

For any two discrete probability distributions P and Q such that $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$ over a domain X and $\alpha > 1$, we define the Rényi Divergence of orders α and ∞ by

$$R_\alpha(P\|Q) = \left(\sum_{x \in X} \frac{P(x)^\alpha}{Q(x)^{\alpha-1}} \right)^{\frac{1}{\alpha-1}} \quad \text{and} \quad R_\infty(P\|Q) = \max_{x \in X} \frac{P(x)}{Q(x)},$$

with the convention that the fraction is zero when both numerator and denominator are zero. A convenient choice for computations (as also used in [LPR13]) is $\alpha = 2$, in which case we omit α. Note that $R_\alpha(P\|Q)^{\alpha-1} = \sum_x P(x) \cdot (P(x)/Q(x))^{\alpha-1} \leq R_\infty(P\|Q)^{\alpha-1}$. We list several properties of the Rényi divergence that can be considered the multiplicative analogues of those of the statistical distance.

Lemma 1.4. *Let P_1, P_2, P_3 and Q_1, Q_2, Q_3 denote discrete distributions on a domain X and let $\alpha \in (1, \infty]$. Then the following properties hold:*

- ***Log. Positivity:*** $R_\alpha(P_1\|Q_1) \geq R_\alpha(P_1\|P_1) = 1$.
- ***Data Processing Inequality:*** $R_\alpha(P_1^f\|Q_1^f) \leq R_\alpha(P_1\|Q_1)$ *for any function* f*, where* P_1^f *(resp.* Q_1^f*) denotes the distribution of* $f(y)$ *induced by sampling* $y \hookleftarrow P_1$ *(resp.* $y \hookleftarrow Q_1$*).*
- ***Multiplicativity:*** *Let* P *and* Q *denote any two distributions of a pair of random variables* (Y_1, Y_2) *on* $X \times X$*. For* $i \in \{1, 2\}$*, assume* P_i *(resp.* Q_i*) is the marginal distribution of* Y_i *under* P *(resp.* Q*), and let* $P_{2|1}(\cdot|y_1)$ *(resp.* $Q_{2|1}(\cdot|y_1)$*) denote the conditional distribution of* Y_2 *given that* $Y_1 = y_1$*. Then we have:*
  - $R_\alpha(P\|Q) = R_\alpha(P_1\|Q_1) \cdot R_\alpha(P_2\|Q_2)$ *if* Y_1 *and* Y_2 *are independent.*
  - $R_\alpha(P\|Q) \leq R_\infty(P_1\|Q_1) \cdot \max_{y_1 \in X} R_\alpha(P_{2|1}(\cdot|y_1)\|Q_{2|1}(\cdot|y_1))$.
- ***Weak Triangle Inequality:*** *We have:*

$$R_\alpha(P_1\|P_3) \leq \begin{cases} R_\alpha(P_1\|P_2) \cdot R_\infty(P_2\|P_3), \\ R_\infty(P_1\|P_2)^{\frac{\alpha}{\alpha-1}} \cdot R_\alpha(P_2\|P_3). \end{cases}$$

- R_∞ ***Triangle Inequality:*** *If* $R_\infty(P_1\|P_2)$ *and* $R_\infty(P_2\|P_3)$ *are defined, then*

$$R_\infty(P_1\|P_3) \leq R_\infty(P_1\|P_2) \cdot R_\infty(P_2\|P_3).$$

- ***Probability Preservation:*** *Let* $A \subseteq X$ *be an arbitrary event. Then*

$$Q_1(A) \geq P_1(A)^{\frac{\alpha}{\alpha-1}} / R_\alpha(P_1\|Q_1).$$

Proof. The log. positivity and data processing inequalities are proved in [EH12, Th. 8, Th. 9].

For multiplicativity, we have

$$R_\alpha(P\|Q)^{\alpha-1} = \sum_{x_1,x_2} \frac{(P_1(x_1) \cdot P_{2|1}(x_2|x_1))^\alpha}{(Q_1(x_1) \cdot Q_{2|1}(x_2|x_1))^{\alpha-1}} = \sum_{x_1} \frac{P_1(x_1)^\alpha}{Q_1(x)^{\alpha-1}} \cdot R_\alpha(P_{2|1}(\cdot|x_1)\|Q_{2|1}(\cdot|x_1))^{\alpha-1}.$$

If X_1 and X_2 are independent, we have $P_{2|1}(x_2|x_1) = P_2(x_2)$ and $Q_{2|1}(x_2|x_1) = Q_2(x_2)$ for all x_1, and the result follows. More generally, since $R_\alpha(P\|Q)^{\alpha-1}$ is the expected value of $f(x_1) = \frac{P_1(x_1)^{\alpha-1}}{Q_1(x)^{\alpha-1}} \cdot R_\alpha(P_{2|1}(\cdot|x_1)\|Q_{2|1}(\cdot|x_1))^{\alpha-1}$ over x_1 sampled from P_1, it follows that $R_\alpha(P\|Q)^{\alpha-1} \leq \max_{x_1} f(x_1)$, which gives the second multiplicativity property.

For the first weak triangle inequality, we have

$$R_\alpha(P_1\|P_3)^{\alpha-1} = \sum_x \frac{P_1(x)^\alpha}{P_3(x)^{\alpha-1}} = \sum_x \frac{P_1(x)^\alpha}{P_2(x)^{\alpha-1}} \cdot \frac{P_2(x)^{\alpha-1}}{P_3(x)^{\alpha-1}} \leq \left(\sum_x \frac{P_1(x)^\alpha}{P_2(x)^{\alpha-1}}\right) \cdot \max_x \frac{P_2(x)^{\alpha-1}}{P_3(x)^{\alpha-1}},$$

which gives the desired result. Similarly, for the second weak triangle inequality,

$$R_\alpha(P_1\|P_3)^{\alpha-1} = \sum_x \frac{P_1(x)^\alpha}{P_3(x)^{\alpha-1}} = \sum_x \frac{P_1(x)^\alpha}{P_2(x)^\alpha} \cdot \frac{P_2(x)^\alpha}{P_3(x)^{\alpha-1}} \leq \left(\max_x \frac{P_1(x)^\alpha}{P_2(x)^\alpha}\right) \cdot \sum_x \frac{P_2(x)^\alpha}{P_3(x)^{\alpha-1}},$$

as required. For the R_∞ triangle inequality, we have

$$R_\infty(P_1\|P_3) = \max_x \frac{P_1(x)}{P_3(x)} = \max_x \frac{P_1(x)}{P_2(x)} \cdot \frac{P_2(x)}{P_3(x)} \leq \left(\max_x \frac{P_1(x)}{P_2(x)}\right) \cdot \max_x \frac{P_2(x)}{P_3(x)}.$$

Finally, the probability preservation property is proved in [LPR13, Claim 5.11] for the case $\alpha = 2$ using the Cauchy-Schwarz inequality. The general case follows by replacing the latter with the more general Holder inequality, which states that $\sum_{x\in A} |f(x)g(x)| \leq (\sum_{x\in A} f(x)^p)^{1/p} \cdot (\sum_{x\in A} g(x)^{1/(1-1/p)})^{1-1/p}$ for real-valued functions f, g and $p \geq 1$. Taking $f(x) = \frac{P_1(x)}{Q_1(x)^{1-1/\alpha}}$, $g(x) = Q_1(x)^{1-1/\alpha}$, and $p = \alpha$, we get $P_1(A) \leq (\sum_{x\in A} \frac{P(x)^\alpha}{Q(x)^{\alpha-1}})^{1/\alpha} \cdot Q(A)^{1-1/\alpha}$, and using $\sum_{x\in A} \frac{P(x)^\alpha}{Q(x)^{\alpha-1}} \leq R_\alpha(P_1\|Q_1)^{\alpha-1}$ provides the result. □

We note that the Rényi divergence does not satisfy the (multiplicative) triangle inequality $R(P_1\|P_3) \leq R(P_1\|P_2) \cdot R(P_2\|P_3)$ in general (see [EH12]), but a weaker inequality holds if one of the pairs of distributions has a bounded R_∞ divergence, as shown above. We also observe that R_∞ *does* satisfy the triangle inequality.

1.1.5 Leftover Hash Lemma

The Leftover Hash Lemma [HILL99] is a very classic cryptographic tool. We recall here some particular cases that we will use in this thesis.

Lemma 1.5. *Let $m, n, q \geq 1$ be integers such that $m \geq 4n \log q$ and q prime, and let $\boldsymbol{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and $\boldsymbol{r} \hookleftarrow U(\{0,1\}^m)$. Then the pair $(\boldsymbol{A}, \boldsymbol{r}^T\boldsymbol{A})$ is within statistical distance $\leq 2^{-n}$ from the uniform distribution on $\mathbb{Z}_q^{m\times n} \times mZ_q^n$.*

We recall the following lemma which is an immediate corollary of the leftover hash lemma.

Lemma 1.6. *Let $m, n, q \geq 1$ be integers, and $\epsilon > 0$ be such that $m \geq n \ln q + 2\ln(1/\varepsilon)$. For $\boldsymbol{H} \hookleftarrow U(\mathbb{T}_q^{m\times n})$, $\boldsymbol{z} \hookleftarrow U(\{0,1\}^m)$, $\boldsymbol{u} \hookleftarrow U(\mathbb{T}_q^n)$, the distributions of $(\boldsymbol{H}, \boldsymbol{z}^T\boldsymbol{H})$ and $(\boldsymbol{H}, \boldsymbol{u})$ are within statistical distance at most ε.*

Here $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ denote the torus, i.e., the additive group of reals modulo 1.

1.2 Algebraic number theory

In the following, we recall a few facts on elementary algebraic theory in this special case we use in Chapter 5, 9 and 10. We refer the reader to [Mol99] and [LPR10, LPR13] for thorough introductions to the topics covered in this section.

1.2.1 Number fields and cyclotomic fields

Every complex root of a polynomial $g(X) \in \mathbb{Q}[X]$ is an *algebraic number*. The *minimal polynomial* of an algebraic number ξ is the unique irreducible monic polynomial f of minimal degree such that ξ is one of its roots. An *algebraic integer* is an algebraic number whose minimal polynomial belongs to $\mathbb{Z}[X]$. Let ξ be an algebraic number, the *number field* $K = \mathbb{Q}(\xi)$ is a finite extension of the rational number field $\mathbb{Q}$. It is also an n-dimensional vector space over $\mathbb{Q}$ with basis $\{1, \xi, \ldots, \xi^{n-1}\}$, where n is the degree of f. We call n the degree of K. Let R be the set of the algebraic integers belonging to K. This is a ring, called the ring of integers (or maximal order) of K. If ξ is an algebraic integer, then $\mathbb{Z}[\xi] = \sum_{j=1}^n \mathbb{Z} \cdot \xi^j \subseteq R$. In general, this inclusion can be strict.

A cyclotomic field is a field $K = \mathbb{Q}(\xi)$ where ξ is a root of unity. If ξ is a primitive ν-th root of unity, then it is a root of the ν-th cyclotomic polynomial Φ_ν. The degree $n = \phi(\nu)$ of Φ_ν is the degree of K (here $\phi(\cdot)$ denotes Euler's totient function). In the case of cyclotomic fields, we have $R = \mathbb{Z}[\xi]$.

In this work, all number fields will be cyclotomic fields.

1.2.2 Complex embeddings

The canonical embeddings are the n ring homomorphisms $\sigma_j : K \to \mathbb{C}$ that fix every element of $\mathbb{Q}$. In our particular case of cyclotomic fields, all n embeddings are complex: They are defined by $\sigma_j : \xi \mapsto \xi^j$ for any $j \in \mathbb{Z}_\nu^\times$. Note that if j is invertible modulo ν, then so is $\nu - j$, and $\sigma_{\nu-j} = \overline{\sigma_j}$. For notational simplicity, we let $\mathbb{J}$ denote $[\nu/2] \cap \mathbb{Z}_\nu^\times$. We call *canonical embedding vector* the ring homomorphism $\sigma_C : K \to \mathbb{C}^n$ defined as: $\sigma_C(y) = (\sigma_j(y))_{j \in \mathbb{Z}_\nu^\times}$, where addition and multiplication in $\mathbb{C}^n$ are component-wise. Indeed, for any $x, y \in K$, we have that $\sigma_C(x \cdot y)$ is the component-wise product of $\sigma_C(x)$ and $\sigma_C(y)$. By elementary linear algebra, we observe that an element of K is fully specified by its canonical embedding vector.

The *trace* $\mathrm{Tr} : K \to \mathbb{Q}$ and the (algebraic) *norm* $\mathrm{N} : K \to \mathbb{Q}$ are defined as follows: $\mathrm{Tr}(x) = \sum_{j \in \mathbb{Z}_\nu^\times} \sigma_j(x)$ and $\mathrm{N}(x) = \prod_{j \in \mathbb{Z}_\nu^\times} \sigma_j(x)$. For any $x, y \in K$ we have $\mathrm{Tr}(x \cdot y) = \sum_{j \in \mathbb{Z}_\nu^\times} \sigma_j(x) \cdot \sigma_j(y) = \langle \sigma_C(x), \overline{\sigma_C(y)} \rangle$ where $\langle \cdot, \cdot \rangle$ is the canonical Hermitian product on $\mathbb{C}^n$.

1.2.3 Space H

We use the following subspace of $\mathbb{C}^n$, as in [LPR10]:

$$H = \{(x_j)_{j \in \mathbb{Z}_\nu^\times} \in \mathbb{C}^n : \forall j \in \mathbb{J}, x_{\nu-j} = \overline{x_j}\}.$$

Let $\mathbf{h}_j = \frac{1}{\sqrt{2}}(\mathbf{e}_j + \mathbf{e}_{\nu-j})$ and $\mathbf{h}_{\nu-j} = \frac{i}{\sqrt{2}}(\mathbf{e}_j - \mathbf{e}_{\nu-j})$ for $j \in \mathbb{J}$. The $\mathbf{h}_j$'s form a basis of H as a real vector space. An element $x \in K$ can be represented according to the basis $(\mathbf{h}_j)_j$: For $x \in K$, we define $\sigma_H(x)$ by $\sigma_H(x) = (x_j)_j \in \mathbb{R}^n$ such that $\sigma_C(x) = \sum_j x_j \cdot \mathbf{h}_j$. As $\sigma_C(x) = (\sigma_j(x))_j$, we have, for $j \in \mathbb{J}$:

$$\begin{bmatrix} x_j \\ x_{\nu-j} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix} \begin{bmatrix} \sigma_j(x) \\ \sigma_{\nu-j}(x) \end{bmatrix} \text{ and } \begin{bmatrix} \sigma_j(x) \\ \sigma_{\nu-j}(x) \end{bmatrix} = \begin{bmatrix} 1 & i \\ 1 & -i \end{bmatrix} \begin{bmatrix} x_j \\ x_{\nu-j} \end{bmatrix}.$$

The addition in H is component wise. Let $\sigma_H(x) = (x_j)_j$ and $\sigma_H(y) = (y_j)_j$, the multiplication is given by $\sigma_H(x \cdot y) = (z_j)_j$ where, for $j \in \mathbb{J}$:

$$\begin{bmatrix} z_j \\ z_{\nu-j} \end{bmatrix} = \begin{bmatrix} x_j & -x_{\nu-j} \\ x_{\nu-j} & x_j \end{bmatrix} \begin{bmatrix} y_j \\ y_{\nu-j} \end{bmatrix} \text{ or } \begin{bmatrix} z_j \\ z_{\nu-j} \end{bmatrix} = \begin{bmatrix} y_j & -y_{\nu-j} \\ y_{\nu-j} & y_j \end{bmatrix} \begin{bmatrix} x_j \\ x_{\nu-j} \end{bmatrix}.$$

To ease the presentation, in Chapter 5 we identify elements of K with their σ_H embeddings.

1.2.4 Ideals

An (integral) *ideal* I of R is a non-zero additive subgroup of R that is closed under multiplication by every element of R. The smallest ideal of R containing the set S is denoted by (S). The quotient R/I is the set of the equivalence classes $g + I$ of R modulo I. For any nonzero ideal, the *norm* $\mathcal{N}(I)$ of the ideal is the number of elements of the quotient ring R/I. We have $\mathcal{N}((x)) = \mathcal{N}(x)$, for all $x \in K$.

Let I and J be ideals of R. We define the *product* of two ideals by $IJ = \{\sum_i \alpha_i \beta_i : \alpha_i \in I, \beta_i \in J\}$ and their *sum* by $I + J = \{\alpha + \beta : \alpha \in I, \beta \in J\}$. An ideal $I \subsetneq R$ is *prime* if for any $ab \in I$ then $a \in I$ or $b \in I$. Every ideal of R can be represented as a unique product of prime ideals, and for a prime ideal I, the quotient ring R/I is the finite field of order $\mathcal{N}(I)$. A *fractional ideal* $I \subseteq K$ is a set such that $dI \subseteq R$ is an (integral) ideal for a nonzero $d \in R$. The *inverse* of

a fractional I is defined by $I^{-1} = \{\alpha \in K : \alpha I \subseteq R\}$ and is itself a fractional ideal. We have $II^{-1} = R$. The *dual* of an ideal is defined as $I^\vee = \{x \in K : \mathrm{Tr}(xI) \subseteq \mathbb{Z}\}$. We have $I^\vee = I^{-1} \cdot R^\vee$.

In our setup of cyclotomic fields, if q is a prime integer, the prime ideal factorization of $(q) \subseteq R$ can be computed efficiently. In particular, if $q = 1 \bmod \nu$, then $(q) = \prod_{j \in \mathbb{Z}_\nu^\times} \mathfrak{q}_j$ where each $\mathfrak{q}_j$ if a prime ideal with norm $\mathcal{N}(\mathfrak{q}_j) = q$. The field K has n automorphisms $\tau_j : K \to K$ defined by $\tau_j(\xi) = \xi^j$ (for $j \in \mathbb{Z}_\nu^\times$). As noted in [LPR10, Le. 2.16], the automorphism group of the $\{\tau_j\}$ acts transitively on the set $\{\mathfrak{q}_j\}_j$.

1.2.5 An isomorphism of quotient rings

Lyubashevsky et al. [LPR10, Se. 2.3.9] used the Chinese Remainder Theorem to make explicit an isomorphism between I/qI and R/qR for an arbitrary positive integer q, which we recall now. Let R_q and $R_q^\vee$ respectively denote R/qR and $R^\vee/qR^\vee$.

Let $t \in I$ be such that $(t) + qI = I$ (such a t exists and can be found efficiently given I and the prime ideal factorization of (q), see [LPR10, Le. 2.14]). The function $\theta_I : K \to K$ defined as $\theta_I(x) = t \cdot x$ induces an isomorphism from R_q to I/qI. Moreover, this isomorphism may be efficiently inverted using $\theta_I^{-1} : I/qI \to R_q$ defined by $\theta_I^{-1}(y) = t^{-1} \cdot y' \bmod qR$ where $y' = y \bmod qI$ and $y' \in (t)$. The function θ_I also induces an isomorphism from $I^\vee/qI^\vee$ to $R_q^\vee$ that may be efficiently inverted using $\theta_I^{-1} : R_q^\vee \to I^\vee/qI^\vee$ with $\theta_I^{-1}(y) = t^{-1} \cdot y' \bmod qR$ where $y' = y \bmod qI^\vee$ and $y' \in (t)$.

1.2.6 Modules

A subset $M \subseteq K^d$ is an R-module if it is closed under addition and under multiplication by elements of R. It is a finitely generated module if there exists a finite family $(\mathbf{b}_k)_k$ of vectors in K^d such that $M = \sum_k R \cdot \mathbf{b}_k$. In general, if the ring R is arbitrary, an R-module may not have a basis. But here K is a number field, so R is a Dedekind domain, and we have the existence of so-called pseudo-bases (see, e.g., [Coh00, Ch. 1]): For every module M, there exist $(I_k)_{1 \le k \le d'}$ nonzero ideals of R and $(\mathbf{b}_k)_{1 \le k \le d'}$ linearly independent vectors of $K^{d'}$ such that $M = \sum_{1 \le k \le d'} I_k \cdot \mathbf{b}_k$. We say that $[(I_k)_k, (\mathbf{b}_k)_k]$ is a *pseudo-basis* of M. The word pseudo-basis is used as the coefficient ideals $(I_k)_k$ can be non-principal. The representation of the elements of M with respect to a pseudo-basis is unique. Two pseudo-bases can generate the same module and then, they have the same cardinality. The latter is called *rank* of the module. In this work, we will restrict ourselves to full-rank modules, i.e., with $d' = d$.

We define the dual of a module by $M^\vee = \{\mathbf{x} \in K^d, \forall \mathbf{y} \in M : \mathrm{Tr}(\langle \mathbf{x}, \overline{\mathbf{y}} \rangle) \in \mathbb{Z}\}$, where $\langle \cdot, \cdot \rangle$ is the Hermitian product on K^d. We have the following property:

Lemma 1.7. *If $M = \sum_{k=1}^d I_k \cdot \boldsymbol{b}_k$, then $M^\vee = \sum_{k=1}^d I_k^\vee \cdot \boldsymbol{b}_k^\vee$, where the $\boldsymbol{b}_\ell^\vee$'s are defined by $\langle \boldsymbol{b}_k, \overline{\boldsymbol{b}_\ell^\vee} \rangle = 1$ if $k = \ell$ and $\langle \boldsymbol{b}_k, \overline{\boldsymbol{b}_\ell^\vee} \rangle = 0$ otherwise.*

Proof. We first show that $\sum_{k=1}^d I_k^\vee \cdot \mathbf{b}_k^\vee \subseteq M^\vee$. Let $\mathbf{x} \in \sum_{k=1}^d I_k^\vee \cdot \mathbf{b}_k^\vee$. Then for each i there exists $x_k \in I_k^\vee$ such that $\mathbf{x} = \sum_{k=1}^d x_k \cdot \mathbf{b}_k^\vee$. Let $\mathbf{y} = \sum_{k=1}^d y_k \cdot \mathbf{b}_k \in M$. Then by linearity, we have $\mathrm{Tr}(\langle \mathbf{x}, \overline{\mathbf{y}} \rangle) = \sum_{k=1}^d \mathrm{Tr}(x_k y_k)$. For all i, we have $x_k \in I_k^\vee$ and $y_k \in I_k$, and thus $\mathrm{Tr}(x_k y_k) \in \mathbb{Z}$. Therefore, we have $\mathrm{Tr}(\langle \mathbf{x}, \overline{\mathbf{y}} \rangle) \in \mathbb{Z}$ and $\mathbf{x} \in M^\vee$.

We now show that $M^\vee \subseteq \sum_{k=1}^d I_k^\vee \cdot \mathbf{b}_k^\vee$. Let $\mathbf{x} \in M^\vee \subseteq K^d$. We can write $\mathbf{x} = \sum_{k=1}^d x_k \cdot \mathbf{b}_k^\vee$, for some x_k's in K. It suffices to show that $x_k \in I_k^\vee$. Let $y_k \in I_k$ be arbitrary. By linearity, we have $\mathrm{Tr}(\langle \mathbf{x}, \overline{y_k \mathbf{b}_k} \rangle) = \mathrm{Tr}(x_k y_k) \in \mathbb{Z}$. This implies that $x_k \in I_k^\vee$. □

We generalize the isomorphism θ_I defined above to modules. Let $M = \sum_{k=1}^d I_k \cdot \mathbf{b}_k$, $f : I_1/qI_1 \times \ldots \times I_d/qI_d \to M/qM$ be such that $f(x_1, \ldots, x_n) = \sum_{k=1}^d x_k \cdot \mathbf{b}_k$ and $g : M/qM \to$

$I_1/qI_1 \times \ldots \times I_d/qI_d$ be such that $g(\sum_{k=1}^{d} x_k \cdot \mathbf{b}_k) = (x_1, \ldots, x_n)$. The functions f and g are ring isomorphisms and $g = f^{-1}$. Let $\theta_{I_1}, \ldots, \theta_{I_d}$ be as described above. We define the functions Θ and Θ^{-1} as follows: $\Theta = f \circ (\theta_{I_1} \times \ldots \times \theta_{I_d})$ and $\Theta^{-1} = (\theta_{I_1}^{-1} \times \ldots \times \theta_{I_d}^{-1}) \circ g$. The function Θ induces an isomorphism from R_q^d to M/qM with inverse Θ^{-1}.

1.3 Lattices

In this section we recall the notion of Euclidean lattice and give some of their properties, we also introduce the particular case of ideal and module lattices.

1.3.1 Definition

A *euclidean lattice* $\Lambda \subseteq \mathbb{R}^n$ is the set of all integer linear combinations $\sum_{i=1}^{p} x_i \mathbf{b}_i$ of some linearly independent vectors $(\mathbf{b}_i)_{1 \leq i \leq p} \in \mathbb{R}^n$. We write $\mathcal{L}(\mathbf{B})$ for the lattice spanned by the basis $\mathbf{B} = (\mathbf{b}_i)_{i \leq p}$. We call p the dimension of the lattice. In this work, we will often restrict ourselves to full-rank lattices, i.e., with $p = n$ (except in Chapters 9 and 10).

The determinant $\det(\Lambda)$ is defined as $\sqrt{\det(\mathbf{B}^T\mathbf{B})}$, where $\mathbf{B} = (\mathbf{b}_i)_i$ is any such *basis* of Λ.

We recall that for a set $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset \mathbb{R}^n$ of linearly independent vectors, we denote by $\widetilde{\mathbf{S}}$ its Gram-Schmidt orthogonalization, in which $\|\widetilde{\mathbf{s}}_i\| \leq \|\mathbf{s}_i\|$ for all i. The following lemma from [MG02] states that for any full-rank set of vectors in a lattice, one can efficiently find a basis of this lattice, without increasing the norm of the Gram-Schmidt vectors.

Lemma 1.8 ([MG02, Lemma 7.1])**.** *There is a deterministic polynomial-time algorithm that, given an arbitrary basis $\boldsymbol{B}$ of an n-dimensional lattice $\mathcal{L}(\boldsymbol{B})$ and a full-rank set of lattice vectors $\boldsymbol{S} \subset \mathcal{L}(\boldsymbol{B})$, outputs a basis $\boldsymbol{T}$ of $\mathcal{L}(\boldsymbol{B})$ such that $\|\widetilde{\boldsymbol{t}}_i\| \leq \|\widetilde{\boldsymbol{s}}_i\|$ for all $1 \leq i \leq n$.*

The *minimum* $\lambda_1(\Lambda)$ of a lattice Λ is the norm of any of its shortest nonzero vectors. More generally, the *ith successive minimum* $\lambda_i(\Lambda)$ is the smallest radius r such that Λ contains i linearly independent vectors of norm at most r. The *dual lattice* of $\Lambda \subseteq \mathbb{R}^n$ is $\Lambda^* = \{\mathbf{x} \in \mathbb{R}^n : \forall \mathbf{y} \in \Lambda, \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}$. If $\Lambda = \mathcal{L}(\mathbf{B})$ then $\Lambda^* = \mathcal{L}(\mathbf{B}^*)$ with $\mathbf{B}^* = \mathbf{B}^{-T}$.

Lemma 1.9 (Minkowski's second theorem)**.** *Let Λ be an n-dimensional lattice. Then:*

$$\Big(\prod_{1 \leq n} \lambda_i(\Lambda)\Big)^{1/n} \leq \sqrt{n} \det(\Lambda)^{1/n}.$$

The following result links the determinants of a lattice and its orthogonal.

Lemma 1.10 ([NS97, Cor. 2])**.** *Let $\Lambda \subseteq \mathbb{Z}^n$ be a lattice, and let $\Lambda^\perp = (\mathrm{Span}(\Lambda))^\perp \bigcap \mathbb{Z}^n$ denote the orthogonal lattice of Λ. Then $\det(\Lambda^\perp) \leq \det(\Lambda)$.*

1.3.2 Ideal and module lattices

As σ_H is a group homomorphism from $(K, +)$ to $(\mathbb{R}^n, +)$ and I an ideal of R, the set $\sigma_H(I)$ is a lattice. We call it ideal lattice with respect to K. To ease the presentation, we often identify I and $\sigma_H(I)$.

We define module lattices similarly. The map $(\sigma_H, \ldots, \sigma_H)$ is an embedding from K^d to $\mathbb{R}^N$, with $N = nd$, and $M \subseteq K^d$ a module of R. By abuse of notation, we also call it σ_H. The set $\sigma_H(M)$ is a module lattice. Note that if M is a rank d module and if K has degree n, then the corresponding module lattice has dimension $N = nd$. For any $\mathbf{x} \in K^d$, we

define $\|\mathbf{x}\| = (\sum_{k=1}^{d} \sum_{j \in \mathbb{Z}_\nu^\times} |\sigma_j(x_k)|^2)^{1/2}$. We also define $\|\mathbf{x}\|_\infty = \max_{j,k} |\sigma_j(x_k)|$, $\|\mathbf{x}\|_{2,\infty} = \max_j (\sum_k |\sigma_j(x_k)|^2)^{1/2}$ and $\|\mathbf{x}\|_{\infty,2} = \max_k (\sum_j |\sigma_j(x_k)|^2)^{1/2}$.

When a module is given as input of a problem, we consider that we give a lattice basis of the corresponding module lattice. Note that it is equivalent to give a basis of the module lattice and a pseudo-basis of the module because from the first representation, the second representation is computable in polynomial time [BP91, Coh00]. All asymptotic statements involving modules (including hardness results) will be given for N growing to infinity.

1.3.3 Computational problems

Shortest and Closest Vector Problems. The Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) are the two fundamentals problems over lattices. Given an n-dimensional lattice, represented by one of its base, and a target vector $\mathbf{t} \in \mathbb{R}^n$ the goal of SVP is to find a shortest non-zero vector in this lattice. The goal of CVP is to find a closest vector of the lattice to the target vector. We consider the version of these problem with the euclidean norm (unless another norm is specified).

Definition 1.11 (Shortest Vector Problem and Closest Vector Problem)**.** We recall the definitions of the following problems:

SVP: Given a lattice basis $\mathbf{B}$, find a non-zero vector $\mathbf{s} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{s}\| = \lambda_1(\mathcal{L}(\mathbf{B}))$.

Approx SVP$_\gamma$: Let $\gamma \geq 1$ be a function of the dimension n. Given a lattice basis $\mathbf{B}$, find a non-zero vector $\mathbf{s} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{s}\| \leq \gamma \cdot \lambda_1(\mathcal{L}(\mathbf{B}))$.

CVP: Given a lattice basis $\mathbf{B}$, and a target vector $\mathbf{t} \in \mathbb{R}^n$, find a vector $\mathbf{s} \in \mathcal{L}(\mathbf{B})$ minimizing $\|\mathbf{s}-\mathbf{t}\|$.

Approx CVP$_\gamma$ Given a lattice basis $\mathbf{B}$, and a target vector $\mathbf{t} \in \mathbb{R}^n$, find a vector $\mathbf{s} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{s} - \mathbf{t}\| \leq \gamma \cdot \min_{\mathbf{x} \in \mathcal{L}(\mathbf{B})} \|\mathbf{x} - \mathbf{t}\|$.

As shown in [GMSS99], the problem CVP$_\gamma$ is at least as hard as the problem SVP$_\gamma$ for any approximation factor.

Definition 1.12 (Gap Shortest Vector Problem)**.** We recall the definition of the following problems:

GapSVP: Given a lattice basis $\mathbf{B}$ and a number d, decide if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$ or $\lambda_1(\mathcal{L}(\mathbf{B})) \geq d$.

GapSVP$_\gamma$: Given a lattice basis $\mathbf{B}$ and a number d, decide if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$ or $\lambda_1(\mathcal{L}(\mathbf{B})) \geq \gamma \cdot d$.

If $\lambda_1(\mathcal{L}(\mathbf{B})) \in (d, \gamma \cdot d)$, then the algorithm does not have to answer, or may answer something wrong.

SIVP and GIVP. The Shortest Independent Vectors problem (SIVP) is a generalization of SVP, where instead of finding one short vector, one has to find n linearly independent short vectors.

Definition 1.13 (Shortest Independent Vectors Problem)**.** Let $\gamma \geq 1$ be a function of the dimension n. The SIVP$_\gamma$ is as follows: Given a lattice basis $\mathbf{B}$, find $n = \dim(\mathcal{L}(\mathbf{B}))$ linearly independent vectors $\mathbf{s}_1, \ldots, \mathbf{s}_n \in \mathcal{L}(\mathbf{B})$ such that $\max_i \|\mathbf{s}_i\| \leq \gamma \cdot \lambda_n(\mathcal{L}(\mathbf{B}))$.

We consider the following generalization of SIVP.

Definition 1.14 (Generalized Independent Vectors Problem)**.** Let ϕ denote an arbitrary real-valued function of a lattice. Let $\gamma \geq 1$ be a function of the dimension n. The $\mathrm{GIVP}^{\phi}_{\gamma}$ is as follows: Given a lattice basis $\mathbf{B}$, find $n = \dim(\mathcal{L}(\mathbf{B}))$ linearly independent vectors $\mathbf{s}_1, \ldots, \mathbf{s}_n \in \mathcal{L}(\mathbf{B})$ such that $\max_i \|\mathbf{s}_i\| \leq \gamma \cdot \phi(\mathcal{L}(\mathbf{B}))$.

For $\phi = \lambda_n$, we recover SIVP_{γ}. We let Id-GIVP denote the restriction of GIVP to ideal lattices. Similarly, we let Mod-GIVP denote the restriction of GIVP to module lattices.

Complexity of lattice problems. We refer to [Reg10b] for more details about the complexity of solving the lattice problems. They are NP-hard for any approximation factor $\gamma \leq O(1)$ (classically for CVP, under a randomized reduction for SVP, see [vEB81, Mic98, BS99]). The currently best polynomial time algorithms [Sch87, GN08, HPS11] only solve those problems with an approximation factor $2^{n \log\log n / \log n}$. On the other hand, it has been proven [HR07] that there is no efficient algorithm that approximate lattice problems with an approximation factor $n^{c/\log\log n}$ for some constant $c > 0$, unless P=NP.

There is a gap between those two approximation factors, and the best known algorithms (even quantum) for an approximation to within any polynomial factor γ all have exponential complexities [MR09]. This motivates the following conjecture: There is no polynomial time (quantum) algorithm that approximates lattice problems to within a polynomial factor. It seems that we can even go further by saying that there is no sub-exponential time (quantum) algorithm that approximate those problems to within polynomial factor. Those conjectures are the foundation of lattice-based cryptography.

Intermediate problems and reductions. We now define intermediate problems on lattices, that we will use further in Chapters 2 and 5. We also recall the principle of the reduction from GIVP to these problems.

Definition 1.15. The *Incremental Independent Vectors Decoding* problem $\mathit{IncIVD}^{\phi}_{\gamma,g}$ is as follows: Given a tuple $(\mathbf{B}, \mathbf{S}, \mathbf{t})$, where $\mathbf{B}$ is a basis of a full-rank lattice in $\mathbb{R}^n$, $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$ is a full-rank set of lattice vectors such that $\|\mathbf{S}\| \geq \gamma \cdot \phi(\mathcal{L}(\mathbf{B}))$, and $\mathbf{t}$ is a target point. The goal is to find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \frac{\|\mathbf{S}\|}{g}$.

This problem is a variant of the Incremental Guaranteed Distance Decoding problem, defined in [MR07] as follows:

Definition 1.16. The *Incremental Guaranteed Distance Decoding* problem $\mathit{IncGDD}^{\phi}_{\gamma,g}$ is as follows: Given a tuple $(\mathbf{B}, \mathbf{S}, \mathbf{t}, r)$, where $\mathbf{B}$ is a basis of a full-rank lattice in $\mathbb{R}^n$, $\mathbf{S} \subset \mathcal{L}(\mathbf{B})$ is a full-rank set of lattice vectors, and r is a real such that $r > \gamma(n) \cdot \phi(\mathcal{L}(\mathbf{B}))$. The goal is to find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \frac{\|\mathbf{S}\|}{g} + r$.

The goal of those two problems is to find a lattice vector within a certain distance from a given target. In the two problems the distance is larger than $\frac{\gamma\phi(\mathcal{L}(\mathbf{B}))}{g}$, where ϕ is usually the smoothing parameter of the lattice or its n-th minima.

The following intermediate result is proven in [Mic07].

Theorem 1.17. *For any $\gamma > 1$ and any ϕ, there exists a reduction from $\mathrm{GIVP}^{\phi}_{8\gamma}$ to $\mathit{IncGDD}^{\phi}_{\gamma,8}$.*

Proof. We now give the sketch of the proof given by [MR07, Lemma 5.10].

Given a basis $\mathbf{B}$, the goal is to solve $\mathrm{GIVP}^{\phi}_{8\gamma}$ using a IncGDD oracle, i.e., to find n linearly independent vectors $\mathbf{S}$ such that $\|\mathbf{S}\| \leq 8\gamma \cdot \phi(\mathcal{L}(\mathbf{B}))$. The process is iterative. Initially $\mathbf{S} = \mathbf{B}$, then at each step:

* Identify $\mathbf{s}_i$ as the the longest vector in $\mathbf{S}$,
* Take $\mathbf{t}$ orthogonal to $\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}, \mathbf{s}_{i+1}, \ldots, \mathbf{s}_n$ (of length $\|\mathbf{S}\|/2$),
* Apply the IncGDD oracle to the instance $(\mathbf{B}, \mathbf{S}, \mathbf{t}, \|\mathbf{S}\|/8)$.
* If it fails, output $\mathbf{S}$. Otherwise we replace $\mathbf{s}_i$ by the output $\mathbf{u}$ of the oracle, and repeat the process.

Note that at each step, the vector $\mathbf{u}$ is at distance at most $\|\mathbf{S}\|/4$ from $\mathbf{t}$, $\|\mathbf{u}\| \leq 3\|\mathbf{S}\|/4$ and $\mathbf{u}$ is linearly independent from the vectors $\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}, \mathbf{s}_{i+1}, \ldots, \mathbf{s}_n$.

If the oracle fails, it must be that $\|\mathbf{S}\|/8 \leq \gamma \cdot \phi(\mathcal{L}(\mathbf{B}))$, then $\mathbf{S}$ is as required to solve the GIVP problem. Moreover this algorithm terminates after a polynomial number of steps, as $\log \prod_i \|\mathbf{s}_i\|$ decreases by a constant at each step, and is initially polynomial in the input size. □

1.4 Gaussian measures

We start by recalling the definitions of continuous and discrete Gaussian functions over lattices for different types of parameters. We then define the smoothing parameter which is used to study discrete Gaussians. Finally, we give some tail bounds and properties about adding and multiplying Gaussians.

1.4.1 Continuous Gaussian distributions

We will consider different types of Gaussian distributions.

Definition 1.18 (Spherical continuous Gaussian)**.** For a vector $\mathbf{c} \in \mathbb{R}^n$ and a real $s > 0$, the Gaussian function is defined by $\rho_{s,\mathbf{c}}(\mathbf{x}) = e^{-\pi \|\frac{\mathbf{x}-\mathbf{c}}{s}\|^2}$, for all $\mathbf{x} \in \mathbb{R}^n$.

By normalizing the Gaussian function, we obtain the (spherical) continuous Gaussian probability distribution:

$$D_{s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{s^n} = \frac{1}{s^n} e^{-\pi \|\frac{\mathbf{x}-\mathbf{c}}{s}\|^2},$$

of parameter s, centered in $\mathbf{c}$.

Remark 1.19. For $\mathbf{r} = (r_1, \ldots, r_n)^T \in (\mathbb{R}^+)^n$, a sample from $D_{\mathbf{r},\mathbf{c}}$ over $\mathbb{R}^n$ is given by $(D_{r_i,c_i})_i$.

We extend this definition to elliptical Gaussian distributions with respect to the axes $(\mathbf{h}_i)_i$ defined in Section 1.2.3, the parameter of the Gaussian is now a vector.

Definition 1.20 (Elliptical continuous Gaussian [LPR10])**.** The elliptical Gaussian distributions in the basis $\{\mathbf{h}_j\}_{j \in \mathbb{Z}_\nu^\times}$ is defined as follows: For $(r_j)_{j \in \mathbb{Z}_\nu^\times} \in \mathbb{R}^n$ such that $r_j = r_{\nu-j}$ for all $j \in \mathbb{J}$, a sample x from $D_{\mathbf{r}}$ is given by $\sigma_C(x) = \sum_j x_j \cdot \mathbf{h}_j$, where each x_j is independently chosen from the Gaussian distribution D_{r_j} over $\mathbb{R}$.

We define $\Psi_{[\alpha,\alpha']}$ for $0 \leq \alpha < \alpha'$, as the set of Gaussian distributions $D_{\mathbf{r}}$ with $\alpha < r_i \leq \alpha'$, for all i. We write $\Psi_{\leq \alpha'}$ when $\alpha = 0$.

We recall the distribution Υ_α used in [LPR10]. The gamma distribution $\Gamma(2,1)$ with shape parameter 2 and scale parameter 1 has density $t\exp(-t)$ for $t \geq 0$ and zero for $t < 0$. For $\alpha > 0$, a distribution sampled from Υ_α is an elliptical Gaussian distribution $D_{\mathbf{r}}$ whose parameters are $r_j = r_{\nu-j} = \alpha\sqrt{1 + \sqrt{n}x_j}$, where the x_j's for $j \in \mathbb{J}$ are chosen independently from $\Gamma(2,1)$. We will use the following result on $\Gamma(2,1)$.

Lemma 1.21 ([LPR10, Claim 5.10])**.** *Let P be the distribution $\Gamma(2,1)^n$ and Q be the distribution $(\Gamma(2,1) - z_1) \times \ldots \times (\Gamma(2,1) - z_n)$ for some $0 \leq z_1, \ldots, z_n \leq 1/\sqrt{n}$. Then for any measurable set $A \subseteq \mathbb{R}^n$, we have $\int_A Q \geq \frac{1}{\mathrm{poly}(n)} \cdot (\int_A P)^2$ (i.e., $R(P\|Q) \leq \mathrm{poly}(n)$).*

Finally we define the *ellipsoid* continuous Gaussian distribution where the parameter of the Gaussian is a non-singular matrix in $\mathbf{S} \in \mathbb{R}^{n\times n}$ of a rank-n matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$. In the two cases, the symmetric matrix $\Sigma = \mathbf{S}^T\mathbf{S}$ is definite positive, i.e., we have $\mathbf{x}\Sigma\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^n$.

Definition 1.22 (Ellipsoid continuous Gaussian)**.** For a rank-n matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ and a vector $\mathbf{c} \in \mathbb{R}^n$, the *ellipsoid* Gaussian distribution with parameter $\mathbf{S}$ and center $\mathbf{c}$ is defined as:

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

This distribution may also be denoted by $D_{\sqrt{\Sigma},\mathbf{c}}$ with $\Sigma = \mathbf{S}^T\mathbf{S}$ if the matrix $\mathbf{S}$ is seen as the square root of Σ. If we let $\rho_{\mathbf{S},\mathbf{c}}$ be the associated Gaussian function. Note that

$$\begin{aligned}\rho_{\mathbf{S},\mathbf{c}}(x) &= \exp\left(-\pi(\mathbf{x}-\mathbf{c})^T\Sigma^{-1}(\mathbf{x}-\mathbf{c})\right)\\ &= \exp\left(-\pi\|(\mathbf{S}^T)^\dagger(\mathbf{x}-\mathbf{c})\|\right) = \exp\left(-\pi\langle(\mathbf{S}^T)^\dagger(\mathbf{x}-\mathbf{c}),(\mathbf{S}^T)^\dagger(\mathbf{x}-\mathbf{c})\rangle\right),\end{aligned}$$

where $\mathbf{X}^\dagger$ denotes the pseudo-inverse of $\mathbf{X}$.

1.4.2 Discrete Gaussian distributions

Those continuous functions can be extended to any countable set $A \subseteq \mathbb{R}^n$ in the usual way:

$$\rho_{s,\mathbf{c}}(A) = \sum_{\mathbf{x}\in A} \rho_{s,\mathbf{c}}(\mathbf{x}).$$

For a n-dimensional lattice Λ and a vector $\mathbf{u} \in \mathbb{R}^n$, we now define the discrete Gaussian distribution as the discrete distribution with support on the coset $\Lambda + \mathbf{u}$ whose probability mass is proportional to the Gaussian function.

Definition 1.23 (Discrete Gaussian)**.** For all $\mathbf{c} \in \mathbb{R}^n$, $s > 0$ (resp. $\mathbf{r} \in \mathbb{R}^n$ and $\mathbf{S} \in \mathbb{R}^{m\times n}$) and lattice Λ, the discrete Gaussian probability distribution with support Λ, center $\mathbf{c}$ and parameter s (resp. $\mathbf{r}$, $\mathbf{S}$) is defined by:

$$\forall \mathbf{x} \in \Lambda,\ D_{\Lambda,s,\mathbf{c}}(\mathbf{x}) = \frac{\rho_{s,\mathbf{c}}(\mathbf{x})}{\rho_{s,\mathbf{c}}(\Lambda)}.$$

(resp. $\mathbf{x} \in \Lambda,\ D_{\Lambda,\mathbf{r},\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\mathbf{r},\mathbf{c}}(\mathbf{x})}{\rho_{\mathbf{r},\mathbf{c}}(\Lambda)}$ and $D_{\Lambda,\mathbf{S},\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\mathbf{S},\mathbf{c}}(\mathbf{x})}{\rho_{\mathbf{S},\mathbf{c}}(\Lambda)}$).

The distribution is the same as a continuous one, but all the sampled vectors belong to the support Λ. Figure 1.1 is an example for $\Lambda = \mathbb{Z}$.

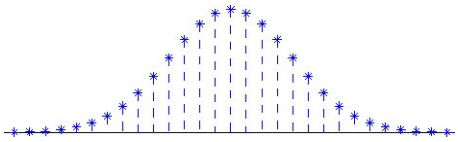

Figure 1.1: Discrete Gaussian on $\mathbb{Z}$.

As we will see further, the properties satisfied by continuous Gaussian distributions are often satisfied by discrete Gaussian distributions.

There exists an efficient procedure that samples within negligible statistical distance of any (not too narrow) discrete Gaussian distribution ([GPV08, Theorem 4.1]; see also [Pei10]). In the next theorem, proved in Section 1.4.3, we modify this sampler so that the output is distributed exactly as a discrete Gaussian. This also allows us to sample from slightly narrower Gaussians.

Theorem 1.24 ([GPV08, Th. 4.1] and Lemmma 1.25)**.** *There is a probabilistic polynomial time algorithm* GPVSample *that, given a basis $\boldsymbol{B}$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\boldsymbol{B})$, a standard deviation $s \geq \|\widetilde{\boldsymbol{B}}\| \cdot \sqrt{\log n}$, and a center $\boldsymbol{c} \in \mathbb{R}^n$, outputs a sample whose distribution is $D_{\Lambda,s,\boldsymbol{c}}$.*

Here $\widetilde{\mathbf{B}}$ denotes the Gram-Schmidt orthogonalisation of $\mathbf{B}$, and $\|\widetilde{\mathbf{B}}\|$ is the length of the longest vector in it. We recall that if $\mathbf{B} = [\mathbf{b}_1, \cdots, \mathbf{b}_n]$, then $\widetilde{\mathbf{B}} = [\widetilde{\mathbf{b}}_1, \cdots, \widetilde{\mathbf{b}}_n]$ with $\widetilde{\mathbf{b}}_1 = \mathbf{b}_1$ and for all $i = 2, \cdots, n$, the vector $\widetilde{\mathbf{b}}_i$ is the projection of $\mathbf{b}_i$ orthogonal to $\mathsf{span}(\mathbf{b}_1, \cdots, \mathbf{b}_{i-1})$.

1.4.3 Exact Gaussian sampler

Theorem 1.24 follows from the following lemma.

Lemma 1.25. *There exists a* PPT *algorithm that takes as inputs a basis $\boldsymbol{B}$ of a lattice $L \subseteq \mathbb{Z}^n$ and a rational $\sigma \geq \|\widetilde{\boldsymbol{B}}\| \cdot \Omega(\sqrt{\log n})$, and outputs vectors $\boldsymbol{b} \in L$ with distribution $D_{L,\sigma}$.*

We now prove this lemma.

Proof. As in [GPV08], the proof consists of two parts. In the first we consider the one-dimensional case, and in the second we use it recursively to sample from arbitrary lattices. Our one-dimensional sampler is based on rejection sampling, just like the one in [GPV08]. Unlike [GPV08], we use the continuous normal distribution as the source distribution which allows us to avoid truncation, and as a result obtain an exact sample. Our second part uses the same recursive routine as in [GPV08], but adds a rejection sampling step to it in order to take care of the deviation of its output from the desired distribution.

The one-dimensional case. Here we show how to sample from the discrete Gaussian distribution on arbitrary cosets of one-dimensional lattices. We use a standard rejection sampling procedure (see, e.g. [Dev86, Page 117] for a very similar procedure).

By scaling, we can restrict without loss of generality to the lattice $\mathbb{Z}$, i.e., we consider the task of sampling from $D_{\mathbb{Z}+c,r}$ for a given coset representative $c \in [0, 1)$ and parameter $r > 0$. The sampling procedure is as follows. Let $Z_0 = \int_c^\infty \rho_r(x)\mathrm{d}x$, and $Z_1 = \int_{-\infty}^{c-1} \rho_r(x)\mathrm{d}x$. These two numbers can be computed efficiently by expressing them in terms of the error function. Let $Z = Z_0 + Z_1 + \rho_r(c) + \rho_r(c-1)$. The algorithm repeats the following until it outputs an answer:

- With probability $\rho_r(c)/Z$ it outputs c;
- With probability $\rho_r(c-1)/Z$ it outputs $c-1$;
- With probability Z_0/Z it chooses x from the restriction of the continuous normal distribution D_r to the interval $[c, \infty)$. Let y be the smallest element in $\mathbb{Z} + c$ that is larger than x. With probability $\rho_r(y)/\rho_r(x)$ output y, and otherwise repeat;
- With probability Z_1/Z it chooses x from the restriction of the continuous normal distribution D_r to the interval $(-\infty, c-1]$. Let y be the largest element in $\mathbb{Z} + c$ that is smaller than x. With probability $\rho_r(y)/\rho_r(x)$ output y, and otherwise repeat.

Consider now one iteration of the procedure. The probability of outputting c is $\rho_r(c)/Z$, that of outputting $c-1$ is $\rho_r(c-1)/Z$, that of outputting $c+k$ for some $k \geq 1$ is

$$\frac{Z_0}{Z} \cdot \frac{1}{Z_0} \int_{c+k-1}^{c+k} \rho_r(x) \cdot \frac{\rho_r(c+k)}{\rho_r(x)} \mathrm{d}x = \frac{\rho_r(c+k)}{Z},$$

and similarly, that of outputting $c-1-k$ for some $k \geq 1$ is $\rho_r(c-1-k)/Z$. From this it follows immediately that conditioned on outputting something, the output distribution has support on $\mathbb{Z}+c$ and probability mass function proportional to ρ_r, and is therefore the desired discrete Gaussian distribution $D_{\mathbb{Z}+c,r}$. Moreover, the probability of outputting something is

$$\frac{\rho_r(\mathbb{Z}+c)}{Z} = \frac{\rho_r(\mathbb{Z}+c)}{Z_0 + Z_1 + \rho_r(c) + \rho_r(c-1)} \geq \frac{\rho_r(\mathbb{Z}+c)}{\rho_r(\mathbb{Z}+c) + \rho_r(c) + \rho_r(c-1)} \geq \frac{1}{2}.$$

Therefore at each iteration the procedure has probability of at least $1/2$ to terminate. As a result, the probability that the number of iterations is greater than t is at most 2^{-t}, and in particular, the expected number of iterations is at most 2.

The general case. For completeness, we start by recalling the SampleD procedure described in [GPV08]. This is a recursive procedure that gets as input a basis $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_n)$ of an n-dimensional lattice $\Lambda = \mathcal{L}(\mathbf{B})$, a parameter $r > 0$, and a vector $\mathbf{c} \in \mathbb{R}^n$, and outputs a vector in $\Lambda + \mathbf{c}$ whose distribution is close to that of $D_{\Lambda+\mathbf{c},r}$. Let $\widetilde{\mathbf{b}}_1, \ldots, \widetilde{\mathbf{b}}_n$ be the Gram-Schmidt orthogonalization of $\mathbf{b}_1, \ldots, \mathbf{b}_n$, and let $\overline{\mathbf{b}}_1, \ldots, \overline{\mathbf{b}}_n$ be the normalized Gram-Schmidt vectors, i.e., $\overline{\mathbf{b}}_i = \widetilde{\mathbf{b}}_i / \|\widetilde{\mathbf{b}}_i\|$. The procedure is the following.

1. Let $\mathbf{c}_n \leftarrow \mathbf{c}$. For $i \leftarrow n, \ldots, 1$, do:
   a) Choose v_i from $D_{\|\widetilde{\mathbf{b}}_i\|\mathbb{Z}+\langle \mathbf{c}_i, \overline{\mathbf{b}}_i \rangle, r}$ using the exact one-dimensional sampler.
   b) Let $\mathbf{c}_{i-1} \leftarrow \mathbf{c}_i + (v_i - \langle \mathbf{c}_i, \overline{\mathbf{b}}_i \rangle) \cdot \mathbf{b}_i / \|\widetilde{\mathbf{b}}_i\| - v_i \overline{\mathbf{b}}_i$.
2. Output $\mathbf{v} := \sum_{i=1}^n v_i \overline{\mathbf{b}}_i$.

It is easy to verify that the procedure always outputs vectors in the coset $\Lambda + \mathbf{c}$. Moreover, the probability of outputting any $\mathbf{v} \in \Lambda + \mathbf{c}$ is

$$\prod_{i=1}^n \frac{\rho_r(v_i)}{\rho_r(\|\widetilde{\mathbf{b}}_i\|\mathbb{Z} + \langle \mathbf{c}_i, \overline{\mathbf{b}}_i \rangle)} = \frac{\rho_r(\mathbf{v})}{\prod_{i=1}^n \rho_r(\|\widetilde{\mathbf{b}}_i\|\mathbb{Z} + \langle \mathbf{c}_i, \overline{\mathbf{b}}_i \rangle)},$$

where $\mathbf{c}_i$ are the values computed in the procedure when it outputs $\mathbf{v}$. Notice that by Lemma 1.28 and our assumption on r, we have that $r \geq \eta_{1/(n+1)}(\|\widetilde{\mathbf{b}}_i\|\mathbb{Z})$ for all i. Therefore, by Lemma 1.31, we have that for all $c \in \mathbb{R}$,

$$\rho_r(\|\widetilde{\mathbf{b}}_i\|\mathbb{Z} + c) \in \left[1 - \frac{2}{n+2}, 1\right] \rho_r(\|\widetilde{\mathbf{b}}_i\|\mathbb{Z}).$$

In order to get an exact sample, we combine the above procedure with rejection sampling. Namely, we apply SampleD to obtain some vector $\mathbf{v}$. We then output $\mathbf{v}$ with probability

$$\frac{\prod_{i=1}^n \rho_r(\|\widetilde{\mathbf{b}}_i\|\mathbb{Z} + \langle \mathbf{c}_i, \overline{\mathbf{b}}_i \rangle)}{\prod_{i=1}^n \rho_r(\|\widetilde{\mathbf{b}}_i\|\mathbb{Z})} \in \left(\left(1 - \frac{2}{n+2}\right)^n, 1\right] \subseteq (e^{-2}, 1], \tag{1.1}$$

and otherwise repeat. This probability can be efficiently computed, as we will show below. As a result, in any given iteration the probability of outputting the vector $\mathbf{v} \in \Lambda + \mathbf{c}$ is

$$\frac{\rho_r(\mathbf{v})}{\prod_{i=1}^{n} \rho_r(\|\widetilde{\mathbf{b}}_i\|\mathbb{Z})}.$$

Since the denominator is independent of $\mathbf{v}$, we obtain that in any given iteration, conditioned on outputting something, the output is distributed according to the desired distribution $D_{\Lambda+\mathbf{c},r}$, and therefore this is also the overall output distribution of our sampler. Moreover, by (1.1), the probability of outputting something in any given iteration is at least e^{-2}, and therefore, the probability that the number of iterations is greater than t is at most $(1-e^{-2})^t$, and in particular, the expected number of iterations is at most e^2.

It remains to show how to efficiently compute the probability in (1.1). By scaling, it suffices to show how to compute

$$\rho_r(\mathbb{Z}+c) = \sum_{k\in\mathbb{Z}} \exp(-\pi(k+c)^2/r^2)$$

for any $r > 0$ and $c \in [0,1)$. If $r < 1$, the sum decays very fast, and we can achieve any desired t bits of accuracy in time $\mathrm{poly}(t)$, which agrees with our notion of efficiently computing a real number (following, e.g., the treatment in [Lov86, Section 1.4]). For $r \geq 1$, we use the Poisson summation formula (see, e.g., [MR07, Lemma 2.8]) to write

$$\rho_r(\mathbb{Z}+c) = r \cdot \sum_{k\in\mathbb{Z}} \exp(-\pi k^2 r^2 + 2\pi i c k) = r \cdot \sum_{k\in\mathbb{Z}} \exp(-\pi k^2 r^2)\cos(2\pi c k),$$

which again decays fast enough so we can compute it to within any desired t bits of accuracy in time $\mathrm{poly}(t)$. □

1.4.4 Smoothing parameter

The *smoothing parameter* of a lattice was introduced by [MR07].

Definition 1.26. For an n-dimensional lattice Λ and positive real $\epsilon > 0$, the smoothing parameter $\eta_\varepsilon(\Lambda)$ is the smallest real $s > 0$ such that $\rho_{1/s}(\Lambda^* \setminus \{0\}) \leq \varepsilon$.

This parameter gives a threshold above which many properties for continuous Gaussians also carry over to discrete Gaussians. We recall a few standard properties on the smoothing parameter and on discrete Gaussians.

Lemma 1.27 ([MR07, Lemma 3.3])**.** *Let Λ be an n-dimensional lattice and $\varepsilon > 0$. Then*

$$\eta_\varepsilon(\Lambda) \leq \lambda_n(\Lambda) \cdot \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}}.$$

Lemma 1.28 ([GPV08, Lemma 3.1])**.** *Let Λ be an n-dimensional lattice with basis $\boldsymbol{B}$ and $\varepsilon > 0$. Then*

$$\eta_\varepsilon(\Lambda) \leq \|\widetilde{\boldsymbol{B}}\| \cdot \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}}.$$

This first lemma implies a (trivial) reduction from SIVP_γ to $\mathrm{GIVP}^{\eta_\varepsilon}_{\gamma'}$, with $\gamma' = \gamma/\sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}}$.

Lemma 1.29 ([Pei08, Lemma 3.5]). *Let Λ be an n-dimensional lattice and $\varepsilon > 0$. Then*

$$\eta_\varepsilon(\Lambda) \leq \sqrt{\frac{\ln(2n(1+1/\varepsilon))}{\pi}} / \lambda_1^\infty(\Lambda^*),$$

where λ_1^∞ refers to the lattice minimum with respect to the infinity norm.

Lemma 1.30 ([MR07, Lemma 4.1]). *For any n-dimensional lattice Λ, $\varepsilon > 0$, $s \geq \eta_\varepsilon(\Lambda)$, the distribution of $\boldsymbol{x} \bmod \Lambda$ where $\boldsymbol{x}$ is sampled from D_s is within statistical distance $\varepsilon/2$ of the uniform distribution on cosets of Λ.*

Lemma 1.31 ([Reg09, Claim 3.8]). *For any n-dimensional lattice Λ, $\varepsilon > 0$, $s \geq \eta_\varepsilon(\Lambda)$, and $\boldsymbol{c} \in \mathbb{R}^n$, we have $\rho_r(\Lambda + \boldsymbol{c}) \in [\frac{1-\epsilon}{1+\epsilon}, 1] \cdot \rho_s(\Lambda)$.*

Lemma 1.32 (Adapted from [MR07, Lemma 2.7]). *Let Λ be an n-dimensional lattice and $\varepsilon \in (0,1)$. Then for any $c \in \mathbb{R}^n$ and $s \geq \eta_\varepsilon(\Lambda)$ we have $\rho_{s,c}(\Lambda) \in [1-\varepsilon, 1+\varepsilon] \cdot \det(\Lambda)^{-1}$.*

Lemma 1.33 ([GPV08, Corollary 2.8]). *Let $\Lambda' \subseteq \Lambda$ be n-dimensional lattices. Then for any $\varepsilon \in (0,1)$, any $s \geq \eta_\varepsilon(\Lambda')$, and any $\boldsymbol{c} \in \mathbb{R}^n$, the distribution $(D_{\Lambda,s,\boldsymbol{c}} \bmod \Lambda')$ is within statistical distance at most 2ε of the uniform distribution over Λ/Λ'. And for any $x \in \Lambda/\Lambda'$ we have:*

$$(D_{\Lambda,s,c} \bmod \Lambda')(x) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, \frac{1+\varepsilon}{1-\varepsilon}\right] \cdot \frac{\det(\Lambda)}{\det(\Lambda')}.$$

Lemma 1.34 ([MR04, Lemma 4.4]). *Let Λ be an n-dimensional lattice, $s > 2\eta_\varepsilon(\Lambda)$ for $\varepsilon \leq 1/100$, and $\boldsymbol{c} \in \mathbb{R}^n$. Then for any $(n-1)$-dimensional hyperplane $\mathcal{H}$, the probability that $x \notin \mathcal{H}$ where x is chosen from $D_{\Lambda,s,c}$ is $\geq 1/100$.*

Lemma 1.35 ([PR06]). *Let n and $q \geq 2$ be integers. Let $m \geq 2n \log q$, and $\sigma \geq \eta_\varepsilon(\mathbb{Z}^m)$ for $\varepsilon < 1/3$. Then the min-entropy of $D_{\mathbb{Z}^m,\sigma}$ is at least $m - 1$.*

1.4.5 Tail bounds

An important property on Gaussian distributions is that a sample from a continuous or a discrete Gaussian distribution is short with overwhelming probability.

Lemma 1.36 ([Ban93, Le. 1.5]). *For any lattice $\Lambda \subseteq \mathbb{R}^n$, vector $\boldsymbol{c} \in \mathbb{R}^n$, and parameter $s > 0$, we have*

$$\Pr_{\boldsymbol{b} \hookleftarrow D_{\Lambda,s,\boldsymbol{c}}}[\|\boldsymbol{b} - \boldsymbol{c}\| \leq \sqrt{n}s] \geq 1 - 2^{-\Omega(n)}.$$

Lemma 1.37 (Adapted from [Pei08, Cor. 5.3]). *For any n-dimensional lattice $\Lambda \subseteq \mathbb{R}^n$, $\boldsymbol{c} \in \mathbb{R}^n$, $\varepsilon \in (0,1)$, $t \geq \sqrt{2\pi}$, unit vector $\boldsymbol{u} \in \mathbb{R}^n$ and $s \geq \eta_\varepsilon(\Lambda)$, we have:*

$$\Pr_{\boldsymbol{b} \hookleftarrow D_{\Lambda,s,\boldsymbol{c}}}[|\langle \boldsymbol{b} - \boldsymbol{c}, \boldsymbol{u} \rangle| \geq st] \leq \frac{1+\varepsilon}{1-\varepsilon} t\sqrt{2\pi e} \cdot e^{-\pi t^2}.$$

Lemma 1.38 (Adapted from [Pei08, Cor. 5.3]). *Let Λ be an n-dimensional lattice, $\varepsilon \in (0,1)$ and $\boldsymbol{r} \in \mathbb{R}^n$ with $r_i \geq \eta_\varepsilon(\Lambda)$ for all $i \leq n$. Then we have*

$$Pr_{\boldsymbol{x} \hookleftarrow D_{\Lambda,\boldsymbol{r}}}\left[\|\boldsymbol{x}\|_\infty \geq (\max_i r_i) \cdot t\right] \leq 2en \cdot \exp(-\pi t^2),$$

for all $t > 0$. In particular, for $t = \omega(\sqrt{\log n})$ (resp. $t = \Omega(\sqrt{n})$) the above probability is at most $n^{-\omega(1)}$ (resp. $2^{-\Omega(n)}$).

We now generalize [Pei08, Cor. 5.3] and [SS11, Le. 2.9] to the case of module lattices (over the ring of integers of a cyclotomic number field).

Lemma 1.39. *Let $\varepsilon \in (0, \frac{1}{2m+1})$ and $z_1, \ldots, z_m \in R$. Let $M \subseteq K^d$ be a rank d module on R, $s \geq \eta_\varepsilon(M)$ and $\mathbf{c}_1, \ldots, \mathbf{c}_m \in R^d$. If the $\boldsymbol{y}_\ell$'s are independently sampled from the $D_{M,s,\mathbf{c}_\ell}$'s, then, for all $t \geq 0$:*

$$\Pr\left[\| \sum_{\ell \in [m]} z_\ell \cdot (\boldsymbol{y}_\ell - \mathbf{c}_\ell) \|_\infty \geq st\|\boldsymbol{z}\|\right] \leq 2\frac{1+\varepsilon}{1-\varepsilon} tN\sqrt{2\pi e} \cdot e^{-\pi t^2}.$$

In particular, for $t = \omega(\sqrt{\log N})$ the above probability is negligible with respect to N.

Proof. The proof builds upon that of [Pei08, Cor. 5.3]. The principle is to interpret the m Gaussian samples from the N-dimensional lattice M as one Gaussian sample from the (Nm)-dimensional lattice L and then apply Lemma 1.37, where $L = M \times \cdots \times M$ (i.e., the Cartesian product of m copies of M). We also define $\vec{\mathbf{c}} = (\mathbf{c}_1, \ldots, \mathbf{c}_m)^T \in (R^d)^m$ and $\vec{\mathbf{y}} = (\mathbf{y}_1, \ldots, \mathbf{y}_m)^T \in (R^d)^m$. We have $\rho_{s,\vec{\mathbf{c}}}(L) = \prod_{\ell \in [m]} \rho_{s,\mathbf{c}_\ell}(M)$. The vector $\vec{\mathbf{y}}$ has distribution $D_{L,s,\vec{\mathbf{c}}}$. We have:

$$\sigma_C\Big(\sum_{\ell \in [m]} z_\ell \cdot (\mathbf{y}_\ell - \mathbf{c}_\ell)\Big) = \begin{bmatrix} \sum_{\ell=1}^m \sigma_C(z_\ell \cdot (y_\ell^{(1)} - c_\ell^{(1)})) \\ \vdots \\ \sum_{\ell=1}^m \sigma_C(z_\ell \cdot (y_\ell^{(d)} - c_\ell^{(d)})) \end{bmatrix} = \begin{bmatrix} \left(\langle \sigma_j(\vec{z}), \overline{\sigma_j(\vec{y}^{(1)} - \vec{c}^{(1)})}\rangle\right)_{j \in \mathbb{Z}_\nu^\times} \\ \vdots \\ \left(\langle \sigma_j(\vec{z}), \overline{\sigma_j(\vec{y}^{(d)} - \vec{c}^{(d)})}\rangle\right)_{j \in \mathbb{Z}_\nu^\times} \end{bmatrix}$$

with $\vec{z} = (z_1, \ldots, z_m)^T \in R^m$, $\vec{y}^{(k)} - \vec{c}^{(k)} = (y_1^{(k)} - c_1^{(k)}, \ldots, y_m^{(k)} - c_m^{(k)})^T \in R^m$ for $k \in [d]$, and, for any $j \in \mathbb{Z}_\nu^\times$ and $\vec{x} \in R^m$, $\sigma_j(\vec{x}) = (\sigma_j(x_\ell))_{\ell \in [m]}$.

By applying the union bound over all $j \in \mathbb{Z}_\nu^\times$ and all $k \in [d]$, it suffices to obtain a probabilistic upper bound on the Hermitian product between $\sigma_j(\vec{z})$ and $\overline{\sigma_j(\vec{y}^{(k)} - \vec{c}^{(k)})}$ for any fixed j and k. For the rest of the proof, we fix $j \in \mathbb{Z}_\nu^\times$ and $k \in [d]$. Wlog (by complex conjugation), we take $j \in \mathbb{J}$.

For $\ell \in [m]$, let $\mathbf{u}_\ell = (u_\ell^{(1)}, \ldots, u_\ell^{(d)})^T \in \mathbb{C}^{nd}$ with $u_\ell^{(k')} = (0, \ldots, 0)^T$ for $k' \neq k$, and:

$$u_\ell^{(k)} = (0, \ldots, 0, \sigma_j(z_\ell), 0, \ldots, 0, -i \cdot \sigma_j(z_\ell), 0, \ldots, 0)^T,$$

i.e., the coordinate of index j is equal to $\sigma_j(z_\ell)$, the coordinate of index $\nu - j$ is equal to $-i \cdot \sigma_j(z_\ell)$, and all the others are 0. We now define $\vec{\mathbf{u}} \in \mathbb{C}^{ndm}$ as the concatenation of the $\mathbf{u}_\ell$'s (for $\ell \in [m]$), and $\sigma_H(\vec{\mathbf{y}} - \vec{\mathbf{c}}) \in \mathbb{R}^{ndm}$ as the concatenation of the $\sigma_H(\mathbf{y}_\ell - \mathbf{c}_\ell)$'s. We have:

$$\langle \sigma_j(\vec{z}), \overline{\sigma_j(\vec{y}^{(k)} - \vec{c}^{(k)})}\rangle = \sum_\ell \sigma_j(z_\ell)\overline{\sigma_j(y_\ell^{(k)} - c_\ell^{(k)})} = \langle \vec{\mathbf{u}}, \sigma_H(\vec{\mathbf{y}} - \vec{\mathbf{c}})\rangle.$$

Now, we define $\vec{\mathbf{v}} = \vec{\mathbf{u}}/\|\vec{\mathbf{u}}\| \in \mathbb{C}^{ndm}$. By Lemma 1.37, we have:

$$\begin{aligned} \Pr_{\vec{\mathbf{y}} \hookleftarrow D_{L,s,\vec{\mathbf{c}}}}\left[|\langle \sigma_H(\vec{\mathbf{y}} - \vec{\mathbf{c}}), \Re(\vec{\mathbf{v}})\rangle| \geq st\right] &\leq \frac{1+\varepsilon}{1-\varepsilon} t\sqrt{2\pi e} \cdot e^{-\pi t^2}, \\ \Pr_{\vec{\mathbf{y}} \hookleftarrow D_{L,s,\vec{\mathbf{c}}}}\left[|\langle \sigma_H(\vec{\mathbf{y}} - \vec{\mathbf{c}}), \Im(\vec{\mathbf{v}})\rangle| \geq st\right] &\leq \frac{1+\varepsilon}{1-\varepsilon} t\sqrt{2\pi e} \cdot e^{-\pi t^2}, \end{aligned}$$

where $\Re$ and $\Im$ respectively denote the real and imaginary parts of a complex number. By using the union bound and scaling by $\|\vec{\mathbf{u}}\| \leq \|\mathbf{z}\|$, we obtain that:

$$\Pr\left[|\langle \vec{\mathbf{u}}, \sigma_H(\vec{\mathbf{y}}' - \vec{\mathbf{c}}')\rangle| \geq st\|\mathbf{z}\|\right] \leq 2\frac{1+\varepsilon}{1-\varepsilon} t\sqrt{2\pi e} \cdot e^{-\pi t^2}.$$

This leads to the claimed result. □

When the parameter of the Gaussian is a matrix, we compare the smallest singular value of this matrix to the smoothing parameter to obtain a tail on the discrete Gaussian distribution which depends on the largest singular value of the parameter.

Lemma 1.40 ([AGHS13, Le. 3]). *For a rank-n lattice Λ, constant $0 < \varepsilon < 1$, vector $\mathbf{c}$ and matrix $\boldsymbol{S}$ with $\sigma_n(\boldsymbol{S}) \geq \eta_\varepsilon(\Lambda)$, if $\mathbf{x}$ is sampled from $D_{\Lambda,\boldsymbol{S},\mathbf{c}}$ then $\|\mathbf{x}\| \leq \sigma_1(\boldsymbol{S})\sqrt{n}$, except with probability $\leq \frac{1+\varepsilon}{1-\varepsilon} \cdot 2^{-n}$.*

1.4.6 Linear combinations of Gaussians

The sum of two continuous Gaussians with parameters s and r is a continuous Gaussian with parameter $\sqrt{s^2 + r^2}$. We have the following similar result for the sum of a continuous Gaussian and a discrete one.

Lemma 1.41 ([Reg09, Claim 3.9]). *Let Λ be an n-dimensional lattice, let $\boldsymbol{u} \in \mathbb{R}^n$ be arbitrary, let $r, s > 0$ and let $t = \sqrt{r^2 + s^2}$. Assume that $rs/t = 1/\sqrt{1/r^2 + 1/s^2} \geq \eta_\varepsilon(\Lambda)$ for some $\varepsilon < 1/2$. Consider the continuous distribution Y on $\mathbb{R}^n$ obtained by sampling from $D_{\Lambda+\boldsymbol{u},r}$ and then adding a noise vector taken from D_s. Then we have $\Delta(Y, D_{\boldsymbol{t}}) \leq 4\varepsilon$.*

We adapt this Lemma in the case of elliptical Gaussian distributions.

Lemma 1.42 (Adapted from [Reg09, Claim 3.9]). *Let Λ be an n-dimensional lattice, $\boldsymbol{u} \in \mathbb{R}^n$, $\boldsymbol{r} \in (\mathbb{R}^+)^n$, $s > 0$ and $t_i = \sqrt{r_i^2 + s^2}$ for all i. Assume that $\min_i(r_i\sigma/t_i) \geq \eta_\varepsilon(\Lambda)$ for some $\varepsilon \leq 1/2$. Consider the discrete distribution Y on $\mathbb{R}^n$ obtained by sampling from $D_{\Lambda+\boldsymbol{u},\boldsymbol{r}}$ and then adding a vector taken from D_s. Then we have $\Delta(Y, D_{\boldsymbol{t}}) \leq 4\varepsilon$.*

Proof. This proof follows the same principle as the one of [Reg09, Claim 3.9], the only difference being that [Reg09, Claim 3.9] considers the case where all r_i's are equal. Using the Poisson summation formula, one obtains that the probability density function Y can be written as:

$$\forall \mathbf{x} \in \mathbb{R}^n : \; Y(\mathbf{x}) = \frac{\rho_{\mathbf{t}}(\mathbf{x})}{\prod_i t_i} \cdot \frac{\left(\prod_i \frac{t_i}{sr_i}\right) \cdot \widehat{\rho_{\mathbf{t}',\mathbf{x}'-\mathbf{u}}}(\Lambda^*)}{\left(\prod_i \frac{1}{r_i}\right) \cdot \widehat{\rho_{\mathbf{r},-\mathbf{u}}}(\Lambda^*)},$$

where $t'_i = r_i s/t_i$ and $x'_i = r_i^2 x_i / t_i^2$ for all i, and where $\widehat{f}$ denotes the Fourier transform of f. Then, we have:

$$\begin{aligned}
\left|1 - \left(\prod_i \frac{t_i}{sr_i}\right) \widehat{\rho_{\mathbf{t}',\mathbf{x}'-\mathbf{u}}}(\Lambda^*)\right| &\leq \rho_{\mathbf{t}''}(\Lambda^* \setminus \{\mathbf{0}\}), \quad \text{with} \quad t''_i = 1/t'_i \text{ for all } i, \\
\left|1 - \left(\prod_i \frac{1}{r_i}\right) \widehat{\rho_{\mathbf{r},-\mathbf{u}}}(\Lambda^*)\right| &\leq \rho_{\mathbf{r}''}(\Lambda^* \setminus \{\mathbf{0}\}), \quad \text{with} \quad r''_i = 1/r_i \text{ for all } i.
\end{aligned}$$

Let $\mathbf{s}'$ and $s' > 0$ be such that $s'_i \geq s'$ for all i. We have that for any vector $\mathbf{x}$:

$$\frac{\rho_{1/s'}(\mathbf{x})}{\rho_{(1/s'_i)_i}(\mathbf{x})} = \exp\left(-\pi \sum_i ((s')^2 x_i^2 - (s'_i)^2 x_i^2)\right) \geq 1.$$

This implies that $\rho_{\mathbf{t}''}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \varepsilon$ and $\rho_{\mathbf{r}''}(\Lambda^* \setminus \{\mathbf{0}\}) \leq \varepsilon$, which completes the proof. □

Lemma 1.43 (Special case of [Pei10, Theorem 3.1])**.** *Let Λ be a lattice and $r, s > 0$ be such that $s \geq \eta_\varepsilon(\Lambda)$ for some $\varepsilon \leq 1/2$. Then if we choose $\boldsymbol{x}$ from the continuous Gaussian D_r and then choose $\boldsymbol{y}$ from the discrete Gaussian $D_{\Lambda-\boldsymbol{x},s}$ then $\boldsymbol{x}+\boldsymbol{y}$ is within statistical distance 8ε of the discrete Gaussian $D_{\Lambda,\sqrt{r^2+s^2}}$.*

1.4.7 Product and inner product

The product of a continuous Gaussian on $\mathbb{R}$ with parameter s and a scalar $x \in \mathbb{R}$ is a continuous Gaussian with parameter $|x|s$. This can be generalized to the ring and module settings. The following result is given in [LPR10], but without proof.

Lemma 1.44. *Let $\boldsymbol{r} \in (\mathbb{R}^+)^n$ with $r_j = r_{\nu-j}$ for all $j \in \mathbb{Z}_\nu^\times$, $x \in K$ sampled from $D_{\boldsymbol{r}}$ and $e \in K$ fixed. Then $x \cdot e$ is distributed from $D_{\boldsymbol{r}'}$ with $r'_j = r_j|\sigma_j(e)|$ for all j.*

Proof. Let us write $\sigma_C(x) = \sum_j x_j \cdot \mathbf{h}_j$ where each x_j is sampled from D_{r_j}. By definition of the $\mathbf{h}_j$'s, we have $\sigma_j(x) = (x_j + ix_{\nu-j})$ and $\sigma_{\nu-j}(x) = (x_j - ix_{\nu-j})$, for $j \in \mathbb{J}$. Let $\sigma_C(e) = \sum_j e_j \cdot \mathbf{h}_j$ and $\sigma_C(e \cdot x) = \sum_j y_j \cdot \mathbf{h}_j$. We have, for $j \in \mathbb{J}$

$$\begin{bmatrix} y_j \\ y_{\nu-j} \end{bmatrix} = \begin{bmatrix} e_j & -e_{\nu-j} \\ e_{\nu-j} & e_j \end{bmatrix} \begin{bmatrix} x_j \\ x_{\nu-j} \end{bmatrix}$$

The vector $(y_j, y_{\nu-j})^T$ is an orthogonal transformation of the vector $(x_j, x_{\nu-j})$, and thus y_j and $y_{\nu-j}$ are statistically independent. Further, the reals y_j and $y_{\nu-j}$ are samples of $D_{r'_j}$ and $D_{r'_{\nu-j}}$ respectively, with $r'_j = r'_{\nu-j} = (e_j^2 r_j^2 + e_{\nu-j}^2 r_{\nu-j}^2)^{1/2} = r_j|\sigma_j(e)|$. □

The following lemma generalizes the previous result to the module setting.

Lemma 1.45. *Let $\boldsymbol{r} \in (\mathbb{R}^+)^n$ with $r_j = r_{\nu-j}$ for all $j \in \mathbb{Z}_\nu^\times$, $\boldsymbol{x} \in K^d$ sampled from $D_{\boldsymbol{s},\ldots,\boldsymbol{s}}$ and $\boldsymbol{e} \in K^d$ fixed. Then $\sum_k x_k e_k$ is distributed from $D_{\boldsymbol{r}'}$ with $r'_j = r_j \cdot (\sum_{k\in[d]} |\sigma_j(e_k)|^2)^{1/2}$ for all j.*

Proof. By Lemma 1.44, we have that $x_k \cdot e_k$ has distribution $D_{\mathbf{r}'_k}$ with $r'_{k,j} = r'_{k,\nu-j} = r_j|\sigma_j(e_k)|$ for all j. The quantity under scope is the sum of independent Gaussians. □

Lemma 1.46 ([Reg09, Corollary 3.10])**.** *Let Λ be an n-dimensional lattice, let $\boldsymbol{u}, \boldsymbol{z} \in \mathbb{R}^n$ be arbitrary, and let $r, \alpha > 0$. Assume that $(1/r^2 + (\|\boldsymbol{z}\|/\alpha)^2)^{-1/2} \geq \eta_\varepsilon(\Lambda)$ for some $\varepsilon < 1/2$. Then the distribution of $\langle \boldsymbol{z}, \boldsymbol{v}\rangle + e$ where $\boldsymbol{v}$ is sampled from $D_{\Lambda+\mathbf{u},r}$ and e is sampled from D_α, is within statistical distance 4ε of D_β for $\beta = \sqrt{(r\|\boldsymbol{z}\|)^2 + \alpha^2}$.*

Chapter 2

Small Integer Solution and Learning with Errors Problem

Two main problems serve as the foundation of numerous lattice-based cryptographic protocols. The first one, introduced by Ajtai in 1996 [Ajt96], is the *Short Integer Solution problem* (SIS): For parameters n, m and q positive integers, the problem is to find a short nonzero solution $\mathbf{z} \in \mathbb{Z}^m$ to the homogeneous linear system $\mathbf{Az} = \mathbf{0} \bmod q$ for uniformly random $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ (the notation $\mathbb{Z}_q$ denotes the ring of integers modulo q). The second one, introduced by Regev in 2005 [Reg05, Reg09], is the *Learning With Errors problem* (LWE). The search version of LWE is as follows: For parameters n and q positive integers and χ a probability density function on $\mathbb{T} = \mathbb{R}/\mathbb{Z} \simeq [0, 1)$, the problem is to find $\mathbf{s}$, given arbitrarily many independent pairs $(\mathbf{a}, \frac{1}{q}\langle \mathbf{a}, \mathbf{s} \rangle + e)$ for a vector $\mathbf{a} \in \mathbb{Z}_q^n$ chosen uniformly at random, and $e \in \mathbb{T}$ sampled from χ. It is possible to interpret LWE in terms of linear algebra: If m independent samples $(\mathbf{a}_i, \frac{1}{q}\langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$ are considered, the goal is to find $\mathbf{s}$ from $(\mathbf{A}, \frac{1}{q}\mathbf{As} + \mathbf{e})$, where the rows of $\mathbf{A}$ correspond to the $\mathbf{a}_i$'s and $\mathbf{e} = (e_1, \ldots, e_m)^T$. The decision counterpart of LWE consists in distinguishing between arbitrarily many independent pairs $(\mathbf{a}, \frac{1}{q}\langle \mathbf{a}, \mathbf{s} \rangle + e)$ sampled as in the search version and the same number of uniformly random and independent pairs.

Ajtai [Ajt96] proposed the first worst-case to average-case reduction for a lattice problem, by providing a reduction from SIVP_γ to SIS, where γ depends on the shortness of the SIS solution. Later, Regev [Reg05, Reg09] showed the hardness of the LWE problem by describing a (quantum) reduction from SIVP_γ to LWE. Cryptographic protocols relying on SIS or LWE therefore enjoy the property of being provably at least as secure as a worst-case problem which is strongly suspected of being extremely hard. However, on the other hand, the cryptographic applications of SIS and LWE are inherently inefficient due to the size of the associated key (or public data), which typically consists of the matrix $\mathbf{A}$.

In this chapter, we give the formal definition of those two problems, and we describe the existing hardness results. In Chapter 3, we provide examples of cryptographic primitives constructed using those two problems.

2.1 Small Integer Solution problem

We first describe the Small Integer Solution problem.

2.1.1 Definition

The SIS problem was first introduced by Ajtai [Ajt96] and formalized in [MR07].

Definition 2.1. The *Small Integer Solution problem* $SIS_{q,m,\beta}$ is as follows: Given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ chosen from the uniform distribution, find $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{z}^T\mathbf{A} = 0 \bmod q$ and $0 < \|\mathbf{z}\| \leq \beta$.

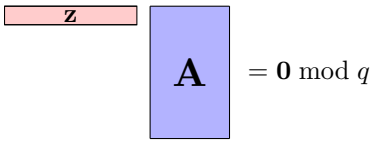

Figure 2.1: The Small Integer Solution problem.

As observed in [MR07, Le. 5.2], for any q, $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and $\beta \geq \sqrt{m}q^{n/m}$, the SIS instance $(q, \mathbf{A}, \beta)$ admits a solution.

ISIS. There exists a variant of the SIS problem: the *Inhomogeneous Small Integer Solution* problem. Instead of finding a solution of a homogeneous system we now look for a solution of a inhomogeneous one.

Definition 2.2. The *Inhomogeneous Small Integer Solution problem* $ISIS_{q,m,\beta}$ is as follows: Given $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ both chosen from the uniform distribution, find $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{z}^T\mathbf{A} = \mathbf{u}^T \bmod q$ and $\|\mathbf{z}\| \leq \beta$.

We denote by $\mathrm{SIS}^p_{n,m,q,\beta}$ (respectively $\mathrm{ISIS}^p_{n,m,q,\beta}$) the SIS (respectively the ISIS) problem in the ℓ_p norm.

2.1.2 Hardness of SIS

There are several reductions from GIVP to SIS (see, e.g., [Ajt96, MR04, GPV08]). The strongest known result is the following.

Theorem 2.3 (Adapted from [GPV08, Th. 9.2])**.** *For $\varepsilon(n) = n^{-\omega(1)}$, there is a probabilistic polynomial time reduction from solving $\mathrm{GIVP}^{\eta_\varepsilon}_\gamma$ in polynomial time (in the worst case, with high probability) to solving $\mathrm{SIS}_{q,m,\beta}$ (or $\mathrm{ISIS}_{q,m,\beta}$) in polynomial time with non-negligible probability, for any $m(n), q(n), \beta(n)$ and $\gamma(n)$ such that $\gamma \geq \beta\sqrt{n}\cdot\omega(\sqrt{\log n})$, $q \geq \beta\sqrt{n}\cdot\omega(\log n)$ and $m, \log q \leq \mathrm{poly}(n)$.*

It then follows from the relationship between the ℓ_2 and ℓ_∞ norms that the $\mathsf{SIS}^\infty_{q,m,\beta}$ and $\mathsf{ISIS}^\infty_{q,m,\beta}$ problems are at least as hard as SIVP^2_γ (in the ℓ_2 norm) for some $\gamma = \beta\cdot\widetilde{O}(n)$.

Proof. We now sketch the proof given by [GPV08] for the SIS problem. The full proof for the ISIS problem follows the same principle and is given in [GPV08]. The authors use an intermediate problem called the Incremental Independent Vectors Decoding problem defined in Definition 1.15. The reduction from GIVP to this problem is recalled in Chapter 1, Theorem 1.17.

We now give the principle of the reduction from $\mathrm{IncIVD}^{\eta_\varepsilon}_{\gamma,g}$ to $\mathrm{SIS}_{q,m,\beta}$. Given a tuple $(\mathbf{B}, \mathbf{S}, \mathbf{t})$, the goal is to find a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v} - \mathbf{t}\| \leq \frac{\|\mathbf{S}\|}{g}$, given a $\mathrm{SIS}_{q,m,\beta}$ oracle.

- Choose $j \in [m]$ and $\alpha \in \{-\beta, \ldots, \beta\}$ uniformly at random.
  Let $\mathbf{c}_j = \mathbf{t} \cdot q/\alpha$ and $\mathbf{c}_i = \mathbf{0}$ otherwise.
  Using Lemma 1.8, convert $(\mathbf{B}, \mathbf{S})$ into a basis $\mathbf{T}$ of $\mathcal{L}(\mathbf{B})$ such that $\|\widetilde{\mathbf{T}}\| \leq \|\widetilde{\mathbf{S}}\| \leq \|\mathbf{S}\|$.
- Let $s = \|\mathbf{S}\| \cdot q/\gamma$. For each $i \in [m]$, sample $\mathbf{y}_i \hookleftarrow D_{\mathcal{L}(\mathbf{B}),s,\mathbf{c}_i}$ using $\mathbf{T}$ in Theorem 1.24. Then let $\mathbf{A} = \mathbf{B}^{-1}\mathbf{Y} \bmod q$, where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m] \in \mathbb{R}^{n \times m}$.
- Invoke the $\mathrm{SIS}_{q,m,\beta}$ oracle on $\mathbf{A}$, yielding $\mathbf{e} \in \mathbb{Z}^n$, and output $\mathbf{v} = \mathbf{Ye}/q$.

We now have to prove that the distribution of the matrix $\mathbf{A}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n \times m}$, that $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ and finally that $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{S}\|/g$. The first result comes from Lemma 1.33. As $\|\mathbf{S}\| \geq \eta_\varepsilon(q\mathcal{L}(\mathbf{B}))$. We have that $\mathbf{y}_i \bmod q\mathcal{L}(\mathbf{B})$ is statistically close to uniform over $\mathcal{L}(\mathbf{B})/q\mathcal{L}(\mathbf{B})$, and then that $\mathbf{a}_i = \mathbf{B}^{-1}\mathbf{y}_i \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$. As the $\mathbf{y}_i$ are independent and $m = \mathrm{poly}(n)$, the distribution of the matrix $\mathbf{A}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n \times m}$. We then have that the oracle outputs a non-zero solution $\mathbf{e}$ such that, with probability $1/(2\beta m)$, the coordinate $e_j = \alpha$. We know that $\mathbf{Ae} = \mathbf{0} \bmod q$, which implies that $\mathbf{B}^{-1}\mathbf{Ye} = \mathbf{0} \bmod q$. Then we have $\mathbf{B}^{-1}\mathbf{Ye} \in q\mathcal{L}(\mathbf{B})$ which implies that $\mathbf{v} \in \mathcal{L}(\mathbf{B})$.

Finally, if $\mathbf{e}_j = \alpha$ we have $\mathbf{t} = \mathbf{Ce}/q$ where $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_m]$. For each $\mathbf{y}_i$ we let $\mathbf{w}_i = \mathbf{y}_i \bmod q\mathcal{L}(\mathbf{B})$. Then, conditioned on any fixed value of $\mathbf{w}_i$, the vector $\mathbf{y}_i$ is distributed as $\mathbf{w}_i + D_{q\mathcal{L}(\mathbf{B}),s,\mathbf{c}_i - \mathbf{w}_i}$. As a consequence, for any fixed $\mathbf{e}$, the vector $\mathbf{v} - \mathbf{t} = (\mathbf{Y} - \mathbf{C})\mathbf{e}/q$ is distributed as

$$\frac{1}{q}\left((\mathbf{W} - \mathbf{C})\mathbf{e} + \sum_i e_i \cdot D_{q\mathcal{L}(\mathbf{B}),s,\mathbf{c}_i - \mathbf{w}_i}\right)$$

As $s \geq \eta_\varepsilon(q\mathcal{L}(\mathbf{B}))$, this is distributed as a Gaussian centered in $\mathbf{0}$ and of parameter $\|\mathbf{e}\| \cdot s/q$. This gives that $\|\mathbf{v} - \mathbf{t}\| \leq \|\mathbf{S}\|/g$. □

Recently, Micciancio and Peikert [MP13] gave another hardness result for the SIS problem for small parameters. Indeed, Theorem 2.3 required $q \geq \beta\sqrt{n} \cdot \omega(\sqrt{\log n})$ and in [MP13] they only need $q \geq \beta \cdot n^\delta$ for any constant $\delta > 0$.

Theorem 2.4 ([MP13, Theorem 1.1])**.** *Let n and $m = \mathrm{poly}(n)$ be integers, and let $q \geq \beta \cdot n^\delta$ for any constant $\delta > 0$. Then there is a polynomial time reduction from solving $\mathrm{GIVP}_\gamma^{\eta_\varepsilon}$ to solving $\mathrm{SIS}_{q,m,\beta}$ for $\gamma = \max\{1, \beta^2/q\} \cdot \widetilde{O}(\beta\sqrt{n})$.*

2.2 Learning with Errors problem

We now introduce the second fundamental problem in lattice-based cryptography, the Learning with Errors problem, and give the principle of its existing hardness reduction from hard problems on lattices.

2.2.1 Definition

The Learning with Errors problem has been introduced by Regev [Reg05] in 2005.

There are several versions of this problem, we will describe two of them which are equivalent. The main parameters of this problem are the dimension n and the modulus q. In the first version all elements will have coordinates in $\mathbb{Z}_q$, all the operations will be modulo q. In the second version, part of the elements will be in $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ which denotes the segment $[0, 1)$ with addition modulo 1. We also recall that we denote by $\mathbb{T}_q$ its cyclic subgroup of order q, i.e., the subgroup given by $\{0, 1/q, \ldots, (q-1)/q\}$.

We recall the following definitions from [Reg05, Reg09].

Distribution $A_{q,\mathbf{s},\phi}$. For a probability density function ϕ on $\mathbb{T}$ and a vector $\mathbf{s} \in \mathbb{Z}_q^n$, we let $A_{q,\mathbf{s},\phi}$ denote the distribution on $\mathbb{Z}_q^n \times \mathbb{T}$ obtained by choosing a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \in \mathbb{T}$ according to ϕ, and returning $(\mathbf{a}, \frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle + e)$.

This distribution could also be defined by choosing $e \in \mathbb{Z}$ according to ϕ on $\mathbb{Z}$, and returning $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$. Alternatively one can also choose a vector $\mathbf{a} \in \mathbb{T}_q^n$ uniformly at random, $e \in \mathbb{T}$ according to ϕ on $\mathbb{T}$, and return $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e) \in \mathbb{T}_q^n \times \mathbb{T}$.

Definition 2.5. The *search* version of the *Learning With Error problem* $SLWE_{n,q,\phi}$ is as follows: Let $\mathbf{s} \in \mathbb{Z}_q^n$ be secret; Given arbitrarily many samples from $A_{q,\mathbf{s},\phi}$, the goal is to find $\mathbf{s}$.

The *decision* version of the *Learning With Error problem* $LWE_{n,q,\phi}$ is as follows: Let $\mathbf{s} \in \mathbb{Z}_q^n$ be uniformly random; The goal is to distinguish between arbitrarily many independent samples from $A_{q,\mathbf{s},\phi}$ and the same number of independent samples from $U(\mathbb{Z}_q^n \times \mathbb{Z}_q)$.

The LWE problem is equivalently defined for a $A_{q,\mathbf{s},\phi}$ distribution in $\mathbb{T}_q^n \times \mathbb{T}$ or in $\mathbb{Z}_q^n \times \mathbb{T}$. In Chapter 4, we use the representation of samples in $\mathbb{T}_q^n \times \mathbb{T}$, whereas in Chapter 5, we use the representation of samples in $\mathbb{Z}_q^n \times \mathbb{T}$.

It is also possible to interpret LWE in terms of linear algebra: Suppose the number of requested samples $(\mathbf{a}_i, \langle \mathbf{a}_i, \mathbf{s}\rangle + e_i)$ from $A_{q,\mathbf{s},\phi}$ is m, then we consider the matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ whose rows are the $\mathbf{a}_i$'s, and we create the vector $\mathbf{e} = (e_1, \ldots, e_m)^T$. When the number of samples is known and if needed, we may use the notation $\mathrm{LWE}_{n,m,q,\phi}$. Then SLWE is as follows:

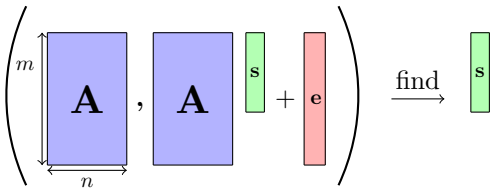

Figure 2.2: The Learning With Errors problem.

2.2.2 Hardness of search LWE

Theorem 2.6 ([Reg09]). *Let* $\varepsilon(n) = n^{-\omega(1)}$, $\alpha \in (0,1)$ *and* $q \geq 2$ *such that* $\alpha q > 2\sqrt{n}$. *There exists a quantum reduction from solving* $\mathrm{GIVP}^{\eta_\varepsilon}_{\sqrt{8}n/\alpha}$ *in polynomial time (in the worst case, with high probability) to solving* $\mathrm{SLWE}_{q,D_\alpha}$ *in polynomial time with non-negligible probability.*

Assume that q *is prime,* $q \leq poly(n)$, *and that* ϕ *is a probability density function on* $\mathbb{T}$. *Then there exists a polynomial-time reduction from* $\mathrm{SLWE}_{q,\phi}$ *to* $\mathrm{LWE}_{q,\phi}$.

A quantum reduction from SIVP. The first main result of [Reg09] is the reduction from GIVP to the computational version of LWE. It makes use of the following intermediary problem, where ϕ denotes an arbitrary real-valued function on lattice, called *Discrete Gaussian Sampling problem* (DGS_ϕ): Given an n-dimensional lattice Λ and a number $r > \phi(\Lambda)$, output a sample from $D_{\Lambda,r}$. Regev's reduction proceeds in two steps:

$$\mathrm{GIVP}_{\frac{\sqrt{8}n}{\alpha}} \xrightarrow{\text{[Reg09, Le. 3.17]}} \mathrm{DGS}_{\frac{\sqrt{2}n}{\alpha}} \xrightarrow{\text{Lemma 2.7}} \mathrm{SLWE}_{q,D_\alpha}$$

The first reduction is lattice-preserving and also works for the structured versions of LWE to be considered later. In contrast, the second one will need to be modified. It comes from the following result:

Lemma 2.7 ([Reg09, Le. 3.3]). *Let $\varepsilon(n) = n^{-\omega(1)}$ (resp. $\varepsilon(n) = 2^{-\Omega(n)}$), $\alpha \in (0,1)$ and $q \geq 2$. Assume that we have access to an oracle that solves $\mathrm{SLWE}_{q,D_\alpha}$ in polynomial time with non-negligible probability (resp. in sub-exponential time with non-exponentially small probability). Then there exists a polynomial time (resp. sub-exponential time) quantum algorithm that, given an n-dimensional lattice Λ, a number $r > \sqrt{2}q \cdot \eta_\varepsilon(\Lambda)$ and $\mathrm{poly}(n)$ (resp. $2^{o(n)}$) samples from $D_{\Lambda,r}$, produces a sample from $D_{\Lambda,\frac{r\sqrt{n}}{\alpha q}}$ with non-negligible (resp. non-exponentially small) probability.*

The principle of the Regev's reduction from DGS to SLWE is to use Lemma 2.7 several times to progressively decrease the value of r. Take $r > \sqrt{2}q \cdot \eta_\varepsilon(\Lambda)$ and $r_i = r \cdot (\alpha q/\sqrt{n})^i$. The first iteration starts with $r_{3n} > 2^{3n} > 2^{2n}\lambda_n(\Lambda)$ (using a LLL-reduction algorithm beforehand). Then it obtains $\mathrm{poly}(n)$ (resp. $2^{o(n)}$) samples of $D_{\Lambda,r_{3n}}$ by Theorem 1.24, and finishes with $\mathrm{poly}(n)$ (resp. $2^{o(n)}$) samples of $D_{\Lambda,r_{3n-1}}$ (the reduction repeats $\mathrm{poly}(n)$ (resp. $2^{o(n)}$) times the same iteration with the same samples in input to obtain sufficiently many different samples in output). It iterates until having $\mathrm{poly}(n)$ (resp. $2^{o(n)}$) samples of D_{Λ,r_1} with $r_1 = r\alpha q/\sqrt{n} > \sqrt{2}q \cdot \eta_\varepsilon(\Lambda)$ then it iterates a last time to obtain samples of D_{Λ,r_0} with $r_0 = r > \sqrt{2n} \cdot \eta_\varepsilon(\Lambda)/\alpha$. These samples are solutions to $\mathrm{DGS}_{\sqrt{2n}\cdot\eta_\varepsilon(\Lambda)/\alpha}$.

To prove Lemma 2.7, Regev uses the intermediary problems called q-BDD_δ: Given a lattice Λ and any point $\mathbf{y} \in \mathbb{R}^n$ within distance $\delta < \lambda_1(\Lambda)/2$ of the lattice, output the coset of $\Lambda/q\Lambda$ of the closest vector to $\mathbf{y}$. The proof of Lemma 2.7 consists also of a sequence of reductions:

$$\mathrm{DGS}_{\Lambda,\frac{r\sqrt{n}}{\alpha q}} \xrightarrow[\text{(quantum)}]{\text{[Reg09, Le. 3.14 \& 3.5]}} q\text{-}\mathrm{BDD}_{\Lambda^*,\frac{\alpha q}{\sqrt{2}r}} \xrightarrow{\text{Lemma 2.8}} \begin{array}{c}\mathrm{SLWE}_{q,D_\alpha}\\ +\\ \text{samples from } D_{\Lambda,r}\end{array}$$

The first reduction also works for the structured versions of LWE to be considered later. However, we will modify the second reduction, by proving an adaptation of the following result.

Lemma 2.8 ([Reg09, Le. 3.4]). *Let $\varepsilon(n) = n^{-\omega(1)}$ (resp. $\varepsilon(n) = 2^{-\Omega(n)}$), $\alpha \in (0,1)$ and $q \geq 2$. Let Λ be a n-dimensional lattice and $r \geq \sqrt{2}q \cdot \eta_\varepsilon(\Lambda)$. Given access to an oracle sampling from the distribution $D_{I,r}$, there exists a probabilistic reduction from solving q-$\mathrm{BDD}_{\Lambda^*,\frac{\alpha q}{\sqrt{2}r}}$ in polynomial time with non-negligible probability (resp. in sub-exponential time with non-exponentially small probability) to solving $\mathrm{SLWE}_{q,D_\alpha}$ in polynomial time with non-negligible probability (resp. in sub-exponential time with non-exponentially small probability).*

2.2.3 From search LWE to decisional LWE

The second main result from [Reg09] is a reduction from the computational problem SLWE to its decisional counterpart LWE. Note that this reduction does not carry over to the structured variants of LWE that we will consider in Chapter 5.

The most recent search-to-decision reduction is by Micciancio and Peikert [MP12], which essentially subsumes all previous reductions, requires the modulus q to be smooth. Below we give the special case when the modulus is a power of 2, which suffices for our purposes. It follows from the results in Chapter 4 that (decision) LWE is hard not just for a smooth modulus q, as follows from [MP12], but actually for all moduli q, including prime moduli, with a deterioration in the noise of $\widetilde{O}(\sqrt{n})$ (see Corollary 4.5).

Theorem 2.9 (Special case of [MP12, Theorem 3.1]). *Let q be a power of 2, and α satisfy $1/q < \alpha < 1/\omega(\sqrt{\log n})$. Then there exists an efficient reduction from search $\mathrm{LWE}_{n,q,\alpha}$ to (decision) $\mathrm{LWE}_{n,q,\alpha'}$ for $\alpha' = \alpha \cdot \omega(\log n)$.*

A classical reduction from GapSVP for q exponential. In [Pei09], Peikert gave the first classical reduction from GapSVP to show the hardness of LWE, but this reduction only allows q exponential in the dimension n.

Theorem 2.10 ([Pei09, Th. 3.1]). *For $\varepsilon(n) = n^{-\omega(1)}$, there is a probabilistic polynomial time reduction from solving $\text{Gap-SVP}_{n,\gamma}$ in polynomial time (in the worst case, with high probability) to solving $\mathrm{LWE}_{n,q,\alpha}$ in polynomial time with non-negligible probability, for any $m(n), q(n), \alpha(n)$ and $\gamma(n)$ such that $\gamma \geq \frac{n}{\alpha \log n}$, $q \geq 2^{n/2} \cdot \omega(\sqrt{\log n/n})$ and $m \leq \mathrm{poly}(n)$.*

2.2.4 Unknown (bounded) noise rate

We also consider a variant of LWE in which the amount of noise is some unknown $\beta \leq \alpha$ (as opposed to exactly α), with β possibly depending on the secret $\mathbf{s}$. As the following lemma shows, this does not make the problem significantly harder.

Definition 2.11. For integers $n, q \geq 1$ and $\alpha \in (0, 1)$, $\mathrm{LWE}_{n,q,\leq\alpha}$ is the problem of solving $\mathrm{LWE}_{n,q,\beta}$ for any $\beta \leq \alpha$, where β possibly depends on $\mathbf{s}$.

Lemma 2.12. *Let $\mathcal{A}$ be an algorithm for $\mathrm{LWE}_{n,m,q,\alpha}$ with advantage at least $\varepsilon > 0$. Then there exists an algorithm $\mathcal{B}$ for $\mathrm{LWE}_{n,m',q,\leq\alpha}$ using oracle access to $\mathcal{A}$ and with advantage at least $1/3$, where both m' and its running time are $\mathrm{poly}(m, 1/\epsilon, n, \log q)$.*

The proof is standard (see, e.g., [Reg09, Lemma 3.7] for the analogous statement for the search version of LWE). The idea is to use Chernoff bound to estimate $\mathcal{A}$'s success probability on the uniform distribution, and then add noise in small increments to our given distribution and estimate $\mathcal{A}$'s behaviour on the resulting distributions. If there is a gap between any of these and the uniform behaviour, the input distribution is deemed non-uniform. The proof is omitted.

2.2.5 Distribution of the secret vector s

We denote by $\mathrm{LWE}_{n,q,\phi}(\mathcal{D})$ the LWE problem for any distribution over secrets $\mathcal{D}$. When the noise is a Gaussian with parameter $\alpha > 0$, i.e., $\phi = D_\alpha$, we use the shorthand $\mathrm{LWE}_{n,q,\alpha}(\mathcal{D})$. Since the case when $\mathcal{D}$ is uniform over $\{0,1\}^n$ plays an important role in Chapter 4, we will denote it by $\mathrm{binLWE}_{n,q,\phi}$ (and by $\mathrm{binLWE}_{n,m,q,\phi}$ when the algorithm only gets m samples). Finally, as we show in the following lemma, one can efficiently reduce LWE to the case in which the secret is distributed according to the (discretized) error distribution and is hence somewhat short. This latter form of LWE, known as the "normal form," was first shown hard in [ACPS09] for the case of prime q. Here we observe that the proof extends to non-prime q, the new technical ingredient being Claim 2.14 below.

Lemma 2.13. *For any $q \geq 25$, $n, m \geq 1$, $\alpha > 0$, $\varepsilon < 1/2$ and $s \geq \sqrt{\ln(2n(1+1/\varepsilon)/\pi)}/q$, there is an efficient (transformation) reduction from $\mathrm{LWE}_{n,m,q,\alpha}$ to $\mathrm{LWE}_{n,m',q,\alpha}(\mathcal{D})$ where $m' = m - (16n + 4\ln\ln q)$ and $\mathcal{D} = D_{\mathbb{Z}^n, q(\alpha^2+s^2)^{1/2}}$, that turns advantage ζ into an advantage of at least $(\zeta - 8\varepsilon)/4$. In particular, assuming $\alpha \geq \sqrt{\ln(2n(1+1/\varepsilon)/\pi)}/q$, we can take $s = \alpha$, in which case $\mathcal{D} = D_{\mathbb{Z}^n, \sqrt{2}q\alpha}$.*

Proof. Consider the first $16n + 4\ln\ln q$ samples $(\mathbf{a}, b)$. Using Claim 2.14, with probability at least $1 - 2e^{-1} \geq 1/4$, we can efficiently find a subsequence of the samples such that the matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times n}$ whose columns are formed by the $\mathbf{a}$ in the subset (scaled up by q) has an inverse $\mathbf{A}_0^{-1} \in \mathbb{Z}_q^{n\times n}$ modulo q. If we cannot find such a subsequence, we abort. Let $\mathbf{b}_0 \in \mathbb{T}^n$ be the vector formed by the corresponding b in the subsequence. Let also $\mathbf{b}'_0 \in \mathbb{T}_q^n$ be $\mathbf{b}_0 + \mathbf{x}$ where $\mathbf{x}$ is chosen from $D_{q^{-1}\mathbb{Z}^n - \mathbf{b}_0, s}$. (Notice that the coset $q^{-1}\mathbb{Z}^n - \mathbf{b}_0$ is well defined because $\mathbf{b}_0$ is a coset of $\mathbb{Z}^n \subseteq q^{-1}\mathbb{Z}^n$.) From each of the remaining m' samples $(\mathbf{a}, b) \in \mathbb{T}_q^n \times \mathbb{T}$ we produce a pair

$$\left(\mathbf{a}' = \mathbf{A}_0^{-1}\mathbf{a}, b' = b - \langle \mathbf{A}_0^{-1} \cdot q\mathbf{a}, \mathbf{b}'_0\rangle\right) \in \mathbb{T}_q^n \times \mathbb{T}.$$

We then apply the given LWE oracle to the resulting m' pairs and output its result.

We now analyze the reduction. First notice that the construction of $\mathbf{A}_0$ depends only on the $\mathbf{a}$ component of the input samples, and hence the probability of finding it is the same in case the input is uniform and in case it consists of LWE samples. It therefore suffices in the following to show that there is a distinguishing gap conditioned on successfully finding an $\mathbf{A}_0$. To that end, first observe that if the input samples $(\mathbf{a}, b)$ are uniform in $\mathbb{T}_q^n \times \mathbb{T}$ then so are the output samples $(\mathbf{a}', b')$. Next consider the case that the input samples are distributed according to $A_{q,\mathbf{s},D_\alpha}$ for some $\mathbf{s} \in \mathbb{Z}^n$. Then since $s \geq \eta_\varepsilon(q^{-1}\mathbb{Z})$ by Lemma 1.28, using Lemma 1.43 we get that $\mathbf{b}'_0 = q^{-1}\mathbf{A}_0^T\mathbf{s} + \mathbf{e}_0$ where $\mathbf{e}_0$ is distributed within statistical distance 8ε from $D_{q^{-1}\mathbb{Z}^n, (\alpha^2+s^2)^{1/2}}$. Therefore, for each output sample $(\mathbf{a}', b')$ we have

$$b' = b - \langle \mathbf{A}_0^{-1} \cdot q\mathbf{a}, \mathbf{b}'_0\rangle = \langle \mathbf{a}, \mathbf{s}\rangle + e - \langle \mathbf{a}, \mathbf{s}\rangle - \langle \mathbf{A}_0^{-1} q\mathbf{a}, \mathbf{e}_0\rangle = \langle -q\mathbf{e}_0, \mathbf{a}'\rangle + e,$$

where e is an independent error from D_α. Therefore, the output samples are distributed according to $A_{q,-q\mathbf{e}_0,D_\alpha}$, completing the proof. □

Claim 2.14. *For any $q \geq 25$, $n \geq 1$, and $t_1 \geq 4, t_2 \geq 1$, given a sequence of $t_1 n + t_2 \ln\ln q$ vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots$ chosen uniformly and independently from $\mathbb{Z}_q^n$, except with probability $e^{-t_1 n/16} + e^{-t_2/4}$, there exists a subsequence of n vectors such that the $n \times n$ matrix they form is invertible modulo q. Moreover, such a subsequence can be found efficiently.*

Proof. We consider the following procedure. Let k be a counter, initialized to 0, indicating the number of vectors currently in the subsequence, and let $\mathbf{A} \in \mathbb{Z}_q^{n\times k}$ be the matrix whose columns are formed by the current subsequence. We also maintain a unimodular matrix $\mathbf{U} \in \mathbb{Z}^{n\times n}$, initially set to the identity, satisfying the invariant that $\mathbf{U} \cdot \mathbf{A} \in \mathbb{Z}_q^{n\times k}$ has the following form: its top $k \times k$ submatrix is upper triangular with each diagonal coefficient coprime with q; its bottom $(n-k) \times k$ submatrix is zero. The procedure considers the vectors $\mathbf{a}_i$ one by one. For each vector $\mathbf{a}$, if it is such that the gcd of the last $n-k$ entries of $\mathbf{Ua}$, call it g, is coprime with q, then it does the following: it adds $\mathbf{a}$ to the subsequence, computes (using, say, the extended GCD algorithm) a unimodular matrix V that acts as identity on the first k coordinates and for which the last $n-k$ coordinates of $\mathbf{VUa}$ are $(g, 0, \ldots, 0)$, replaces $\mathbf{U}$ with $\mathbf{VU}$, and increments k.

It is easy to see that the procedure's output is correct if it reaches $k = n$. It therefore suffices to analyse the probability that this event happens. For this we use the following two facts to handle the cases $k < n-1$ and $k = n-1$, respectively. First, the probability that the gcd of two uniformly random numbers modulo q is coprime with q is

$$\prod_{p|q,\ p \text{ prime}} (1 - p^{-2}) \geq \prod_{p \text{ prime}} (1 - p^{-2}) = \zeta(2)^{-1} \approx 0.61,$$

where ζ is the Riemann zeta function.

Second, the probability that one uniformly random number modulo q is coprime with q is $\varphi(q)/q$, where φ is Euler's totient function. By [BS96, Theorem 8.8.7], this probability is at least $(e^\gamma \ln\ln q + 3/(\ln\ln q))^{-1}$ where γ is Euler's constant, which for $q \geq 25$ is at least $(4 \ln\ln q)^{-1}$.

Using the (multiplicative) Chernoff bound, the first fact, and the fact that $\mathbf{Ua}$ is uniform in $\mathbb{Z}_q^n$ since $\mathbf{U}$ is unimodular, we see that the probability that $k < n-1$ after considering $t_1 n$ vector is at most $e^{-t_1 n/16}$. Moreover, once $k = n-1$, using the second fact we get that the probability that after considering $t_2 \ln\ln q$ additional vectors we still have $k = n-1$ is at most $e^{-t_2/4}$. □

Chapter 3

Simple Lattice-Based Cryptographic Primitives

In the previous chapter, we described the LWE and SIS problems, which are fundamental in lattice-based cryptography as most of the schemes are based on these problems.

The LWE problem is the basis of many cryptographic constructions such as encryption scheme secure against chosen-plaintext attacks [Reg05, GPV08, PVW08, LP11], chosen-ciphertext attacks [PW08, Pei09, MP12], oblivious transfer [PVW08], identity based encryption (IBE) [GPV08, CHKP10, ABB10a, ABB10b], various forms of leakage-resilient cryptography [AGV09], attribute based encryption [GVW13], and Fully Homomorphic Encryption [BV11, BGV12, Bra12] (first introduced, but not based on LWE, by Gentry [Gen09]).

The SIS problem is also fundamental in lattice-based cryptography, the lattice-based one-way and collision resistant hash-functions [Ajt96, GGH96, Mic02a, PR06, LM06, LM08, Lyu08] and signature schemes rely on the hardness of this problem and its ring variant. Attempts for lattice-based signature schemes started in 1997 with Goldreich et al. [GGH97] but this scheme is unsecure [NR09]. Constructions also followed from one-way hash function with the transformation in [NY89] and via the Fiat-Shamir transform [FS86] as the scheme in [MV03]. The first provably secure lattice based signature was introduced in 2008, by Gentry, Peikert and Vaikuntanathan in [GPV08] (based on trapdoors) and independently by Lyubashevsky and Micciancio in [LM08] (based on collision-resistant hash function). From then, other constructions has been given [Lyu09, Lyu12, GLP12, DDLL13].

In this chapter, which is partially based on [LLS14], we first describe in Section 3.1 two public-key encryption schemes based on LWE, Regev's encryption scheme introduced by Regev in [Reg05, Reg09] and the Dual-Regev encryption scheme introduced by Gentry, Peikert and Vaikuntanathan in [GPV08]. Those two schemes are examples of encryption schemes which are secure against chosen-plaintext attacks under the hardness of the LWE problem (but they are not the only ones). In Section 3.2, we first explain the notion of full trapdoor for lattices as most of the signature schemes and many encryption based on LWE (in particular the Identity Based Encryption [GPV08, CHKP10, ABB10a, ABB10b, MP12]) use this notion. Then we describe three lattice-based signature schemes: the first one described in [GPV08], called the GPV signature, the "Bonsai Tree" signature introduced by [CHKP10] and finally Boyen's signature from [Boy10]. As explained before, there are other lattice-based group signatures, we choose to describe those ones as we will use them in our two group signatures constructions in Chapters 7 and 8.

3.1 Encryption

We start this section by recalling definition and security associated to public-key encryption, then we give two examples of scheme based on LWE.

3.1.1 Cryptographic definition

Public-key encryption was introduced by Diffie and Hellman [DH76]. We first recall its cryptographic definition and associated notions of security.

Definition 3.1 (Encryption scheme)**.** Let λ be the security parameter. An *encryption scheme* Π is given by three probabilistic polynomial time algorithms:

$\mathsf{KeyGen}(1^\lambda) \to (pk, sk)$. This algorithm takes λ as input and outputs a pair of keys (pk, sk) which are the public key and the associated secret key.

$\mathsf{Enc}(1^\lambda, pk, M) \to C$. The encryption algorithm takes as inputs the security parameter λ, a public key pk and a message $M \in \{0,1\}^*$. It outputs a ciphertext C.

$\mathsf{Dec}(1^\lambda, sk, C) \to \{M, \perp\}$. The decryption algorithm takes as inputs the security parameter λ, a secret key sk and a ciphertext C. It outputs either the message M, or the symbol $\perp$ if the ciphertext is deemed invalid.

This encryption protocol must be *correct*, i.e., for any λ large enough and all message $M \in \{0,1\}^*$, if $(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ then $\mathsf{Dec}(1^\lambda, sk, \mathsf{Enc}(1^\lambda, pk, M)) = M$ with probability negligibly close from 1 over the choice of the randomness used to encrypt the message.

Security. There are two main types of security used for encryption in public key cryptography: the security against chosen-plaintext attacks (IND-CPA) and the security against non-adaptive / adaptive chosen-ciphertext attack (IND-CCA1 and IND-CCA2). The security of an encryption scheme Π against an adversary $\mathcal{A}$ is defined in Figure 3.1.

1. Setup. The challenger runs $\mathsf{KeyGen}(1^\lambda)$ to generate (pk, sk), then gives pk to the adversary $\mathcal{A}$.
2. Queries. If the game is CCA, the adversary $\mathcal{A}$ can make the following queries:
   - Decryption: query to decrypt a cipher c_i, the challenger returns $M_i = \mathsf{Dec}(1^\lambda, sk, c_i)$.
3. Challenge. Adversary $\mathcal{A}$ outputs two messages M_0 and M_1, such that $\mathcal{A}$ never made a decryption query those messages. The challenger chooses a bit $b^\star \hookleftarrow U(\{0,1\})$, computes a encryption of M_b as $c = \mathsf{Enc}(1^\lambda, pk, M_{b^\star})$, and returns c to $\mathcal{A}$.
4. Restricted queries. If the game is CCA2, the adversary $\mathcal{A}$ can make the following queries:
   - Decryption: query to decrypt a cipher $c_i \neq c$, the challenger returns $M_i = \mathsf{Dec}(1^\lambda, sk, c_i)$.
5. Output. Eventually, $\mathcal{A}$ outputs a bit b'. Returns 1 if $b = b^\star$, 0 otherwise.

Figure 3.1: IND-CPA or IND-CCA security games.

The advantage of the adversary is defined by

$$\mathsf{Adv}_\Pi^{\mathsf{IND-ATK}}(\mathcal{A}) = \left| \Pr\left(\mathbf{Exp}_\Pi^{\mathsf{IND-ATK}}(\mathcal{A}) = 1 \right) - 1/2 \right|,$$

where ATK is either CPA, CCA1 or CCA2.

An encryption scheme is IND-CPA (respectively IND-CCA1 or IND-CCA2) secure if any probabilistic polynomial adversary against the scheme has a negligible advantage in this security game.

Remark 3.2. We defined an Encryption scheme for any $M \in \{0,1\}^*$, note that the size of the message can also be fixed by the scheme.

3.1.2 Regev's encryption scheme

Description. The first encryption scheme based on LWE has been introduced by Regev [Reg05]. The security parameter of this scheme is function of the parameters of LWE. We describe this scheme in Figure 3.2. All operations are modulo the integer q.

Let n, m, and q be positive integers with q prime and $m \geq 4(n+1)\log_2 q$, and α be a real in $(0, 1/(4m))$.

KeyGen: Given the parameters n, m, q and α.

- **Secret key**: a vector $\mathbf{s}$ such that $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$.
- **Public key**: a pair $(\mathbf{A}, \mathbf{b}) = (\mathbf{A}, \mathbf{As} + \mathbf{e})$, where $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$.

Encryption: Given the parameters n, m, q and α, a message $M \in \{0,1\}$ and a public key $(\mathbf{A}, \mathbf{b})$. Sample $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$ and output the ciphertext:

$$(\mathbf{r}^T\mathbf{A}, \mathbf{r}^T\mathbf{b} + \lfloor q/2 \rfloor \cdot M) \in \mathbb{Z}_q^n \times \mathbb{Z}_q.$$

Decryption: Given the parameters n, m, q and α, a ciphertext $(\mathbf{u}^T, v) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ and a secret key $\mathbf{s}$. Compute $v - \mathbf{u}^T\mathbf{s}$: the decryption is 0 is the result is closer to 0 than to $\lfloor q/2 \rfloor$, otherwise, the decryption is 1.

Figure 3.2: Regev's encryption scheme.

The principle is to give m LWE samples: $(\mathbf{A}, \mathbf{b})$ as a public key, and to keep the vector $\mathbf{s}$ as a secret key. To encrypt one bit $M \in \{0,1\}$, one chooses a vector $\mathbf{r}$ uniformly at random in $\{0,1\}^m$, which allows to choose a random subset of the rows $(\mathbf{a}_i, b_i)$ of the LWE sample. Then we add the chosen rows and add $\lfloor q/2 \rfloor \cdot M$ to the sum of the b_i's. To decrypt a ciphertext $(\mathbf{u}^T, v)$, one uses the secret key $\mathbf{s}$ to compute $v - \mathbf{u}^T\mathbf{s} = \mathbf{r}^T\mathbf{e} + \lfloor q/2 \rfloor \cdot M$. The vector $\mathbf{e}$ is sampled from a discrete Gaussian of parameter αq and then is small enough (such that $\|\mathbf{e}\| \leq \alpha q\sqrt{m}$ with overwhelming probability, by Lemma 1.36) to have $v - \mathbf{u}^T\mathbf{s}$ either close from 0, either close from $\lfloor q/2 \rfloor$, (as $\|\mathbf{r}\| \leq \sqrt{m}$, we have $\|\mathbf{r}^T\mathbf{e}\| \leq \alpha q m \leq q/4$ as $\alpha \leq 1/(4m)$). It allows to find the message M.

Remark 3.3. This scheme only allows to encrypt one bit. To encrypt a message of several bits, one can either use it several times, either use several columns $\mathbf{b}_i$ instead of one.

Security of the scheme. This scheme is IND-CPA secure under the hardness of the LWE problem. The full proof is provided by Regev in [Reg09] and sketched in [Reg10a].

First, the Leftover Hash Lemma (LHL) (see Section 1.1.5) gives that if $m \geq 4n \log q$, the distribution of the pair $(\mathbf{A}, \mathbf{r}^T\mathbf{A})$ for $\mathbf{A}$ uniform in $\mathbb{Z}_q^{m \times n}$ and $\mathbf{r}$ uniform in $\{0,1\}^m$ is statistically close to the uniform distribution in $\mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n$.

Regev shows that if there exists an adversary which can distinguish between a ciphertext of 0 and a ciphertext of 1 in polynomial time with non-negligible advantage then this adversary

can distinguish between the distributions $A_{q,\mathbf{s},D_{\mathbb{Z},\alpha q}}$ and $\mathbb{Z}_q^{n+1}$ with non-negligible advantage over the choice of $\mathbf{s}$, and then be used to solve the decisional variant of LWE. Assume that there exists a probabilistic polynomial-time adversary $\mathcal{A}$, which given a public key $(\mathbf{A}, \mathbf{b})$ sampled from $(A_{q,\mathbf{s},D_{\mathbb{Z},\alpha q}})^m$ can guess the plaintext with probability at least $1/2 + 1/\text{poly}(n)$ for non-negligible subset of secrets $\mathbf{s}$. Now assume that we give to this adversary $\mathcal{A}$ a pair $(\mathbf{A}, \mathbf{b})$ uniformly sampled in $\mathbb{Z}_q^{m\times n} \times \mathbb{Z}^m$ instead of an LWE sampled public key. According to the LHL, the distribution of $(\mathbf{A}, \mathbf{b}, \mathbf{r}^T\mathbf{A}, \mathbf{r}^T\mathbf{b})$ is essentially uniform. As a consequence the ciphertext of 0 and 1 will be statistically indistinguishable for $\mathcal{A}$.

Recall that an instance of the decisional LWE problem is a pair $(\mathbf{A}, \mathbf{b})$ sampled either from $(A_{q,\mathbf{s},D_{\mathbb{Z},\alpha q}})^m$ for a fixed $\mathbf{s}$, either from $U(\mathbb{Z}_q^{m\times n} \times \mathbb{Z}^m)$. The principle is to give to $\mathcal{A}$ an instance of decisional LWE as the public key, and to use it to encrypt a bit given to $\mathcal{A}$. If $\mathcal{A}$ guesses correctly with a sufficiently large probability, then we know it was an LWE instance, otherwise it was a random instance.

3.1.3 Dual-Regev encryption scheme

Description. In 2008, Gentry, Peikert and Vaikuntanathan [GPV08] described another encryption scheme based on LWE called Dual-Regev. The word "dual" is used to refer to the fact that this scheme is obtained by switching the Encryption and Decryption algorithms of Regev's scheme, as illustrated in figure 3.4. This encryption is used for more advanced encryptions scheme, as in the Identity Based Encryption [GPV08].

Let n, m, and q be integers with q prime and $m \geq 4(n+1)\log_2 q$, and α be in $(0, 1/(8m))$. Users share a matrix $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$.

KeyGen: Given the parameters n, m, q and α and the matrix $\mathbf{A}$.

- **Secret key**: a vector $\mathbf{r} \hookleftarrow U(\{0,1\}^m)$.
- **Public key**: a vector $\mathbf{y}^T = \mathbf{r}^T\mathbf{A} \bmod q$.

Encryption: Given the parameters, the matrix $\mathbf{A}$, a message $M \in \{0,1\}$ and a public key $\mathbf{y}$. Sample $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ and $e' \hookleftarrow D_{\mathbb{Z},\alpha q}$. The ciphertext is

$$(\mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{y}^T\mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M) \in \mathbb{Z}_q^m \times \mathbb{Z}_q.$$

Decryption: Given the parameters, the matrix $\mathbf{A}$, a ciphertext $(\mathbf{b}, c)$ and a secret key $\mathbf{r}$. Compute $c - \mathbf{r}^T\mathbf{b}$. The decryption is 0 if the result is closer to 0 than to $\lfloor q/2 \rfloor$. Otherwise the decryption is 1.

Figure 3.3: Dual Regev encryption scheme.

The principle of this scheme is to give as a public key a vector $\mathbf{y}^T = \mathbf{r}^T\mathbf{A} \bmod q$ and to keep $\mathbf{r}$ as a secret key. One then uses LWE to encrypt a message: choose a uniform $\mathbf{s}$, and the ciphertext $(\mathbf{b}, c)$ is exactly $(m+1)$ rows of a LWE sample for the matrix $\mathbf{A}$ with an additional row $\mathbf{y}$ as left member. To decrypt $(\mathbf{b}, c)$, one just has to compute:

$$c - \mathbf{r}^T\mathbf{b} = \mathbf{y}^T\mathbf{s} + e' + \lfloor q/2 \rfloor \cdot M - \mathbf{r}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = e' - \mathbf{r}^T\mathbf{e} + \lfloor q/2 \rfloor \cdot M.$$

As in Regev's encryption, the parameter α has been chosen for $e' - \mathbf{r}^T\mathbf{e}$ to be small compared to $\lfloor q/2 \rfloor$, which allows to recover the message M.

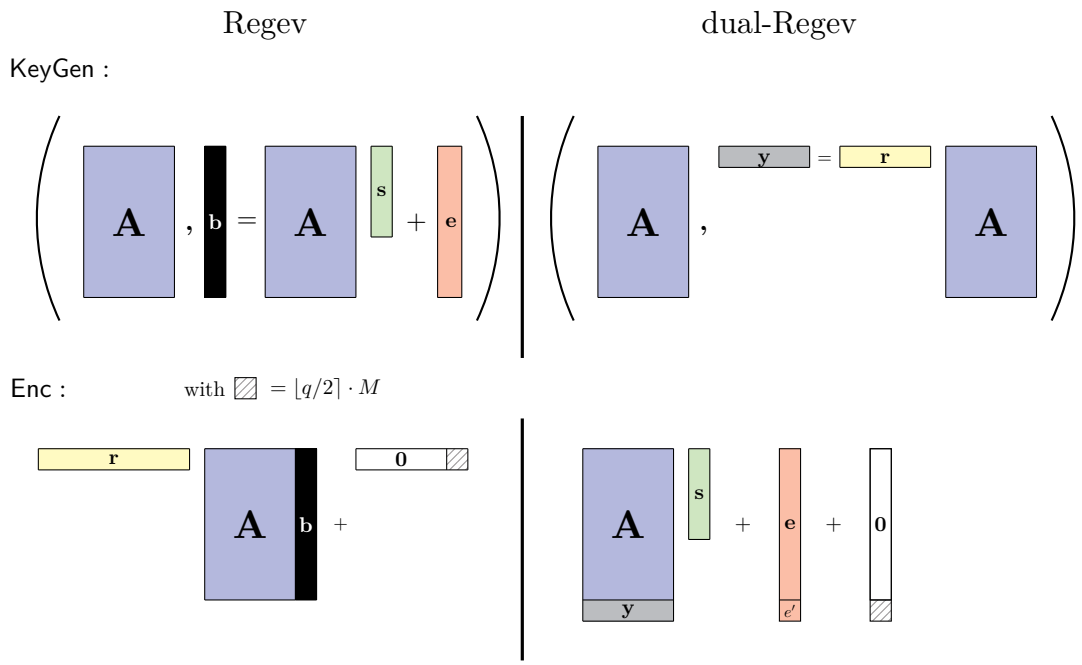

Figure 3.4: Comparison between Regev's encryption and Dual-Regev encryption.

Security. This scheme is IND-CPA secure under the hardness of the LWE problem. The full proof is provided in the original description [GPV08]. The security arguments are the same as for Regev's encryption scheme, but are used in a different order.

The LHL gives that the public key $(\mathbf{A}, \mathbf{y})$ is computationally indistinguishable from a uniform distribution. As a consequence, the pair $(\mathbf{As}+\mathbf{e}, \mathbf{y}^T\mathbf{s}+e')$ used during the encryption is essentially distributed as $(m+1)$ LWE samples $A_{q,\mathbf{s},D_{\mathbb{Z},\alpha q}}$ for a vector $\mathbf{s}$. Given a ciphertext $(\mathbf{b}, c)$, the distribution of $(\mathbf{A}, \mathbf{b}, \mathbf{y}, c)$ is computationally indistinguishable from a uniform distribution under the hardness of decisional LWE.

Remark 3.4. Those two schemes are only IND-CPA secure and not IND-CCA secure. There exists lattice-based encryption schemes which are IND-CCA secure under the hardness of LWE [ABB10a, Pei09, MP12], but we will not describe them here.

3.2 Signature

We first give the cryptographic definition of a signature scheme and associated notions of security. Then we define the notion of trapdoor lattices and we describe three lattice-based signature schemes [GPV08, CHKP10, Boy10]. The security of those signatures is based on the hardness of the SIS problem.

3.2.1 Cryptographic definition

We start with a formal definition for a signature scheme.

Definition 3.5 (Signature)**.** Let λ be the security parameter. A *signature scheme* Σ is given by three probabilistic polynomial time algorithms:

$\mathsf{KeyGen}(1^\lambda) \to (vk, sk)$. This algorithm takes λ as input and outputs a pair of keys (vk, sk) which are the verification key (public) and the associated signing key (secret).

$\mathsf{Sign}(1^\lambda, sk, M) \to \sigma$. The encryption algorithm takes as input the security parameter λ, a signing key sk and a message $M \in \{0,1\}^*$. It outputs a signature $\sigma \in \{0,1\}^*$.

$\mathsf{Verify}(1^\lambda, vk, M, \sigma) \to \{accept, reject\}$. The decryption algorithm takes as input the security parameter λ, a verification key vk, the message M, and a signature σ. It either accepts or rejects.

This signature protocol must be *correct*, i.e., for any λ large enough and all message $M \in \{0,1\}^*$, if $(vk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ then $\mathsf{Verify}(1^\lambda, vk, \mathsf{Sign}(1^\lambda, sk, M)) = accept$ with probability negligibly close from 1 over the choice of the randomness used to sign the message.

Security. There are three main types of security used for signature schemes: the (static) existential unforgeability against chosen message attack (EU-CMA) and the strong unforgeability against chosen message attack. The security of a signature scheme is defined in Figure 3.5.

1. Setup. The challenger runs $\mathsf{KeyGen}(1^\lambda)$ to generate (vk, sk), then gives vk to the adversary $\mathcal{A}$.
2. Queries. Adversary $\mathcal{A}$ can make the following queries:
   - Sign: query to sign a message M_i, the challenger returns $\sigma_i = \mathsf{Sign}(1^\lambda, sk, M_i)$.
3. Forgery: Eventually, $\mathcal{A}$ outputs a message M^* and a signature σ^*. The adversary wins the game if $\mathsf{Verify}(1^\lambda, vk, M^*, \sigma^*) = accept$ and $M^* \neq M_i$ for all i.

Figure 3.5: EU-CMA security game.

In the static (by opposition to adaptive) EU-CMA security game, the attacker give the list of signature queries before receiving the verification key. In the strong unforgeability against chosen message attack variant, the adversary wins if $\mathsf{Verify}(1^\lambda, vk, M^*, \sigma^*) = accept$ and $(M^*, \sigma^*) \neq (M_i, \sigma_i)$ for all i. The advantage of the adversary is defined by

$$\mathsf{Adv}_\Sigma^{\mathsf{EU-CMA}}(\mathcal{A}) = \left| \Pr\left(\mathbf{Exp}_\Sigma^{\mathsf{EU-CMA}}(\mathcal{A}) = win \right) - 1/2 \right|.$$

A signature scheme is EU-CMA (respectively for the strong variant) secure if any probabilistic polynomial adversary against the scheme has a negligible advantage in this security game.

Remark 3.6. As for encryption scheme, we defined a signature scheme for any $M \in \{0,1\}^*$, note that the size of the message can also be fixed by the scheme.

3.2.2 Trapdoors for lattices

The following signature schemes, as many cryptographic primitives, use the notion of *full trapdoor* for lattices [Ajt99, AP11]. Note that there exists signature schemes that do not use trapdoors [Lyu12, GLP12] and that those signatures are asymptotically more efficient. Otherwise, as we mentioned before, we choose to describe those particular signatures because we use them in our constructions of Chapters 6.4.3 and 8.

Definitions and short vectors. Let $m \geq n \geq 1$ and $q \geq 2$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a vector $\mathbf{u} \in \mathbb{Z}^n$, we define the m-dimensional lattices:

$$\begin{aligned} \Lambda_q(\mathbf{A}) &= \{\mathbf{y} \in \mathbb{Z}^n : \mathbf{y} = \mathbf{x}^T \cdot \mathbf{A} \bmod q \text{ for some } \mathbf{x} \in \mathbb{Z}^m\}. \\ \Lambda_q^\perp(\mathbf{A}) &= \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{x}^T \cdot \mathbf{A} = 0 \bmod q\}, \end{aligned}$$

The lattice $\Lambda_q(\mathbf{A})$ is generated by the rows of the matrix $\mathbf{A}$, the lattice $\Lambda_q^\perp(\mathbf{A})$ contains all vectors orthogonal to the matrix $\mathbf{A}$ modulo q. Note that those two lattices are m-dimensional lattices as $q\mathbb{Z}^m \subseteq \Lambda_q(\mathbf{A}) \subseteq \mathbb{Z}^m$ and $q\mathbb{Z}^m \subseteq \Lambda_q^\perp(\mathbf{A}) \subseteq \mathbb{Z}^m$. We also note that find a short vector in those lattices, for a random matrix $\mathbf{A}$, is a hard problem: given $\mathbf{A}$, finding a non-zero short vector in $\Lambda_q^\perp(\mathbf{A})$ corresponds exactly to solve the SIS problem (and is related to the decisional LWE problem, as explained further).

A short vector in one of those two lattices may be viewed as a "partial trapdoor" for the LWE problem. For example, given $(\mathbf{A}, \mathbf{b})$ an instance of the decisional LWE problem: The goal is to distinguish between $(A_{q,\mathbf{s},D_{\alpha q}})^m$ and a uniform distribution in $\mathbb{Z}^m$. If we are given a short element $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$, i.e., such that $\mathbf{x}^T \cdot \mathbf{A} = 0 \bmod q$, then to solve the LWE problem one can simply multiply $\mathbf{b}$ by $\mathbf{x}^T$:

- If $\mathbf{b}$ is uniform in $\mathbb{Z}_q^m$, then $\mathbf{x}^T\mathbf{b}$ will also be uniform in $\mathbb{Z}_q$,
- If $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}$ with $\mathbf{e} \hookleftarrow D_{\alpha q}$, then $\mathbf{x}^T\mathbf{b} = \mathbf{x}^T(\mathbf{A}\mathbf{s} + \mathbf{e}) = \mathbf{x}^T\mathbf{e}$: as $\mathbf{x}$ and $\mathbf{e}$ are small compared to q, $\mathbf{x}^T\mathbf{e}$ is also small.

Then given a short element in $\Lambda_q^\perp(\mathbf{A})$, one can have a non-negligible advantage in solving the decisional version of the LWE problem. But this short element does not allow to solve the computational version as we cannot find the secret $\mathbf{s}$.

Short basis of $\Lambda_q^\perp(\mathbf{A})$. A full trapdoor for LWE is a short basis of $\Lambda_q^\perp(\mathbf{A})$. As we just explain, given $\mathbf{A}$ it is hard to find such a basis but on the other hand, one can jointly sample $\mathbf{A}$ and a short basis of $\Lambda_q^\perp(\mathbf{A})$ simultaneously as shown in [AP11].

Lemma 3.7 ([AP11, Th. 3.2])**.** *There exists a* PPT *algorithm* TrapGen *that takes as inputs* 1^n, 1^m *and an integer* $q \geq 2$ *with* $m \geq \Omega(n \log q)$*, and outputs a matrix* $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$ *and a basis* $\boldsymbol{T_A}$ *of* $\Lambda_q^\perp(\boldsymbol{A})$ *such that* $\boldsymbol{A}$ *is within statistical distance* $2^{-\Omega(n)}$ *to* $U(\mathbb{Z}_q^{m \times n})$*, and* $\|\widetilde{\boldsymbol{T}_{\boldsymbol{A}}}\| \leq \mathcal{O}(\sqrt{n \log q})$.

The principle is to use the LHL to sample rows of $\mathbf{T_A}$ together with rows of $\mathbf{A}$. An example is given in Figure 3.6. To add a row to $\mathbf{T_A}$, one samples a uniform $\mathbf{r}$ and then computes $\mathbf{r}^T\mathbf{A}_i$ for all i (where $\mathbf{A}_i$ are the columns of $\mathbf{A}$), and then the last row of $\mathbf{A}$ is $(-\mathbf{r}^T\mathbf{A}_i \bmod q)_{i \in [n]}$. As $\mathbf{r}$ is uniformly sampled, the new row of $\mathbf{A}$ is also uniformly distributed.

Micciancio and Peikert [MP12] recently proposed a more efficient approach for this combined task, which should be preferred in practice. Lemma 3.7 was later extended by Gordon *et al.* [GKV10] so that the columns of $\mathbf{A}$ lie within a prescribed linear vector subspace of $\mathbb{Z}_q^n$ (for q prime).

Sample a short vector given a short basis. Lemma 3.7 is often combined with the sampler from Lemma 1.24 which allows to sample Gaussian distributions with lattice support given a sufficiently short basis of the lattice.

As a consequence, sampling using GPVSample with input the short basis $\mathbf{T_A}$ of the lattice $\Lambda_q^\perp(\mathbf{A})$, and a parameter s allows to find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\mathbf{x}^T\mathbf{A} = 0 \bmod q$

$$\begin{bmatrix} 2 & -3 & 3 & 0 & -1 & 1 & 4 & -3 & 0 \\ 3 & -2 & -4 & -1 & -2 & -2 & 4 & 2 & 0 \\ 0 & 4 & 4 & -1 & 4 & -3 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 8 & 0 & 0 & 0 \\ 5 & 4 & 2 & -4 & -4 & -2 & 1 & -3 & 0 \\ -5 & 1 & 3 & -1 & -3 & 4 & 6 & 2 & 0 \\ 1 & 3 & -6 & 6 & 4 & -5 & 1 & -2 & 0 \\ -4 & 3 & -4 & -7 & -1 & -2 & 3 & -6 & 0 \\ 1 & 5 & -2 & 2 & 0 & 6 & 1 & -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 185 & 97 & 202 \\ 146 & 148 & 11 \\ 208 & 219 & 164 \\ 218 & 173 & 117 \\ 211 & 176 & 187 \\ 79 & 255 & 112 \\ 47 & 136 & 232 \\ 204 & 172 & 58 \\ 184 & 161 & 135 \end{bmatrix} = \mathbf{0} \bmod 257.$$

Figure 3.6: Example for $\mathbf{T_A}$ and $\mathbf{A}$ with $q = 257$ and $n = 3$.

and $\|\mathbf{x}\| \leq s\sqrt{m}$ with overwhelming probability (by Lemma 1.36, as $\mathbf{x}$ is sampled from a discrete Gaussian with parameter s).

Remark 3.8. We recall the the algorithm GPVSample has a condition on the parameter s: it requires $s \geq \|\widetilde{\mathbf{B}}\| \cdot \sqrt{\log n}$, where $\mathbf{B}$ is the basis of the support lattice. Then to sample a short element in a lattice, one need a somewhat short basis of this lattice. The particular case of $\mathbb{Z}^n$ for $n > 0$ allows to find short elements in $\mathbb{Z}^n$.

Now given $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, $\mathbf{u} \in \mathbb{Z}_q^m$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$, it is possible to find a short vector $\mathbf{x}$ such that $\mathbf{x}^T\mathbf{A} = \mathbf{u}^T \bmod q$, we proceed as follows:

- Take any $\mathbf{x}_0$ such that $\mathbf{x}_0^T\mathbf{A} = \mathbf{u}^T \bmod q$ (using linear algebra),
- Sample $\mathbf{x}_1 \hookleftarrow D_{\Lambda_q^\perp(\mathbf{A}),s,\mathbf{x}_0}$, we have with overwhelming probability:

$$\begin{cases} \mathbf{x}_1 \in \Lambda_q^\perp(\mathbf{A}) \\ \|\mathbf{x}_1 - \mathbf{x}_0\| \leq \sqrt{m}s \end{cases},$$

- Return $\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_0$, we have with overwhelming probability:

$$\begin{cases} \mathbf{x}^T\mathbf{A} = \mathbf{u}^T \bmod q \\ \|\mathbf{x}\| \leq \sqrt{m}s \end{cases}.$$

The following lemma states that an element sampled from a discrete Gaussian distribution with a parameter large enough multiplied by a matrix $\mathbf{A}$ uniform in $\mathbb{Z}_q^{m \times n}$ will give an almost uniform vector. This lemma is very used for signature scheme as it will give the same distribution for $(\mathbf{u}, \mathbf{x})$ if one sample a vector $\mathbf{u}$ uniform and use a trapdoor to find a short $\mathbf{x}$ such that $\mathbf{u}^T = \mathbf{x}^T\mathbf{A} \bmod q$ and if one sample a small vector $\mathbf{x}$ in $\mathbb{Z}^m$ and then compute $\mathbf{u}^T = \mathbf{x}^T\mathbf{A} \bmod q$.

Lemma 3.9 ([GPV08, Corollary 5.4])**.** *Let n and $q \geq 2$ be integers. Let $m \geq 2n \log q$, and $\sigma \geq \omega(\sqrt{\log m})$. For all but a $2q^{-n}$ fraction of all $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, for $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, the distribution of $\mathbf{u}^T = \mathbf{x}^T \cdot \mathbf{A} \bmod q$ is statistically close to uniform over $\mathbb{Z}_q^n$. Moreover, let $\boldsymbol{t}$ be an arbitrary element such that $\boldsymbol{u}^T = \boldsymbol{t}^T \cdot \boldsymbol{A} \bmod q$, then the conditional distribution of $\mathbf{x}$ given $\mathbf{u}$ is exactly $\boldsymbol{t} + D_{\Lambda_q^\perp(\mathbf{A}),\sigma,-\boldsymbol{t}}$.*

Randomize and extend. Cash *et al.* [CHKP10] showed how to use GPVSample to randomize the basis of a given lattice. The following statement is obtained by using Lemma 1.24 in the proof of [CHKP10].

Lemma 3.10 (Adapted from [CHKP10, Le. 3.3])**.** *There exists a* PPT *algorithm* RandBasis *that takes as inputs a basis* $\boldsymbol{B}$ *of a lattice* $L \subseteq \mathbb{Z}^n$ *and a rational* $\sigma \geq \|\widetilde{\boldsymbol{B}}\| \cdot \Omega(\sqrt{\log n})$*, and outputs a basis* $\boldsymbol{C}$ *of* L *satisfying* $\|\widetilde{\boldsymbol{C}}\| \leq \sqrt{n}\sigma$ *with probability* $\geq 1 - 2^{-\Omega(n)}$*. Further, the distribution of* $\boldsymbol{C}$ *is independent of the input basis* $\boldsymbol{B}$*.*

Finally, [CHKP10] also gave an algorithm that extends a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ to a trapdoor of any $\mathbf{B} \in \mathbb{Z}_q^{m' \times n}$ whose top $m \times n$ submatrix is $\mathbf{A}$.

Lemma 3.11 ([CHKP10, Le. 3.2])**.** *There exists a* PPT *algorithm* ExtBasis *that takes as inputs a matrix* $\boldsymbol{B} \in \mathbb{Z}_q^{m' \times n}$ *whose first* m *rows span* $\mathbb{Z}_q^n$*, and a basis* $\boldsymbol{T_A}$ *of* $\Lambda_q^\perp(\boldsymbol{A})$ *where* $\boldsymbol{A}$ *is the top* $m \times n$ *submatrix of* $\boldsymbol{B}$*, and outputs a basis* $\boldsymbol{T_B}$ *of* $\Lambda_q^\perp(\boldsymbol{B})$ *with* $\|\widetilde{\boldsymbol{T}_B}\| \leq \|\widetilde{\boldsymbol{T}_A}\|$*.*

Given $\mathbf{A} \in \mathbb{Z}^{m \times n}$, a basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$ and a matrix $\mathbf{B}$ where $\mathbf{A}$ is the top $m \times n$ submatrix of $\mathbf{B}$, the principle is to construct a basis $\mathbf{T_B}$ of $\Lambda_q^\perp(\mathbf{A})$ as showed in Figure 3.7 where the $\mathbf{s}_i$ are chosen such that $-\mathbf{s}_i^T \cdot \mathbf{A} = \mathbf{b}_i^T \bmod q$ for all $m+1 \leq i \leq m'$.

$$\left[\begin{array}{c|c} \mathbf{T_A} & \mathbf{0} \\ \hline \mathbf{s}_{m+1}^T & \\ \cdots & I_{m'-m+1} \\ \mathbf{s}_{m'}^T & \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{A} \\ \hline \mathbf{b}_{m+1}^T \\ \cdots \\ \mathbf{b}_{m'}^T \end{array}\right] = \mathbf{0} \bmod q,$$

Figure 3.7: Extend a trapdoor: construction of $\mathbf{T_B}$.

Note that $\|\widetilde{\mathbf{T}}_\mathbf{B}\| \leq \|\widetilde{\mathbf{T}}_\mathbf{A}\|$ even if the $\mathbf{s}_i$ are not short, as $\widetilde{\mathbf{T}}_\mathbf{A}$ is full-rank and

$$\widetilde{\mathbf{T}}_\mathbf{B} = \left[\begin{array}{c|c} \widetilde{\mathbf{T}}_\mathbf{A} & \mathbf{0} \\ \hline \vdots & \\ \mathbf{0} & I_{m'-m+1} \\ \vdots & \end{array}\right].$$

Adding constraints. Another interesting property showed in [GKV10] is that one can also sample a basis $\mathbf{A}$ and trapdoor $\mathbf{T_A}$ given matrix $\mathbf{B}$ and $\mathbf{c}$ and conditioned to the fact that $\mathbf{B}^T \cdot \mathbf{A} = \mathbf{C} \bmod q$.

Lemma 3.12. *There exists a* PPT *algorithm* SuperSamp *that takes as inputs matrices* $\boldsymbol{B} \in \mathbb{Z}_q^{m \times n}$ *and* $\boldsymbol{C} \in \mathbb{Z}_q^{n \times n}$ *such that the rows of* $\boldsymbol{B}$ *span* $\mathbb{Z}_q^n$*,* $m \geq n \geq 1$*, and* $q \geq 2$ *prime such that* $m \geq \Omega(n \log q)$*. It outputs* $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$ *and a basis* $\boldsymbol{T_A}$ *of* $\Lambda_q^\perp(\boldsymbol{A})$ *such that* $\boldsymbol{A}$ *is within statistical distance* $2^{-\Omega(n)}$ *to* $U(\mathbb{Z}_q^{m \times n})$ *conditioned on* $\boldsymbol{B}^T \cdot \boldsymbol{A} = \boldsymbol{C}$*, and* $\|\widetilde{\boldsymbol{T}_A}\| \leq \mathcal{O}(\sqrt{mn \log q \log m})$*.*

Proof. The algorithm is a simple extension of the one in [GKV10]. It first partitions $\mathbf{B}$ into matrices $\mathbf{B}_1 \in \mathbb{Z}_q^{m_1 \times n}$ and $\mathbf{B}_2 \in \mathbb{Z}_q^{n \times n}$, with $m_1 = m - n$, such that $\mathbf{B}_2$ is invertible over $\mathbb{Z}_q$ and $\mathbf{B}^T = [\mathbf{B}_1^T | \mathbf{B}_2^T]$. Such a partition can always be found by re-arranging the rows of $\mathbf{B}$ if necessary. The execution of $\mathsf{GenSuperSamp}(1^n, 1^m, q, \mathbf{B}, \mathbf{C})$ then proceeds with the following steps.

1. Generate $(\mathbf{A}_1, \mathbf{T}_1) \leftarrow \mathsf{TrapGen}(1^n, 1^{m_1}, q)$. Return $\perp$ if the rows of $\mathbf{A}_1 \in \mathbb{Z}_q^{m_1 \times n}$ do not span $\mathbb{Z}_q^n$.
2. Compute $\mathbf{A}_2 = \mathbf{B}_2^{-T} \cdot (\mathbf{C} - \mathbf{B}_1^T \cdot \mathbf{A}_1) \bmod q$. Note that

$$\mathbf{A} = \left[\frac{\mathbf{A}_1}{\mathbf{A}_2}\right] \text{ satisfies } \mathbf{B}^T \cdot \mathbf{A} = \mathbf{C} \bmod q.$$

3. Extend $\mathbf{T}_1 \in \mathbb{Z}^{m_1 \times m_1}$ to have a basis $\mathbf{T} \in \mathbb{Z}^{m \times m}$ for $\mathbf{A}$ using the basis delegation algorithm from Lemma 3.11. Then, re-randomize $\mathbf{T}'$ to obtain $\mathbf{T}''$ using the basis randomization algorithm $\mathsf{RandBasis}$.

The rest of the proof is exactly identical to the proof of Lemma 4 in [GKV10]. □

Now that we described all the properties of trapdoors that we will use, we will start describing the three signature schemes.

3.2.3 GPV signature

Description. The following signature scheme is described in [GPV08, Section 5.3.2].

Let n, m and q be integers such that $m = \Omega(n \log q)$, and let $\tilde{L} = O(\sqrt{n \log q})$ and $s = \tilde{L} \cdot \omega(\sqrt{\log n})$. Choose a hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_q^n$ which will be modelled as a random oracle.

$\mathsf{KeyGen}(1^n, 1^m) \to (vk, sk)$. Given the parameters n, m, q and s.
Generates $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, where $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is negligibly close from uniform and $\mathbf{T_A}$ is a basis for $\Lambda_q^\perp(\mathbf{A})$ (with $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$).

- **Verification key**: $vk = \mathbf{A}$.
- **Signing key**: $sk = \mathbf{T_A}$.

$\mathsf{Sign}(1^n, 1^m, sk, M \in \{0,1\}^*) \to \mathbf{v}$. Given the parameters n, m, q and s, a signing key $sk = \mathbf{T_A}$ and a message $M \in \{0,1\}^*$. Generate

$$\mathbf{v} \leftarrow \mathsf{GPVSample}(\mathbf{T_A}, \mathcal{H}(M), s),$$

i.e., a short $\mathbf{v}$ such that $\mathbf{v}^T\mathbf{A} = \mathcal{H}(M) \bmod q$. Output $\mathbf{v}$.

$\mathsf{Verify}(1^n, 1^m, vk, \mathbf{v}) \to \{accept, reject\}$. Given the parameters n, m, q and s, a verification key $vk = \mathbf{A}$ and a signature $\mathbf{u}$. Accept if, and only if, $\mathbf{v} \neq 0$, $\|\mathbf{v}\| \leq s \cdot \sqrt{m}$ and $\mathbf{v}^T\mathbf{A} = \mathcal{H}(M) \bmod q$.

Figure 3.8: GPV signature scheme.

The principle of this scheme is to generate a public basis $\mathbf{A}$ together with a trapdoor $\mathbf{T_A}$ which stay secret. To sign a message, the hash function $\mathcal{H}$ and the message M are used to create a random $\mathbf{u} = \mathcal{H}(M)$. Given this random element (which depends on the message), a signature will be a short vector $\mathbf{v}$ such that $\mathbf{v}^T\mathbf{A} = \mathbf{u} \bmod q$. We saw in the previous section that with the trapdoor of $\mathbf{A}$ such a vector is easy to find. Finally to verify a signature, one cheeks if it satisfies all the corresponding conditions. Note that this scheme is correct only with high probability.

Security. As proven in [GPV08], this signature is strongly EU-CMA secure under the hardness of the ISIS or SIS problem in the Random Oracle Model. We sketch here the proof in [GPV08]. If we assume that there exists an adversary $\mathcal{A}$ which breaks the EU-CMA security with probability ε, we construct an adversary $\mathcal{S}$ that breaks the SIS problem with probability negligibly close to ε.

Given a public key, the adversary $\mathcal{S}$ simulates the random oracle and the signing queries for $\mathcal{A}$. We also assume, without loss of generality, that $\mathcal{A}$ queries H on every message m before making a signing query on m. If $\mathcal{A}$ queries H on M: $\mathcal{S}$ sample $\mathbf{u}_M$ from $D_{\mathbb{Z}^m,s}$ (which is a short element), and compute $\mathbf{t}_M = \mathbf{u}_m^T \cdot \mathbf{A} \bmod q$. It stores $(M, \mathbf{u}_m)$ and outputs $H(M) = \mathbf{t}_M$. If $\mathcal{A}$ queries Sign on M: $\mathcal{S}$ looks in the local storage for $(M, \mathbf{u}_M)$ and returns $\mathbf{u}_M$ (which is statistically close to a real signature, see Lemma 3.9). When $\mathcal{A}$ produces a forgery $(M^*, \mathbf{u}^*)$, $\mathcal{S}$ finds $(M^*, \mathbf{u}_{M^*})$ in the local storage, and compute $\mathbf{u} = \mathbf{u}^* - \mathbf{u}_{M^*}$. If $\mathcal{A}$ made a signature query on M^*, $\mathbf{u}$ is non-zero because $(M^*, \mathbf{u}^*)$ is a forgery, otherwise $\mathcal{A}$ made a hash query on M^*, so $\mathcal{S}$ computed $\mathbf{u}_{M^*}$ by sampling it from a discrete Gaussian, and it has min-entropy $\omega(\log n)$ by Lemma 1.35, thus $\mathbf{u}^* \neq \mathbf{u}_{M^*}$. In both cases, it is a solution to the SIS problem.

3.2.4 Bonsai signature

Cash et al. [CHKP10] introduced the first signature scheme that is secure in the standard model under the hardness of SIS. Their scheme relies on a novel structure of random hard lattices: the Bonsai tree.

Description. The following signature scheme is described in [CHKP10, Section 4.2].

Let n, m and q be positive integers such that $m = \Omega(n \log q)$, and let $\tilde{L} = \Omega(\sqrt{n \log q})$ and $s = \tilde{L} \cdot \omega(\sqrt{\log n})$. Also let ℓ be the message length.

$\mathsf{KeyGen}(1^n, 1^m) \to (vk, sk)$. Given the parameters n, m, q, s and ℓ.

Generates $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, where $\mathbf{A}_0 \in \mathbb{Z}^{m \times n}$ is negligibly close from uniform and $\mathbf{T}_{\mathbf{A}_0}$ is a basis for $\Lambda_q^\perp(\mathbf{A}_0)$ (with $\|\widetilde{\mathbf{T}}_{\mathbf{A}_0}\| \leq \mathcal{O}(\sqrt{n \log q})$). Then for each $(b, j) \in \{0,1\} \times [\ell]$, sample independent $\mathbf{A}_j^{(b)} \hookleftarrow U(\mathbb{Z}^{m \times n})$.

- **Verification key**: A pair $vk = (\mathbf{A}_0, \{\mathbf{A}_j^{(b)}\}_{(b,j) \in \{0,1\} \times [\ell]})$.
- **Signing key**: A pair $sk = (\mathbf{T}_{\mathbf{A}_0}, vk)$.

$\mathsf{Sign}(1^n, 1^m, sk, M \in \{0,1\}^\ell) \to \mathbf{v}$. Given the parameters n, m, q, s and ℓ, a signing key $sk = (\mathbf{T}_{\mathbf{A}_0}, vk)$ and a message $M \in \{0,1\}^\ell$. Let

$$\mathbf{A}_M = \left[\begin{array}{c} \mathbf{A}_0 \\ \hline \mathbf{A}_1^{(M[1])} \\ \hline \cdots \\ \hline \mathbf{A}_\ell^{(M[\ell])} \end{array}\right] \in \mathbb{Z}_q^{(\ell+1)m \times n}.$$

where $M[1], \ldots, M[\ell]$ are the bits of M. Generate

$$\mathbf{v} \leftarrow \mathsf{GPVSample}(\mathsf{ExtBasis}(\mathbf{T}_{\mathbf{A}_0}, \mathbf{A}_M), 0, s),$$

i.e., a short $\mathbf{v}$ such that $\mathbf{v}^T \mathbf{A}_M = 0 \bmod q$. Output $\mathbf{v}$.

$\mathsf{Verify}(1^n, 1^m, vk, \mathbf{v}) \to \{accept, reject\}$. Given the parameters n, m, q, s and ℓ, a verification key $vk = (\mathbf{A}_0, \{\mathbf{A}_j^{(b)}\})$ and a signature $\mathbf{v}$. Let $\mathbf{A}_M$ be as above. Accept if, and only if, $\mathbf{v} \neq 0$, $\|\mathbf{v}\| \leq s \cdot \sqrt{(\ell+1)m}$, and $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A}_M)$.

Figure 3.9: Bonsai signature scheme.

If the message space is $\{0,1\}^\ell$, then the signer publishes a Bonsai tree, that is a matrix $\mathbf{A} = \left[\mathbf{A}_0 | \mathbf{A}_1^{(0)} | \mathbf{A}_1^{(1)} | \dots | \mathbf{A}_\ell^{(0)} | \mathbf{A}_\ell^{(1)}\right]^T \in \mathbb{Z}_q^{(2\ell+1)m \times n}$, where the "root" $\mathbf{A}_0 \in \mathbb{Z}_q^{m \times n}$ is generated together with a trapdoor, and the "main tree" $\left[\mathbf{A}_1^{(0)} | \mathbf{A}_1^{(1)} | \dots | \mathbf{A}_\ell^{(0)} | \mathbf{A}_\ell^{(1)}\right]$ is uniformly random in $\mathbb{Z}_q^{2\ell m \times n}$. Then, to sign a message $M = M[1] \dots M[\ell] \in \{0,1\}^\ell$, the signer uses the secret trapdoor to produce a signature $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$, which is a small vector satisfying $\mathbf{z}^T \cdot \mathbf{A}_M = 0 \bmod q$, where $\mathbf{A}_M$ is a subtree defined by M. An example is given in Figure 3.10.

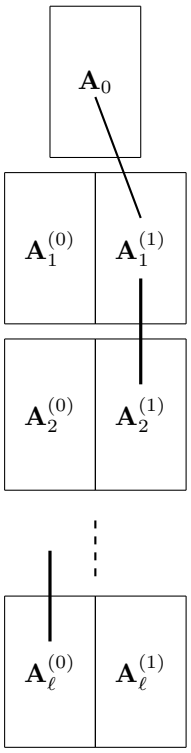

Figure 3.10: Bonsai Tree $\mathbf{A}_M$ for $M = 11 \cdots 00$.

Security. As shown in [CHKP10], this signature scheme is static EU-CMA secure under the hardness of the SIS problem. As for the GPV signature one can see that forging a signature for a given message without the trapdoor is exactly solving the SIS problem for the matrix $\mathbf{A}_M$.

Rückert [Rüc10b] later demonstrated that the Bonsai signature can be modified to satisfy the *strong* unforgeability security level. In his proposal, the public key additionally contains a vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$, and the signature is a small vector $\mathbf{v}$ satisfying $\mathbf{v}^T \cdot \mathbf{A}_M = \mathbf{u}^T \bmod q$.

3.2.5 Boyen's signature

Boyen [Boy10] introduced another signature scheme that is secure in the standard model. We describe here a variant, given in [MP12, Se. 6.2].

As in the Bonsai Tree signature, the signer and the verifier compute a matrix $\mathbf{A}_M$ which depends on the message M and on $\ell + 1$ (instead of $2\ell + 1$ in the Bonsai Tree) public matrices. Note that compared to the one in the previous signature, this matrix is smaller (it has size $2m \times n$

Let n, m and q be positive integers such that $m = \Omega(n \log q)$, $\tilde{L} = \Omega(\sqrt{n \log q})$ and $s = \tilde{L} \cdot \omega(\sqrt{\log n})$ and let ℓ be the message length.

$\mathsf{KeyGen}(1^\lambda) \to (vk, sk)$. Given the parameters n, m, q, s and ℓ.
Generates $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, where $\mathbf{A} \in \mathbb{Z}^{m \times n}$ is negligibly close from uniform and $\mathbf{T_A}$ is a basis for $\Lambda_q^\perp(\mathbf{A})$ (with $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$). Then for each j from 0 to ℓ, sample independent $\mathbf{A}_j \hookleftarrow U(\mathbb{Z}^{m \times n})$.

* **Verification key**: A pair $vk = (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell})$.
* **Signing key**: A pair $sk = (\mathbf{T_A}, vk)$.

$\mathsf{Sign}(1^\lambda, sk, M \in \{0,1\}^\ell) \to \mathbf{v}$. Given the parameters n, m, q, s and ℓ, a signing key $sk = (\mathbf{T}_{\mathbf{A}_0}, vk)$ and a message $M \in \{0,1\}^\ell$. Let

$$\mathbf{A}_M = \left[\frac{\mathbf{A}}{\mathbf{A}_0 + \sum_{j=1}^{\ell} M[j]\mathbf{A}_j} \right] \in \mathbb{Z}_q^{2m \times n},$$

where $M[1], \ldots, M[\ell]$ are the bits of M. Generate

$$\mathbf{v} \leftarrow \mathsf{GPVSample}(\mathsf{ExtBasis}(\mathbf{T_A}, \mathbf{A}_M), 0, s),$$

i.e., a short $\mathbf{v}$ such that $\mathbf{v}^T \mathbf{A}_M = 0 \bmod q$. Output $\mathbf{v}$.

$\mathsf{Verify}(1^n, 1^m, vk, \mathbf{v}) \to \{accept, reject\}$. Given the parameters n, m, q, s and ℓ, a verification key $vk = (\mathbf{A}_0, \{\mathbf{A}_j^{(b)}\})$ and a signature $\mathbf{v}$. Let $\mathbf{A}_M$ be as above. Accept if, and only if, $\mathbf{v} \neq 0$, $\|\mathbf{v}\| \leq s \cdot \sqrt{2m}$, and $\mathbf{v} \in \Lambda_q^\perp(\mathbf{A}_M)$.

Figure 3.11: Boyen's signature scheme.

instead of $(\ell + 1)m \times n$, where ℓ may be large). Given this matrix $\mathbf{A}_M$, the principle is still the same: use the trapdoor of the root $\mathbf{A}$ to find a trapdoor of $\mathbf{A}_M$ and then, given this trapdoor, find a short vector in $\Lambda_q^\perp(\mathbf{A}_M)$.

Security. This signature scheme is EU-CMA secure based on the hardness of the SIS problem [Boy10].

Part Two

Worst-Case to Average-Case Reductions for Lattice Problems

Starting with Ajtai [Ajt96] and with Regev [Reg05], the hardness results of the Small Integer Solution problem and the Learning With Errors problem are the foundation of lattice-based cryptography. Indeed the security of a major part of the existing cryptosystems is based on those two problems and on their variants. In this second part of the thesis, we describe several worst-case to average-case reductions for variants of the SIS and the LWE problem.

Chapter 4 is dedicated to the Learning With Error problem. The results of this chapter have been published in [BLP+13], which is a joint work with Z. Brakerski, C. Peikert, O. Regev and D. Stehlé. We show that LWE is *classically* at least as hard as standard worst-case lattice problems, even with polynomial modulus. Previously, this was only known under *quantum* reductions or for an exponential modulus. Our techniques capture a tradeoff between the dimension and the modulus of LWE instances, leading to a much better understanding of the landscape of the problem. The proof is inspired by techniques from several recent cryptographic constructions, most notably fully homomorphic encryption schemes.

We then study two variants of SIS and LWE. The efficiency of schemes based on SIS and LWE can be drastically improved by switching the hardness assumptions to the more compact Ring-SIS and Ring-LWE problems. However, this change of hardness assumptions comes along with a possible security weakening: SIS and LWE are known to be at least as hard as standard (worst-case) problems on euclidean lattices, whereas Ring-SIS and Ring-LWE are only known to be at least as hard as their restrictions to special classes of ideal lattices, corresponding to ideals of some polynomial rings. Chapter 5 is a joint work with D. Stehlé and is derived from [LS]. In this chapter, we define the Module-SIS and Module-LWE problems , which bridge SIS with Ring-SIS, and LWE with Ring-LWE, respectively. We prove that these average-case problems are at least as hard as standard lattice problems restricted to module lattices (which themselves bridge arbitrary and ideal lattices). As these new problems enlarge the toolbox of the lattice-based cryptographer, they could prove useful for designing new schemes. Importantly, the worst-case to average-case reductions for the module problems are (qualitatively) sharp, in the sense that there exist converse reductions. This property is not known to hold in the context of Ring-SIS/Ring-LWE: Ideal lattice problems could reveal easy without impacting the hardness of Ring-SIS/Ring-LWE.

Chapter 4

Classical Hardness of LWE

To summarize the hardness results of Chapter 2, which pre-date the work described in this chapter (corresponding to a joint work with Z. Brakerski, C. Peikert, O. Regev, and D. Stehlé, published in [BLP+13]), the existence of an efficient algorithm for LWE with polynomial modulus was only known to imply an efficient *quantum* algorithm for lattice problems, or an efficient classical algorithm for a non-standard lattice problem. While both consequences are unlikely, they are arguably not as earth-shattering as an efficient classical algorithm for lattice problems. Hence, some concern about the hardness of LWE persisted, tainting the plethora of cryptographic applications based on it.

We provide the first classical hardness reduction of LWE with polynomial modulus. Our reduction is the first to show that the existence of an efficient classical algorithm for LWE with any subexponential modulus would indeed have earth-shattering consequences: it would imply an efficient algorithm for worst-case instances of standard lattice problems.

Theorem 4.1 (Informal)**.** *Solving n-dimensional LWE with $\mathrm{poly}(n)$ modulus implies an equally efficient solution to a worst-case lattice problem in dimension $\sqrt{n}$.*

As a result, we establish the hardness of all known applications of polynomial-modulus LWE based on classical worst-case lattice problems, previously only known under a quantum assumption. This result is formally given in Theorem 4.2.

Techniques. Even though our main theorem has the flavor of a statement in computational complexity, its proof crucially relies on a host of ideas coming from recent progress in cryptography, most notably recent breakthroughs in the construction of fully homomorphic encryption schemes.

At a high level, our main theorem is a "modulus reduction" result: we show a reduction from LWE with large modulus q and dimension n to LWE with (small) modulus $p = \mathrm{poly}(n)$ and dimension $n \log_2 q$. Theorem 4.1 now follows from the main result in [Pei09], which shows that the former problem with $q = 2^n$ is as hard as n-dimensional GapSVP. We note that the increase in dimension from n to $n \log_2 q$ is to be expected, as it essentially preserves the number of possible secrets (and hence the running time of the naive brute-force algorithm).

The main idea in modulus reduction is to map $\mathbb{Z}_q$ into $\mathbb{Z}_p$ through the naive mapping that sends any $a \in \{0, \ldots, q-1\}$ to $\lfloor pa/q \rfloor \in \{0, \ldots, p-1\}$. This basic idea is confounded by two issues. The first is that if carried out naively, this transformation introduces rounding artifacts into LWE, ruining the distribution of the output. We resolve this issue by using a more careful Gaussian randomized rounding procedure (Section 4.2). A second serious issue is that in order for the rounding errors not to be amplified when multiplied by the LWE secret $\mathbf{s}$, it is essential to

assume that $\mathbf{s}$ has small coordinates. A major part of our reduction (Section 4.3) is therefore dedicated to showing a reduction from LWE (in dimension n) with arbitrary secret in $\mathbb{Z}_q^n$ to LWE (in dimension $n \log_2 q$) with a secret chosen uniformly over $\{0,1\}$. This follows from a careful hybrid argument (Section 4.3.3) combined with a hardness reduction to the so-called "extended-LWE" problem, which is a variant of LWE in which we have some control over the error vector (Section 4.3.2). We stress that even though our proof is inspired by and has analogues in the cryptographic literature, the details of the reductions are very different.

In particular, the idea of modulus reduction plays a key role in recent work on fully homomorphic encryption schemes, giving a way to control the noise growth during homomorphic operations [BV11, BGV12, Bra12]. However, since the goal there is merely to preserve the functionality of the scheme, their modulus reduction can be performed in a rather naive way similar to the one outlined above, and so the output of their procedure does not constitute a valid LWE instance. In our reduction we need to perform a much more delicate modulus reduction, which we do using Gaussian randomized rounding, as mentioned above.

The idea of reducing LWE to have a $\{0,1\}$ secret also exists already in the cryptographic literature: precisely such a reduction was shown by Goldwasser et al. [GKPV10] who were motivated by questions in leakage-resilient cryptography. Their reduction, however, incurred a severe blow-up in the noise rate, making it useless for our purposes. In more detail, not being able to faithfully reproduce the LWE distribution in the output, they resort to hiding the faults in the output distribution under a huge independent fresh noise, in order to make it close to the correct one. The trouble with this "noise flooding" approach is that the amount of noise one has to add depends on the running time of the algorithm solving the target $\{0,1\}$-LWE problem, which in turn forces the modulus to be equally big. So while in principle we could use the reduction from [GKPV10] (and shorten our proof by about a half), this would lead to a qualitatively much weaker result: the modulus and the approximation ratio for the worst-case lattice problem would both grow with the running time of the $\{0,1\}$-LWE algorithm. In particular, we would not be able to show that for some fixed polynomial modulus, LWE is a hard problem; instead, in order to capture all polynomial time algorithms, we would have to take a super-polynomial modulus, and rely on the hardness of worst-case lattice problem to within super-polynomial approximation factors. In contrast, with our reduction, the modulus and the approximation ratio both remain fixed independently of the target $\{0,1\}$-LWE algorithm.

As mentioned above, our alternative to the reduction in [GKPV10] is based on a hybrid argument combined with a new hardness reduction for the *extended-LWE* problem, which is a variant of LWE in which in addition to the LWE samples, we also get to see the inner product of the vector of error terms with a vector $\mathbf{z}$ of our choosing. This problem has its origins in the cryptographic literature, namely in the work of O'Neill, Peikert, and Waters [OPW11] on (bi)deniable encryption and the later work of Alperin-Sheriff and Peikert [ASP12] on key-dependent message security. The hardness reductions included in those papers are not sufficient for our purposes, as they cannot handle large moduli or error terms, which is crucial in our setting. We therefore provide an alternative reduction which is conceptually much simpler, and essentially subsumes both previous reductions. Our reduction works equally well with exponential moduli and correspondingly long error vectors, a case earlier reductions could not handle.

Broader perspective. As a byproduct of the proof of Theorem 4.1, we obtain several results that shed new light on the hardness of LWE. Most notably, our modulus reduction result in Section 4.2 is actually far more general, and can be used to show a "modulus expansion/dimension reduction" tradeoff. Namely, it shows a reduction from LWEin dimension n and modulus p to LWE in dimension n/k and modulus p^k (see Corollary 4.6). Combined with our modulus reduction, this has the following interesting consequence: the hardness of n-dimensional LWE with modulus q is

a function of the quantity $n \log_2 q$. In other words, varying n and q individually while keeping $n \log_2 q$ fixed essentially preserves the hardness of LWE.

Although we find this statement quite natural (since $n \log_2 q$ represents the number of bits in the secret), it has some surprising consequences. One is that n-dimensional LWE with modulus 2^n is essentially as hard as n^2-dimensional LWE with polynomial modulus. As a result, n-dimensional LWE with modulus 2^n, which was shown in [Pei09] to be as hard as n-dimensional lattice problems using a classical reduction, is actually as hard as n^2-dimensional lattice problems using a quantum reduction. The latter is presumably a much harder problem, requiring $\exp(\widetilde{\Omega}(n^2))$ time to solve. This corollary highlights an inherent quadratic loss in the classical reduction of [Pei09] (and as a result also our Theorem 4.1) compared to the quantum one in [Reg09].

A second interesting consequence is that 1-dimensional LWE with modulus 2^n is essentially as hard as n-dimensional LWE with polynomial modulus. The 1-dimensional version of LWE is closely related to the Hidden Number Problem of Boneh and Venkatesan [BV96]. It is also essentially equivalent to the Ajtai-Dwork-type [AD97] cryptosystem in [Reg], as follows from simple reductions similar to the one in the appendix of [Reg10a]. Moreover, the 1-dimensional version can be seen as a special case of the Ring-LWE problem introduced in [LPR10] (for ring dimension 1, i.e., ring equal to $\mathbb{Z}$). This allows us, via the ring switching technique from [GHPS12], to obtain the first hardness proof of Ring-LWE, with arbitrary ring dimension and exponential modulus, under the hardness of problems on general lattices (as opposed to just ideal lattice problems). In addition, this leads to the first hardness proof for the Ring-SIS problem [LM06, PR06] with exponential modulus under the hardness of general lattice problems, via the standard LWE-to-SIS reduction. (We note that since both results are obtained by scaling up from a ring of dimension 1, the hardness does not improve as the ring dimension increases.)

A final interesting consequence of our reductions is that (the decision form of) LWE is hard with an arbitrary huge modulus, e.g., a prime; see Corollary 4.5. Previous results (e.g., [Reg09, Pei09, MM11, MP12]) required the modulus to be *smooth*, i.e., all its prime divisors had to be polynomially bounded.

4.1 Classical Hardness of LWE

In this section we give the formal result that we obtain about the classical hardness of the Learning with Errors problem and we sketch the proof of this result.

Theorem 4.2. *Let $\varepsilon(n) = n^{-\omega(1)}$, $\alpha \in (0,1)$ and $q \geq 2$ such that $\alpha q \geq \sqrt{n}$. There is a probabilistic polynomial time reduction from solving* Gap-SVP$_{\sqrt{n},\gamma}$ *in polynomial time (in the worst case, with high probability) to solving* LWE$_{n,m,q,\alpha}$ *in polynomial time with non-negligible probability, for any $m(n) \leq \mathrm{poly}(n)$ and $\gamma(n)$ such that*

$$\gamma \geq \frac{\sqrt{n}}{\sqrt{10}\alpha \cdot \sqrt{\log n}}$$

For $\alpha = \frac{1}{\mathrm{poly}(n)}$ and $q \geq O(\sqrt{n}/\alpha)$, we obtain $\gamma = \mathrm{poly}(n)$.

To prove this result, we proceed by a sequence of reductions described in Figure 4.1. We omit the number m of samples in the notation of LWE as it is the same for the three LWE variants.

In this figure, the first reduction is the one of Peikert [Pei09] from GapSVP to LWE in the same dimension but with an exponential modulus (here $q = 2^{\sqrt{n}}$). Then we provide a reduction from LWE with a secret uniformly chosen in $\mathbb{Z}_q^n$, where n is the dimension of LWE, to binLWE in dimension $n \log q$ but with the same modulus. Finally we give a reduction from binLWE with an exponential modulus to LWE with a polynomial modulus, which preserves the dimension.

$$\text{Gap-SVP}_{\sqrt{n},\text{poly}(n)} \xrightarrow{\text{Theorem 2.10}} \text{LWE}_{\sqrt{n},2^{\sqrt{n}},\frac{1}{\text{poly}(n)}} \xrightarrow{\text{Theorem 4.8}} \text{binLWE}_{n,2^{\sqrt{n}},\frac{1}{\text{poly}(n)}} \xrightarrow{\text{Corrolary 4.4}} \text{LWE}_{n,\text{poly}(n),\frac{1}{\text{poly}(n)}}$$

Figure 4.1: Sequence of reductions to prove the classical hardness of LWE.

4.2 Modulus-Dimension Switching

The main results of this section are Corollaries 4.4 and 4.6 below. Both are special cases of the following technical theorem. We say that a distribution $\mathcal{D}$ over $\mathbb{Z}^n$ is (B,δ)-bounded for some reals $B,\delta \geq 0$ if the probability that $\mathbf{x} \leftarrow \mathcal{D}$ has norm greater than B is at most δ.

Theorem 4.3. *Let $m,n,n',q,q' \geq 1$ be integers, let $\boldsymbol{G} \in \mathbb{Z}^{n' \times n}$ be such that the lattice $\Lambda = \frac{1}{q'}\boldsymbol{G}^T\mathbb{Z}^{n'} + \mathbb{Z}^n$ has a known basis $\boldsymbol{B}$, and let $\mathcal{D}$ be an arbitrary (B,δ)-bounded distribution over $\mathbb{Z}^n$. Let $\alpha,\beta > 0$ and $\varepsilon \in (0,1/2)$ satisfy*

$$\beta^2 \geq \alpha^2 + (4/\pi)\ln(2n(1+1/\varepsilon)) \cdot (\max\{q^{-1}, \|\tilde{\boldsymbol{B}}\|\} \cdot B)^2.$$

Then there is an efficient (transformation) reduction from $\text{LWE}_{n,m,q,\leq\alpha}(\mathcal{D})$ to $\text{LWE}_{n',m,q',\leq\beta}(\boldsymbol{G}\cdot\mathcal{D})$ that reduces the advantage by at most $\delta + 14\varepsilon m$.

Here we use the notation $\|\widetilde{\mathbf{B}}\|$ from Lemma 1.24. We also note that if needed, the distribution on secrets produced by the reduction can always be turned into the uniform distribution on $\mathbb{Z}_{q'}^{n'}$, as mentioned after Definition 2.5. Also, we recall that there exists an elementary reduction from $LWE_{n',q',\leq\beta}$ to $LWE_{n',q',\beta}$ (see Lemma 2.12).

Here we state two important corollaries of the theorem. The first corresponds to just modulus reduction (the LWE dimension is preserved), and is obtained by letting $n' = n$, $\mathbf{G} = \mathbf{I}$ be the n-dimensional identity matrix, and $\mathbf{B} = \mathbf{I}/q'$. For example, we can take $q \geq q' \geq \sqrt{2\ln(2n(1+1/\varepsilon))} \cdot (B/\alpha)$ and $\beta = \sqrt{2}\alpha$, which corresponds to reducing an arbitrary modulus to almost B/α, while increasing the initial error rate α by just a small constant factor.

Corollary 4.4. *For any $m,n \geq 1$, $q \geq q' \geq 1$, (B,δ)-bounded distribution $\mathcal{D}$ over $\mathbb{Z}^n$, $\alpha,\beta > 0$ and $\varepsilon \in (0,1/2)$ such that*

$$\beta^2 \geq \alpha^2 + (4/\pi)\ln(2n(1+1/\varepsilon)) \cdot (B/q')^2,$$

there is an efficient reduction from $\text{LWE}_{n,m,q,\leq\alpha}(\mathcal{D})$ to $\text{LWE}_{n,m,q',\leq\beta}(\mathcal{D})$ that reduces the advantage by at most $\delta + 14\varepsilon m$.

In particular, by using the normal form of LWE (Lemma 2.13), in which the secret has distribution $\mathcal{D} = D_{\mathbb{Z}^n,\sqrt{2}\alpha q}$, we can switch to a power-of-2 modulus with only a small loss in the noise rate, as described in the following corollary. Together with the known search-to-decision reduction (Theorem 2.9), this extends the known hardness of (decision) LWE to *any* modulus q. Here we use that $\mathcal{D} = D_{\mathbb{Z}^n,r}$ is $(Cr\sqrt{n\log(n/\delta)},\delta)$-bounded for some universal constant $C > 0$, which follows by taking union bound over the n coordinates. (Alternatively, one could use that it is $(r\sqrt{n},2^{-n})$-bounded, as follows from Lemma 1.36, leading to a slightly tighter statement for large n.)

Corollary 4.5. *Let $\delta \in (0,1/2)$, $m \geq n \geq 1$, $q' \geq 25$. Let also $q \in [q',2q')$ be the smallest power of 2 not smaller than q' and $\alpha \geq \sqrt{\ln(2n(1+16/\delta)/\pi)}/q$. There exists an efficient (transformation) reduction from $\text{LWE}_{n,m,q,\alpha}$ to $\text{LWE}_{n,m',q',\leq\beta}$ where $m' = m - (16n + 4\ln\ln q)$ and*

$$\beta = C\alpha\sqrt{n}\sqrt{\log(n/\delta)\log(m/\delta)}$$

for some universal constant $C > 0$, that turns advantage of ζ into an advantage of at least $(\zeta - \delta)/4$.

Another corollary illustrates a modulus-dimension tradeoff. Assume $n = kn'$ for some $k \geq 1$, and let $q' = q^k$. Let $\mathbf{G} = \mathbf{I}_{n'} \otimes \mathbf{g}$, where $\mathbf{g} = (1, q, q^2, \ldots, q^{k-1})^T \in \mathbb{Z}^k$. We then have $\Lambda = q^{-k}\mathbf{G}^T\mathbb{Z}^{n'} + \mathbb{Z}^n$. A basis of Λ is given by

$$\mathbf{B} = \mathbf{I}_{n'} \otimes \begin{bmatrix} q^{-1} & q^{-2} & \cdots & q^{-k} \\ & q^{-1} & \cdots & q^{1-k} \\ & & \ddots & \vdots \\ & & & q^{-1} \end{bmatrix} \in \mathbb{R}^{n \times n};$$

this is since the column vectors of $\mathbf{B}$ belong to Λ and the determinants match. Orthogonalizing from left to right, we have $\tilde{\mathbf{B}} = q^{-1}\mathbf{I}$ and so $\|\tilde{\mathbf{B}}\| = q^{-1}$. We therefore obtain the following corollary, showing that we can trade off the dimension against the modulus, holding $n \log q = n' \log q'$ fixed. For example, letting $\mathcal{D} = D_{\mathbb{Z}^n, \alpha q}$ (corresponding to a secret in normal form, see Lemma 2.13), which is $(\alpha q\sqrt{n}, 2^{-n})$-bounded, the reduction increases the error rate by about a $\sqrt{n}$ factor.

Corollary 4.6. *For any $n, m, q \geq 1$, $k \geq 1$ that divides n, (B, δ)-bounded distribution $\mathcal{D}$ over $\mathbb{Z}^n$, $\alpha, \beta > 0$, and $\varepsilon \in (0, 1/2)$ such that*

$$\beta^2 \geq \alpha^2 + (4/\pi)\ln(2n(1 + 1/\varepsilon)) \cdot (B/q)^2,$$

there is an efficient reduction from $\mathrm{LWE}_{n,m,q,\leq\alpha}(\mathcal{D})$ to $\mathrm{LWE}_{n/k,m,q^k,\leq\beta}(\mathbf{G} \cdot \mathcal{D})$ that reduces the advantage by at most $\delta + 14\varepsilon m$, where $\boldsymbol{G} = \boldsymbol{I}_{n/k} \otimes (1, q, q^2, \ldots, q^{k-1})^T$.

Theorem 4.3 follows immediately from the following lemma.

Lemma 4.7. *Adopt the notation of Theorem 4.3, and let*

$$r \geq \max\{q^{-1}, \|\tilde{\boldsymbol{B}}\|\} \cdot \sqrt{2\ln(2n(1 + 1/\epsilon))/\pi}.$$

There is an efficient mapping from $\mathbb{T}_q^n \times \mathbb{T}$ to $\mathbb{T}_{q'}^{n'} \times \mathbb{T}$, which has the following properties:

- *If the input is uniformly random, then the output is within statistical distance 4ϵ from the uniform distribution.*
- *If the input is distributed according to $A_{q,\boldsymbol{s},D_\alpha}$ for some $\boldsymbol{s} \in \mathbb{Z}^n$ with $\|\boldsymbol{s}\| \leq B$, then the output distribution is within statistical distance 10ϵ from $A_{q',\boldsymbol{Gs},D_{\alpha'}}$, where $(\alpha')^2 = \alpha^2 + r^2(\|\boldsymbol{s}\|^2 + B^2) \leq \alpha^2 + 2(rB)^2$.*

Proof. The main idea behind the reduction is to encode $\mathbb{T}_q^n$ into $\mathbb{T}_{q'}^{n'}$, so that the mod-1 inner products between vectors in $\mathbb{T}_q^n$ and a short vector $\mathbf{s} \in \mathbb{Z}^n$, and between vectors in $\mathbb{T}_{q'}^{n'}$ and $\mathbf{Gs} \in \mathbb{Z}^{n'}$, are nearly equivalent. In a bit more detail, the reduction will map its input vector $\mathbf{a} \in \mathbb{T}_q^n$ (from the given LWE-or-uniform distribution) to a vector $\mathbf{a}' \in \mathbb{T}_{q'}^{n'}$, so that

$$\langle \mathbf{a}', \mathbf{Gs} \rangle = \langle \mathbf{G}^T\mathbf{a}', \mathbf{s} \rangle \approx \langle \mathbf{a}, \mathbf{s} \rangle \bmod 1$$

for any (unknown) $\mathbf{s} \in \mathbb{Z}^n$. To do this, it randomly samples $\mathbf{a}'$ so that $\mathbf{G}^T\mathbf{a}' \approx \mathbf{a} \bmod \mathbb{Z}^n$, where the approximation error will be a discrete Gaussian of parameter r.

We can now formally define the reduction, which works as follows. On an input pair $(\mathbf{a}, b) \in \mathbb{T}_q^n \times \mathbb{T}$, it does the following:

- Choose $\mathbf{f} \leftarrow D_{\Lambda-\mathbf{a},r}$ using Lemma 1.24 with basis $\mathbf{B}$, and let $\mathbf{v} = \mathbf{a} + \mathbf{f} \in \Lambda/\mathbb{Z}^n$. (The coset $\Lambda - \mathbf{a}$ is well defined since $\mathbf{a} = \bar{\mathbf{a}} + \mathbb{Z}^n$ is some coset of $\mathbb{Z}^n \subseteq \Lambda$.) Choose a uniformly random solution $\mathbf{a}' \in \mathbb{T}_{q'}^{n'}$ to the equation $\mathbf{G}^T\mathbf{a}' = \mathbf{v} \bmod \mathbb{Z}^n$. This can be done by computing a basis of the solution set $\mathbf{G}^T\mathbf{a}' = \mathbf{0} \bmod \mathbb{Z}^n$, and adding a uniform element from that set to an arbitrary solution to the equation $\mathbf{G}^T\mathbf{a}' = \mathbf{v} \bmod \mathbb{Z}^n$.
- Choose $e' \leftarrow D_{rB}$ and let $b' = b + e' \in \mathbb{T}$.
- Output $(\mathbf{a}', b')$.

We now analyze the reduction. First, if the distribution of the input is uniform, then it suffices to show that $\mathbf{a}'$ is (nearly) uniformly random, because both b and e' are independent of $\mathbf{a}'$, and $b \in \mathbb{T}$ is uniform. To prove this claim, notice that it suffices to show that the coset $\mathbf{v} \in \Lambda/\mathbb{Z}^n$ is (nearly) uniformly random, because each $\mathbf{v}$ has the same number of solutions $\mathbf{a}'$ to $\mathbf{G}^T\mathbf{a}' = \mathbf{v} \bmod \mathbb{Z}^n$. Next, observe that for any $\bar{\mathbf{a}} \in \mathbb{T}_q^n$ and $\bar{\mathbf{f}} \in \Lambda - \bar{\mathbf{a}}$, we have by Lemma 1.31 (using that $r \geq \eta_\varepsilon(\Lambda)$ by Lemma 1.28) that

$$\begin{aligned}\Pr[\mathbf{a} = \bar{\mathbf{a}} \wedge \mathbf{f} = \bar{\mathbf{f}}] &= q^{-n} \cdot \rho_r(\bar{\mathbf{f}})/\rho_r(\Lambda - \bar{\mathbf{a}}) \\ &\in C[1, \tfrac{1+\epsilon}{1-\epsilon}] \cdot \rho_r(\bar{\mathbf{f}}). \end{aligned} \tag{4.1}$$

where $C = q^{-n}/\rho_r(\Lambda)$ is a normalizing value that does not depend on $\bar{\mathbf{a}}$ or $\bar{\mathbf{f}}$. Therefore, by summing over all $\bar{\mathbf{a}}, \bar{\mathbf{f}}$ satisfying $\bar{\mathbf{a}} + \bar{\mathbf{f}} = \bar{\mathbf{v}}$, we obtain that for any $\bar{\mathbf{v}} \in \Lambda/\mathbb{Z}^n$,

$$\Pr[\mathbf{v} = \bar{\mathbf{v}}] \in C[1, \tfrac{1+\epsilon}{1-\epsilon}] \cdot \rho_r(q^{-1}\mathbb{Z}^n + \bar{\mathbf{v}}).$$

Since $r \geq \eta_\varepsilon(q^{-1}\mathbb{Z}^n)$ (by Lemma 1.28), Lemma 1.31 implies that $\Pr[\mathbf{v} = \bar{\mathbf{v}}] \in [\frac{1-\epsilon}{1+\epsilon}, \frac{1+\epsilon}{1-\epsilon}]C'$ for a constant C' that is independent of $\bar{\mathbf{v}}$. By Claim 1.3, this shows that $\mathbf{a}'$ is within statistical distance $1 - ((1-\varepsilon)/(1+\varepsilon))^2 \leq 4\epsilon$ of the uniform distribution.

It remains to show that the reduction maps $A_{q,\mathbf{s},D_\alpha}$ to $A_{q',\mathbf{G}\mathbf{s},D_\beta}$. Let the input sample from the former distribution be $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s}\rangle + e)$, where $e \hookleftarrow D_\alpha$. As argued above, the output $\mathbf{a}'$ is (nearly) uniform over $\mathbb{T}_{q'}^{n'}$. So condition now on any fixed value $\overline{\mathbf{a}'} \in \mathbb{T}_{q'}^{n'}$ of $\mathbf{a}'$, and let $\bar{\mathbf{v}} = \mathbf{G}^T\overline{\mathbf{a}'} \bmod \mathbb{Z}^n$. We have

$$b' = \langle \mathbf{a}, \mathbf{s}\rangle + e + e' = \langle \overline{\mathbf{a}'}, \mathbf{G}\mathbf{s}\rangle + e + \langle -\mathbf{f}, \mathbf{s}\rangle + e' \bmod 1.$$

By Claim 1.3 and (4.1) (and noting that if $\mathbf{f} = \bar{\mathbf{f}}$ then $\mathbf{a} = \bar{\mathbf{v}} - \bar{\mathbf{f}} \bmod \mathbb{Z}^n$), the distribution of $-\mathbf{f}$ is within statistical distance $1 - (1-\varepsilon)/(1+\varepsilon) \leq 2\varepsilon$ of $D_{q^{-1}\mathbb{Z}^n - \bar{\mathbf{v}}, r}$. By Lemma 1.46 (using $r \geq \sqrt{2}\eta_\varepsilon(q^{-1}\mathbb{Z}^n)$ and $\|\mathbf{s}\| \leq B$), the distribution of $\langle -\mathbf{f}, \mathbf{s}\rangle + e'$ is within statistical distance 6ε from D_t, where $t^2 = r^2(\|\mathbf{s}\|^2 + B^2)$. It therefore follows that $e + \langle -\mathbf{f}, \mathbf{s}\rangle + e'$ is within statistical distance 6ε from $D_{(t^2+\alpha^2)^{1/2}}$, as required. □

4.3 Hardness of LWE with Binary Secret

The following is the main theorem of this section.

Theorem 4.8. *Let $k, q \geq 1$, and $m \geq n \geq 1$ be integers, and let $\epsilon \in (0, 1/2)$, $\alpha, \delta > 0$, be such that $n \geq (k+1)\log_2 q + 2\log_2(1/\delta)$, $\alpha \geq \sqrt{\ln(2n(1+1/\varepsilon))/\pi}/q$. There exist three (transformation) reductions from $\mathrm{LWE}_{k,m,q,\alpha}$ to $\mathrm{binLWE}_{n,m,q,\leq\sqrt{10n}\alpha}$, such that for any algorithm for the latter problem with advantage ζ, at least one of the reductions produces an algorithm for the former problem with advantage at least*

$$(\zeta - \delta)/(3m) - 41\varepsilon/2 - \sum_{p|q,\ p\ prime} p^{-k-1}. \tag{4.2}$$

By combining Theorem 4.8 with the reduction in Corollary 4.4 (and noting that $\{0,1\}^n$ is $(\sqrt{n}, 0)$ bounded), we can replace the binLWE problem above with $\mathrm{binLWE}_{n,m,q',\beta}$ for any $q' \geq 1$ and $\xi > 0$ where

$$\beta := \left(10n\alpha^2 + \frac{4n}{\pi q'^2}\ln(2n(1+1/\xi))\right)^{1/2},$$

while decreasing the advantage in (4.2) by $14\xi m$. Recalling that LWE of dimension $k = \sqrt{n}$ and modulus $q = 2^{k/2}$ (assume k is even) is known to be classically as hard as $\sqrt{n}$-dimensional lattice problems (Theorems 2.10 and 2.9), this gives Theorem 4.2. The modulus q' can be taken almost as small as $\sqrt{n}$.

For most purposes the sum over prime factors of q in (4.2) is negligible. For instance, in deriving the formal statement of Theorem 4.1 above, we used a q that is a power of 2, in which case the sum is $2^{-k-1} = 2^{-\sqrt{n}-1}$, which is negligible. If needed, one can improve this by applying the modulus switching reduction (Corollary 4.5) before applying Theorem 4.8 in order to make q prime. (Strictly speaking, one also needs to apply Lemma 2.12 to replace the "unknown noise" variant of LWE given by Corollary 4.5 with the fixed noise variant.) This improves the advantage loss to $q^{-\sqrt{n}-1}$ which is roughly 2^{-n}.

In a high level, the proof of the theorem follows by combining three main steps. The first, given in Section 4.3.1, reduces LWE to a variant in which the first equation is errorless. The second, given in Section 4.3.2, reduces the latter to the intermediate problem extLWE, another variant of LWE in which some information on the noise elements is leaked. Finally, in Section 4.3.3, we reduce extLWE to LWE with $\{0,1\}$ secret. We note that the first reduction is relatively standard; it is the other two that we consider as the main contribution of this section. We now proceed with more details (see also Figure 4.2).

Proof. First, since $m \geq n$, Lemma 4.10 provides a transformation reduction from $\mathrm{LWE}_{k,m,q,\alpha}$ to first-is-errorless $\mathrm{LWE}_{k+1,n,q,\alpha}$, while reducing the advantage by at most 2^{-k+1}. Next, Lemma 4.14 with $\mathcal{Z} = \{0,1\}^n$, which is of quality $\xi = 2$ by Claim 4.13, reduces the latter problem to $\mathrm{extLWE}_{k+1,n,q,\sqrt{5}\alpha,\{0,1\}^n}$ while reducing the advantage by at most $33\varepsilon/2$. Then, Lemma 4.15 reduces the latter problem to $\mathrm{extLWE}^m_{k+1,n,q,\sqrt{5}\alpha,\{0,1\}^n}$, while losing a factor of m in the advantage. Finally, Lemma 4.16 provides three reductions to $\mathrm{binLWE}_{n,m,q,\leq\sqrt{10n\alpha}}$: two from the latter problem, and one from $\mathrm{LWE}_{k+1,m,q,\sqrt{5n}\alpha}$, guaranteeing that the sum of advantages is at least the original advantage minus $4m\varepsilon + \delta$. Together with the trivial reduction from $\mathrm{LWE}_{k,m,q,\alpha}$ to $\mathrm{LWE}_{k+1,m,q,\sqrt{5n}\alpha}$ (which incurs no loss in advantage), this completes the proof. □

4.3.1 First-is-errorless LWE

We first define a variant of LWE in which the first equation is given without error, and then show in Lemma 4.10 that it is still hard.

Definition 4.9. For integers $n, q \geq 1$ and an error distribution ϕ over $\mathbb{R}$, the "first-is-errorless" variant of the LWE problem is to distinguish between the following two scenarios. In the first, the first sample is uniform over $\mathbb{T}_q^n \times \mathbb{T}_q$ and the rest are uniform over $\mathbb{T}_q^n \times \mathbb{T}$. In the second, there is an unknown uniformly distributed $\mathbf{s} \in \{0, \ldots, q-1\}^n$, the first sample we get is from $A_{q,\mathbf{s},\{0\}}$ (where $\{0\}$ denotes the distribution that is deterministically zero) and the rest are from $A_{q,\mathbf{s},\phi}$.

Lemma 4.10. *For any $n \geq 2$, $m, q \geq 1$, and error distribution ϕ, there is an efficient (transformation) reduction from $\mathrm{LWE}_{n-1,m,q,\phi}$ to the first-is-errorless variant of $\mathrm{LWE}_{n,m,q,\phi}$ that reduces the advantage by at most $\sum_p p^{-n}$, with the sum going over all prime factors of q.*

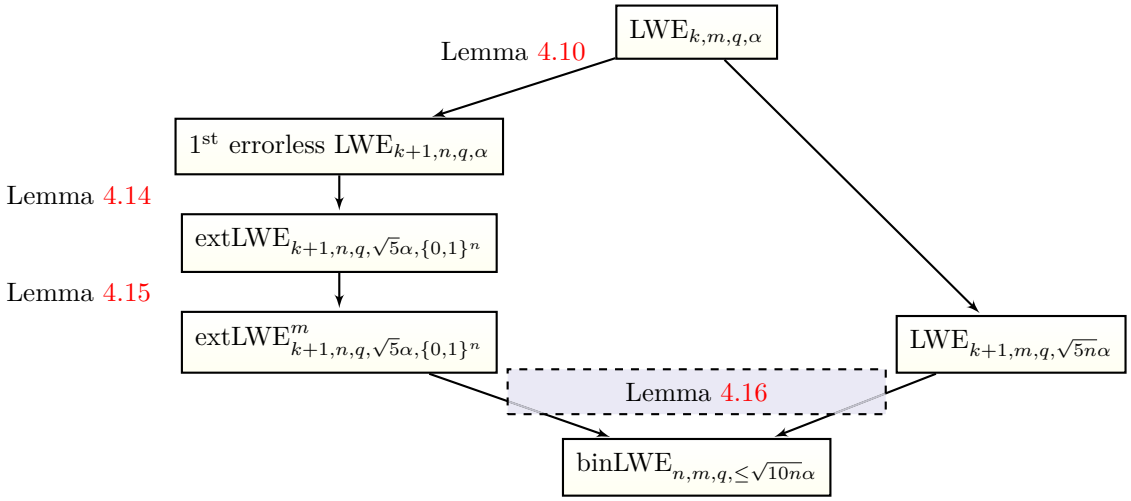

Figure 4.2: Summary of reductions used in Theorem 4.8.

Notice that if q is prime the loss in advantage is at most q^{-n}. Alternatively, for any number q we can bound it by

$$\sum_{k\geq 2} k^{-n} \leq 2^{-n} + \int_2^\infty t^{-n}\mathrm{d}t \leq 2^{-n+2},$$

which might be good enough when n is large.

Proof. The reduction starts by choosing a vector $\mathbf{a}'$ uniformly at random from $\{0,\ldots,q-1\}^n$. Let r be the greatest common divisor of the coordinates of $\mathbf{a}'$. If it is not coprime to q, we abort. The probability that this happens is at most

$$\sum_{p \text{ prime},\ p|q} p^{-n}.$$

Assuming we do not abort, we proceed by finding a matrix $\mathbf{U} \in \mathbb{Z}^{n\times n}$ that is invertible modulo q and whose leftmost column is $\mathbf{a}'$. Such a matrix exists, and can be found efficiently. For instance, using the extended GCD algorithm, we find an $n \times n$ unimodular matrix $\mathbf{R}$ such that $\mathbf{R}\mathbf{a}' = (r, 0, \ldots, 0)^T$. Then $\mathbf{R}^{-1} \cdot \mathrm{diag}(r, 1, \ldots, 1)$ is the desired matrix. We also pick a uniform element $s_0 \in \{0,\ldots,q-1\}$. The reduction now proceeds as follows. The first sample it outputs is $(\mathbf{a}'/q, s_0/q)$. The remaining samples are produced by taking a sample $(\mathbf{a}, b)$ from the given oracle, picking a fresh uniformly random $d \in \mathbb{T}_q$, and outputting $(\mathbf{U}(d|\mathbf{a}), b + (s_0 \cdot d))$ with the vertical bar denoting concatenation. It is easy to verify correctness: given uniform samples, the reduction outputs uniform samples (with the first sample's b component uniform over $\mathbb{T}_q$), up to statistical distance 2^{-n+1}; and given samples from $A_{q,\mathbf{s},\phi}$, the reduction outputs one sample from $A_{q,\mathbf{s}',\{0\}}$ and the remaining samples from $A_{q,\mathbf{s}',\phi}$, up to statistical distance 2^{-n+1}, where $\mathbf{s}' = (\mathbf{U}^{-1})^T(s_0|\mathbf{s}) \bmod q$. This proves correctness since $\mathbf{U}$, being invertible modulo q, induces a bijection on $\mathbb{Z}_q^n$, and so $\mathbf{s}'$ is uniform in $\{0,\ldots,q-1\}^n$. □

4.3.2 Extended LWE

We next define the intermediate problem extLWE. (This definition is of an easier problem than the one considered in previous work [ASP12], which makes our hardness result stronger.)

Definition 4.11. For $n, m, q, t \geq 1$, $\mathcal{Z} \subseteq \mathbb{Z}^m$, and a distribution χ over $\frac{1}{q}\mathbb{Z}^m$, the $\mathrm{extLWE}^t_{n,m,q,\chi,\mathcal{Z}}$ problem is as follows. The algorithm gets to choose $\mathbf{z} \in \mathcal{Z}$ and then receives a tuple

$$(\mathbf{A}, (\mathbf{b}_i)_{i=1}^t, (\langle \mathbf{e}_i, \mathbf{z} \rangle)_{i=1}^t) \in \mathbb{T}_q^{n \times m} \times (\mathbb{T}_q^m)^t \times (\tfrac{1}{q}\mathbb{Z})^t.$$

Its goal is to distinguish between the following two cases. In the first, $\mathbf{A} \in \mathbb{T}_q^{n \times m}$ is chosen uniformly, $\mathbf{e}_i \in \frac{1}{q}\mathbb{Z}^m$ are chosen from χ, and $\mathbf{b}_i = \mathbf{A}^T \mathbf{s}_i + \mathbf{e}_i \bmod 1$ where $\mathbf{s}_i \in \{0, \ldots, q-1\}^n$ are chosen uniformly. The second case is identical, except that the $\mathbf{b}_i$ are chosen uniformly in $\mathbb{T}_q^m$ independently of everything else.

When $t = 1$, we omit the superscript t. Also, when χ is $D_{q^{-1}\mathbb{Z}^m, \alpha}$ for some $\alpha > 0$, we replace the subscript χ by α. We note that a discrete version of LWE can be defined as a special case of extLWE by setting $\mathcal{Z} = \{0^m\}$. We next define a measure of quality of sets $\mathcal{Z}$.

Definition 4.12. For a real $\xi > 0$ and a set $\mathcal{Z} \subseteq \mathbb{Z}^m$ we say that $\mathcal{Z}$ is of *quality* ξ if given any $\mathbf{z} \in \mathcal{Z}$, we can efficiently find a unimodular matrix $\mathbf{U} \in \mathbb{Z}^{m \times m}$ such that if $\mathbf{U}' \in \mathbb{Z}^{m \times (m-1)}$ is the matrix obtained from $\mathbf{U}$ by removing its leftmost column then all of the columns of $\mathbf{U}'$ are orthogonal to $\mathbf{z}$ and its largest singular value is at most ξ.

The idea in this definition is that the columns of $\mathbf{U}'$ form a basis of the lattice of integer points that are orthogonal to $\mathbf{z}$, i.e., the lattice $\{\mathbf{b} \in \mathbb{Z}^m : \langle \mathbf{b}, \mathbf{z} \rangle = 0\}$. The quality measures how "short" we can make this basis.

Claim 4.13. *The set $\mathcal{Z} = \{0,1\}^m$ is of quality 2.*

Proof. Let $\mathbf{z} \in \mathcal{Z}$ and assume without loss of generality that its first $k \geq 1$ coordinates are 1 and the remaining $m - k$ are 0. Then consider the upper bidiagonal matrix $\mathbf{U}$ whose diagonal is all 1s and whose diagonal above the main diagonal is $(-1, \ldots, -1, 0, \ldots, 0)$ with -1 appearing $k - 1$ times. The matrix is clearly unimodular and all the columns except the first one are orthogonal to $\mathbf{z}$. Moreover, by the triangle inequality, we can bound the operator norm of $\mathbf{U}$ by the sum of that of the diagonal 1 matrix and the off-diagonal matrix, both of which clearly have norm at most 1. □

Lemma 4.14. *Let $\mathcal{Z} \subseteq \mathbb{Z}^m$ be of quality $\xi > 0$. Then for any $n, q \geq 1$, $\varepsilon \in (0, 1/2)$, and $\alpha, r \geq (\ln(2m(1 + 1/\varepsilon))/\pi)^{1/2}/q$, there is a (transformation) reduction from the first-is-errorless variant of $\mathrm{LWE}_{n,m,q,\alpha}$ to $\mathrm{extLWE}_{n,m,q,(\alpha^2\xi^2 + r^2)^{1/2},\mathcal{Z}}$ that reduces the advantage by at most $33\varepsilon/2$.*

Proof. We first describe the reduction. Assume we are asked to provide samples for some $\mathbf{z} \in \mathcal{Z}$. We compute a unimodular $\mathbf{U} \in \mathbb{Z}^{m \times m}$ for $\mathbf{z}$ as in Definition 4.12, and let $\mathbf{U}' \in \mathbb{Z}^{m \times (m-1)}$ be the matrix formed by removing the first column of $\mathbf{U}$. We then take m samples from the given distribution, resulting in $(\mathbf{A}, \mathbf{b}) \in \mathbb{T}_q^{n \times m} \times (\mathbb{T}_q \times \mathbb{T}^{m-1})$. We also sample a vector $\mathbf{f}$ from the m-dimensional continuous Gaussian distribution $D_{\alpha(\xi^2 \mathbf{I} - \mathbf{U}'\mathbf{U}'^T)^{1/2}}$, which is well defined since $\xi^2 \mathbf{I} - \mathbf{U}'\mathbf{U}'^T$ is a positive semidefinite matrix by our assumption on $\mathbf{U}$. The output of the reduction is the tuple

$$(\mathbf{A}' = \mathbf{A}\mathbf{U}^T, \mathbf{b}' + \mathbf{c}, \langle \mathbf{z}, \mathbf{f} + \mathbf{c} \rangle) \in \mathbb{T}_q^{n \times m} \times \mathbb{T}_q^m \times \tfrac{1}{q}\mathbb{Z}, \tag{4.3}$$

where $\mathbf{b}' = \mathbf{U}\mathbf{b} + \mathbf{f}$, and $\mathbf{c}$ is chosen from the discrete Gaussian distribution $D_{q^{-1}\mathbb{Z}^m - \mathbf{b}', r}$ (using Theorem 1.24).

We now prove the correctness of the reduction. Consider first the case that we get valid LWE equations, i.e., $\mathbf{A}$ is uniform in $\mathbb{T}_q^{n \times m}$ and $\mathbf{b} = \mathbf{A}^T \mathbf{s} + \mathbf{e} \in \mathbb{T}^m$ where $\mathbf{s} \in \{0, \ldots, q-1\}^n$ is uniformly chosen, the first coordinate of $\mathbf{e} \in \mathbb{R}^m$ is 0, and the remaining $m - 1$ coordinates are

chosen from D_α. Since $\mathbf{U}$ is unimodular, $\mathbf{A}' = \mathbf{A}\mathbf{U}^T$ is uniformly distributed in $\mathbb{T}_q^{n\times m}$ as required. From now on we condition on an arbitrary $\mathbf{A}'$ and analyze the distribution of the remaining two components of (4.3). Next,

$$\mathbf{b}' = \mathbf{U}\mathbf{b} + \mathbf{f} = \mathbf{A}'^T\mathbf{s} + \mathbf{U}\mathbf{e} + \mathbf{f}.$$

Since $\mathbf{U}\mathbf{e}$ is distributed as a continuous Gaussian $D_{\alpha\mathbf{U}'}$, the vector $\mathbf{U}\mathbf{e}+\mathbf{f}$ is a distributed as a *spherical* continuous Gaussian $D_{\alpha\xi}$. Moreover, since $\mathbf{A}'^T\mathbf{s} \in \mathbb{T}_q^m$, the coset $q^{-1}\mathbb{Z}^m - \mathbf{b}'$ is identical to $q^{-1}\mathbb{Z}^m - (\mathbf{U}\mathbf{e}+\mathbf{f})$, so we can see $\mathbf{c}$ as being chosen from $D_{q^{-1}\mathbb{Z}^m-(\mathbf{U}\mathbf{e}+\mathbf{f}),r}$. Therefore, by Lemma 1.43 and using that $r \geq \eta_\varepsilon(q^{-1}\mathbb{Z}^m)$ by Lemma 1.28, the distribution of $\mathbf{U}\mathbf{e}+\mathbf{f}+\mathbf{c}$ is within statistical distance 8ε of $D_{q^{-1}\mathbb{Z}^m,(\alpha^2\xi^2+r^2)^{1/2}}$. This shows that the second component in (4.3) is also distributed correctly. Finally, for the third component, by our assumption on $\mathbf{U}$ and the fact that the first coordinate of $\mathbf{e}$ is zero,

$$\langle \mathbf{z}, \mathbf{f}+\mathbf{c}\rangle = \langle \mathbf{z}, \mathbf{U}\mathbf{e}+\mathbf{f}+\mathbf{c}\rangle,$$

and so the third component gives the inner product of the noise with $\mathbf{z}$, as desired.

We now consider the case where the input is uniform, i.e., that $\mathbf{A}$ is uniform in $\mathbb{T}_q^{n\times m}$ and $\mathbf{b}$ is independent and uniform in $\mathbb{T}_q \times \mathbb{T}^{m-1}$. We first observe that by Lemma 1.30, since $\alpha \geq \eta_{\varepsilon/m}(q^{-1}\mathbb{Z})$ (by Lemma 1.28), the distribution of $(\mathbf{A},\mathbf{b})$ is within statistical distance $\varepsilon/2$ of the distribution of $(\mathbf{A}, \mathbf{e}'+\mathbf{e})$ where $\mathbf{e}'$ is chosen uniformly in $\mathbb{T}_q^m$, the first coordinate of $\mathbf{e}$ is zero, and its remaining $m-1$ coordinates are chosen independently from D_α. So from now on assume our input is $(\mathbf{A}, \mathbf{e}'+\mathbf{e})$. The first component of (4.3) is uniform in $\mathbb{T}_q^{n\times m}$ as before, and moreover, it is clearly independent of the other two. Moreover, since $\mathbf{b}' = \mathbf{U}\mathbf{e}' + \mathbf{U}\mathbf{e} + \mathbf{f}$ and $\mathbf{U}\mathbf{e}' \in \mathbb{T}_q^m$, the coset $q^{-1}\mathbb{Z}^m - \mathbf{b}'$ is identical to $q^{-1}\mathbb{Z}^m - (\mathbf{U}\mathbf{e}+\mathbf{f})$, and so $\mathbf{c}$ is distributed identically to the case of a valid LWE equation, and in particular is independent of $\mathbf{e}'$. This establishes that the third component of (4.3) is correctly distributed; moreover, since $\mathbf{e}'$ is independent of the first and third components, and $\mathbf{U}\mathbf{e}'$ is uniform in $\mathbb{T}_q^m$ (since $\mathbf{U}$ is unimodular), we get that the second component is uniform and independent of the other two, as desired. □

We end this section by stating the standard reduction to the multi-secret ($t \geq 1$) case of extended LWE.

Lemma 4.15. *Let $n, m, q, \chi, \mathcal{Z}$ be as in Definition 4.11 with χ efficiently sampleable, and let $t \geq 1$ be an integer. Then there is an efficient (transformation) reduction from* $\mathrm{extLWE}_{n,m,q,\chi,\mathcal{Z}}$ *to* $\mathrm{extLWE}^t_{n,m,q,\chi,\mathcal{Z}}$ *that reduces the advantage by a factor of t.*

The proof is by a standard hybrid argument. We bring it here for the sake of completeness. We note that the distribution of the secret vector $\mathbf{s}$ needs to be sampleable but otherwise it plays no role in the proof. The lemma therefore naturally extends to any (sampleable) distribution of $\mathbf{s}$.

Proof. Let $\mathcal{A}$ be an algorithm for $\mathrm{extLWE}^t_{n,m,q,\chi,\mathcal{Z}}$, let $\mathbf{z}$ be the vector output by $\mathcal{A}$ in the first step (note that this is a random variable) and let H_i denote the distribution

$$\left(\mathbf{A}, \{\mathbf{b}_1, \ldots, \mathbf{b}_i, \mathbf{u}_{i+1}, \ldots, \mathbf{u}_t\}, \mathbf{z}, \{\langle \mathbf{z}, \mathbf{e}_i\rangle\}_{i=1}^t\right) \ ,$$

where $\mathbf{u}_{i+1}, \ldots, \mathbf{u}_t$ are sampled independently and uniformly in $\mathbb{T}_q^m$. Then by definition $\mathrm{Adv}[\mathcal{A}] = |\Pr[\mathcal{A}(H_0)] - \Pr[\mathcal{A}(H_t)]|$.

We now describe an algorithm $\mathcal{B}$ for $\mathrm{extLWE}_{n,m,q,\chi,\mathcal{Z}}$: First, $\mathcal{B}$ runs $\mathcal{A}$ to obtain $\mathbf{z}$ and sends it to the challenger as its own $\mathbf{z}$. Then, given an input $(\mathbf{A}, \mathbf{d}, \mathbf{z}, y)$ for $\mathrm{extLWE}_{n,m,q,\chi,\mathcal{Z}}$, the distinguisher $\mathcal{B}$ samples $i^* \hookleftarrow U(\{1,\ldots,t\})$, and in addition $\mathbf{s}_1, \ldots, \mathbf{s}_{i^*-1} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{u}_{i^*+1}, \ldots, \mathbf{u}_t \hookleftarrow$

$U(\mathbb{T}_q^m)$, $\mathbf{e}_1, \ldots, \mathbf{e}_{i^*-1}, \mathbf{e}_{i^*+1}, \ldots, \mathbf{e}_t \hookleftarrow \chi^m$. It sets $\mathbf{b}_i = \mathbf{A}^T \cdot \mathbf{s}_i + \mathbf{e}_i \pmod 1$, and sends the following to $\mathcal{A}$:

$$(\mathbf{A}, \{\mathbf{b}_1, \ldots, \mathbf{b}_{i^*-1}, \mathbf{d}, \mathbf{u}_{i^*+1}, \ldots, \mathbf{u}_t\}, \mathbf{z}, \{\langle \mathbf{z}, \mathbf{e}_1 \rangle, \ldots, \langle \mathbf{z}, \mathbf{e}_{i^*-1} \rangle, y, \langle \mathbf{z}, \mathbf{e}_{i^*+1} \rangle, \ldots, \langle \mathbf{z}, \mathbf{e}_t \rangle\}) \ .$$

Finally, $\mathcal{B}$ outputs the same output as $\mathcal{A}$ did.

Note that when the input to $\mathcal{B}$ is distributed as $P_0 = (\mathbf{A}, \mathbf{b}, \mathbf{z}, \mathbf{z}^T \cdot \mathbf{e})$ with $\mathbf{b} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e} \pmod 1$, then $\mathcal{B}$ feeds $\mathcal{A}$ with exactly the distribution H_{i^*}. On the other hand, if the input to $\mathcal{B}$ is $P_1 = (\mathbf{A}, \mathbf{u}, \mathbf{z}, \mathbf{z}^T \cdot \mathbf{e})$ with $\mathbf{u} \hookleftarrow U(\mathbb{T}_q^m)$, then $\mathcal{B}$ feeds $\mathcal{A}$ with H_{i^*-1}.

Since i^* is uniform in $\{1, \ldots, t\}$, we get that

$$\begin{aligned} t \operatorname{Adv}[\mathcal{B}] &= t \left| \Pr[\mathcal{B}(P_0)] - \Pr[\mathcal{B}(P_1)] \right| \\ &= \left| \sum_{i^*=1}^{t} \Pr[\mathcal{A}(H_{i^*})] - \sum_{i^*=1}^{t} \Pr[\mathcal{A}(H_{i^*-1})] \right| \\ &= \left| \Pr[\mathcal{A}(H_t)] - \Pr[\mathcal{A}(H_0)] \right| \\ &= \operatorname{Adv}[\mathcal{A}] \ , \end{aligned}$$

and the result follows. □

4.3.3 Reducing to binary secret

Lemma 4.16. *Let $k, n, m, q \in \mathbb{N}$, $\epsilon \in (0, 1/2)$, and $\delta, \alpha, \beta, \gamma > 0$ be such that $n \geq k \log_2 q + 2 \log_2(1/\delta)$, $\beta \geq \sqrt{2 \ln(2n(1+1/\varepsilon))/\pi}/q$, $\alpha = \sqrt{2n}\beta$, $\gamma = \sqrt{n}\beta$. Then there exist three efficient (transformation) reductions to* $\mathrm{binLWE}_{n,m,q,\leq\alpha}$ *from* $\mathrm{extLWE}^m_{k,n,q,\beta,\{0,1\}^n}$, $\mathrm{extLWE}_{k,m,q,\gamma}$, *and* $\mathrm{extLWE}^m_{k,n,q,\beta,\{0^n\}}$, *such that if $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ are the algorithms obtained by applying these reductions (respectively) to an algorithm $\mathcal{A}$, then*

$$Adv[\mathcal{A}] \leq Adv[\mathcal{B}_1] + Adv[\mathcal{B}_2] + Adv[\mathcal{B}_3] + 4m\epsilon + \delta \, .$$

Pointing out the transformation reduction from $\mathrm{extLWE}^m_{k,n,q,\beta,\{0,1\}^n}$ to $\mathrm{extLWE}^m_{k,n,q,\beta,\{0^n\}}$, the lemma implies the hardness of $\mathrm{binLWE}_{n,m,q,\leq\alpha}$ based on the hardness of $\mathrm{extLWE}^m_{k,n,q,\beta,\{0,1\}^n}$ and $\mathrm{LWE}_{k,m,q,\gamma}$.

We note that our proof is actually more general, and holds for any binary distribution of min-entropy at least $k \log_2 q + 2 \log_2(1/\delta)$, and not just a uniform binary secret as in the definition of binLWE.

Proof. The proof follows by a sequence of hybrids. Let $k, n, m, q, \varepsilon, \alpha, \beta, \gamma$ be as in the lemma statement. We consider $\mathbf{z} \hookleftarrow U(\{0,1\}^n)$ and $\mathbf{e} \hookleftarrow D^m_{\alpha'}$ for $\alpha' = \sqrt{\beta^2 \|\mathbf{z}\|^2 + \gamma^2} \leq \sqrt{2n}\beta = \alpha$. In addition, we let $\mathbf{A} \hookleftarrow U(\mathbb{T}_q^{n \times m})$, $\mathbf{u} \hookleftarrow U(\mathbb{T}^m)$, and define $\mathbf{b} := \mathbf{A}^T \cdot \mathbf{z} + \mathbf{e} \pmod 1$. We consider an algorithm $\mathcal{A}$ that distinguishes between $(\mathbf{A}, \mathbf{b})$ and $(\mathbf{A}, \mathbf{u})$.

We let H_0 denote the distribution $(\mathbf{A}, \mathbf{b})$ and H_1 the distribution

$$H_1 = (\mathbf{A}, \mathbf{A}^T\mathbf{z} - \mathbf{N}^T\mathbf{z} + \widehat{e} \bmod 1),$$

where $\mathbf{N} \hookleftarrow D^{n \times m}_{q^{-1}\mathbb{Z},\beta}$ and $\widehat{e} \hookleftarrow D^m_\gamma$. Using $\|\mathbf{z}\| \leq \sqrt{n}$ and that $\beta \geq \sqrt{2}\eta_\varepsilon(\mathbb{Z}^n)/q$ (by Lemma 1.28), it follows by Lemma 1.46 that the statistical distance between $-\mathbf{N}^T\mathbf{z} + \widehat{e}$ and $D^m_{\alpha'}$ is at most $4m\epsilon$. It thus follows that

$$\left| \Pr[\mathcal{A}(H_0)] - \Pr[\mathcal{A}(H_1)] \right| \leq 4m\epsilon \ . \tag{4.4}$$

We define a distribution H_2 as follows. Let $\mathbf{B} \leftarrow \mathbb{T}_q^{k\times m}$ and $\mathbf{C} \leftarrow \mathbb{T}_q^{k\times n}$. Let $\widehat{\mathbf{A}} := q\mathbf{C}^T \cdot \mathbf{B} + \mathbf{N} \pmod 1$. Finally,

$$H_2 = (\widehat{\mathbf{A}}, \widehat{\mathbf{A}}^T \cdot \mathbf{z} - \mathbf{N}^T\mathbf{z} + \widehat{e}) = (\widehat{\mathbf{A}}, q\mathbf{B}^T \cdot \mathbf{C} \cdot \mathbf{z} + \widehat{e}) \ .$$

We now argue that there exists an adversary $\mathcal{B}_1$ for problem $\mathrm{extLWE}^m_{k,n,q,\beta,\{0,1\}^n}$, such that

$$\mathrm{Adv}[\mathcal{B}_1] = |\Pr[\mathcal{A}(H_1)] - \Pr[\mathcal{A}(H_2)]| \ . \tag{4.5}$$

This is because H_1, H_2 can be viewed as applying the same efficient transformation on the distributions $(\mathbf{C}, \mathbf{A}, \mathbf{N}^T\mathbf{z})$ and $(\mathbf{C}, \widehat{\mathbf{A}}, \mathbf{N}^T\mathbf{z})$ respectively. Since distinguishing the latter distributions is exactly the $\mathrm{extLWE}^m_{k,n,q,\beta,\{0,1\}^n}$ problem (where the columns of $q \cdot \mathbf{B}$ are interpreted as the m secret vectors), the distinguisher $\mathcal{B}_1$ follows by first applying the aforementioned transformation and then applying $\mathcal{A}$.

For the next hybrid, we define $H_3 = (\widehat{\mathbf{A}}, \mathbf{B}^T \cdot \mathbf{s} + \widehat{e})$, for $\mathbf{s} \leftarrow \mathbb{Z}_q^k$. It follows that

$$|\Pr[\mathcal{A}(H_2)] - \Pr[\mathcal{A}(H_3)]| \le \delta \tag{4.6}$$

by the leftover hash lemma (see Lemma 1.6), since H_2, H_3 can be derived from $(\mathbf{C}, q\mathbf{C} \cdot \mathbf{z})$ and $(\mathbf{C}, \mathbf{s})$ respectively, whose statistical distance is at most δ.

Our next hybrid makes the second component uniform: $H_4 = (\widehat{\mathbf{A}}, \mathbf{u})$. There exists an algorithm $\mathcal{B}_2$ for $\mathrm{LWE}_{k,m,q,\gamma}$ such that

$$\mathrm{Adv}[\mathcal{B}_2] = |\Pr[\mathcal{A}(H_3)] - \Pr[\mathcal{A}(H_4)]| \ , \tag{4.7}$$

since H_3, H_4 can be computed efficiently from $(\mathbf{B}, \mathbf{B}^T\mathbf{s} + \widehat{e}), (\mathbf{B}, \mathbf{u})$.

Lastly, we change $\widehat{\mathbf{A}}$ back to uniform: $H_5 = (\mathbf{A}, \mathbf{u})$. There exists an algorithm $\mathcal{B}_3$ for $\mathrm{extLWE}^m_{k,n,q,\beta,\{0^n\}}$ such that

$$\mathrm{Adv}[\mathcal{B}_3] = |\Pr[\mathcal{A}(H_4)] - \Pr[\mathcal{A}(H_5)]| \ . \tag{4.8}$$

Eq. (4.8) is derived very similarly to Eq. (4.5): We notice that H_4, H_5 can be viewed as applying the same efficient transformation on the distributions $(\mathbf{C}, \widehat{\mathbf{A}})$ and $(\mathbf{C}, \mathbf{A})$ respectively. Since distinguishing the latter distributions is exactly the $\mathrm{extLWE}^m_{k,n,q,\beta,\{0^n\}}$ problem (where the columns of $q \cdot \mathbf{B}$ are interpreted as the m secret vectors), the distinguisher $\mathcal{B}_3$ follows by first applying the aforementioned transformation and then applying $\mathcal{A}$.

Putting together Eq. (4.4), (4.5), (4.6), (4.7), (4.8), the lemma follows. □

Chapter 5

Hardness of Module-SIS and Module-LWE

In this chapter, which is part of a joint work with D. Stehlé, published in [LS], we bridge the reductions from SIVP to SIS and Id-SIVP to R-SIS on the first hand, and from SIVP to LWE and Id-SIVP to R-LWE on the second hand. We consider two problems M-SIS and M-LWE, where the letter M stands for module. A module is an algebraic structure generalizing rings and vector spaces, whereas module lattices (corresponding to finitely generated modules over the ring of integers of a number field) generalize both arbitrary lattices and ideal lattices. Note that M-LWE has recently been introduced (although not studied) in [BGV11], where it is called Generalized-LWE. We describe two new worst-case to average-case reductions: A reduction from Mod-SIVP (i.e., SIVP restricted to module lattices) to M-SIS in the proof of Theorem 5.10, and a (quantum) reduction from Mod-SIVP to M-LWE in both its search and decision versions in the proofs of Theorems 5.17 and 5.19. We also show that the Mod-SIVP to M-SIS/M-LWE reductions admit converse reductions (with a module rank degradation): from M-SIS to Mod-SIVP in Theorem 5.43 and from M-LWE to Mod-SIVP in Theorem 5.44.

The Mod-SIVP to M-SIS and Mod-SIVP to M-LWE reductions are smooth generalizations of the existing reductions: By setting the module dimension and the field degree appropriately, we recover the former reductions. When doing so, the conditions on the approximation factor γ and the modulus q required for the results to hold match with the conditions of the existing reductions,[1] up to logarithmic factors with respect to the lattice dimension. These parameters quantify the quality of the reductions: The hardness of the SIVP problem is given by the approximation factor γ, whereas the bit-size of the average-case instances is proportional to $\log q$.

To achieve these results, we carefully combine and adapt the existing reductions and their proofs of correctness ([GPV08] and [LM06] for M-SIS, and [Reg09] and [LPR10] for M-LWE). At a high level, the module structure can be seen as a "tensor" between the lattice and ideal algebraic structures, leading to reductions and proof that can heuristically be seen as "tensors" of the former reductions and proofs.

On the way, we improve the state-of-the-art results on the hardness of R-SIS and R-LWE. Concerning R-SIS: We improve the reduction from Id-SIVP$_\gamma$ by allowing for smaller values of q; this improvement is obtained by adapting a technique based on the Chinese Remainder Theorem and developed by Lyubashesvky et al. in [LPR10] in the context of R-LWE; its application to R-SIS was suggested in [LPR10] but left open. Concerning R-LWE: We show that R-LWE is hard for all sufficiently large q, independently of the arithmetic properties of q with respect to the ring dimension n; this improvement is obtained by adapting the modulus-switching technique

[1] with the exception of the recent result of Micciancio and Peikert [MP13] on the hardness of SIS and LWE with small parameters.

we developed in Chapter 4 in the context of LWE.

A larger toolbox for cryptographic design. The hardness results for M-SIS and M-LWE possibly enlarge the toolbox for devising lattice-based cryptosystems. Let us consider small examples. The following is an instance of M-SIS for which we can prove hardness for specific values of the parameters n, q and β. Given $a_{i,j}$'s sampled uniformly and independently from the uniform distribution over $\mathbb{Z}_q[x]/(x^n+1)$, the goal is to find z_i's in $\mathbb{Z}[x]/(x^n+1)$ not all zero, with coefficients smaller than a prescribed bound β and such that:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = 0 \mod q.$$

Similarly, our results on M-LWE imply that for specific values of n, q and for a specific error distribution ψ taking small values in $\mathbb{Z}[x]/(x^n+1)$ (or, actually, a specific distribution over such distributions), the following pair is computationally indistinguishable from uniform over its range:

$$\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}, \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \mod q \right),$$

where the a_{ij}'s and s_i's are sampled uniformly in $\mathbb{Z}_q[x]/(x^n+1)$, and the e_i's are sampled from ψ.

Note that the existing results on R-LWE and R-SIS already imply, via naive reductions, that these problems are no easier than some SIVP instances: For example, one can embed an R-SIS instance into the first row of an M-SIS instance, and generate the other row(s) independently. However, with this approach, the hardness of the corresponding worst-case instances is related to n-dimensional instances of SIVP. By tailoring our reduction to the module case, we can show that the M-SIS instance above is no easier than solving SIVP for a $(2n)$-dimensional lattice (or, more generally, a (dn)-dimensional lattice, if the number of rows of the M-SIS matrix is d).

From the cryptographic construction viewpoint, we expect that most constructions based on R-SIS and R-LWE can be adapted to M-SIS and M-LWE, with an efficiency slowdown (in terms of memory requirements, communication costs and algorithm run-times) bounded by a constant factor when $d = O(1)$.

Hedging against a possible non-hardness of Id-SIVP. Our results lead to cryptographic primitives whose efficiencies are within a constant factor of those based on R-SIS/R-LWE, but for which the security relies on Mod-SIVP instead of Id-SIVP. We argue here that Mod-SIVP is possibly a harder problem than Id-SIVP.

As a first observation, we emphasize that there exists a naive reduction from Id-SIVP to Mod-SIVP, as any Id-SIVP instance can be embedded into a Mod-SIVP instance of higher dimension (e.g., the Id-SIVP instance may be duplicated into two orthogonal subspaces), but no converse reduction is currently known.

Further, Id-SIVP has been much less studied than SIVP, and attacks on SIVP working only in the case of ideal lattices cannot be fully ruled out. Such attacks could, for example, exploit the multiplicative structure of the ideals, and fail to hold as soon as the rank d of the module is greater than 1 (i.e., a phase transition between $d = 1$ and $d > 1$). Such weaknesses due to the multiplicative structure actually exist for some lattice problems. Consider for example the task of estimating, within a factor γ, the euclidean norm of the shortest nonzero vector in the lattice (known as GapSVP_γ). This problem is suspected to be extremely hard in the worst case for values of γ that are polynomial in the lattice dimension. But it is easy for ideal lattices, as Minkowski's

bound on the lattice minimum is known to be essentially sharp in that case (see, e.g., [PR07, Se. 6]). Further, we suspect that this problem is hard in the worst case for module lattices with module rank greater than 1, as it would allow one to efficiently solve two notable problems. For module rank 2, it would lead to an efficient algorithm for the Decisional Small Polynomial Ratio problem (DSPR) from [LATV12], inspired from the NTRU encryption scheme [HPS98]. The goal of DSPR is to determine whether a given $h \in \mathbb{Z}_q[x]/(x^n+1)$ is uniformly sampled or sampled of the form $h = g/f \bmod q$ where both f and g have small coefficients. A GapSVP oracle used on the (rank 2) module $\{(k_1, k_2) \in (Z[x]/(x^n+1))^2 : k_1 h + k_2 = 0 \bmod q\}$ would allow one to solve DSPR: for a uniform h, the lattice minimum is expected to be of the order of $\approx \sqrt{nq}$ (as the determinant is q^n and the dimension is $2n$), whereas for $h = g/f$, the minimum is $\leq \sqrt{\|f\|^2 + \|g\|^2}$. For module rank 3, it would lead to an efficient algorithm for R-LWE. With the same notations as above, R-LWE with three sample pairs consists in deciding whether $(a_1, b_1), (a_2, b_2), (a_3, b_3)$ are uniformly and independently sampled in $(\mathbb{Z}_q[x]/(x^n+1))^2$, or whether there exists $s \in \mathbb{Z}_q[x]/(x^n+1)$ such that $b_i - a_i \cdot s$ has small coefficients for all $i \in \{1, 2, 3\}$. In the first case, the lattice minimum of the (rank 3) module $(a_1, a_2, a_3) \cdot \mathbb{Z}_q[x]/(x^n+1) + (b_1, b_2, b_3) \cdot \mathbb{Z}_q[x]/(x^n+1) + (q\mathbb{Z}[x]/(x^n+1))^3$ is expected to be of the order of $\approx \sqrt{n} q^{1/3}$ (the determinant is q^{2n},unless we are in a degenerate case, which occurs with small probability). In the second case, the minimum is unexpectedly small. We note that no such phase transition is known for Id-SIVP, but it cannot be ruled out given our current knowledge.

From an algorithmic viewpoint, attempts [Nap96, FP96, GLM09] have been made to adapt the existing algorithms for integer lattices to modules over the rings of integers of number fields. However, no norm bound is known for the output of the algorithm of [FP96], and the results of [Nap96, GLM09] hold for very limited cases. The main approach to handle these lattices is to discard their module structure and view the modules as integer lattices of much higher dimensions [Coh00, FS10].

Related works. Most SIVP to SIS reductions (including ours) consider the euclidean norm. Peikert [Pei08] described an SIVP to SIS reduction that handles all ℓ_p norms. Independently, many variants of LWE have been shown as hard as Regev's original LWE: These variants may consist in sampling the secret vector $\mathbf{s}$ from the same distribution as the errors [ACPS09], in sampling the error vectors from other distributions [Pei09, GKV10] and in relaxing the conditions on the factorisation of the modulus [MP12, Se. 3] (see also the references therein). Other cryptographically useful variants of SIS and LWE proven as secure as SIVP include k-SIS [BF11], ISIS [GPV08], subspace-LWE [KPC+11, Pie12] and extended-LWE [OPW11, ASP12] and Chapter 4.

In [Pei09] and Chapter 4, Peikert and Brakerski et al. partially dequantized Regev's proof of hardness of LWE [Reg09], by proposing a reduction from the decisional GapSVP_γ problem to LWE. Peikert's classical reduction is restricted to large LWE moduli q (that are additionally required to be products of many small primes in the case of the decisional variant of LWE), unless one considers a variant of GapSVP that is somewhat unusual. Peikert's dequantization carries over to the module case, by giving a reduction from GapSVP restricted to module lattices to M-LWE (using Lemma 5.26 from Section 5.2.3). Note that it also carries over to ideal/R-LWE setting but is meaningless in this situation as GapSVP is easy for ideal lattices and the involved approximation factors γ (as a good approximation to the minimum known). The reduction of Brakerski et al. consists of several steps, the first one being Peikert's reduction. It is thus equally useless in the case of ideal lattices.

Some computational aspects of module lattices have been investigated in [BP91, FS10] (see also [Coh00, Ch. 1]). These results show that the additional algebraic structure may be exploited to obtain compact representations of modules (namely, pseudo-bases) similar to lattice bases in

Hermite Normal Form and LLL-reduced lattice bases. None hints that SIVP would be any easier when restricted to module lattices.

Peikert and Rosen [PR07] observed that solving R-SIS exactly consists in finding a short nonzero vector in a module lattice.

Road-map. In Section 5.1 we give a reduction from Mod-SIVP to M-SIS. Then, in Section 5.2, we describe a (quantum) reduction from Mod-SIVP to both the computational and the decisional variants of M-LWE. Finally, we give converse reductions in Section 5.3, i.e., reductions from both M-SIS and M-LWE to Mod-SIVP.

Note. We refer to Chapter 1, Section 1.2 for algebraic number theory reminders. The ring R is the ring of integers of a number field K which is also a cyclotomic field, for example one can take $R \simeq \mathbb{Z}[x]/(x^n+1)$ for n a power of 2.

We propose in this chapter a unified analysis of R-SIS/M-SIS and R-LWE/M-LWE by only considering the complex canonical embeddings of the ring elements. Note that all prior works on R-SIS except [PR07] used the polynomial embedding. However, the canonical embedding representation is mathematically sounder, and the unification leads to a more natural connection between R-SIS and R-LWE.

5.1 Hardness of Module-SIS

5.1.1 Variants of SIS

The Short Integer Solution problem (SIS) is fully described in Chapter 2, we now consider two structured variants: Ring-SIS and Module-SIS.

SIS over rings. The R-SIS problem was concurrently introduced in [PR06] and [LM06].

Definition 5.1. The problem $R\text{-}SIS_{q,m,\beta}$ is as follows: Given $a_1, \ldots, a_m \in R_q$ chosen independently from the uniform distribution, find $z_1, \ldots, z_m \in R$ such that $\sum_{i=1}^{m} a_i \cdot z_i = 0 \bmod q$ and $0 < \|\mathbf{z}\| \leq \beta$, where $\mathbf{z} = (z_1, \ldots, z_m)^T \in R^m$.

This problem over rings can be interpreted in terms of structured integer matrices. For example, when n is a power of 2, then R and R_q are isomorphic to $\mathbb{Z}[x]/(x^n+1)$ and $\mathbb{Z}_q[x]/(x^n+1)$ respectively, and the ring multiplication $a_i \cdot z_i$ can be written as the multiplication of the vector of $\mathbb{Z}^n$ whose entries are the coefficients of z_i and, with a nega-circulant matrix whose entries are derived from the coefficients of a_i. In this setup, R-SIS is a variant of SIS where $\mathbf{A}$ is restricted to being block nega-circulant: $\mathbf{A} = [\mathrm{Rot}(a_1)|\ldots|\mathrm{Rot}(a_m)]$, with:

$$\mathrm{Rot}(b) := \begin{bmatrix} b_0 & -b_{n-1} & \cdots & -b_1 \\ b_1 & b_0 & \cdots & -b_2 \\ \vdots & \vdots & \ddots & \vdots \\ b_{n-1} & b_{n-2} & \cdots & b_0 \end{bmatrix}, \text{ for } b = \sum_{i=0}^{n-1} b_i x^i \in R.$$

SIS over modules. The problem M-SIS generalizes both SIS and R-SIS. We use the following notations: the variable n denotes the dimension of the ring R and the variable d corresponds to the rank of the module $M \subseteq R^d$; we let $N = nd$ denote the dimension of the corresponding module lattice, and give the complexity statements for N growing to infinity.

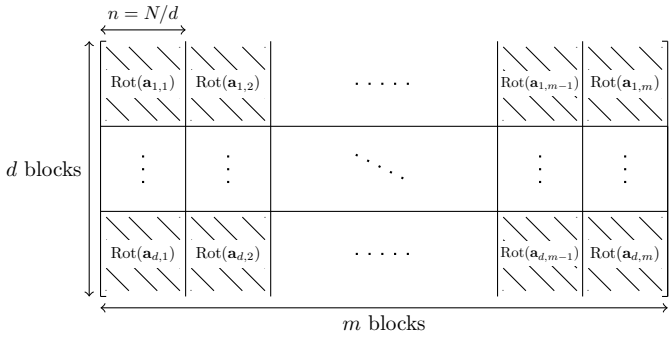

Figure 5.1: Structured $\mathbf{A}$ matrix for Module-SIS.

Definition 5.2. The problem *M-SIS*$_{q,m,\beta}$ is as follows: Given $\mathbf{a}_1, \ldots, \mathbf{a}_m \in R_q^d$ chosen independently from the uniform distribution, find $z_1, \ldots, z_m \in R$ such that $\sum_{i=1}^m \mathbf{a}_i \cdot z_i = 0 \bmod q$ and $0 < \|\mathbf{z}\| \leq \beta$, where $\mathbf{z} = (z_1, \ldots, z_m)^T \in R^m$.

Like R-SIS, M-SIS can be interpreted in terms of matrices. In the same setting as above for R-SIS, it consists in taking a SIS matrix $\mathbf{A}$ of the form described in Figure 5.1.

5.1.2 Hardness of Ring-SIS

The hardness of the Ring-SIS problem is proven in [PR06] and [LM06].

Theorem 5.3 (Adapted from [LM06])**.** *For $\varepsilon(n) = n^{-\omega(1)}$, there is a probabilistic polynomial time reduction from solving* Ideal-GIVP$_\gamma^{\eta_\varepsilon}$ *in polynomial time (in the worst case, with high probability) to solving* R-SIS$_{q,m,\beta}$ *in polynomial time with non-negligible probability, for any $m(n), q(n), \beta(n)$ and $\gamma(n)$ such that $\gamma \geq \beta\sqrt{n} \cdot \omega(\sqrt{\log n})$, $q \geq \beta n^{1.5} \cdot \omega(\log n)$ and $m, \log q \leq \mathrm{poly}(n)$.*

By using a technique from [LPR10] (namely, the isomorphism between I/qI and R_q described in Subsection 1.2) into the proof of [LM06], we obtain the following result.

Theorem 5.4. *For $\varepsilon(n) = n^{-\omega(1)}$, there is a probabilistic polynomial time reduction from solving Ideal-*GIVP$_\gamma^{\eta_\varepsilon}$ *in polynomial time (in the worst case, with high probability) to solving* R-SIS$_{q,m,\beta}$ *in polynomial time with non-negligible probability, for any $m(n), q(n), \beta(n)$ and $\gamma(n)$ such that $\gamma \geq \beta\sqrt{n} \cdot \omega(\sqrt{\log n})$, $q \geq \beta\sqrt{n} \cdot \omega(\log n)$ and $m, \log q \leq \mathrm{poly}(n)$.*

In the rest of this section, we provide the reduction.

In order to prove that R-SIS (and in Section 5.1 for the new problem M-SIS) is at least as hard as GIVP restricted to ideal (respectively module) lattices, we use the following intermediate problem, introduced in [MR04].

Definition 5.5 ([MR04, Definition 5.3])**.** The *Incremental Independent Vectors Problem Inc-GIVP*$_\gamma^{\eta_\varepsilon}$, is as follows: Given a tuple $(\mathbf{B}, \mathbf{S}, \mathcal{H})$ where $\mathbf{B}$ is a basis of an n-dimensional lattice, $\mathbf{S} \subseteq \mathcal{L}(\mathbf{B})$ is a full-rank set of vectors such that $\|\mathbf{S}\| \geq \gamma \cdot \eta_\varepsilon(\mathcal{L}(\mathbf{B}))$ and $\mathcal{H}$ is a hyperplane, find $\mathbf{h} \in \mathcal{L}(\mathbf{B}) \setminus \mathcal{H}$ such that $\|\mathbf{h}\| \leq \|\mathbf{S}\|/2$.

Theorem 5.6 ([Mic04, Th. 6.3])**.** *For any function ε and γ, there is a probabilistic polynomial time reduction from solving* GIVP$_\gamma^{\eta_\varepsilon}$ *(in the worst case, with high probability) to solving* IncGIVP$_\gamma^{\eta_\varepsilon}$ *(in the worst case, with high probability).*

As the latter reduction preserves the lattice, it induces a reduction from Ideal-GIVP$^{\eta_\varepsilon}_{\gamma}$ to Ideal-IncGIVP$^{\eta_\varepsilon}_{\gamma}$, i.e., IncGIVP$^{\eta_\varepsilon}_{\gamma}$ restricted to ideal lattices. To prove Theorem 5.4, we provide a reduction from Ideal-IncGIVP$^{\eta_\varepsilon}_{\gamma}$ to R-SIS$_{q,m,\beta}$.

Suppose that an oracle $\mathcal{O}$ solves R-SIS$_{q,m,\beta}$ in polynomial time with probability $n^{-O(1)}$. The algorithm for Ideal-IncGIVP proceeds as follows on input $(\mathbf{B}, \mathbf{S}, \mathcal{H})$. We write $I = \mathcal{L}(\mathbf{B})$. Let s be such that

$$\max\left(\frac{2q}{\gamma}, \sqrt{\log n}\right) \|\mathbf{S}\| \leq s \leq \frac{q\|\mathbf{S}\|}{2\beta\sqrt{n}\cdot\omega(\sqrt{\log n})}$$

- For all $\ell \leq m$,
  - Get a fresh y_ℓ distributed as $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{0}}$ (using Theorem 1.24),
  - Let $a_\ell = \theta_I^{-1}(y_\ell \mod qI) \in R_q$ (see the definition of θ_I in Section 1.2.6).
- Invoke the oracle $\mathcal{O}$ on input $(a_1, \ldots, a_m)$. If $\mathcal{O}$ succeeds, it returns $\mathbf{z} = (z_1, \ldots, z_m)^T \in R^m$ such that $\sum_{\ell=1}^m a_\ell \cdot z_\ell = 0 \bmod q$ and $0 < \|\mathbf{z}\| \leq \beta$.
- Output $h = \frac{1}{q}\sum_{\ell=1}^m z_\ell \cdot y_\ell$.

This algorithm runs in polynomial time. Also, thanks to the parameter constraints, the interval to which the standard deviation s must belong is nonempty. Moreover, the standard deviation s is sufficiently large for the assumptions of Theorem 1.24 to hold. Indeed, by Lemma 1.8 and given I and $\mathbf{S}$, it is possible to compute (in polynomial time) a basis $\mathbf{T}$ of I such that $\|\widetilde{\mathbf{T}}\| \leq \|\widetilde{\mathbf{S}}\| \leq \|\mathbf{S}\|$. We use this basis and we have that $s \geq \|\widetilde{\mathbf{T}}\| \cdot \sqrt{\log n}$.

The following lemmata are particular cases of Lemmata 5.11, 5.12 and 5.13 (we then refer to those lemmata for the proofs).

Lemma 5.7. *The statistical distance between the distribution of $(a_1, \ldots, a_m)$ and the uniform distribution over R_q is at most $2m\varepsilon$.*

As a consequence, the oracle $\mathcal{O}$ succeeds with probability $n^{-O(1)}$. In the following, we assume we are in that situation.

Lemma 5.8. *For any hyperplane $\mathcal{H}$, the probability that the output vector h does not belong to $\mathcal{H}$ is $\geq 1/100$.*

Lemma 5.9. *We have $h \in I$ and, with probability close to 1, we have that $\|h\| \leq \|\boldsymbol{S}\|/2$.*

It completes the proof of Theorem 5.4.

5.1.3 Hardness of Module-SIS

In this section, we provide our worst-case to average-case reduction from Mod-GIVP to M-SIS. We will now prove the following result.

Theorem 5.10. *For any $d \geq 1$ and $\varepsilon(N) = N^{-\omega(1)}$, there is a probabilistic polynomial time reduction from solving* Mod-GIVP$^{\eta_\varepsilon}_{\gamma}$ *in polynomial time (in the worst case, with high probability) to solving* M-SIS$_{q,m,\beta}$ *in polynomial time with non-negligible probability, for any $m(N), q(N), \beta(N)$ and $\gamma(N)$ such that $\gamma \geq \beta\sqrt{N}\cdot\omega(\sqrt{\log N})$, $q \geq \beta\sqrt{N}\cdot\omega(\log N)$ and $m, \log q \leq \mathrm{poly}(N)$.*

In the case of a sub-exponential oracle (and with $\varepsilon(N) = 2^{-\Omega(N)}$), the result still holds and the conditions on the parameters become $\gamma \geq \beta \cdot \Omega(N)$ and $q \geq \beta \cdot \Omega(N^{3/2})$.

Taking $n = N$ and $d = 1$ in Theorem 5.10 allows us to recover Theorem 5.4. Also, by taking $n = 1$ and $d = N$ in Theorem 5.10, we obtain a hardness result for SIS that is as good as that of Theorem 2.3.

A reduction from Mod-GIVP to M-SIS. In order to prove that the new problem M-SIS is at least as hard as GIVP restricted to module lattices, we also use the intermediate problem IncGIVP, see Definition 5.5, introduced in [MR04].

As the reduction of Theorem 5.6 preserves the lattice, it induces a reduction from Mod-GIVP$^{\eta_\varepsilon}_{\gamma}$ to Mod-IncGIVP$^{\eta_\varepsilon}_{\gamma}$, i.e., IncGIVP$^{\eta_\varepsilon}_{\gamma}$ restricted to module lattices. To prove Theorem 5.10, we provide a reduction from Mod-IncGIVP$^{\eta_\varepsilon}_{\gamma}$ to M-SIS$_{q,m,\beta}$.

Suppose that an oracle $\mathcal{O}$ solves M-SIS$_{q,m,\beta}$ in polynomial time with probability $N^{-O(1)}$. The algorithm for Mod-IncGIVP proceeds as follows on input $(\mathbf{B}, \mathbf{S}, \mathcal{H})$. We write $M = \mathcal{L}(\mathbf{B})$. Let s be such that

$$\max\left(\frac{2q}{\gamma}, \sqrt{\log N}\right)\|\mathbf{S}\| \leq s \leq \frac{q\|\mathbf{S}\|}{2\beta\sqrt{N}\cdot\omega(\sqrt{\log N})}$$

- For all $\ell \leq m$,
  - Get a fresh $\mathbf{y}_\ell$ distributed as $D_{\mathcal{L}(\mathbf{B}),s,\mathbf{0}}$ (using Theorem 1.24),
  - Let $\mathbf{a}_\ell = \Theta^{-1}(\mathbf{y}_\ell \mod qM)$ (see the definition of Θ in Section 1.2.6).
- Invoke the oracle $\mathcal{O}$ on input $(\mathbf{a}_1, \ldots, \mathbf{a}_m)$. If $\mathcal{O}$ succeeds, it returns $\mathbf{z} = (z_1, \ldots, z_m)^T \in R^m$ such that $\sum_{\ell=1}^m \mathbf{a}_\ell \cdot z_\ell = \mathbf{0} \bmod q$ and $0 < \|\mathbf{z}\| \leq \beta$.
- Output $\mathbf{h} = \frac{1}{q}\sum_{\ell=1}^m z_\ell \cdot \mathbf{y}_\ell$.

This algorithm runs in polynomial time. Also, thanks to the parameter constraints, the interval to which the standard deviation s must belong is nonempty. Moreover, the standard deviation s is sufficiently large for the assumptions of Theorem 1.24 to hold. Indeed, by Lemma 1.8 and given M and $\mathbf{S}$, it is possible to compute (in polynomial time) a basis $\mathbf{T}$ of M such that $\|\widetilde{\mathbf{T}}\| \leq \|\widetilde{\mathbf{S}}\| \leq \|\mathbf{S}\|$. We use this basis and we have that $s \geq \|\widetilde{\mathbf{T}}\| \cdot \sqrt{\log N}$.

Lemma 5.11. *The statistical distance between the distribution of $(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m)$ and the uniform distribution over R_q^d is at most $2m\varepsilon$.*

Proof. We have $s \geq \frac{2q}{\gamma} \cdot \|\mathbf{S}\|$ and $\|\mathbf{S}\| \geq \gamma \cdot \eta_\varepsilon(M)$. This implies that $s \geq q \cdot \eta_\varepsilon(M) = \eta_\varepsilon(qM)$. By Lemma 1.33 applied to the lattices M and qM, the statistical distance between the distribution of $(\mathbf{y}_\ell \bmod qM)$ and the uniform distribution on M/qM is at most 2ε. As Θ^{-1} is an isomorphism from M/qM to $(R/qR)^d$, the statistical distance between the distribution of the $\mathbf{a}_\ell = \Theta^{-1}(\mathbf{y}_\ell)$ and the uniform distribution on $(R/qR)^d$ is also at most 2ε. The result follows. □

As a consequence, the oracle $\mathcal{O}$ succeeds with probability $N^{-O(1)}$. In the following, we assume we are in that situation.

Lemma 5.12. *For any hyperplane $\mathcal{H}$, the probability that the output vector $\boldsymbol{h}$ does not belong to $\mathcal{H}$ is $\geq 1/100$.*

Proof. As $\mathcal{O}$ succeeded, the vector $\mathbf{z}$ is nonzero. By definition of $\mathbf{h}$, for every $\mathbf{y}'_1$ we have:

$$\begin{aligned}\mathbf{h} \in \mathcal{H} &\Leftrightarrow \sum_{\ell=1}^m z_\ell \cdot \mathbf{y}_\ell \in \mathcal{H} \Leftrightarrow\ z_1 \cdot \mathbf{y}_1 \in -\sum_{i=2}^m z_\ell \cdot \mathbf{y}_\ell + \mathcal{H} \\ &\Leftrightarrow (\mathbf{y}_1 - \mathbf{y}'_1) \in -\mathbf{y}'_1 + \frac{1}{z_1}(\mathcal{H} - \sum_{i=2}^m z_\ell \cdot \mathbf{y}_\ell)\ =\ \mathcal{H}'.\end{aligned}$$

Assume that we fix $\mathbf{y}'_1 = \mathbf{y}_1 \bmod qM$, then $\mathbf{y}_1 = \mathbf{y}'_1 + \mathbf{y}''_1$, with $\mathbf{y}'_1$ fixed and the vector $\mathbf{y}''_1$ statistically independent of all the $\mathbf{a}_\ell$'s, z_ℓ's and $\mathbf{y}_\ell$'s for $\ell > 1$. The conditional distribution of $\mathbf{y}''_1 = (\mathbf{y}_1 - \mathbf{y}'_1)$ is $D_{qM,s,-\mathbf{y}'_1}$. Therefore:

$$\Pr\left[(\mathbf{y}_1 - \mathbf{y}'_1) \notin \mathcal{H}' | \mathbf{y}'_1, (\mathbf{a}_1, \ldots, \mathbf{a}_m), (z_1, \ldots, z_m)\right] = \Pr_{\mathbf{y}''_1 \hookleftarrow D_{qM,s,-\mathbf{y}'_1}}[\mathbf{y}''_1 \notin \mathcal{H}'].$$

As $s \geq 2q \cdot \eta_\varepsilon(M) = 2\eta_\varepsilon(qM)$, Lemma 1.34 gives that this probability is $\geq 1/100$. □

The following completes the proof of Theorem 5.10.

Lemma 5.13. *We have $\boldsymbol{h} \in M$ and, with probability close to 1, we have that $\|\boldsymbol{h}\| \leq \|\boldsymbol{S}\|/2$.*

Proof. Let us first show that $\mathbf{h} \in M$. We have, modulo qM:

$$\sum_{\ell=1}^{m} z_\ell \cdot \mathbf{y}_\ell = \sum_{\ell=1}^{m} z_\ell \cdot \Theta(\mathbf{a}_\ell) = \Theta(\sum_{\ell=1}^{m} z_\ell \mathbf{a}_\ell) = \mathbf{0}.$$

This implies that $\mathbf{h} = (\sum_{\ell=1}^{m} z_\ell \cdot \mathbf{y}_\ell)/q$ belongs to M.

We now show that $\|\mathbf{h}\| \leq \|\mathbf{S}\|/2$. We have $\|\mathbf{h}\| = \|\sum_{\ell=1}^{m} z_\ell \cdot \mathbf{y}_\ell\|/q$. As in the previous proof, we define $\mathbf{y}'_\ell = \mathbf{y}_\ell \bmod qM$. Then, we have $\mathbf{y}_\ell = \mathbf{y}''_\ell + \mathbf{y}'_\ell$ with $\mathbf{y}''_\ell$ statistically independent from the z_ℓ's and distributed as $D_{qM,s,-\mathbf{y}'_\ell}$. By Lemma 1.39, for $s \geq \eta_\varepsilon(qM)$ and $t = \omega(\sqrt{\log N})$, we know that:

$$\Pr_{\forall \ell : \mathbf{y}''_\ell \hookleftarrow D_{qM,s,-\mathbf{y}'_\ell}} \left[\left\|\sum_{\ell=1}^{m} z_\ell \cdot (\mathbf{y}''_\ell + \mathbf{y}'_\ell)\right\| \geq st\sqrt{N} \cdot \|\mathbf{z}\|\right] \leq N^{-\omega(1)}.$$

So, with probability close to 1, we have $\|\sum_{\ell=1}^{m} z_\ell \cdot \mathbf{y}_\ell\| \leq st\sqrt{N} \cdot \|\mathbf{z}\|$. As $0 < \|\mathbf{z}\| \leq \beta$, we obtain:

$$\|\mathbf{h}\| = \frac{1}{q}\left\|\sum_{\ell=1}^{m} z_\ell \cdot \mathbf{y}_\ell\right\| \leq \frac{st\beta\sqrt{N}}{q}.$$

Finally, since $s \leq \frac{q \cdot \|\mathbf{S}\|}{2\beta t\sqrt{N}}$, we obtain $\|\mathbf{h}\| \leq \frac{\|\mathbf{S}\|}{2}$. □

5.2 Hardness of Module-LWE

5.2.1 Variants of LWE

We first define the two variants of the Learning with Errors problem over rings and over modules.

LWE over rings. The R-LWE problem was introduced by Lyubashevsky et al. in [LPR10]. Let ψ be a distribution on $\mathbb{T}_{R^\vee} = K_\mathbb{R}/R^\vee$ and $s \in R_q^\vee$. We let $A^{(R)}_{s,\psi}$ denote the distribution on $R_q \times \mathbb{T}_{R^\vee}$ obtained by choosing $a \in R_q$ uniformly at random and $e \in \mathbb{T}_{R^\vee}$ according to ψ, and returning $(a, (a \cdot s)/q + e)$.

Definition 5.14. Let $q \geq 2$ and Ψ be a family of distributions on $\mathbb{T}_{R^\vee}$. The *search* version of the *Ring Learning With Error problem R-SLWE*$_{q,\Psi}$ is as follows: Let $s \in R_q^\vee$ be secret and $\psi \in \Psi$; Given arbitrarily many samples from $A^{(R)}_{s,\psi}$, the goal is to find s.

Let Υ be a distribution over a family of noise distributions over $K_\mathbb{R}$. The *decision* version of the *Ring Learning With Error problem R-LWE*$_{q,\Upsilon}$ is as follows: Let $s \in R_q^\vee$ be uniformly random and ψ be sampled from Υ; The goal is to distinguish between arbitrarily many independent samples from $A^{(R)}_{\mathbf{s},\psi}$ and the same number of independent samples from $U(R_q, \mathbb{T}_{R^\vee})$.

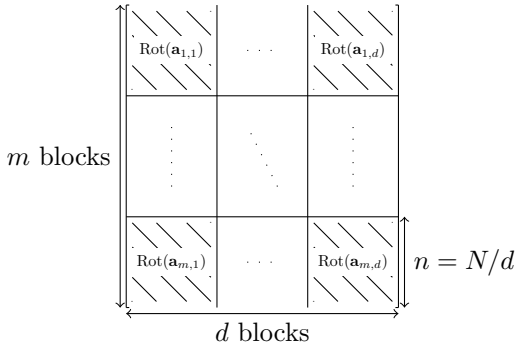

Figure 5.2: Structured $\mathbf{A}$ matrix for Module-LWE.

As for R-SIS, this problem can be interpreted in terms of linear algebra. In the same example setting as in the case of R-SIS (i.e., the parameter ν is set to a power of 2, implying that $R \simeq \mathbb{Z}[x]/(x^n + 1)$), R-SIS is a variant of LWE where the matrix $\mathbf{A}$ is restricted to being block-negacirculant: $\mathbf{A} = [\mathrm{Rot}(a_1)|\ldots|\mathrm{Rot}(a_m)]^T$. The two main results from [LPR10] are a reduction from Id-GIVP to R-SLWE and a reduction from the search version R-SLWE to the decision version R-LWE.

Theorem 5.15 ([LPR10, Th. 4.1 and Th. 5.1])**.** *Let $\varepsilon(n) = n^{-\omega(1)}$, $\alpha \in (0,1)$ and $q \geq 2$ of known factorization such that $\alpha q > \omega(\sqrt{\log n})$. There exists a quantum reduction from solving* Id-GIVP$^{\eta_\varepsilon}_{\gamma}$ *in polynomial time (in the worst case, with high probability) to solving* R-SLWE$_{q,\Psi_{\leq\alpha}}$ *in polynomial time with non-negligible probability with $\gamma = \sqrt{n} \cdot \omega(\sqrt{\log n})/\alpha$.*

Assume that q is prime, $q \leq \mathrm{poly}(n)$, and that $q = 1 \bmod \nu$. Then there exists a polynomial time reduction from R-SLWE$_{q,\Psi_{\leq\alpha}}$ *to* R-LWE$_{q,\Upsilon_\alpha}$*.*

LWE over modules. The M-LWE problem generalizes both LWE and R-LWE, and was recently introduced in [BGV11]. As in Chapter 5, the variable n and d respectively denote the dimension of R and the rank of the module $M \subseteq R^d$; We still let $N = nd$ denote the dimension of the corresponding module lattice.

Let ψ be some probability distribution on $\mathbb{T}_{R^\vee}$ and $\mathbf{s} \in (R_q^\vee)^d$ be a vector. We define $A^{(M)}_{q,\mathbf{s},\psi}$ as the distribution on $(R_q)^d \times \mathbb{T}_{R^\vee}$ obtained by choosing a vector $\mathbf{a} \in (R_q)^d$ uniformly at random, and $e \in \mathbb{T}_{R^\vee}$ according to ψ, and returning $(\mathbf{a}, \frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle + e)$.

Definition 5.16. Let $q \geq 2$ and Ψ be a family of distributions on $\mathbb{T}_{R^\vee}$. The *search* version of the *Module Learning With Error problem M-SLWE*$_{q,\Psi}$ is as follows: Let $\mathbf{s} \in (R_q^\vee)^d$ be secret and $\psi \in \Psi$; Given arbitrarily many samples from $A^{(M)}_{q,\mathbf{s},\psi}$, the goal is to find $\mathbf{s}$.

For an integer $q \geq 2$ and a distribution Υ over a family of distributions over $K_\mathbb{R}$. The *decision* version of the *Module Learning With Error problem M-LWE*$_{q,\Upsilon}$ is as follows: Let $\mathbf{s} \in (R_q^\vee)^d$ be uniformly random and ψ be sampled from Υ; The goal is to distinguish between arbitrarily many independent samples from $A^{(M)}_{q,\mathbf{s},\psi}$ and the same number of independent samples from $U(R_q^d, \mathbb{T}_{R^\vee})$.

As for LWE and R-LWE, the problem M-LWE can be interpreted in terms of linear algebra. Still in the same example setting, it consists in taking the LWE matrix $\mathbf{A}$ of the form:

We now give our two main results concerning the hardness of M-LWE, in the following two theorems.

Theorem 5.17. *Let $\varepsilon(N) = N^{-\omega(1)}$, $\alpha \in (0,1)$ and $q \geq 2$ of known factorization such that $\alpha q > 2\sqrt{d} \cdot \omega(\sqrt{\log n})$. There is a quantum reduction from solving* Mod-GIVP$^{\eta_\varepsilon}_{\gamma}$ *in polynomial time (in the worst case, with high probability) to solving* M-SLWE$_{q,\Psi_{\leq\alpha}}$ *in polynomial time with non-negligible advantage with $\gamma = \sqrt{8Nd} \cdot \omega(\sqrt{\log n})/\alpha$.*

Assume that q is prime, $q \leq \mathrm{poly}(N)$ and that $q = 1 \bmod \nu$. Then there exists a polynomial time reduction from M-SLWE$_{q,\Psi_{\leq\alpha}}$ *to* M-LWE$_{q,\Upsilon_\alpha}$.

In the case of a sub-exponential oracle (and with $\varepsilon(N) = 2^{-\Omega(N)}$), the result still holds and the conditions on the parameters become $\alpha q > 2\sqrt{d} \cdot \Omega(\sqrt{n})$ and $\gamma = d \cdot \Omega(n)/\alpha$.

When $n = N$ and $d = 1$, our theorem is identical to Theorem 5.15 [LPR10, Th. 3.1]. When $n = 1$ and $d = N$, it is identical to Theorem 2.6 [Reg09, Th. 4.1 and 5.1] (apart from a minor detail with the error distribution which can easily be handled: we use Υ_α rather than D_α).

In [LPR10], this result is completed by [LPR10, Th. 5.2], that states that R-LWE with spherical noise is also hard if the number of samples is limited. We adapt it to the module setting.

Theorem 5.18. *Let $\varepsilon(N) = N^{-\omega(1)}$, $\alpha \in (0,1)$ and $q \geq 2$ prime, with $q \leq \mathrm{poly}(N)$ and $q = 1 \bmod \nu$ such that $\alpha q > 2\sqrt{d} \cdot \omega(\sqrt{\log n})$. There is a quantum reduction from solving* M-SLWE$_{q,\Psi_{\leq\alpha}}$ *in polynomial time (in the worst case, with high probability) to solving* M-DLWE$_{q,D_\xi}$*, given only ℓ samples, in polynomial time with non-negligible advantage with $\xi = \alpha(n\ell/\log(n\ell))^{1/4}$.*

The proof relies on [LPR10, Le. 5.16], which carries over directly from the ring setting to the module setting.

Our second main result is the following:

Theorem 5.19. *Let $p, q \in [2, 2^{N^{O(1)}}]$ and $\alpha, \beta \in (0,1)$ such that $\beta \geq \alpha \cdot \max(1, \frac{q}{p}) \cdot n^{1/4} N^{1/2} \cdot \omega(\log^2 N)$ and $\alpha q \geq \omega(\sqrt{\log(N)/n})$. There exists a polynomial time reduction from* M-LWE$_{q,\Upsilon_\alpha}$ *to* M-LWE$_{p,\Upsilon_\beta}$.

In the case of a sub-exponential oracle (and with $\varepsilon(N) = 2^{-\Omega(N)}$), the result still holds and the conditions on the parameters become $\beta \geq \alpha \cdot \max(1, \frac{q}{p}) \cdot \Omega(n^{1/4} N^{5/2})$ and $\alpha q \geq \Omega(\sqrt{d})$.

Note that the condition on αq from Theorem 5.19 is always weaker than the one from Theorem 5.17. Combined with Theorem 5.17 by using a q prime close to p with $q = 1 \bmod \nu$, Theorem 5.19 provides a reduction from Mod-SIVP to M-LWE with a modulus p of arbitrary arithmetic form. As M-LWE is a generalization of both LWE and R-LWE, this theorem also provides a reduction from Id-SIVP$_\gamma$ to RLWE$_{p,\Upsilon_\beta}$, for a modulus p of arbitrary arithmetic shape. Note the in the case of LWE, this theorem is almost identical to Corollary 4.5.

The remainder of this chapter is devoted to proving Theorems 5.17 (Section 5.2.3 and 5.2.4) and 5.19 (Section 5.2.5).

5.2.2 Hardness of Ring-LWE

The Lyubashevsky et al reduction from Id-GIVP to R-SLWE relies on the same sequence of reductions as Regev's proof of hardness of SLWE (recalled in Section 2.2.2), but with problems restricted to ideal lattices. The only step in Regev's reduction that fails to carry over to the ideal/ring setting is Lemma 2.8. Lyubashevsky et al circumvent it by proving the following. In this Lemma, the problem q-Id-BDD is the restriction of q-BDD to ideal lattice lattices and instead of using the Euclidean norm for bounding the distance to the lattice, they use the infinity norm.

Lemma 5.20 ([LPR10, Le. 4.4])**.** *Let $\varepsilon = n^{-\omega(1)}$, $\alpha \in (0,1)$ and $q \geq 2$ of known factorization. Let $I \subseteq R$ be an ideal and $r \geq \sqrt{2}q \cdot \eta_\varepsilon(I)$. Given access to an oracle sampling from the distribution $D_{I,r}$, there exists a probabilistic reduction from solving q-*Id-BDD$_{I^\vee, \frac{\alpha q}{\sqrt{2}r}}$ *in polynomial*

time with non-negligible probability to solving R-SLWE$_{q,\Psi_{\leq\alpha}}$ *in polynomial time with non-negligible probability.*

The reduction from R-SLWE to R-LWE from [LPR10, Th. 5.2] proceeds by several reductions between intermediates problems, which we will also consider in our reduction for the module variant of LWE. Let $q = 1 \bmod 2n$ be prime, then $(q) = \prod_{i=1}^{n} \mathfrak{q}_i$ where any $\mathfrak{q}_i$ is a prime ideal with norm $N(\mathfrak{q}_i) = q$. Lyubashevsky et al define:

- $\mathfrak{q}_i$**-RLWE**$_{q,\Psi}$, with parameters Ψ a family of distributions over $\mathbb{T}_{R^\vee}$ and $i \leq n$: Given access to an oracle sampling from $A_{s,\psi}^{(R)}$ for an arbitrary $s \in R_q^\vee$ and $\psi \in \Psi$, find $s \bmod \mathfrak{q}_i R_q^\vee$.
- **Hybrid distribution** $A_{s,\psi}^{(R,i)}$, with parameters ψ a distribution over $\mathbb{T}_{R^\vee}$, $s \in R_q^\vee$, and $i \leq n$: The distribution $A_{s,\psi}^{(R,i)}$ over $R_q \times \mathbb{T}_{R^\vee}$ is defined as follows: Choose (a, b) from $A_{s,\psi}^{(R)}$ and return $(a, b + r/q)$ where r is uniformly random and independent in $R_q^\vee/\mathfrak{q}_j R^\vee$ for all $j \leq i$, and is 0 modulo the remaining $\mathfrak{q}_j R^\vee$'s.
- **DecRLWE**$_{q,\Psi}^{i}$, with parameters Ψ a family of distributions on $\mathbb{T}_{R^\vee}$ and $i \leq n$: Given access to an oracle sampling from $A_{s,\psi}^{(R,j)}$ for an arbitrary $s \in R_q^\vee$, $\psi \in \Psi$ and $j \in \{i-, i\}$, find j.

The sequence of reductions is as follows:

$$\text{R-SLWE}_{q,\Psi} \xrightarrow{\text{[LPR10, Le. 5.5]}} \mathfrak{q}_i\text{-RLWE}_{q,\Psi} \xrightarrow{\text{[LPR10, Le. 5.8]}} \text{DecRLWE}_{q,\Psi}^{i} \xrightarrow{\text{[LPR10, Le. 5.11 \& 5.13]}} \text{R-LWE}_{q,\Upsilon}$$

The oracle for M-LWE might only let us deduce the value of secret s relative to one ideal factor of (q). The first reduction allows us, if we know $s \bmod \mathfrak{q}_i R_q^\vee$ for a prime ideal $\mathfrak{q}_i$ factor of (q), to know it for all ideals factors of (q) and then we can recover s.

The second reduction allows us to pass from a decisional version to a computational one, indeed it shows that if we can distinguish between two hybrids distributions (the hybrids distributions are between $A_{s,\psi}^{(R)}$ and the uniform distribution according to i), then we can find each coordinate modulo $\mathfrak{q}_i R_q^\vee$.

The third reduction works on the noise problem, we go from a problem on a distribution over noise distributions Υ_α to a problem under the family of distributions $\Psi_{\leq\alpha}$. The principle is to show that, if we know how to distinguish for a function ϕ distributed according to Υ_α, then we know how to distinguish for any function $\psi \in \Psi_{\leq\alpha}$.

Finally, the last reduction only use the hybrid argument: if we know how to distinguish the distribution $A_{s,\psi}^{R}$ and the uniform one, then there exists an i such that we know how to distinguish between two successive hybrid distributions.

In our adaptation to modules, we will keep the general structure of this reduction. The two first intermediates reductions will be modified, while the reduction from DecRLWE to R-LWE will be kept as it also works in the case of modules.

5.2.3 Hardness of search Module-LWE

We show the hardness of the search version of M-LWE by providing a reduction from Mod-GIVP to M-SLWE. This reduction follows the same design principle as Regev's reduction from GIVP to SLWE. It makes use of the following intermediate problems, where ϕ denotes an arbitrary real-valued function on lattices and where γ is a function of the dimension, called *Module Discrete Gaussian Sampling problem* (M-DGS$_\gamma^\phi$): Given an N-dimensional module lattice M and a number $r > \gamma \cdot \phi(M)$, the goal is to output a sample from $D_{M,r}$. The reduction proceeds in two steps:

$$\text{Mod-GIVP}^{\eta_\varepsilon}_{\sqrt{8Nd}\cdot\omega(\sqrt{\log n})/\alpha} \xrightarrow{\text{Lemma 5.21}} \text{M-DGS}^{\eta_\varepsilon}_{\sqrt{2d}\cdot\omega(\sqrt{\log n})/\alpha} \xrightarrow{\text{Lemma 5.22}} \text{M-SLWE}_{q,\Psi_{\leq\alpha}}$$

The first reduction comes directly from the reduction from GIVP to DGS given by [Reg09, Le. 3.17]: It is lattice-preserving and thus also works when we consider the problems restricted to any family of lattices.

Lemma 5.21 (Adapted from [Reg09, Le. 3.17])**.** *For any $\varepsilon = \varepsilon(N) \leq 1/10$ and any γ and ϕ such that $\gamma \cdot \phi(M) \leq \sqrt{2}\eta_\varepsilon(M)$, there is a polynomial time reduction from* $\text{Mod-GIVP}^{\phi}_{2\sqrt{N}\cdot\gamma}$ *to* $\text{M-DGS}^{\phi}_{\gamma}$.

In contrast, the second one needs to be adapted.

Lemma 5.22. *Let $\varepsilon(N) = N^{-\omega(1)}$, $\alpha \in (0,1)$ and q be some integer such that $\alpha q \geq 2\sqrt{d} \cdot \omega(\sqrt{\log n})$. Assume that we have access to an oracle that solves* $\text{M-SLWE}_{q,\psi_{\leq\alpha}}$ *given a polynomial number of samples. Then there exists a polynomial time quantum algorithm for* $\text{M-DGS}^{\eta_\varepsilon}_{\sqrt{2d}\cdot\omega(\sqrt{\log n})/\alpha}$.

Proof. We use the same principle as Regev's reduction [Reg09, Th. 3.1]. We consider a module lattice M and a number $r \geq \sqrt{2d} \cdot \omega(\sqrt{\log n}) \cdot \eta_\varepsilon(M)/\alpha$. The idea is to produce samples for $D_{M,r'}$ with r' large enough, and then to use Lemma 5.23 several times to progressively decrease the value of r'. Take $r_i = r \cdot (\alpha q/\sqrt{d} \cdot \omega(\sqrt{\log n}))^i$. The first iteration starts with $r_{3N} > 2^{3N} > 2^{2N}\lambda_N(M)$ (using a LLL-reduction algorithm beforehand). Then it obtains $\text{poly}(N)$ samples of $D_{M,r_{3N}}$ using the algorithm of Theorem 1.24, and finishes with $\text{poly}(N)$ samples of $D_{M,r_{3N-1}}$ (the reduction repeats $\text{poly}(N)$ times the same iteration with the same samples in input to obtain sufficiently many different samples in output). It iterates until having $\text{poly}(N)$ samples of D_{M,r_1} with $r_1 = r \cdot \alpha q/(\sqrt{d} \cdot \omega(\sqrt{\log n})) > \sqrt{2}q \cdot \eta_\varepsilon(M)$ then it iterates a last time to obtain samples of D_{M,r_0} with $r_0 = r > \sqrt{d} \cdot \omega(\sqrt{\log n}) \cdot \eta_\varepsilon(M)/\alpha$. These samples are solutions to $\text{M-DGS}_{\sqrt{2d}\cdot\omega(\sqrt{\log n})\cdot\eta_\varepsilon(M)/\alpha}$. □

We now describe the iterative step:

Lemma 5.23. *Let $\varepsilon(N) = N^{-\omega(1)}$, $\alpha \in (0,1)$ and $q \geq 2$. Assume that we have access to an oracle that solves* $\text{M-SLWE}_{q,\Psi_{\leq\alpha}}$ *in polynomial time with non-negligible probability. Then there exists a polynomial time quantum algorithm that, given an N-dimensional module lattice M, a number $r > \sqrt{2}q \cdot \eta_\varepsilon(M)$ and $\text{poly}(N)$ samples from $D_{M,r}$, produces a sample from $D_{M,\frac{r\sqrt{d}\cdot\omega(\sqrt{\log n})}{\alpha q}}$ with non-negligible probability.*

To prove Lemma 5.23, we use the intermediate problem Mod-BDD_δ: Given a module lattice M, $\delta < \lambda_1(M)/2$ and any point $\mathbf{y} \in \mathbb{R}^n$ of the form $\mathbf{y} = \mathbf{x} + \mathbf{e}$ for some $\mathbf{x} \in M$ and $\|\mathbf{e}\|_{2,\infty} \leq \delta$, find $\mathbf{x}$. Note that we use the $\ell_{2,\infty}$ rather than the euclidean norm, as it is more convenient in Lemma 5.26.

As in [Reg09], we use another intermediate problem called q-Mod-BDD_δ: Given a module lattice M and a point $\mathbf{y} \in \mathbb{R}^n$ within distance (with respect to $\ell_{2,\infty}$ norm) δ of M, output the coset in M/qM of the closest vector to $\mathbf{y}$. The proof of Lemma 5.23 consists of a sequence of reductions (note that δ is set to $\frac{\alpha q\cdot\omega(\sqrt{\log n})}{\sqrt{2n}r}$).

$$\text{Samples from } D_{M,\frac{r\sqrt{d}\cdot\omega(\sqrt{\log n})}{\alpha q}} \xrightarrow[\text{(quantum)}]{\text{Lemma 5.24}} \text{Mod-BDD}_{M^\vee,\delta} \xrightarrow{\text{Lemma 5.25}} q\text{-Mod-BDD}_{M^\vee,\delta} \xrightarrow{\text{Lemma 5.26}} \begin{matrix}\text{M-SLWE}_{q,D_\alpha}\\ +\\ \text{Samples from } D_{M,r}\end{matrix}$$

The first reduction of Regev's proof is quantum and also lattice-preserving. It is adapted to the $\ell_{2,\infty}$ norm rather than the euclidean norm (note that $\lambda_1(M^\vee)$ is still with respect to the euclidean norm). For the adaptation, we used the fact that an N-dimensional vector sampled from D_s has $\ell_{2,\infty}$ norm at most $s\sqrt{d}\omega(\sqrt{\log n})$, except with negligible probability.

Lemma 5.24 (Adapted from [Reg09, Le. 3.14])**.** *There exists an efficient quantum algorithm that, given any N-dimensional module lattice M, a number $\delta < \lambda_1(M^\vee)\omega(\sqrt{\log n})/(2\sqrt{n})$, and an oracle that solves* Mod-BDD$_\delta$ *on $M^\vee$, outputs samples from $D_{M,\sqrt{d}\omega(\sqrt{\log n})/(\sqrt{2}\delta)}$.*

Note that by Lemma 1.29, as $r > \sqrt{2}q \cdot \eta_\varepsilon(M)$, we have that:

$$\delta = \frac{\alpha q \cdot \omega(\sqrt{\log n})}{\sqrt{2n}r} < \frac{\omega(\sqrt{\log n})}{\sqrt{n} \cdot \eta_\varepsilon(M)} < \frac{\lambda_1(M^\vee)\omega(\sqrt{\log n})}{2\sqrt{n}}.$$

The second reduction is a special case of [Reg09, Le. 3.5], which is lattice-preserving (and hence also applies to module lattices).

Lemma 5.25 ([Reg09, Le. 3.5])**.** *For any $q \geq 2$, there is a polynomial time reduction from* Mod-BDD$_\delta$ *to* q-Mod-BDD$_\delta$*.*

We will modify the last reduction, by proving the following adaptation of [Reg09, Le. 3.4]. The following lemma is the main modification of the proof of the first part of Theorem 5.17.

Lemma 5.26. *Let $\varepsilon(N) = N^{-\omega(1)}$, $\alpha \in (0,1)$ and $q \geq 2$. Let $M \subseteq R^d$ be an R-module, and $r > \sqrt{2}q \cdot \eta_\varepsilon(M)$. Given access to an oracle sampling from the distribution $D_{M,r}$, there exists a probabilistic reduction from* q-Mod-BDD$_{M^\vee, \frac{\alpha q \cdot \omega(\sqrt{\log n})}{\sqrt{2n}r}}$ *to* M-SLWE$_{q,\Psi_{\leq\alpha}}$*.*

The principle of the reduction is to construct from $\mathbf{y}$, the input of q-Mod-BDD, and from some discrete and continuous Gaussian samples, the pairs $(\mathbf{a}, b)$ distributed as $A^{(M)}_{q,\mathbf{s},\psi}$, where $\mathbf{s}$ will directly depend on the closest vector $\mathbf{x}$ to $\mathbf{y}$. To produce such samples $(\mathbf{a}, b)$ with the desired distribution, we combine the corresponding proofs for LWE and R-LWE (those of Lemmata [Reg09, Le. 3.4] and [LPR10, Le. 4.5]). Then a call to the oracle of M-SLWE returns $\mathbf{s}$ and lets us recover information on $\mathbf{x}$.

Proof of Lemma 5.26. Let $\mathcal{O}$ be the oracle which, given $m \leq \text{poly}(N)$ samples $(\mathbf{a}, b)$ from $A^{(M)}_{q,\mathbf{s},\psi}$ for $\psi \in \Psi_{\leq\alpha}$, outputs $\mathbf{s}$ in polynomial time with probability $N^{-O(1)}$. Given $M = \sum_{k=1}^d I_k \cdot \mathbf{b}_k$, the input of the reduction is $\mathbf{y} = \mathbf{x} + \mathbf{e}$ such that $\mathbf{x} \in M^\vee$ and $\|\mathbf{e}\|_{2,\infty} \leq \delta = \frac{\alpha q \cdot \omega(\sqrt{\log n})}{\sqrt{2n}r}$. The goal is to find $\mathbf{x} \bmod qM^\vee$. The reduction is as follows:

- For all $\ell \leq m$:
  - Get a fresh $\mathbf{z}_\ell$ distributed as $D_{M,r}$ and a fresh e'_ℓ distributed as $D_{\alpha/\sqrt{2}}$,
  - Let $\mathbf{a}_\ell = \Theta^{-1}(\mathbf{z}_\ell \bmod qM)$ and $b_\ell = \frac{1}{q}\langle \mathbf{z}_\ell, \mathbf{y}\rangle + e'_\ell \bmod R^\vee$ (see the definition of Θ in Section 1.2.5).
- Invoke the oracle $\mathcal{O}$ on input $\{(\mathbf{a}_\ell, b_\ell)\}_{\ell=1}^m$. If $\mathcal{O}$ succeeds, it returns some $\mathbf{s} \in (R_q^\vee)^d$.
- Output $\Theta^{-1}(\mathbf{s}) \in M^\vee/qM^\vee$.

We show that the oracle $\mathcal{O}$ is used properly, i.e., that its input follows the distribution $A^{(M)}_{q,\mathbf{s},\psi}$.

Lemma 5.27. *Let $\varepsilon > 0$ and $\mathbf{s} = \Theta(\mathbf{x} \bmod qM^\vee)$. There exists $\psi \in \Psi_{\leq\alpha}$ such that the statistical distance between $A^{(M)}_{q,\mathbf{s},\psi}$ and the distribution of $(\mathbf{a}, b)$ is at most 6ε.*

Proof. We first show that the statistical distance between $\mathbf{a}$, the first component of each sample, and the uniform distribution on R_q^d is at most 2ε. By Lemma 1.33, the statistical distance between the distribution of $\mathbf{z}$ and the uniform distribution on M_q is at most 2ε, because $r \geq q \cdot \eta_\varepsilon(M) = \eta_\varepsilon(qM)$. Then, as Θ^{-1} induces a bijection from M_q to R_q^d, the statistical distance between the distribution of $\mathbf{a} = \Theta^{-1}(\mathbf{z} \bmod qM)$ and the uniform distribution on $(R_q)^d$ is at most 2ε.

Now, we show that b is of the shape $b = \frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle + f$, where f distributed from $D_{\mathbf{r}'}$ with $r'_i \leq \alpha$ for all i. We have:

$$b = \frac{1}{q}\langle \mathbf{z}, \mathbf{y}\rangle + e' = \frac{1}{q}\langle \mathbf{z}, \mathbf{x} + \mathbf{e}\rangle + e' = \frac{1}{q}\langle \mathbf{z}, \mathbf{x}\rangle + \langle \frac{1}{q}\mathbf{z}, \mathbf{e}\rangle + e'.$$

By definition, we have $\mathbf{z} = \Theta(\mathbf{a}) = \sum_{k=1}^d (t_k \cdot a_k) \cdot \mathbf{b}_k \bmod qM$ with $t_k \in I_k$ and $a_k \in R_q$. By Lemma 1.7, we have $M^\vee = \sum_{k=1}^d I_k^\vee \cdot \mathbf{b}_k^\vee$. Let $\mathbf{x} = \sum_{k=1}^d x_k \cdot \mathbf{b}_k^\vee$. We have that $x_k \in I_k^\vee = I_k^{-1} \cdot R^\vee$ for all k. We also have $\langle \mathbf{b}_k, \mathbf{b}_{k'}^\vee\rangle = 1$ if $k = k'$ and $\langle \mathbf{b}_k, \mathbf{b}_{k'}^\vee\rangle = 0$ otherwise. Then, modulo $qR^\vee$:

$$\langle \mathbf{z}, \mathbf{x}\rangle = \sum_{k,k'=1}^d (t_k \cdot a_k) \cdot x_{k'} \cdot \langle \mathbf{b}_k, \mathbf{b}_{k'}^\vee\rangle = \sum_{k=1}^d (t_k \cdot a_k) \cdot x_k = \sum_{k=1}^d a_k \cdot (t_k \cdot x_k).$$

Because $\mathbf{s} = \Theta(\mathbf{x} \bmod qM^\vee) = (t_1 \cdot x_1 \bmod qR^\vee, \ldots, t_d \cdot x_d \bmod qR^\vee)^T$, we have:

$$\langle \mathbf{a}, \mathbf{s}\rangle = \sum_{k=1}^d a_k \cdot (t_k \cdot x_k) = \langle \mathbf{z}, \mathbf{x}\rangle \mod qR^\vee.$$

As a consequence, we obtain that $\frac{1}{q}\langle \mathbf{z}, \mathbf{x}\rangle = \frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle \bmod R^\vee$.

We now show that, conditioned on $\mathbf{a}$, the quantity $\langle \frac{1}{q}\mathbf{z}, \mathbf{e}\rangle + e'$ has distribution $D_{\mathbf{r}'}$ with $r'_i \leq \alpha$ for all i. First, let us analyse the distribution of $\mathbf{z}' = \frac{1}{q}\mathbf{z}$ knowing $\mathbf{a}$. We know that $\mathbf{z}$ has distribution $D_{M,r}$ and that $\mathbf{a} = \Theta^{-1}(\mathbf{z} \bmod qM)$. Let $\mathbf{u} = \Theta(\mathbf{a}) \bmod qM$, then the residual distribution of $\mathbf{z}' = \frac{1}{q}\mathbf{z}$ knowing $\mathbf{a}$ is $D_{M+\mathbf{u}/q, r/q}$ (with $r/q \geq \sqrt{2}\eta_\varepsilon(M)$).

We next show that e' is following the same distribution as $\langle \mathbf{e}'', \mathbf{e}\rangle$ with $\mathbf{e}'' \hookleftarrow D_{\mathbf{s},\ldots,\mathbf{s}}$, $\mathbf{s} = (s_i)_i$ and $s_i = s_{\nu-i} = \alpha/\sqrt{2\sum_{k=1}^d |\sigma_i(e_k)|^2}$ for $i \in \mathbb{J}$. By Lemma 1.45, as the vector $\mathbf{e}''$ is distributed from $D_{\mathbf{s},\ldots,\mathbf{s}}$ and $\mathbf{e} \in K^d$ is fixed, we have that $\langle \mathbf{e}'', \mathbf{e}\rangle$ has distribution $D_{\mathbf{s}'}$ with $s'_i = s'_{\nu-i} = s_i\sqrt{\sum_{k=1}^d |\sigma_i(e_k)|^2} = \alpha/\sqrt{2}$, which is exactly the distribution of e' as claimed.

We are now led to considering the distribution of $\langle \mathbf{z}' + \mathbf{e}'', \mathbf{e}\rangle$. We write $\mathbf{e}'' = \mathbf{e}''_1 + \mathbf{e}''_2$ with $\mathbf{e}''_1 \hookleftarrow D_{\alpha/(\sqrt{2}\delta)}$ and $\mathbf{e}''_2 \hookleftarrow D_{\mathbf{s}''}$ with $(s''_i)^2 = s_i^2 - \alpha^2/(2\delta^2)$ (which is positive, by the assumption on $\|\mathbf{e}\|_{2,\infty}$). As we have $\alpha/(\sqrt{2}\delta) = r/q$ and $r/q \geq \sqrt{2}\eta_\varepsilon(M)$, Lemma 1.42 gives us that the statistical distance between the distribution of $\mathbf{z}' + \mathbf{e}''_1$ and $D_{\alpha/\delta}$ is at most 4ε. As a consequence, the statistical distance between the distribution of $\mathbf{z}' + \mathbf{e}''_1 + \mathbf{e}''_2$ and $D_{\mathbf{r}'',\ldots,\mathbf{r}''}$ is at most 4ε, with

$$(r''_i)^2 = \frac{\alpha^2}{\delta^2} + (s''_i)^2 = \frac{\alpha^2}{\delta^2} + s_i^2 - \frac{\alpha^2}{2\delta^2} = \frac{\alpha^2}{2\sum_{k=1}^d |\sigma_i(e_k)|^2} + \frac{\alpha^2}{2\delta^2}.$$

By using Lemma 1.45 again with the fixed vector $\mathbf{e}$, we obtain that the statistical distance between the distribution of $\langle \mathbf{z} + \mathbf{e}'', \mathbf{e}\rangle$ and $D_{\mathbf{r}'}$ is at most 4ε, where

$$r'_i = \sqrt{\frac{\alpha^2}{2} + \frac{\alpha^2 \sum_{k=1}^d |\sigma_i(e_k)|^2}{2\delta^2}}.$$

Since $\delta \geq \sqrt{\sum_{k=1}^{d} |\sigma_i(e_k)|^2}$, we have $r'_i \leq \alpha$, as desired. □

As the input of $\mathcal{O}$ is within negligible statistical distance from $A^{(M)}_{q,\mathbf{s},\psi}$ for a distribution $\psi \in \Psi_{\leq\alpha}$ and $\mathbf{s} = \Theta(\mathbf{x} \bmod qM^\vee)$, oracle $\mathcal{O}$ succeeds with non-negligible probability. If it does succeed, then the output of our reduction is $\mathbf{x} \bmod qM^\vee$, which completes the proof of Lemma 5.26. □

This conclude the proof of the first part of Theorem 5.17.

5.2.4 Hardness of decisional Module-LWE

We now describe a reduction from the search version M-SLWE to the decision version M-LWE. The reduction of Regev from SLWE to LWE in [Reg09] does not carry over to the structured variants of LWE. We instead use the line of proof of Lyubashevsky et al. in [LPR10]. Let $q = 1 \bmod \nu$ be prime. Then $(q) = \prod_{i\in\mathbb{Z}_\nu^\times} \mathfrak{q}_i$ where any $\mathfrak{q}_i$ is a prime ideal with norm $\mathcal{N}(\mathfrak{q}_i) = q$. For $i \in \mathbb{Z}_\nu^\times$, we let $i-$ denote the largest element in $\mathbb{Z}_\nu^\times$ less than i (and we define $1-$ as 0). We define the following intermediate problems:

- $\mathfrak{q}_i$-**MLWE**$_{q,\Psi}$, with parameters Ψ a family of distributions over $\mathbb{T}_{R^\vee}$ and $i \in \mathbb{Z}_\nu^\times$: Given access to an oracle sampling from $A^{(M)}_{q,\mathbf{s},\psi}$ for some arbitrary $\mathbf{s} \in (R_q^\vee)^d$ and $\psi \in \Psi$, find $\mathbf{s} \bmod \mathfrak{q}_i R_q^\vee$.

- **Hybrid distribution** $A^{(M,i)}_{q,\mathbf{s},\psi}$, with parameters ψ a distribution over $\mathbb{T}_{R^\vee}$, $\mathbf{s} \in (R_q^\vee)^d$ and $i \in \mathbb{Z}_\nu^\times$: The distribution $A^{(M,i)}_{q,\mathbf{s},\psi}$ over $(R_q)^d \times \mathbb{T}_{R^\vee}$ is defined as follows: Choose $(\mathbf{a}, b)$ from $A^{(M)}_{q,\mathbf{s},\psi}$ and return $(\mathbf{a}, b + r/q)$ where $r \in R_q^\vee$ is uniformly random and independent in $R_q^\vee/\mathfrak{q}_j R^\vee$ for all $j \leq i$, and is 0 modulo the remaining $\mathfrak{q}_j R^\vee$'s.

- **DecMLWE**$^i_{q,\Psi}$, with parameters Ψ a family of distributions on $\mathbb{T}_{R^\vee}$ and $i \in \mathbb{Z}_\nu^\times$: Given access to an oracle sampling from $A^{(M,j)}_{q,\mathbf{s},\psi}$ for arbitrary $\mathbf{s} \in (R_q^\vee)^d$, $\psi \in \Psi$ and $j \in \{i-, i\}$, find j.

- **M-DLWE**$^i_{q,\Upsilon}$, with parameters a distribution Υ over errors distributions and $i \in \mathbb{Z}_\nu^\times$: Given access to an oracle sampling from $A^{(M,j)}_{q,\mathbf{s},\psi}$ for $\mathbf{s}$ uniform in $(R_q^\vee)^d$, ψ sampled from Υ and arbitrary $j \in \{i-, i\}$, find j.

We consider the following sequence of reductions:

$$\text{M-SLWE}_{q,\Psi} \xrightarrow{5.28} \mathfrak{q}_i\text{-MLWE}_{q,\Psi} \xrightarrow{5.30} \text{DecMLWE}^i_{q,\Psi} \xrightarrow{5.31} \text{M-DLWE}^i_{q,\Upsilon} \xrightarrow{5.33} \text{M-LWE}_{q,\Upsilon}$$

We explain the first two reductions. The following result is adapted from [LPR10].

Lemma 5.28. *For any $i \in \mathbb{Z}_\nu^\times$, there is a polynomial time reduction from* M-SLWE$_{q,\Psi_{\leq\alpha}}$ *to* $\mathfrak{q}_i$-MLWE$_{q,\Psi_{\leq\alpha}}$.

Proof. We will show below that given the $\mathfrak{q}_i$-MLWE oracle (where i is fixed), we can find the values of $\mathbf{s} \bmod \mathfrak{q}_j R^\vee$ for every $j \in \mathbb{Z}_\nu^\times$. This would complete the proof since by the Chinese Remainder Theorem, this allows us to construct $\mathbf{s} \bmod R^\vee$ and to solve M-SLWE.

We use the K-automorphisms, defined by $\tau_j(\xi) = \xi^j$ for all $j \in \mathbb{Z}_\nu^\times$. We choose $j_i \in \mathbb{Z}_\nu^\times$ such that $\tau_{j_i}(\mathfrak{q}_j) = \mathfrak{q}_i$. The reduction is as follows:

- For every sample $(\mathbf{a}, b)$, create the sample $(\mathbf{a}', b')$ with $\mathbf{a}' = (\tau_{j_i}(a_1), \ldots, \tau_{j_i}(a_d))^T$ and $b' = \tau_{j_i}(b)$.

- Use the oracle of $\mathfrak{q}_i$-MLWE with these samples, and get $\mathbf{t} \in (R^\vee/\mathfrak{q}_i R^\vee)^d$.

- Return $(\tau_{j_i}^{-1}(t_1), \ldots, \tau_{j_i}^{-1}(t_d)) \in (R^\vee/\mathfrak{q}_j R^\vee)^d$.

We show that $\tau_{j_i}^{-1}(t_k) = s_k \bmod \mathfrak{q}_j R^\vee$ for all $k \in \{1, \ldots, d\}$. By definition, we have $b = \frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle + e \bmod R^\vee$ with $\langle \mathbf{a}, \mathbf{s}\rangle = \sum_{k=1}^d a_k \cdot s_k$. As a consequence, we have:

$$b' = \tau_{j_i}(b) = \frac{1}{q}\sum_{k=1}^{d} \tau_{j_i}(a_k)\cdot \tau_{j_i}(s_k) + \tau_{j_i}(e) = \frac{1}{q}\langle \mathbf{a}', \mathbf{s}'\rangle + \tau_{j_i}(e) \bmod R^\vee,$$

with $\mathbf{s}' = (\tau_{j_i}(s_1), \ldots, \tau_{j_i}(s_d))^T$. As τ_{j_i} is an automorphism, the vector $\mathbf{a}'$ is uniformly distributed in R_q^d. Also, as $\Psi_{\leq\alpha}$ is closed under the automorphisms of K (see Lemma 5.29), we have $\psi' := \tau_{j_i}(\psi) \in \Psi_{\leq\alpha}$. Overall, the pairs $(\mathbf{a}', b')$ are distributed as $A^{(M)}_{q,\mathbf{s}',\psi'}$. If successful, the $\mathfrak{q}_i$-MLWE oracle outputs $\mathbf{t} = \mathbf{s}' \bmod \mathfrak{q}_i R^\vee = (\tau_{j_i}(s_1) \bmod \mathfrak{q}_i R^\vee, \ldots, \tau_{j_i}(s_d) \bmod \mathfrak{q}_i R^\vee)$. Then our reduction returns $(\tau_{j_i}^{-1}(t_1), \ldots, \tau_{j_i}^{-1}(t_d))^T \in (R^\vee/\mathfrak{q}_j R^\vee)^d$, which is equal to $\mathbf{s} \bmod \mathfrak{q}_j R^\vee$. □

By [LPR10, Le. 5.6], we know that $\Psi_{\leq\alpha}$ satisfies the property required by Lemma 5.28.

Lemma 5.29 ([LPR10, Le. 5.6])**.** *For any $\alpha > 0$, the family $\Psi_{\leq\alpha}$ is closed under every automorphism τ of K, i.e., $\psi \in \Psi_{\leq\alpha} \Rightarrow \tau(\psi) \in \Psi_{\leq\alpha}$.*

We now describe the next reduction.

Lemma 5.30. *Assume that $q \leq \mathrm{poly}(n)$, then for any $i \in \mathbb{Z}_\nu^\times$, there is a polynomial time reduction from $\mathfrak{q}_i$-MLWE$_{q,\Psi}$ to* DecMLWE$^i_{q,\Psi}$.

Proof. We want to find $\mathbf{s} \bmod \mathfrak{q}_i R^\vee$ from samples from $A^{(M)}_{q,\mathbf{s},\psi}$, by using an oracle that solves the DecMLWE$^i_{q,\Psi}$ problem. The principle of the proof is to find, one by one, each one of the d coordinates of $\mathbf{s} \bmod \mathfrak{q}_i R^\vee$ by using the oracle of DecMLWE$^i_{q,\Psi}$. For each coordinate, there are $\mathcal{N}(\mathfrak{q}_i) = q \leq \mathrm{poly}(n)$ possibilities. Therefore, it is possible to try them all in order to find the correct one. To check that a guess is correct, we use the same approach as in [Reg09, Le. 4.2] and randomize a coordinate of $\mathbf{a}$.

To find $s_1 \bmod \mathfrak{q}_i R^\vee$, we proceed as follows. Let $(\mathbf{a}, b)$ be distributed as $A^{(M)}_{q,\mathbf{s},\psi}$ and let $x \in R_q^\vee$; we want to know if $x = s_1 \bmod \mathfrak{q}_i R^\vee$. We construct the following pair:

$$(\mathbf{a}', b') := \left(\mathbf{a} + (y, 0, \ldots, 0), b + \frac{1}{q}(r + xy)\right),$$

where $y \in R_q$ is sampled uniformly modulo $\mathfrak{q}_i$, and is 0 modulo all the remaining $\mathfrak{q}_j$'s, and where $r \in R_q^\vee$ is uniformly random and independent modulo $\mathfrak{q}_j R^\vee$ for all $j < i$, and 0 modulo all the remaining $\mathfrak{q}_j R^\vee$'s.

Now, we show that if $x = s_1 \bmod \mathfrak{q}_i R^\vee$, then the pair $(\mathbf{a}', b')$ is distributed from $A^{(M,i-)}_{q,\mathbf{s},\psi}$ and if $x \neq s_1 \bmod \mathfrak{q}_i R^\vee$, it is distributed from $A^{(M,i)}_{q,\mathbf{s},\psi}$. First, notice that the vector $\mathbf{a}'$ is uniformly distributed in $(R_q)^d$. Now, we write b' as follows:

$$b' = b + \frac{1}{q}(r + xy) = \frac{1}{q}\left(\sum_{k=1}^{d} a_k \cdot s_k + r + xy\right) + e = \left(\frac{1}{q}\langle \mathbf{a}', \mathbf{s}\rangle + e\right) + \frac{1}{q}\left(r + y(x - s_1)\right).$$

We have two cases:

- If $x = s_1 \bmod \mathfrak{q}_i R^\vee$, then by the Chinese Remainder Theorem we have $y(x - s_1) = 0 \in R_q^\vee$. As r is chosen uniformly random and independent modulo $\mathfrak{q}_j R^\vee$ for all $j < i$, and is 0 modulo all the remaining $\mathfrak{q}_j R^\vee$'s, we obtain that the pair $(\mathbf{a}', b')$ has distribution $A^{(M,i-)}_{q,\mathbf{s},\psi}$.

- If $x \neq s_1 \bmod \mathfrak{q}_i R^\vee$, then $y(x - s_1)$ is uniformly distributed modulo $\mathfrak{q}_i R^\vee$, because $R^\vee/\mathfrak{q}_i R^\vee$ is a field (the ideal $\mathfrak{q}_i$ is prime). Also, the quantity $y(x - s_1)$ is 0 modulo the other $\mathfrak{q}_j R^\vee$'s. As a consequence, we have that $(r + y(x - s_1))$ is uniformly random and independent modulo $\mathfrak{q}_j R^\vee$ for all $j \leq i$ and is 0 modulo all the remaining $\mathfrak{q}_j R^\vee$'s. We obtain that the pair $(\mathbf{a}', b')$ is distributed as $A^{(M,i)}_{q,\mathbf{s},\psi}$.

We repeat this process d times (once for each coordinate of $\mathbf{s}$), to obtain $\mathbf{s} \bmod \mathfrak{q}_i R^\vee$. □

The last reductions carry over directly from the ring setting [LPR10, Le. 5.12 and 5.14] to the module setting (the proof randomizes the noise distribution Ψ, which is the same in the ring and module settings).

Lemma 5.31 (Adapted from [LPR10, Le. 5.12])**.** *For any $\alpha > 0$ and every $i \in \mathbb{Z}_\nu^\times$, there is a polynomial time reduction from* Dec-MLWE$^i_{q,\Psi_{\leq\alpha}}$ *to* M-DLWE$^i_{q,\Upsilon_\alpha}$.

Lemma 5.32 (Adapted from [LPR10, Le. 5.13])**.** *Let $\alpha > (1/q)\eta_\varepsilon(R^\vee)^d$ for some ε. Then for any ψ in the support of Υ_α and $\mathbf{s} \in (R^\vee)^d$, the distribution $A^{(M,\nu-1)}_{q,\mathbf{s},\psi}$ is within statistical distance $\varepsilon/2$ of the uniform distribution over $((R_q)^d, \mathbb{T}_{R^\vee})$.*

Lemma 5.33 (Adapted from [LPR10, Le. 5.14])**.** *Let Υ be a distribution over noise distributions satisfying that for any ψ in the support of Υ and any $\mathbf{s} \in (R_q^\vee)^d$, the distribution $A^{(M,\nu-1)}_{q,\mathbf{s},\psi}$ is within negligible statistical distance from uniform. Then for any oracle solving the* M-LWE$_{q,\Upsilon}$ *problem, there exists an $i \in \mathbb{Z}_\nu^\times$ and an efficient algorithm that solves the* M-DLWE$^i_{q,\Upsilon}$ *using the oracle.*

This completes the proof of Theorem 5.17 and Theorem 5.18.

5.2.5 A modulus-switching self-reduction for Module-LWE

The aim of the present section is to give the proof of Theorem 5.19: For any $p, q \geq 2$, and under some conditions on α and β, M-LWE$_{p,\Upsilon_\beta}$ is no easier than M-LWE$_{q,\Upsilon_\alpha}$. We proceed by a sequence of reductions:

$$\text{M-LWE}_{q,\Upsilon_\alpha} \xrightarrow[5.34 \text{ and } 5.36]{\text{Lemmata}} \text{HNF-MLWE}_{q,D_{\frac{1}{q}R^\vee,[\alpha,\alpha']}} \xrightarrow{\text{Lemma 5.38}} \text{M-LWE}_{p,\Psi_{\leq\beta}} \xrightarrow{\text{Lemma 5.41}} \text{M-LWE}_{p,\Upsilon_\beta}$$

In Lemmata 5.34 and 5.36, we first reduce M-LWE$_{q,\Upsilon_\alpha}$ to the HNF version (i.e., with a secret $\mathbf{s}$ of small euclidean norm) of M-LWE$_{q,D_{(1/q)R^\vee,[\alpha,\alpha']}}$, where $\alpha' \approx \alpha n^{1/4}$. Then, in Lemma 5.38 we reduce HNF-MLWE$_{q,D_{(1/q)R^\vee,[\alpha,\alpha']}}$ to M-LWE$_{p,\Psi_{\leq\beta}}$, by switching the modulus and handling the right hand sides of the M-LWE samples so that the distribution of the error term belongs to $\Psi_{\leq\beta}$. Finally, in Lemma 5.41, we re-randomize the noise distribution, thus providing a reduction from M-LWE$_{p,\Psi_{\leq\beta}}$ to M-LWE$_{p,\Upsilon_\beta}$.

Reducing M-LWE$_{q,\Upsilon_\alpha}$ to HNF-MLWE$_{q,D_{(1/q)R^\vee,[\alpha,\alpha']}}$. We first reduce M-LWE$_{q,\Upsilon_\alpha}$ to M-LWE$_{q,\Psi_{[\alpha,\alpha']}}$. We consider a sample ϕ from Υ_α: we have $\phi = D_{\mathbf{r}}$ with $r_i = r_{\nu-i} = \alpha\sqrt{1 + \sqrt{n}x_i}$ and x_i sampled from $\Gamma(2,1)$, for all $i \in \mathbb{J}$. By definition of $\Gamma(2,1)$, we have $\Pr_{x\hookleftarrow\Gamma(2,1)}[x \leq t] = 1 - (1+t)e^{-t}$, from which We derive that $x \leq \omega(\log N)$ with probability negligibly close to 1. As a consequence, with the same probability we have that $\alpha < r_i \leq \alpha' = \alpha \cdot n^{1/4}\omega(\log N)$ for all i. Therefore, M-LWE$_{q,\Psi_{[\alpha,\alpha']}}$ is no easier than M-LWE$_{q,\Upsilon_\alpha}$.

Now, for any distribution $D_{\mathbf{r}}$ arbitrarily chosen in $\Psi_{[\alpha,\alpha']}$, we discretize the noise distribution by proving that M-LWE$_{q,D_{(1/q)R^\vee,\sqrt{2}\mathbf{r}}}$ is no easier than M-LWE$_{q,D_{\mathbf{r}}}$. Here, by abuse of notation,

M-LWE$_{q,D_{(1/q)R^\vee,\sqrt{2}\mathbf{r}}}$ denotes the M-LWE problem where the distribution $\psi = D_{(1/q)R^\vee,\sqrt{2}\mathbf{r}}$ is a discrete distribution on $(1/q)R^\vee$ and where the goal is to distinguish between arbitrarily many independent samples from $A^{(M)}_{q,\mathbf{s},\psi}$ and the same number of independent samples from $U(R_q^d \times \mathbb{T}_{q,R^\vee})$, with $\mathbb{T}_{q,R^\vee} = ((1/q)R^\vee)/R^\vee$.

Lemma 5.34 (Adapted from [GKV10, Le. 2])**.** *For any $q \geq 2$, $\varepsilon \in (0,1)$, $\mathbf{r} \in (\mathbb{R}^+)^n$ with $r_{\nu-i} = r_i$ for all i, and $\alpha \in [\eta_\varepsilon(R^\vee)/q, \min_i r_i]$, there is a polynomial time reduction from* M-LWE$_{q,D_\mathbf{r}}$ *to* M-LWE$_{q,D_{(1/q)R^\vee,\sqrt{2}\mathbf{r}}}$.

The proof is following the same design as the proof of [GKV10, Le. 2].

Proof. We consider the following transformation: Given $(\mathbf{a}, b) \in R_q^d \times \mathbb{T}_{R^\vee}$, sample $f \hookleftarrow D_{(1/q)R^\vee - b,\mathbf{r}}$ and returns $(\mathbf{a}, b + f \bmod R^\vee)$.

If the sample $(\mathbf{a}, b)$ is uniform over $R_q^d \times \mathbb{T}_{R^\vee}$, then $(b + f \bmod R^\vee)$ is uniform in $\mathbb{T}_{q,R^\vee}$. Now, assume that $(\mathbf{a}, b)$ is distributed according to $A^{(M)}_{q,\mathbf{s},D_\mathbf{r}}$: We have $b = \frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle + e$, where $e \hookleftarrow D_\mathbf{r}$. Since $\frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle$ belongs to $\frac{1}{q}R^\vee$, we have $D_{(1/q)R^\vee - b,\mathbf{r}} = D_{(1/q)R^\vee - e,\mathbf{r}}$. By [Pei10, Th. 3.1], as e is sampled from $D_\mathbf{r}$ and $e' = e + f$ with f sampled from $D_{(1/q)R^\vee - e,\mathbf{r}}$, the distribution of e' is statistically close to $D_{(1/q)R^\vee,\sqrt{2}\mathbf{r}}$. We conclude that, in this case, the transformation returns a sample of $A^{(M)}_{q,\mathbf{s},D_{(1/q)R^\vee,\sqrt{2}\mathbf{r}}}$. □

Finally, Lemma 5.36 allows us to reduce the M-LWE$_{q,D_{(1/q)R^\vee,\sqrt{2}\mathbf{r}}}$ problem to a variant in which the secret is chosen from $D_{(R^\vee)^d,\sqrt{2}q\mathbf{r}}$. We call this new problem the Hermite Normal Form (HNF) of M-LWE.

Definition 5.35. Let $q \geq 2$, and Υ be a set of distributions over $(1/q)R^\vee$. The *Hermite Normal Form* of the decision version of the *Module Learning With Error problem HNF-MLWE*$_{q,\Upsilon}$ is as follows: Let ψ be arbitrarily chosen from Υ and $\mathbf{s} \in (R_q^\vee)^d$ be sampled from $(q \cdot \psi)^d$. The goal is to distinguish between arbitrarily many independent samples from $A^{(M)}_{q,\mathbf{s},\psi}$ and the same number of independent samples from $U(R_q^d \times \mathbb{T}_{q,R^\vee})$.

We have the following result:

Lemma 5.36 (Adapted from [ACPS09, Le. 2])**.** *There is a polynomial time transformation that, for arbitrary $\mathbf{s} \in (R_q^\vee)^d$ and error distribution $D_{(1/q)R^\vee,\mathbf{r}}$, maps $A^{(M)}_{q,\mathbf{s},D_{(1/q)R^\vee,\mathbf{r}}}$ to $A^{(M)}_{q,\overline{\mathbf{x}},D_{(1/q)R^\vee,\mathbf{r}}}$ with $\overline{\mathbf{x}} \hookleftarrow D_{(R^\vee)^d,q\mathbf{r}}$, and maps $U(R_q^d \times \mathbb{T}_{q,R^\vee})$ to itself.*

The proof is following the same principle as the proof of [ACPS09, Le. 2].

Proof. We are given samples from a distribution D that is either the uniform over $R_q^d \times \mathbb{T}_{q,R^\vee}$, or $A^{(M)}_{q,\mathbf{s},D_{(1/q)R^\vee,\mathbf{r}}}$.

In a first stage, we take several samples $(\mathbf{a}, b)$ from D and construct a set of d pairs $\{(\mathbf{a}_k, b_k)\}$ such that the $\mathbf{a}_k$'s are linearly independent over R_q and generate R_q^d (recall that R_q is not a field). A polynomial number of samples suffices to obtain such $\mathbf{a}_k$'s. This can be observed by considering the CRT components of $R_q \simeq (\mathbb{F}_{q^\ell})^{n/\ell}$ independently: An equivalent condition is that the n/ℓ matrices corresponding to each component are invertible over the corresponding finite field. We define $\overline{\mathbf{A}} = (\mathbf{a}_1^T, \ldots, \mathbf{a}_d^T)$ and $\overline{\mathbf{b}} = (b_1, \ldots, b_d)^T$. By construction, the map $\mathbf{y} \mapsto \overline{\mathbf{A}}\mathbf{y}$ is a bijection of R_q^d, and if $D = A^{(M)}_{q,\mathbf{s},D_{(1/q)R^\vee,\mathbf{r}}}$ then we have $\overline{\mathbf{b}} = \frac{1}{q}\left(\overline{\mathbf{A}}\mathbf{s} + \overline{\mathbf{x}}\right)$, where $\overline{\mathbf{x}}$ is sampled from $D_{(R^\vee)^d,q\mathbf{r}}$.

In a second stage, we map the fresh samples $(\mathbf{a}, b)$ from D, to samples $(\mathbf{a}', b')$ with $\mathbf{a}' = -(\overline{\mathbf{A}})^{-T} \cdot \mathbf{a} \in R_q^d$ and $b' = b + \langle \mathbf{a}', \overline{\mathbf{b}}\rangle$. As the map $\mathbf{y} \mapsto \overline{\mathbf{A}}\mathbf{y}$ is a bijection of R_q^d and as $\mathbf{a}$ is

uniform in R_q^d, we have that $\mathbf{a}'$ is uniform in R_q^d. For the right hand side b', we consider two cases:

- If D is the uniform distribution on $R_q^d \times \mathbb{T}_{q,R^\vee}$, then $(\mathbf{a}', b')$ is also uniform on $(R_q)^d \times \mathbb{T}_{q,R^\vee}$.
- If D is $A^{(M)}_{q,\mathbf{s},D_{(1/q)R^\vee,\mathbf{r}}}$, then $b' = \frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle + e - \frac{1}{q}\langle (\overline{\mathbf{A}})^{-T}\mathbf{a}, \overline{\mathbf{A}}\mathbf{s}\rangle + \frac{1}{q}\langle \mathbf{a}', \overline{\mathbf{x}}\rangle = \frac{1}{q}\langle \mathbf{a}', \overline{\mathbf{x}}\rangle + e$. As a consequence, the pair $(\mathbf{a}', b')$ is distributed as $A^{(q)}_{\overline{\mathbf{x}},D_{(1/q)R^\vee,\mathbf{r}}}$, with $\overline{\mathbf{x}}$ sampled from $D_{(R^\vee)^d,q\mathbf{r}}$.

□

This completes the reduction from $\text{M-LWE}_{q,\Upsilon_\alpha}$ to $\text{HNF-MLWE}_{q,D_{(1/q)R^\vee,[\alpha,\alpha']}}$.

Reducing $\text{HNF-MLWE}_{q,D_{(1/q)R^\vee,[\alpha,\alpha']}}$ to $\text{M-LWE}_{p,\Psi_{\leq\beta}}$. This is the main component of the proof of Theorem 5.19. In Lemma 5.37, we first give a bound on $\|\mathbf{s}\|_{2,\infty}$.

Lemma 5.37. *Let $\varepsilon = N^{-\omega(1)}$, $\alpha' > \alpha > 0$ and q an integer such that $\alpha q \geq \eta_\varepsilon(R^\vee)$. Let $\mathbf{r} \in (\mathbb{R}^+)^n$ with $r_i \in [\alpha, \alpha']$ for all i. If $\mathbf{s}$ is sampled from $D_{(1/q)R^\vee,\mathbf{r}}$, then $\|\mathbf{s}\|_{2,\infty} \leq \alpha' q \cdot \sqrt{d} \cdot \omega(\sqrt{\log N})$ with probability $\geq 1 - \varepsilon$.*

Proof. First, we know that $\|\mathbf{s}\|_{2,\infty} \leq \sqrt{d}\|\mathbf{s}\|_\infty$. Let $\varepsilon = N^{-\omega(1)}$, by assumption, we have that $\alpha q \geq \eta_\varepsilon(R^\vee)$. By Lemma 1.38 we know that $\|\mathbf{s}\|_\infty \leq \alpha' q \cdot \omega(\sqrt{\log N})$ with probability $\geq 1 - \varepsilon$. The result follows. □

In the following lemma, we transform a sample from $A^{(M)}_{q,\mathbf{s},D_{(1/q)R^\vee,[\alpha,\alpha']}}$ to a sample of $A^{(M)}_{p,\mathbf{s},\Psi_{\leq\beta}}$, assuming that $\|\mathbf{s}\|_{2,\infty}$ is bounded.

Lemma 5.38 (Adapted from Lemma 4.7)**.** *Let $\varepsilon = N^{-\omega(1)}$, $p, q > 2$, $\alpha, \alpha' \in (0, 1)$, and $s_{\max} > 0$. There is an efficient mapping from $R_q^d \times \mathbb{T}_{q,R^\vee}$ to $R_p^d \times \mathbb{T}_{p,R^\vee}$ which has the following properties:*

- *If the input is uniformly random, then the output is within negligible statistical distance from the uniform distribution.*
- *If the input is distributed from $A^{(M)}_{q,\mathbf{s},D_{(1/q)R^\vee,[\alpha,\alpha']}}$, where $\mathbf{s} \in (R^\vee)^d$ with $\|\mathbf{s}\|_{2,\infty} \leq s_{\max}$, then the output distribution is within negligible statistical distance from $A^{(M)}_{p,\mathbf{s}',\Psi_{\leq\beta}}$, where $\mathbf{s}'$ is uniform in $(R_p^\vee)^d$ and*

$$\beta^2 \geq 2\left(\alpha' + \omega\left(\frac{p+q}{pq} s_{\max} \cdot \eta_\varepsilon(R^d)\right)\right).$$

Proof. The principle of this reduction is to first map $\mathbf{a} \in R_q^d$ to $\mathbf{a}' \in R_p^d$ by scaling it by p/q, and then randomly rounding it (using a discrete Gaussian distribution). Note that simply multiplying by p/q cannot work as, for example, the caridnality of R_p^d may not divide the cardinality of R_q^d. Then, we study the new error term, modified with the Gaussian rounding, and show that it is still a Gaussian error.

We sample $\mathbf{s}_1$ uniformly in $(R_p^\vee)^d$. On input $(\mathbf{a}, b) \in R_q^d \times \mathbb{T}_{q,R^\vee}$, the mapping is as follows:

- Set $\sigma = \omega\left((1 + p/q)\eta_\varepsilon(R^d)\right)$,
- Sample $\mathbf{d}$ from $D_{R^d - \frac{p}{q}\mathbf{a},\sigma}$ and compute $\mathbf{a}' = \frac{p}{q}\mathbf{a} + \mathbf{d}$,
- Sample e_d from $D_{\sigma \cdot s_{\max}}$ and e' from $D_{\alpha'}$,
- Compute $b' = b + \frac{1}{p}\langle \mathbf{a}', \mathbf{s}_1\rangle + \frac{1}{p}e_d + e'$,
- Return the new sample $(\mathbf{a}', b')$.

The choice of σ is derived from the proof of correctness of the reduction (see Lemmata 5.39 and 5.40 below).

We first show that the second step of the mapping transforms the uniform distribution in R_q^d to the uniform distribution in R_p^d, by considering the joint distribution of the pair $(\mathbf{a}', \mathbf{d})$. The following result can be interpreted as a simple particular case of [Pei10, Se. 3].

Lemma 5.39. *Let $\varepsilon = N^{-\omega(1)}$ and assume that $\sigma \geq \omega((1+\frac{p}{q})\eta_\varepsilon(R^d))$. Then the residual distribution of $\mathbf{a}'$ is within negligible statistical distance to $U(R_p^d)$, and, for any $\overline{\mathbf{a}'} \in R_p^d$, the distribution of $\mathbf{d}$ conditioned on $\mathbf{a}' = \overline{\mathbf{a}'}$ is within negligible statistical distance to $D_{\frac{p}{q}R^d + \overline{\mathbf{a}'}, \sigma}$.*

Proof. Let $\overline{\mathbf{a}'} \in R_p^d$. Since $\mathbf{d} = \mathbf{a}' - \frac{p}{q}\mathbf{a} + p\mathbf{k}$ for some $\mathbf{k} \in R^d$ and $\mathbf{a} \in R_q^d$, we have that $\mathbf{d} - \mathbf{a}' \in \frac{p}{q}R^d$. Let $\overline{\mathbf{d}} \in \frac{p}{q}R^d + \overline{\mathbf{a}'}$. By construction, we have:

$$\Pr\left[\mathbf{a}' = \overline{\mathbf{a}'} \ \wedge \ \mathbf{d} = \overline{\mathbf{d}}\right] = \Pr\left[\mathbf{a} = \frac{q}{p}(\overline{\mathbf{a}'} - \overline{\mathbf{d}}) \ \wedge \ \mathbf{d} = \overline{\mathbf{d}}\right] = \frac{\rho_\sigma(\overline{\mathbf{d}})}{q^n \cdot \rho_\sigma(R^d - \overline{\mathbf{a}'} + \overline{\mathbf{d}})}.$$

In the latter, the denominator is within a factor $1 \pm \varepsilon$ from $q^n \cdot \rho_\sigma(R^d)$, because $\sigma \geq \eta_\varepsilon(R^d)$.

We now consider the residual distribution of $\mathbf{a}'$.

$$\begin{aligned}
\Pr\left[\mathbf{a}' = \overline{\mathbf{a}'}\right] &= \sum_{\overline{\mathbf{d}} \in \frac{p}{q}R^d + \overline{\mathbf{a}'}} \Pr\left[\mathbf{a}' = \overline{\mathbf{a}'} \ \wedge \ \mathbf{d} = \overline{\mathbf{d}}\right] \\
&\in \frac{\rho_\sigma(\frac{p}{q}R^d + \overline{\mathbf{a}'})}{q^n \cdot \rho_\sigma(R^d)} \cdot [1-\varepsilon, 1+\varepsilon] \\
&\subseteq \frac{1}{p^n} \cdot [1-\varepsilon, 1+\varepsilon],
\end{aligned}$$

because $\sigma \geq \eta_\varepsilon(\frac{p}{q}R^d)$.

Finally, we obtain that $\Pr[\mathbf{d} = \overline{\mathbf{d}} | \mathbf{a}' = \overline{\mathbf{a}'}]$ is within a factor $1 \pm \varepsilon$ from a quantity that is proportional to $\rho_\sigma(\overline{\mathbf{d}})$. This completes the proof of the claim. □

We now study the right hand size of the LWE sample. Assume that $b \in \mathbb{T}_{q,R^\vee}$ is of the form $\frac{1}{q}\langle \mathbf{a}, \mathbf{s}\rangle + e_q$ with $e_q \hookleftarrow D_{(1/q)R^\vee, [\alpha, \alpha']}$. Then we can write:

$$(\mathbf{a}', b') = \left(\mathbf{a}', \frac{1}{p}\langle \mathbf{a}', \mathbf{s} + \mathbf{s}'\rangle + \frac{1}{p}(\langle \mathbf{d}, \mathbf{s}\rangle + e_d) + e_q + e'\right). \tag{5.1}$$

The new error e_p is equal to $\frac{1}{p}(\langle \mathbf{d}, \mathbf{s}\rangle + e_d) + e_q + e'$. To study this new error, we first study the distribution of $\langle \mathbf{d}, \mathbf{s}\rangle + e_d$ conditioned on $\mathbf{a}'$ in Lemma 5.40 (which generalizes [Reg05, Co. 3.10] to the module case).

Lemma 5.40. *Let $s_{\max} > 0$ and $\mathbf{s} \in K^d$ with $\|\mathbf{s}\|_{2,\infty} < s_{\max}$. Let $\mathbf{d}$ be distributed as $D_{(p/q)R^d - \mathbf{a}, \sigma}$ for some arbitrary $\mathbf{a}$ and $\sigma \geq \sqrt{2}(p/q)\eta_\varepsilon(R^d)$ and e be distributed as D_τ for some $\tau \geq \sigma \cdot s_{\max}$. Then the distribution of $\langle \mathbf{d}, \mathbf{s}\rangle + e$ is within negligible statistical distance of the elliptical Gaussian distribution $D_{\mathbf{t}}$ over K, where $t_i^2 = t_{\nu-i}^2 = \sigma^2 \sum_{k=1}^d |\sigma_i(s_k)|^2 + \tau^2$ for all i.*

Proof. By Lemma 1.45, we have that e is following the same distribution as $\langle \mathbf{e}_s, \mathbf{s}\rangle$ with $\mathbf{e}_s$ distributed from $D_{\mathbf{r}', \ldots, \mathbf{r}'}$ and $r_i' = r_{\nu-i}' = \tau / \sqrt{\sum_{k=1}^d |\sigma_i(s_k)|^2}$ for i.

As a consequence, we have that $\langle \mathbf{d}, \mathbf{s}\rangle + e$ is following the same distribution as $\langle \mathbf{d} + \mathbf{e}_s, \mathbf{s}\rangle$. We write $\mathbf{e}_s = \mathbf{e}_1 + \mathbf{e}_2$ with $\mathbf{e}_1$ distributed from $D_{\tau/S}$ and $\mathbf{e}_2$ distributed from $D_{(\sqrt{(r_i')^2 - (\tau/s_{\max})^2})_i}$.

We now use Lemma 1.42: As $\sigma \geq \sqrt{2}(p/q)\eta_\varepsilon(R^d)$ and $\tau \geq s_{\max} \cdot \sigma$, we have that $\mathbf{d} + \mathbf{e}_1$ is within statistical distance 4ε from $D_{\sqrt{\sigma^2+(\tau/s_{\max})^2}}$. The quantity $\mathbf{d} + \mathbf{e}_s$ can be interpreted as the sum of two continuous Gaussians: It is within statistical distance 4ε from $D_{(\sqrt{\sigma^2+(r'_i)^2})_i}$.

We use Lemma 1.45 once more. We obtain that $\langle \mathbf{d}, \mathbf{s} \rangle + e$ is within statistical distance 4ε from $D_{\mathbf{t}}$ with $t_i^2 = t_{\nu-i}^2 = \sigma^2 \sum_{k=1}^d |\sigma_i(s_k)|^2 + \tau^2$, for all i. □

Let $(\mathbf{a}, b)$ be sampled from $A^{(M)}_{q,\mathbf{s},D_{(1/q)R^\vee,[\alpha,\alpha']}}$ and let $(\mathbf{a}', b')$ be the image of $(\mathbf{a}, b)$ by the mapping. To conclude the proof, we show that $(\mathbf{a}', b')$ is sampled from $A^{(M)}_{p,\mathbf{s}',\Psi_{\leq\beta}}$:

- We recall that $b' = \frac{1}{p}\langle \mathbf{a}', \mathbf{s} + \mathbf{s}_1 \rangle + e_p$.
- We showed that $\mathbf{a}'$ is within negligible statistical distance from the uniform distribution in R_p^d.
- We have that $\mathbf{s}' = \mathbf{s} + \mathbf{s}_1$, where $\mathbf{s}_1$ is uniform in $(R_p^\vee)^d$ and independent from $\mathbf{s}$. This ensures that $\mathbf{s}'$ mod p is uniform in $(R_p^\vee)^d$.
- We now consider $e_p = \frac{1}{p}(\langle \mathbf{d}, \mathbf{s} \rangle + e_d) + e + e'$, where:
  - The component $\frac{1}{p}(\langle \mathbf{d}, \mathbf{s} \rangle + e_d)$ it is within negligible statistical distance from $D_{\mathbf{t}}$ with $t_i^2 = t_{\nu-i}^2 = \frac{1}{p}\sigma^2 \left(\sum_{k=1}^d |\sigma_i(s_k)|^2 + s_{\max}^2 \right)$ by applying Lemma 5.40.
  - The component $e + e'$ is within negligible statistical distance from $D_{\mathbf{t}'}$ with $(t'_i)^2 = (t'_{\nu-i})^2 = r_i^2 + (\alpha')^2$ by Lemma 1.42 and as, for all i, $\alpha' q \geq r_i q > \alpha q \geq \sqrt{2}\eta_\varepsilon(R^\vee)$.

  Then, the error component e_p is within negligible statistical distance from $D_{\mathbf{t}''}$ with $(t''_i)^2 = (t''_{\nu-i})^2 = r_i^2 + (\alpha')^2 + \frac{\sigma^2}{p^2}(\sum_{k=1}^d |\sigma_i(s_k)|^2 + s_{\max}^2)$. As $r_i \leq \alpha'$ holds for all i, and as $\|\mathbf{s}\|_{2,\infty} \leq s_{\max}$, we have:

$$t''_i = t''_{\nu-i} \leq \sqrt{2} \cdot \sqrt{(\alpha')^2 + \frac{\sigma^2}{p^2} s_{\max}^2} \leq \beta, \quad \text{for all } i.$$

□

Reducing M-LWE$_{p,\Psi_{\leq\beta}}$ to M-LWE$_{p,\Upsilon_\beta}$. This is the last component of the proof of Theorem 5.19. The goal is to re-randomize the error distribution of M-LWE. The proof is adapted from [LPR10, Le. 5.11].

Lemma 5.41. *Let $p \geq 2$ and $\beta \in (0, 1)$. There is a polynomial time reduction from* M-LWE$_{p,\Psi_{\leq\beta}}$ *to* M-LWE$_{p,\Upsilon_\beta}$.

Proof. Let $(\mathbf{a}, b = \frac{1}{p}\langle \mathbf{a}, \mathbf{s} \rangle + e)$ be a sample from $A^{(M)}_{p,\mathbf{s},D_{\mathbf{t}}}$ with $0 < t_i \leq \beta$ and $t_{\nu-i} = t_i$ for all i, and $\mathbf{s} \hookleftarrow U((R_q^\vee)^d)$. Let $(x'_i)_i \in \mathbb{J}$ be independent samples from $\Gamma(2, 1)$. We perform the following transformation:

$$(\mathbf{a}', b') := (\mathbf{a}, b + e'),$$

where e' is sampled from $D_{\mathbf{r}}$, with $\mathbf{r}$ defined by $r_i^2 = r_{\nu-i}^2 = \beta^2\sqrt{n}x'_i$ for all i.

This transformation maps the uniform distribution over $R_p^d \times \mathbb{T}_{R^\vee}$ to itself. On the other hand, it maps $A^{(M)}_{p,\mathbf{s},D_{\mathbf{t}}}$ to $A^{(M)}_{p,\mathbf{s},D_{\mathbf{r}'}}$, with $r'_i = r'_{\nu-i} = \sqrt{t_i^2 + \beta^2\sqrt{n}x'_i}$, for all $i \in \mathbb{J}$.

Let S denote the set of ψ's for which the oracle distinguishes with non-negligible probability between the uniform distribution over $R_p^d \times \mathbb{T}_{p,R^\vee}$ and the distribution $A^{(M)}_{p,\mathbf{s},\psi}$. By assumption, the measure of S under Υ_β is non-negligible. Lemma 1.21 implies that $D_{\mathbf{r}'} \in S$ with non-negligible probability. The result follows. □

5.3 Converse reduction

5.3.1 From Module-SIS to Mod-GIVP

We restrict the analysis to cyclotomic polynomials of the form $x^n + 1$ with n a power of 2, for the sake of simplicity. We expect the result to carry over to all cyclotomic polynomials, but this would add technical complications in the proof of Lemma 5.42. Choosing $x^n + 1$ implies that $R^\vee = \frac{1}{n}R$.

Let $\mathbf{a}_1, \ldots, \mathbf{a}_m$ be sampled uniformly and independently in R_q^d. Finding $\mathbf{z} = (z_1, \ldots, z_m)^T \in R^m \setminus \{\mathbf{0}\}$ such that $\sum_i z_i \mathbf{a}_i = 0 \bmod q$ and $\|\mathbf{z}\| \leq \beta$ corresponds to finding a short vector in the lattice:

$$\mathbf{A}^\perp = \left\{ \mathbf{y} \in R^m : \mathbf{A}^T \mathbf{y} = \mathbf{0} \bmod q \right\},$$

where $\mathbf{A} \in R_q^{d \times m}$ is the matrix whose rows are the $\mathbf{a}_i$'s. As this lattice is a module lattice, if we solve Mod-GIVP$_\gamma^{\eta_\varepsilon}$ given as input an arbitrary basis of $\mathbf{A}^\perp$ (which can be computed efficiently given $\mathbf{A}$), then we obtain a solution to the M-SIS instance, for $\beta = \gamma \cdot \eta_\varepsilon(\mathbf{A}^\perp)$. To assess the effectiveness of this reduction from M-SIS to Mod-GIVP, we are thus led to estimating $\eta_\varepsilon(\mathbf{A}^\perp)$ for $\mathbf{A}$ sampled uniformly in $R_q^{m \times d}$. For this task, it is classical to study the dual lattice, as we have $\eta_\varepsilon(\Lambda) \leq \sqrt{\frac{\ln(2N(1+1/\varepsilon))}{\pi}} / \lambda_1^\infty(\Lambda^*)$ for any N-dimensional lattice Λ (see Lemma 1.29). The dual of the lattice $\mathbf{A}^\perp$ is $\frac{1}{q} L_q(\mathbf{A})$ where

$$L_q(\mathbf{A}) = \left\{ \mathbf{y} \in (R^\vee)^m : \exists \mathbf{s} \in (R_q^\vee)^d, \mathbf{B}\mathbf{s} = \mathbf{y} \bmod q \right\}.$$

Hence, it suffices to obtain a probabilistic lower bound on $\lambda_1^\infty(L_q(\mathbf{A}))$, for $\mathbf{A}$ uniform in $R_q^{m \times d}$.

Similarly, for reducing M-SIS to M-SIVP, one is led to bounding $\lambda_{mn}(\mathbf{A}^\perp)$. As $\lambda_N(\Lambda) \leq N/\lambda_1(\Lambda) \leq N^{3/2}/\lambda_1^\infty(\Lambda^*)$ for any N-dimensional Λ, it is also sufficient to obtain a lower bound for $\lambda_1^\infty(\Lambda^*)$.

Lemma 5.42. *Let n, m, d, q be positive integers with $d \leq m$ and n a power of 2. We have:*

$$\Pr_{A \hookleftarrow U(R_q^{m \times d})} \left[\lambda_1^{\infty,2}(L_q(\boldsymbol{A})) \geq \frac{1}{8\sqrt{n}} q^{1 - \frac{d}{m}} \right] \geq 1 - 2^{-n},$$

where $\lambda_1^{\infty,2}(\cdot)$ refers to the lattice minimum with respect to $\|\cdot\|_{\infty,2}$.

Proof. We generalize and adapt the proof of [SS11, Le. 8] (see also [SS13, Le. 3.2]. By the union bound, the probability that $L_q(\mathbf{A})$ contains a nonzero vector of infinity norm $\leq B := \frac{1}{8\sqrt{n}} q^{1 - \frac{d}{m}}$ is bounded from above by:

$$\sum_{\substack{\mathbf{t} \in (R_q^\vee)^m \\ 0 < \|\mathbf{t}\|_{\infty,2} \leq B}} \sum_{\mathbf{s} \in (R_q^\vee)^d} \Pr_{\mathbf{A} \hookleftarrow U(R_q^{m \times d})} [\mathbf{A}\mathbf{s} = \mathbf{t}] = \sum_{\substack{\mathbf{t} \in (R_q^\vee)^m \\ 0 < \|\mathbf{t}\|_{\infty,2} \leq B}} \sum_{\mathbf{s} \in (R_q^\vee)^d} \prod_{i \leq m} \Pr_{\mathbf{a} \hookleftarrow U(R_q^d)} [\mathbf{a}^T \mathbf{s} = t_i].$$

We now consider the probability (over the randomness of $\mathbf{a}$) that $\mathbf{a}^T \mathbf{s} = t_i$. For this purpose, we consider the decomposition of R_q as a Cartesian product of finite fields. By the Chinese Remainder Theorem, we know that $R_q \simeq R_q^\vee \simeq \mathbb{F}_{q^\delta} \times \ldots \times \mathbb{F}_{q^\delta}$ for some integer δ dividing n (there are n/δ copies of the finite field of q^δ elements). Now, the equality $\mathbf{a}^T \mathbf{s} = t_i$ holds if and only if it holds over all n/δ CRT components. Wlog we consider the first one. If t_i and all the coordinates of $\mathbf{s}$ are zero, then the probability is 1. Otherwise, if t_i or some coordinate of $\mathbf{s}$ is

nonzero on that first CRT component, then the probability is $\leq q^{-\delta}$. As a consequence, the probability under scope is bounded from above by:

$$\sum_{S\subseteq[n/\delta]} \sum_{\substack{\mathbf{s}\in(R_q^\vee)^d \\ \forall i,\, s_i \text{ is 0 on } S}} \sum_{\substack{\mathbf{t}\in(R_q^\vee)^m \\ 0<\|\mathbf{t}\|_{\infty,2}\leq B \\ \forall i,\, t_i \text{ is 0 on } S}} q^{m(|S|\delta-n)} \;\leq\; \sum_{S\subseteq[n/\delta]} \sum_{\substack{\mathbf{t}\in(R_q^\vee)^m \\ 0<\|\mathbf{t}\|_{\infty,2}\leq B \\ \forall i,\, t_i \text{ is 0 on } S}} q^{(m-d)(|S|\delta-n)}.$$

We now attempt to bound the number of t's in $R^\vee$ such that $0<\|t\|\leq B$ and t is 0 on all CRT components corresponding to S. As $R^\vee=\frac{1}{n}R$, it suffices to bound the number of t's in R such that $0<\|t\|\leq nB$ and t is 0 on all CRT components corresponding to S.

The latter condition implies that t is a nonzero element of an ideal I of R of algebraic norm $q^{|S|\delta}$. Let $x\in I$ reaching $\lambda_1(I)$. By the arithmetic-geometric inequality, we have:

$$\lambda_1(I)=\|x\|\geq\sqrt{n}\mathcal{N}(x)^{1/n}=\sqrt{n}\mathcal{N}((x))^{1/n}\geq\sqrt{n}\mathcal{N}(I)^{1/n}=\sqrt{n}q^{|S|\delta/n}.$$

As a result, there is no such t when $|S|\geq(1-d/m)n/\delta$. If $|S|\leq(1-d/m)n/\delta$, then we are looking for the number of points of the (ideal) lattice I in the hyperball of radius nB and center 0. All such points are away from one another by at least $\lambda_1(I)$. Therefore, by the pigeon-hole principle, there are at most $(2nB/\lambda_1(I))^n\leq 4^{-n}q^{n-\frac{nd}{m}-|S|\delta}$ such points.

Now, the probability under scope can be bounded from above as

$$4^{-n}\cdot\sum_{S\subseteq[(1-d/m)n/\delta]} q^{(m-d)(|S|\delta-n)}\cdot q^{mn-nd-m|S|\delta} \;\leq\; 2^{-n}.$$

This completes the proof of the lemma. □

As a consequence of the result above and the preceding discussion, we obtain the following converse to Theorem 5.10. Note that even for $d=1$ (i.e., for an R-SIS instance), the resulting Mod-GIVP instance has module rank m: This result does not provide a reduction from R-SIS to Id-GIVP (the module rank in Mod-GIVP is m, which is possibly much larger than d).

Theorem 5.43. *For any $d\geq 1$ and $\varepsilon(N)=N^{-\omega(1)}$, there is a polynomial time reduction from solving* M-SIS$_{q,m,\beta}$ *to solving* Mod-GIVP$^{\eta_\varepsilon}_{\gamma}$ *(with module rank m), for any $m(N), q(N)$, $\beta(N)$ and $\gamma(N)$ such that $\beta\geq\gamma\sqrt{N}\omega\left(\sqrt{\log(N/\varepsilon)}\right)\cdot q^{\frac{d}{m}}$ and $m,\log q\leq\mathrm{poly}(N)$.*

5.3.2 From Module-LWE to Mod-GIVP

One of the classical ways for solving LWE consists in solving an associated SIS instance [MR09]. We propose an adaptation of this approach to module lattices: We reduce M-LWE to M-SIS and then combine this reduction with Theorem 5.43.

Let us sample $\mathbf{s}$ uniformly in $(R_q^\vee)^d$, and ψ from Υ_α. More precisely, we sample x_i from $\Gamma(2,1)$ for $i\in\mathbb{J}$, define $r_i=r_{\nu-i}=\alpha\sqrt{1+\sqrt{n}x_i}$, and let $\psi=D_{\mathbf{r}}$. Assume that we have access to arbitrarily many samples $(\mathbf{a}_i,b_i)\in R_q^d\times\mathbb{T}_{R^\vee}$ with $\mathbf{a}_i$ uniform in R_q^d and all the b_i's uniform and independent in $\mathbb{T}_{R^\vee}$, or all the b_i's of the form $b_i=\frac{1}{q}\langle\mathbf{a}_i,\mathbf{s}\rangle+e_i$ with the e_i's are sampled from ψ. Our goal is to determine with noticeable advantage which situation we are in.

We consider m such samples (with m to be optimized later). Let $\mathbf{A}\in R_q^{m\times d}$ be the matrix whose rows are the $\mathbf{a}_i$'s. By solving M-SIS$_{q,m,\beta}$ for $\mathbf{A}^T$, we obtain a nonzero vector $\mathbf{z}\in R^m$ such that $\|\mathbf{z}\|\leq\beta$ and $\mathbf{z}^t\cdot\mathbf{A}=\mathbf{0}\bmod q$. Now, we compute $\langle\mathbf{z},\mathbf{b}\rangle$, where $\mathbf{b}\in\mathbb{T}_{R^\vee}^m$ is the vector made of the b_i's. If the b_i's are uniform independent of the $\mathbf{a}_i$'s, then the inner product $\langle\mathbf{z},\mathbf{b}\rangle$ is uniformly distributed in $\mathbb{T}_{R^\vee}$. Otherwise, we have $\langle\mathbf{z},\mathbf{b}\rangle=\langle\mathbf{z},\mathbf{e}\rangle$ (modulo $R^\vee$), where $\mathbf{e}$ is the vector made of

the e_i's. By Lemma 1.45, we have that $\langle \mathbf{z}, \mathbf{e} \rangle$ is distributed as $D_{\mathbf{r}'}$ with $r'_j = r_j \cdot \sqrt{\sum_{k \leq m} |\sigma_j(z_k)|^2}$ for all $j \in \mathbb{Z}_\nu^\times$. As a consequence, we have

$$\begin{aligned} \|\langle \mathbf{z}, \mathbf{b} \rangle\| &\leq t\sqrt{n} \cdot \max_j |r'_j| \\ &\leq t\sqrt{n} \cdot \|\mathbf{z}\| \cdot \max_j |r_j| \leq 2tn^{3/4}\alpha\beta \cdot \max_j |x_j|, \end{aligned}$$

with probability $\geq 1 - 2^{-\Omega(nt^2)}$ over the randomness of the e_i's. Furthermore, as we have $|x_j| \leq t$ with probability $\geq 1 - (2+t)e^{-t}$ for all j, we obtain that the bound above is itself smaller than $2t^2 n^{3/4}\alpha\beta$ with probability $\geq 1 - nt2^{-\Omega(t)}$. As $R^\vee = \frac{1}{n}R$, if the latter upper bound is smaller than $\frac{1}{4n}$, then $\langle \mathbf{z}, \mathbf{b} \rangle$ will be unexpectedly small.

Overall, we have proved that if β is such that $n^{7/4}\omega(\log(N)) \cdot \alpha\beta < 1$, then we can distinguish between the two challenge distributions with non-negligible advantage. By Theorem 5.43, we thus obtain a reduction from Mod-GIVP$_\gamma^{\eta_\varepsilon}$ with module rank m to M-LWE$_{q,\Upsilon_\alpha}$, if γ is such that $\alpha\gamma n^{7/4}\sqrt{N}\omega(\sqrt{\log(N/\varepsilon)})q^{\frac{d}{m}} < 1$. Taking $m = d \log q$ leads to the following result.

Theorem 5.44. *For any $d \geq 1$ and $\varepsilon(N) = N^{-\omega(1)}$, there is a probabilistic polynomial time reduction from solving* M-LWE$_{q,\Upsilon_\alpha}$ *over R_q^d to solving* Mod-GIVP$_\gamma^{\eta_\varepsilon}$ *(with module rank $d \log q$), for any $\alpha(N)$ and $\gamma(N)$ such that $\frac{1}{\alpha} \geq \gamma N^{3/2}\omega(\sqrt{\log(N/\varepsilon)})$ and $\log q \leq \mathrm{poly}(N)$.*

Part Three

Cryptographic Constructions: Group Signature

Group signatures are cryptographic primitives where users can anonymously sign messages in the name of a population they belong to. Gordon *et al.* [GKV10] suggested the first realization of group signatures based on lattice assumptions in the random oracle model. A significant drawback of their scheme is its linear signature size in the cardinality N of the group. A recent extension proposed by Camenisch *et al.* [CNR12] suffers from the same overhead. In Chapter 7, we describe the first lattice-based group signature schemes where the signature and public key sizes are essentially logarithmic in N (for any fixed security level). Our basic construction only satisfies a relaxed definition of anonymity (just like the Gordon *et* al. system) but readily extends into a fully anonymous group signature (i.e., that resists adversaries equipped with a signature opening oracle). We prove the security of our schemes in the random oracle model under the SIS and LWE assumptions.

Support of membership revocation is a desirable functionality for any group signature scheme. Among the known revocation approaches, verifier-local revocation (VLR) seems to be the most flexible one, because it only requires the verifiers to possess some up-to-date revocation information, but not the signers. All of the contemporary VLR group signatures [BS04, NF05, NF06, LV09, BCN+10] operate in the bilinear map setting. In Chapter 8, we introduce the first lattice-based VLR group signature. This scheme has the same logarithmic-size signatures as in Chapter 7, it supports membership revocation, but has weaker security assumption and resists weaker attackers.

Chapter 6

Group Signatures

Group signatures are a core cryptographic primitive that paradoxically combines the properties of authenticity and anonymity. They are useful in many real-life applications including trusted computing platforms, auction protocols or privacy-protecting mechanisms for users in public transportation.

Parties involved in such a system are a special entity, called the group manager, and group members. The manager holds a master secret key, generates a system-wide public key, and administers the group members, by providing to each of them an individual secret key that will allow them to anonymously sign on behalf of the group. In case of dispute, the manager (or a separate authority) is able to determine the identity of a signer via an opening operation. This fundamental primitive has been extensively studied, from both theoretical and practical perspectives: It has been enriched with many useful properties, and it has been implemented in the contexts of trusted computing (using privacy-preserving attestation [Bri03]) and of traffic management (e.g., the Vehicle Safety Communications project of the U.S. Dept. of Transportation [IPWG03]).

Group signatures were originally proposed by Chaum and van Heyst [CvH91] and made scalable by Ateniese *et al.* in [ACJT00]. Proper security models were introduced in [BMW03] and [BSZ05, KY06] (for dynamic groups), whereas more intricate and redundant properties were considered hitherto. The model of Bellare *et al.* [BMW03] requires two main security properties called *full anonymity* and *full traceability*. The former notion means that signatures do not leak the identities of their originators, whereas the latter implies that no collusion of malicious users can produce a valid signature that cannot be traced to one of them. Bellare *et al.* [BMW03] proved that trapdoor permutations suffice to design group signatures, but their theoretical construction was mostly a proof of concept. Nevertheless, their methodology has been adapted in practical constructions: Essentially, a group member signs a message by verifiably encrypting a valid membership certificate delivered by the authority, while hiding its identity. While numerous schemes (e.g., [ACJT00, CL02, CL04, BBS04]) rely on the random oracle model (ROM), others are proved secure in the standard model (e.g., [BMW03, BSZ05, BW06, BW07, Gro07]). Except theoretical constructions [BMW03, BSZ05], all of these rely on the Groth-Sahai methodology to design non-interactive proof systems for specific languages involving elements in bilinear groups [GS08]. This powerful tool led to the design of elegant compact group signatures [BW07, Gro07] whose security relies on pairing-related assumptions. The resulting signatures typically consist in a constant number of elements of a group admitting a secure and efficient bilinear map.

One desirable functionality of group signatures is the support for membership revocation. For example, misbehaving members who issue signatures for documents, which they are not allowed to sign, should be revoked from the group. In these cases, if a group signature scheme does not support revocation, then the whole system has to be re-initialized, which is obviously an unsuitable

solution in practice. Currently there are two main revocation approaches for group signatures. The first approach requires all the unrevoked members to update their signing keys after each revocation ([ACJT00, CL02, BBS04, CG04],...). At the same time, all the signature verifiers need to download the up-to-date group public key. As a consequence, it is sometimes inconvenient to practically implement such schemes. The second approach, that is group signatures with verifier-local revocation (VLR), only requires the verifiers to possess some up-to-date revocation information, but not the signers. Since in most of real-life scenarios, the number of signature verifiers is much smaller than the number of signers, this revocation approach is more flexible and more practical. Moreover, it is akin to that of the traditional Public Key Infrastructures, where the verifiers use the latest Certificate Revocation List to check the public key of the signer. The notion of VLR group signatures was introduced by Brickell [Bri03], then formalized by Boneh and Shacham [BS04], further investigated and extended by Nakanishi and Funabiki [NF05, NF06], Libert and Vergnaud [LV09], and Bichsel et al. [BCN+10]. All the existing VLR group signatures schemes operate in the bilinear map setting.

In this chapter, we first give preliminaries specific to group signatures we construct in this third part of the thesis. In particular we recall the definition of a zero-knowledge proof of knowledge and how to construct such a proof for the SIS and LWE problems. We then give the definition and security properties of the two models of group signatures that we study: The original model of Bellare, Micciancio and Warinschi [BMW03] and the model of Boneh and Shacham [BS04] which allows verifier local revocation.

6.1 Preliminaries

In this section, we recall the definition of one-time signatures, commitment schemes and proofs of knowledge. We also describe a zero-knowledge proof of knowledge for an ISIS solution.

6.1.1 One-time signatures

A one-time signature scheme consists of a triple of PPT algorithms $\Pi^{\text{ots}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ such that, on input of a security parameter 1^λ, $\mathcal{G}$ generates a one-time key pair $(\mathsf{SK}, \mathsf{VK})$; $\mathcal{S}$ is a possibly randomized algorithm that outputs a signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, M)$ on input of SK and M; and $\mathcal{V}(\mathsf{VK}, sig, M)$ is a deterministic algorithm that outputs 1 (for accept) or 0 (for reject). The standard correctness requirement mandates that $\mathcal{V}$ always accepts the signatures generated by $\mathcal{S}$.

In a strongly unforgeable one-time signature, the adversary is not only unable to forge a signature on a new message but, in addition, no PPT adversary can create a new signature for a previously signed message.

Definition 6.1. $\Pi^{\text{ots}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ is a strongly unforgeable one-time signature if the probability

$$\mathbf{Adv}^{\mathsf{OTS}}(n) = \Pr\big[(\mathsf{SK}, \mathsf{VK}) \leftarrow \mathcal{G}(1^n);\ (M, St) \leftarrow \mathcal{F}(\mathsf{VK});\ sig \leftarrow \mathcal{S}(\mathsf{SK}, M);\\ (M', sig') \leftarrow \mathcal{F}(\mathsf{VK}, M, sig, St) : \mathcal{V}(\mathsf{VK}', sig', M') = 1 \wedge (M', sig') \neq (M, sig)\ \big],$$

is negligible for any PPT forger $\mathcal{F}$, where St denotes $\mathcal{F}$'s state information across stages.

6.1.2 The KTX string commitment scheme

We define a commitment scheme. Such a scheme allows a user to commit a chosen value and keep it hidden until the user chooses to reveal this value to the other.

Definition 6.2. A commitment scheme $(\mathsf{KGen}, \mathsf{Com}, \mathsf{Ver})$ consists of three PPT algorithms:

$\mathsf{KGen}(1^\lambda) \to pk$. On input 1^λ, the key generation algorithm outputs a public commitment key pk.

$\mathsf{Com}(m, pk) \to (c, d)$. On input a message $m \in \mathcal{M}$ and pk, the commitment algorithm outputs a commitment/opening pair (c, d).

$\mathsf{Ver}(pk, m, c, d) \to \{0, 1\}$. On input the message m, the key pk and the pair (c, d), the verification algorithm outputs 0 or 1.

In our case, it satisfies the following properties:

- *Correctness*: The algorithm Ver evaluates to 1 whenever the inputs where computed by an honest party:

$$\Pr[\mathsf{Ver}(pk, m, c, d) = 1; pk \leftarrow \mathsf{KGen}(1^\lambda), m \in \mathcal{M}, (c, d) \leftarrow \mathsf{Com}(m, pk)] = 1.$$

- *Computationally binding*: With overwhelming probability over the choice of the key pk, no commitment c can be opened in two different ways:

$$Pr\left[(\mathsf{Com}(pk, m) = \mathsf{Com}(pk, m')) \wedge (m \neq m')\right] \leq \mathrm{negl}(\lambda).$$

- *Statistically hiding*: A commitment c statistically hides the committed message: with overwhelming probability over the choice of pk, for every $m, m' \in \mathcal{M}$ and $(c, d) \leftarrow \mathsf{Com}(m, pk)$, $(c', d') \leftarrow \mathsf{Com}(m', pk)$ the distributions c and c' are statistically indistinguishable.

Kawachi et al. [KTX08] constructed a string commitment scheme $\mathrm{COM} : \{0,1\}^* \times \{0,1\}^{\overline{m}/2} \to \mathbb{Z}_q^n$, such that:

- If $\overline{m} > 2n(1+\delta)\log q$ for some positive constant δ, then COM is statistically hiding.
- If the $\mathsf{SIS}^\infty_{n,\overline{m},q,1}$ problem is hard, then COM is computationally binding.

For simplicity, we will omit the randomness of the commitment when we use this scheme in Chapter 8. Also, we implicitly choose $\overline{m}$ sufficiently large, e.g., $\overline{m} = 4n \log q$, to make COM statistically hiding.

6.2 Zero-knowledge proofs of knowledge

6.2.1 Definition

We use the definitions of [BG92, PV08]. We first define a *non-interactive proof system* for a language $\mathcal{L}$. We denote by λ the security parameter associated to the language.

Definition 6.3. A pair (P, V) is a non-interactive proof system for a language $\mathcal{L}$ if V is polynomial-time and the following two conditions hold for some functions $c(\lambda), s(\lambda) : \mathbb{N} \to [0, 1]$ and for all $\lambda \in \mathbb{N}$:

- *Completeness*: For every $x \in \mathcal{L}$, $\Pr[V(x, r, P(x, r)) \text{ accepts}] \geq 1 - c(\lambda)$,

- *Soundness*: For every $x \notin \mathcal{L}$, $\Pr[\exists \pi : V(x, r, \pi) \text{ accepts}] \leq s(\lambda)$.

The probability are taken over the choice of the random input r and the random choices of P. The function $c(\lambda)$ is called the completeness error, and the function $s(\lambda)$ is called the soundness error. We also require $c(\lambda) + s(\lambda) \leq 1 - 1/\text{poly}(\lambda)$.

We now define the notion of *zero-knowledge*, that we will use in a non-interactive setting.

Definition 6.4. A non-interactive proof system (P, V) for a language $\mathcal{L}$ is (statistically) *zero-knowledge* if there exists a probabilistic polynomial time algorithm S such that for all $x \in \mathcal{L}$, the distributions $S(x)$ and $(r, P(x, r))$ are computationally (resp. statistically) indistinguishable, for a random input r.

The principle of zero knowledge proof of knowledge (ZKPoK) is the following. We consider two parties: a Verifier V and a Prover P, then the ZKPoK allows the Prover to show to the Verifier that he knows an information without revealing anything about it. We use them for binary relations R, where the parties share x and the Prover knows y such that $(x, y) \in R$. We call y the witness of the Prover. The protocol satisfies three properties: completeness (the Verifier always accepts if the Prover is honest), zero-knowledgeableness (as defined above), and finally, soundness (from every Prover P which can make the verifier accept with probability larger than κ (defined in Definition 6.5) a y' can be extracted efficiently such that $(x, y') \in R$).

For a binary relation R we define $R(x) = \{y : (x, y) \in R\}$ and $\mathcal{L}_R = \{x : \exists y \text{ s.t. } (x, y) \in R\}$. We now define formally a proof of knowledge [BG92].

Definition 6.5. Let R be a binary relation, let V be a deterministic polynomial time function, and let $c(\lambda), \kappa(\lambda) : \mathbb{N} \to [0, 1]$ be functions. We say that V is a *knowledge verifier for the relation* R with completeness error c and knowledge error κ if the following conditions hold:

- *Completeness*: There exists a probabilistic function P such that for all $x \in \mathcal{L}_R$,

$$\Pr_{r,P}[V(x, r, P(x, r)) \text{ accepts}] \geq 1 - c(\lambda),$$

- *Validity*: There exists a probabilistic oracle machine K such that for every probabilistic function P and every $x \in \mathcal{L}_R$ where $p_x = \Pr_{r,P}[V(x, r, P(x, r)) \text{ accepts}] > \kappa(\lambda)$, K outputs a string from $R(x)$ in expected time at most $\text{poly}(\lambda)/(p_x - \kappa(\lambda))$.

The oracle machine K is called a *universal knowledge extractor*.

Finally a *witness-indistinguishable* [FS90] proof of knowledge has the same properties as a zero-knowledge proof except the zero-knowledge one. Instead of learning nothing about the witness, this kind of proof guarantees that the Verifier will not be able to distinguish between two different witnesses.

6.2.2 Computational Problems

The security of the schemes presented in Chapters 7 and 8 provably relies (in the ROM) on the assumption that the ISIS (and LWE in Chapter 7) problems are hard. In particular because we use in both scheme a zero-knowledge proof of knowledge constructed on this assumption.

The ISIS problem. We define the following relation:

$$R_{\mathsf{ISIS}_p} = \left\{(\mathbf{A}, \mathbf{y}, \beta; \mathbf{x}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n \times \mathbb{Q} \times \mathbb{Z}^m : \mathbf{x}^T \cdot \mathbf{A} = \mathbf{y}^T \ \wedge \ \|\mathbf{x}\|_p \leq \beta\right\},$$

where p corresponds to the choice of the norm.

We will make use in Chapter 7 of a non-interactive zero-knowledge proof of knowledge (NIZPoK) protocol, which can be rather directly derived from [Lyu08, Lyu12], for the relation R_{ISIS} (for p=2, using the Euclidean norm). And we will use in Chapter 8 an adaptation of the "Stern Extension" proof system of [LNSW13], which is a NIZPoK protocol for the relation R_{ISIS_∞} (for the infinity norm). We discuss this choice in Remark 6.6.

Non-interactive protocol. Those two protocols, detailed in the next two sections, are derived from the parallel repetition of a Σ-protocol with binary challenges. A Σ-protocol has three exchanges between the Prover and the Verifier: First the Prover sends a commitment Comm, then the Verifier sends a challenge Chall, finally the Prover sends the response Resp and the Verifier can check if he accepts or not. To make this kind of protocol non-interactive, one can use the Fiat-Shamir heuristic [FS86] and implements the challenge using the random oracle $H(\cdot)$. We call $\mathsf{Prove}_{\mathsf{ISIS}}$ and $\mathsf{Verify}_{\mathsf{ISIS}}$ the PPT algorithms run by the Prover and the Verifier when the scheme is rendered non-interactive.

- Algorithm $\mathsf{Prove}_{\mathsf{ISIS}}$ takes $(\mathbf{A}, \mathbf{y}, \beta; \mathbf{x})$ as input, and generates a transcript $(\mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp})$.
- Algorithm $\mathsf{Verify}_{\mathsf{ISIS}}$ takes $(\mathbf{A}, \mathbf{y}, \beta)$ and such a transcript as inputs, and returns 0 or 1.

The scheme has completeness error $2^{-\Omega(n)}$: if $\mathsf{Prove}_{\mathsf{ISIS}}$ is given as input an element of R_{ISIS}, then given as input the output of $\mathsf{Prove}_{\mathsf{ISIS}}$, $\mathsf{Verify}_{\mathsf{ISIS}}$ replies 1 with probability $\geq 1 - 2^{-\Omega(n)}$ (over the randomness of $\mathsf{Prove}_{\mathsf{ISIS}}$).

Also, there exists a PPT algorithm $\mathsf{Simulate}_{\mathsf{ISIS}}$ that, by reprogramming the random oracle $H(\cdot)$, takes $(\mathbf{A}, \mathbf{y}, \beta)$ as input and generates a transcript $(\mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp})$ whose distribution is within statistical distance $2^{-\Omega(n)}$ of the genuine transcript distribution. Finally, there also exists a PPT algorithm $\mathsf{Extract}_{\mathsf{ISIS}}$ that given access to a time T algorithm $\mathcal{A}$ that generates transcripts accepted by $\mathsf{Verify}_{\mathsf{ISIS}}$ with probability ε, produces, in time $\mathrm{poly}(T, 1/\varepsilon)$ a vector $\mathbf{x}'$ such that $(\mathbf{A}, \mathbf{y}, \mathcal{O}(\beta \cdot m \cdot n); \mathbf{x}') \in R_{\mathsf{ISIS}}$.

The LWE problem. In Chapter 7, we will also need a NIZKPoK protocol for the following language:

$$R_{\mathsf{LWE}} = \left\{(\mathbf{A}, \mathbf{b}, \alpha; \mathbf{s}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \times \mathbb{Q} \times \mathbb{Z}_q^n : \|\mathbf{b} - \mathbf{A} \cdot \mathbf{s}\| \leq \alpha q \sqrt{m}\right\}.$$

As noted in [Lyu12], we may multiply $\mathbf{b}$ by a parity check matrix $\mathbf{G} \in \mathbb{Z}_q^{(m-n) \times m}$ of $\mathbf{A}$ and prove the existence of small $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{e}^T \cdot \mathbf{G}^T = \mathbf{b}^T \cdot \mathbf{G}^T$. This may be done with the above NIZKPoK protocol for R_{ISIS}. We call $\mathsf{Prove}_{\mathsf{LWE}}$, $\mathsf{Verify}_{\mathsf{LWE}}$, $\mathsf{Simulate}_{\mathsf{LWE}}$ and $\mathsf{Extract}_{\mathsf{LWE}}$ the obtained PPT algorithms.

6.2.3 Proof of Knowledge of an ISIS Solution

In [Lyu08], Lyubashevsky described an identification scheme whose security relies on the hardness of the SIS problem. Given a public vector $\mathbf{y} \in \mathbb{Z}_q^n$ and a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$, the prover holds a short secret $\mathbf{x}$ and generates an interactive witness indistinguishable proof of knowledge of a short vector $\mathbf{x}' \in \mathbb{Z}^m$ such that $\mathbf{x}'^T \cdot \mathbf{A} = \mathbf{y}^T \bmod q$. A variant was later proposed in [Lyu12], which enjoys the property of being zero-knowledge (when the distribution of the transcript is conditioned

on the prover not aborting). We present an adaptation of [Lyu12, Fig. 1] (still enjoying the same zero-knowledgedness property): the secret is a single vector, the challenges are binary (which we use for the extraction procedure), and we increase the standard deviation of the commited vector to lower the rejection probability (we use a parallel repetition of the basic scheme, and want the probability that there is a reject among all the parallel iterations to be sufficiently away from 1).

Assume the prover P wishes to prove knowledge of an $\mathbf{x}$ such that $\mathbf{x}^T \cdot \mathbf{A} = \mathbf{y}^T \bmod q$ and $\|\mathbf{x}\| \leq \beta$, where $\mathbf{y}$ and $\mathbf{A}$ are public. The protocol takes place between the prover P and the verifier V and proceeds by the k-times parallel repetition of a basic Σ-protocol with binary challenges. We set $\sigma = \Theta(\beta\sqrt{mn})$ and M_L as specified by [Lyu12, Th. 4.6]. Thanks to our larger value of σ, we obtain (by adapting [Lyu12, Le. 4.5]) that M_L is now $1 - \Omega(1/n)$.

---

1. **Commitment:** The prover P generates a commitment $\mathsf{Comm} = (\mathbf{w}_i)_{i \leq k}$ where, for each $i \leq k$, $\mathbf{w}_i \in \mathbb{Z}_q^n$ is obtained by sampling $\mathbf{y}_i \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ and computing $\mathbf{w}_i^T = \mathbf{y}_i^T \cdot \mathbf{A} \bmod q$. The message Comm is sent to V.
2. **Challenge:** The verifier V sends a challenge $\mathsf{Chall} \hookleftarrow \{0,1\}^k$ to P.
3. **Response:** For $i \leq k$, the prover P does the following.
   a. Compute $\mathbf{z}_i = \mathbf{y}_i + \mathsf{Chall}[i] \cdot \mathbf{x}$, where $\mathsf{Chall}[i]$ denotes the i^{th} bit of Chall.
   b. Set $\mathbf{z}_i$ to $\perp$ with probability $\min(1, \frac{\exp(-\pi\|\mathbf{z}\|^2/\sigma^2)}{M_L \cdot \exp(-\pi\|\mathsf{Chall}[i]\cdot\mathbf{x}-\mathbf{z}\|^2/\sigma^2)})$.

   Then P sends the response $\mathsf{Resp} = (\mathbf{z}_i)_{i \leq k}$ to V.

Verification: The verifier V checks the transcript $(\mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp})$ as follows:

a. For $i \leq k$, set $d_i = 1$ if $\|\mathbf{z}_i\| \leq 2\sigma\sqrt{m}$ and $\mathbf{z}_i^T \cdot \mathbf{A} = \mathbf{w}_i^T + \mathsf{Chall}[i] \cdot \mathbf{y}^T$. Otherwise, set $d_i = 0$.

b. Return 1 (and accept the transcript) if and only if $\sum_{i \leq k} d_i \geq 0.65k$.

Figure 6.1: Proof of knowledge of an ISIS solution.

The protocol has completeness error $2^{-\Omega(k)}$. Further, by [Lyu12, Th. 4.6], the distribution of the transcript conditioned on $\mathbf{z}_i \neq \perp$ can be simulated efficiently. Note that if we implement the challenge phase with a random oracle, we can compute the $\mathbf{z}_i$'s for increasing values of i, and repeat the whole procedure if $\mathbf{z}_i = \perp$ for some i. Thanks to our choice of σ, for any $k \leq \mathcal{O}(n)$, the probability that $\mathbf{z}_i = \perp$ for some i is $\leq c$, for some constant $c < 1$. Thanks to this random-oracle-enabled rejection, the simulator produces a distribution that is within statistical distance $2^{-\Omega(n)}$ from the transcript distribution.

Finally, the modified protocol provides special soundness in that there is a simple extractor that takes as inputs two valid transcripts $(\mathsf{Comm}, \mathsf{Chall}, \mathsf{Resp})$, $(\mathsf{Comm}, \mathsf{Chall}', \mathsf{Resp}')$ with distinct challenges $\mathsf{Chall} \neq \mathsf{Chall}'$ and obtains a witness $\mathbf{x}'$ such that $\mathbf{x}'^T \cdot \mathbf{A} = \mathbf{y}^T \bmod q$ and $\|\mathbf{x}'\| \leq \mathcal{O}(\sigma\sqrt{m}) \leq \mathcal{O}(\beta mn)$.

6.2.4 The LNSW Proof System

Alternatively, Ling et al. [LNSW13] proposed a Stern-type zero-knowledge proof of knowledge for the $\mathrm{ISIS}^{\infty}_{n,m,q,\beta}$ problem that relies on the ISIS assumption. They achieve this feature by using a

Decomposition-Extension technique. We use an adaptation of this technique in Chapter 8.

The input of this protocol is a pair $(\mathbf{A}, \mathbf{y})$, and the prover also knows a $\mathbf{x}$ such that $\mathbf{x}^T\mathbf{A} = \mathbf{y} \bmod q$. We denote by B_{3m} the set of all vectors in $\{-1, 0, 1\}^{3m}$ having exactly m coordinates -1; m coordinates 0; and m coordinates 1. And we denote by S_k the symmetric group of all permutations of k elements. The prover and the verifier starts by both form the matrix $\mathbf{A}' \in \mathbb{Z}_q^{3m \times n}$, by appending $2m$ zero rows to $\mathbf{A}$. Then the prover uses the following steps on $\mathbf{x}$.

- **Elementary Decomposition.** On input a vector $\mathbf{x} = (x_1, x_2, \ldots, x_m) \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_\infty \leq \beta$, the procedure EleDec outputs $p = \lfloor \log \beta \rfloor + 1$ vectors $\widetilde{\mathbf{z}}_1, \ldots, \widetilde{\mathbf{z}}_p \in \{-1, 0, 1\}^m$, such that $\sum_{j=1}^{p} 2^j \cdot \widetilde{\mathbf{z}}_j = \mathbf{x}$. This procedure works as follows:
  1. For each $i \in \{1, \ldots, m\}$, express x_i as $v_i = 2 \cdot x_{i,1} + 2^2 \cdot x_{i,2} + \ldots + 2^p \cdot x_{i,p}$, where $\forall j \in \{1, \ldots, p\} : x_{i,j} \in \{-1, 0, 1\}$.
  2. For each $j \in \{1, \ldots, p\}$, let $\widetilde{\mathbf{z}}_j := (x_{1,j}, x_{2,j}, \ldots, x_{m,j}) \in \{-1, 0, 1\}^m$. Output $\widetilde{\mathbf{z}}_1, \ldots, \widetilde{\mathbf{z}}_p$.
- **Elementary Extension.** On input a vector $\widetilde{\mathbf{z}} \in \{-1, 0, 1\}^m$, the procedure EleExt extends $\widetilde{\mathbf{z}}$ to a vector $\mathbf{z} \in \mathsf{B}_{3m}$. This procedure works as follows:
  1. Let $\lambda^{(-1)}$, $\lambda^{(0)}$ and $\lambda^{(1)}$ be the numbers of coordinates of $\widetilde{\mathbf{z}}$ that equal to -1, 0, and 1 respectively.
  2. Pick a random vector $\widehat{\mathbf{w}} \in \{-1, 0, 1\}^{2m}$ that has exactly $(m - \lambda^{(-1)})$ coordinates -1, $(m - \lambda^{(0)})$ coordinates 0, and $(m - \lambda^{(1)})$ coordinates 1. Output $\mathbf{z} = (\widetilde{\mathbf{z}} \| \widehat{\mathbf{z}}) \in \mathsf{B}_{3m}$.

We denote by $\mathbf{z}$ the decomposition-extension of the element $\mathbf{x}$ following these two procedures. We have that:

$$\Big(\sum_{j=1}^{p} 2^j \cdot \mathbf{z}_j\Big)^T \cdot \mathbf{A}' = \mathbf{y}^T \bmod q \Leftrightarrow \mathbf{x}^T\mathbf{A} = \mathbf{y}^T \bmod q$$

We describe the interactive proof system of [LNSW13] in Figure 6.2.

It is shown in [LNSW13] that this proof system is statistically zero-knowledge if COM is statistically hiding, and that it is a proof of knowledge for the relation R_{ISIS_∞} with knowledge error $\kappa = 2/3$.

Remark 6.6. In the two constructions we give in Chapters 7 and 8, we need a ZKPoK for the ISIS relation. In Chapter 7, we use the construction of Figure 6.1, as we need the property of special soundness to be able to prove the disjunction of two relations (and the second construction of [LNSW13] does not have this property). Whereas in Chapter 8, we use an adaptation of the [LNSW13] proof system, as we will need to modify the proof by adding some conditions in the relation. This second proof system is more versatile, and then it is possible to adapt it to those new conditions.

6.3 Group signature model

This section recalls the model of Bellare, Micciancio and Warinschi [BMW03], which assumes static groups.

6.3.1 Definition

A group signature scheme $\mathcal{GS}$ consists of a tuple of four PPT algorithms $(\mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ with the following specifications:

1. **Commitment:** The prover samples p permutations $\pi_1, \ldots, \pi_p \hookleftarrow U(\mathcal{S}_m)$, and p vectors $\mathbf{r}_1, \ldots, \mathbf{r}_p \hookleftarrow U(\mathbb{Z}_q^{3m})$. Then it sends the commitment $\mathrm{CMT} = \big(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\big) \in (\mathbb{Z}_q^n)^3$ to the verifier, where

$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}\big(\{\pi_j\}_{j=1}^p, \big(\sum_{j=1}^p 2^j \cdot \mathbf{r}_j\big)^T \cdot \mathbf{A}' \bmod q\big), \\ \mathbf{c}_2 = \mathsf{COM}\big(\{\pi_j(\mathbf{r}_j)\}_{j=1}^p\big), \\ \mathbf{c}_3 = \mathsf{COM}\big(\{\pi_j(\mathbf{z}_j + \mathbf{r}_j)\}_{j=1}^p\big). \end{cases} \tag{6.1}$$

2. **Challenge:** The verifier sends a challenge $Ch \hookleftarrow U(\{1,2,3\})$ to the prover.

3. **Response:** Depending on the challenge, the prover computes the response RSP differently:

   - Case $Ch = 1$: $\forall j \in \{1, \ldots, p\}$, let $\mathbf{v}_j = \pi_j(\mathbf{z}_j)$, $\mathbf{w}_j = \pi_j(\mathbf{r}_j)$ and set:
   $$\mathrm{RSP} = \big(\{\mathbf{v}_j\}_{j=1}^p, \{\mathbf{w}_j\}_{j=1}^p\big). \tag{6.2}$$
   - Case $Ch = 2$: $\forall j \in \{1, \ldots, p\}$, let $\phi_j = \pi_j$, $\mathbf{s}_j = \mathbf{z}_j + \mathbf{r}_j$ and set:
   $$\mathrm{RSP} = \big(\{\phi_j\}_{j=1}^p, \{\mathbf{s}_j\}_{j=1}^p\big). \tag{6.3}$$
   - Case $Ch = 3$: $\forall j \in \{1, \ldots, p\}$, let $\psi_j = \pi_j$, $\mathbf{h}_j = \mathbf{r}_j$ and set:
   $$\mathrm{RSP} = \big(\{\psi_j\}_{j=1}^p, \{\mathbf{h}_j\}_{j=1}^p\big). \tag{6.4}$$

Verification: Receiving the response RSP, the verifier proceeds as follows:

- Case $Ch = 1$: Parse RSP as in (6.2). Check that $\forall j \in \{1, \ldots, p\} : \mathbf{v}_j \in B_{3m}$, and that:
$$\mathbf{c}_2 = \mathsf{COM}\big(\{\mathbf{w}_j\}_{j=1}^p\big) \text{ and } \mathbf{c}_3 = \mathsf{COM}\big(\{\mathbf{v}_j + \mathbf{w}_j\}_{j=1}^p\big).$$
- Case $Ch = 2$: Parse RSP as in (6.3). Check that:
$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}\big(\{\phi_j\}_{j=1}^p, \big(\sum_{j=1}^p 2^j \cdot \mathbf{s}_j\big)^T \cdot \mathbf{A}' - \mathbf{u}^T \bmod q\big) \\ \mathbf{c}_3 = \mathsf{COM}\big(\{\phi_j(\mathbf{s}_j)\}_{j=1}^p\big). \end{cases}$$
- Case $Ch = 3$: Parse RSP as in (6.4). Check that:
$$\begin{cases} \mathbf{c}_1 = \mathsf{COM}\big(\{\psi_j\}_{j=1}^p, (\sum_{j=1}^p 2^j \cdot \mathbf{h}_j)^T \cdot \mathbf{A}' \bmod q\big) \\ \mathbf{c}_2 = \mathsf{COM}\big(\{\psi_j(\mathbf{h}_j)\}_{j=1}^p\big). \end{cases}$$

The verifier outputs Valid if and only if all the conditions hold. Otherwise, he outputs Invalid.

Figure 6.2: The LNSW SternExt proof system.

$\mathsf{Keygen}(1^\lambda, 1^N) \to (\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$. The key generation algorithm takes as inputs the security parameter λ and the maximum number of group members N. It returns a tuple $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$ where gpk is the *group public key*, gmsk is the group manager secret key, and gsk is an N-dimensional vector of secret keys: $\mathsf{gsk}[j]$ is the signing key of the j-th user, for $j \in \{0, \ldots, N-1\}$.

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M) \to \Sigma$. The signing algorithm takes as inputs the group public key gpk, a signing key $\mathsf{gsk}[j]$ and a message $M \in \{0,1\}^*$. Its output is a signature $\Sigma \in \{0,1\}^*$ on M.

$\mathsf{Verify}(\mathsf{gpk}, M, \Sigma) \to \{0, 1\}$. The verifying algorithm is deterministic and takes as inputs the group public key gpk, a message M and a putative signature Σ of M. It outputs either 0 or 1.

$\mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, M, \Sigma) \to \{j, \perp\}$. The opening algorithm is deterministic and takes as inputs the group public key gpk, the group manager secret key gmsk, a message M and a valid group signature Σ w.r.t. gpk. It returns an index $j \in \{0, \ldots, N-1\}$ or a special symbol $\perp$ in case of opening failure.

The group signature scheme must be *correct*, i.e., for sufficiently large integers λ and any integer N, all $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$ obtained from Keygen with $(1^\lambda, 1^N)$ as input, all indices $j \in \{0, \ldots, N-1\}$ and $M \in \{0,1\}^*$:

$$\mathsf{Verify}(\mathsf{gpk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)) = 1 \text{ and } \mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, M, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)) = j,$$

with probability negligibly close to 1 over the internal randomness of Keygen and Sign.

Bellare *et al.* [BMW03] gave a unified security model for group signatures in static groups. The two main security requirements are *traceability* and *anonymity*. The former asks that no coalition of group members be able to create a signature that cannot be traced to one of them. The latter implies that, even if all the private keys are given to the adversary, signatures generated by two distinct group members should be computationally indistinguishable.

6.3.2 Anonymity

Anonymity requires that, without the group manager's secret key, an adversary cannot recognize the identity of a user given its signature. More formally, the attacker, modeled as a two-stage adversary (*choose* and *guess*), is engaged in the first random experiment which runs as follows.

1. Setup. The challenger runs $\mathsf{KeyGen}(1^\lambda, 1^N)$ to generate $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$, then gives gpk and gsk to the adversary $\mathcal{A}$.

2. $\mathsf{Choose\ stage}$. Adversary $\mathcal{A}$ can make queries to the following oracles:

   - Open: Query for opening on any message $M \in \{0,1\}^*$ and signature Σ. The challenger returns the identity i of the user.

3. $\mathsf{Challenge}$. Adversary $\mathcal{A}$ outputs a message M^*, two indices j_0 and j_1 and a state st. The challenger chooses a bit $b \hookleftarrow U(\{0,1\})$, computes a signature of user j_b on M^* as $\Sigma^* = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j_b], M^*)$, and returns Σ^* and st to $\mathcal{A}$.

4. $\mathsf{Guess\ stage}$. After the challenge phase, $\mathcal{A}$ can still make queries as before, but with the following restrictions: it is not allowed to make any open query for user j_0 or user j_1.

5. Output. Eventually, $\mathcal{A}$ outputs a bit b'. It wins the game if $b' = b$.

The *advantage* of such an adversary $\mathcal{A}$ against a group signature $\mathcal{GS}$ with N members is defined as

$$\mathbf{Adv}^{\mathsf{anon}}_{\mathcal{GS},\mathcal{A}}(\lambda, N) = \left|\Pr[\mathbf{Exp}^{\mathsf{anon}-1}_{\mathcal{GS},\mathcal{A}}(\lambda, N) = 1] - \Pr[\mathbf{Exp}^{\mathsf{anon}-0}_{\mathcal{GS},\mathcal{A}}(\lambda, N) = 1]\right|.$$

In our first scheme of Chapter 7, we consider a *weak anonymity* scenario in which the adversary is not allowed to query an opening oracle. This relaxed model is precisely the one considered in [GKV10], and was firstly introduced in [BBS04]. Nonetheless, we provide in Section 7.3 a variant of our scheme enjoying chosen-ciphertext security. The adversary is then granted an access to an opening oracle that can be called on any string except the challenge signature $\Sigma^\star$.

Definition 6.7 (Weak and full anonymity, [BMW03, BBS04])**.** A group signature scheme $\mathcal{GS}$ is said to be *weakly anonymous* (resp. *fully anonymous*) if for all polynomial $N(\lambda)$ and all PPT adversaries $\mathcal{A}$ (resp. PPT adversaries $\mathcal{A}$ with access to an opening oracle which cannot be queried for the challenge signature), $\mathbf{Adv}^{\mathsf{anon}}_{\mathcal{GS},\mathcal{A}}(\lambda, N)$ is a negligible function in the security parameter λ.

6.3.3 Full traceability

Full traceability ensures that all signatures, even those created by a coalition of users *and* the group manager, pooling their secret keys together, can be traced to a member of the forging coalition. Once again, the attacker is modeled as a two-stage adversary who is run within the second experiment as follows.

1. Setup: Run $\mathsf{KeyGen}(1^\lambda, 1^N)$ to obtain $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$. Adversary $\mathcal{A}$ is given $(\mathsf{gpk}, \mathsf{gmsk})$. Set $\mathcal{C} = \emptyset$.
2. Choose stage: Adversary $\mathcal{A}$ can make queries to the following oracles:
   - Signing: On input a message M, and an index j, the oracle returns $\Sigma = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)$.
   - Corruption: On input an index j, the oracle adds j to the set $\mathcal{C}$, and returns $\mathsf{gsk}[j]$.
3. Guess stage: Eventually, $\mathcal{A}$ outputs a message M^* and a signature Σ^*. The adversary wins the game, and the experiment returns 1, if:
   a) $\mathsf{Verify}(\mathsf{gpk}, \Sigma^*, M^*) = 1$.
   b) The opening algorithm outputs $\perp$ or j such that $j \notin \mathcal{C}$.
   c) The signature Σ^* is non-trivial, i.e., $\mathcal{A}$ did not obtain Σ^* by making a signing query on j^* and M^*.

   Otherwise the experiment returns 0.

   Its success probability against $\mathcal{GS}$ is defined as

$$\mathbf{Succ}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(\lambda, N) = \Pr[\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(\lambda, N) = 1].$$

Definition 6.8 (Full traceability, [BMW03])**.** A group signature scheme $\mathcal{GS}$ is said to be *fully traceable* if for all polynomial $N(\lambda)$ and all PPT adversaries $\mathcal{A}$, its success probability $\mathbf{Succ}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(\lambda, N)$ is negligible in the security parameter λ.

6.4 Group signature with VLR model

The presentation in this section follows [BS04] and recalls the definition and properties of a group signature scheme with verifier local revocation.

6.4.1 Definition

A VLR group signature consists of threee following algorithms:

$\mathsf{KeyGen}(1^\lambda, 1^N) \to (\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$. On input a security parameter λ and the number of group users N, this PPT algorithm outputs a group public key gpk, an N-dimensional vector of user secret keys $\mathsf{gsk} = [\mathsf{gsk}[j]]_j$, and an N-dimensional vector of user revocation tokens $\mathsf{grt} = [\mathsf{grt}[j]]_j$, for $j \in \{0, \ldots, N-1\}$.

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M) \to \Sigma$. On input gpk, a user secret key $\mathsf{gsk}[j]$, and a message $M \in \{0,1\}^*$, this PPT algorithm outputs a signature Σ.

$\mathsf{Verify}(\mathsf{gpk}, RL, \Sigma, M) \to \{0,1\}$. On input gpk, a set of revocation tokens $RL \subseteq \{\mathsf{grt}[j]\}_j$, a signature Σ, and the message M, this algorithm outputs either 0 or 1. The output 1 indicates that Σ is a valid signature on message M under gpk, and that the signer has not been revoked.

The VLR group signature scheme must be *correct*, i.e., for all $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$ output by KeyGen, $M \in \{0,1\}^*$, and $d \in \{0, 1, \ldots, N-1\}$:

$$\mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M), M) = 1 \Leftrightarrow \mathsf{grt}[j] \notin RL.$$

Remark 6.9. Any VLR group signature has an *implicit tracing algorithm* using grt as the tracing key. The tracing algorithm works as follows: on input a valid signature Σ on a message M, it reveals the signer of Σ by running $\mathsf{Verify}(\mathsf{gpk}, RL = \mathsf{grt}[j], \Sigma, M)$, for $j = 0, 1, \ldots$, and outputting the first index $j^* \in \{0, 1, \ldots, N-1\}$ for which the verification algorithm returns 0. The tracing algorithm fails if and only if the given signature is properly verified for all j.

6.4.2 Selfless-anonymity

In the following selfless-anonymity game, the adversary's goal is to determine which of the two adaptively chosen keys generated a signature. He is not given access to either key.

1. Setup. The challenger runs $\mathsf{KeyGen}(1^\lambda, 1^N)$ to generate $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$, then gives gpk to the adversary $\mathcal{A}$.

2. $\mathsf{Queries}$. Adversary $\mathcal{A}$ can make queries to the following oracles:

   - Signing: Query for signature of any user d on any message $M \in \{0,1\}^*$. The challenger returns the signature $\Sigma = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)$.
   - Corruption: Query for the secret key of any user of index j. The challenger returns $\mathsf{gsk}[j]$.
   - Revocation: Query for the revocation token of any user of index j. The challenger returns $\mathsf{grt}[j]$.

3. $\mathsf{Challenge}$. Adversary $\mathcal{A}$ outputs a message M^* and two indices j_0 and j_1, such that $\mathcal{A}$ never made a corruption or revocation query for index j_0 or j_1. The challenger chooses a bit $b \hookleftarrow U(\{0,1\})$, computes a signature of user of index j_b on M^* as $\Sigma^* = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j_b], M^*)$, and returns Σ^* to $\mathcal{A}$.

4. $\mathsf{Restricted\ Queries}$. After the challenge phase, $\mathcal{A}$ can still make queries as before, but with the following restrictions: it is not allowed to make any corruption or revocation query for the users of indices j_0 and j_1.

5. Output. Eventually, $\mathcal{A}$ outputs a bit b'. It wins the game if $b' = b$.

We define the adversary's advantage in winning the game as $\mathsf{Adv}_{\mathcal{A}} = \left|\Pr[b' = b] - 1/2\right|$. We say that the VLR group signature is selfless-anonymous if $\mathsf{Adv}_{\mathcal{A}}$ is negligible.

Comparison with weak and full anonymity. The notion of selfless anonymity used in the definition of a VLR group signature is weaker than the notions of weak and full anonymity defined for a group signature: in the two last one, the adversary knows the secret keys of the users used in the challenge. Indeed in the weak and full anonymity, the adversary has all the secret keys of the users $\{\mathsf{gsk}_i\}$ (note that he does not need a signing oracle, as he knows the secret key and then can sign any message with one of them). In comparison, in the selfless anonymity game, the adversary can make signing and corruption queries (to obtain a secret key of a user) but cannot know the secret key of one of its two challenge indices. Moreover, in the full anonymity game the adversary also has an opening oracle.

6.4.3 Traceability

The adversary's goal in the traceability game is to forge a signature that cannot be traced to one of the users in his coalition using the implicit tracing algorithm above. The traceability game is defined as follows:

1. Setup: Run $\mathsf{KeyGen}(1^\lambda, 1^N)$ to obtain $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$. Adversary $\mathcal{A}$ is given $(\mathsf{gpk}, \mathsf{grt})$. Set $U = \emptyset$.
2. Queries: Adversary $\mathcal{A}$ can make queries to the following oracles:
   - Signing: On input a message M, and an index j, the oracle returns $\Sigma = \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)$.
   - Corruption: On input an index j, the oracle adds j to the set U, and returns $\mathsf{gsk}[j]$.
3. Forgery: Eventually, $\mathcal{A}$ outputs a message M^*, a set of revocation tokens RL^* and a signature Σ^*.

   The adversary wins the game if:

   a) $\mathsf{Verify}(\mathsf{gpk}, RL^*, \Sigma^*, M^*) = 1$.
   b) The (implicit) tracing algorithm fails or traces to a user outside of the coalition $U \setminus RL^*$.
   c) The signature Σ^* is non-trivial, i.e., $\mathcal{A}$ did not obtain Σ^* by making a signing query on M^*.

The probability that $\mathcal{A}$ wins the game, denoted by $\mathsf{SuccPT}_{\mathcal{A}}$, is taken over the randomness of $\mathcal{A}$, algorithms KeyGen and Sign. We say that a VLR group signature is traceable if $\mathsf{SuccPT}_{\mathcal{A}}$ is negligible.

Comparison with the traceability game for a group signature. The two traceability games are very similar. The major differences are the informations given to the adversary which are relatives to each scheme. In the group signature, the key used in Open to trace a group member is the master secret key while in the VLR group signature the implicit tracing algorithm uses the set of all the user revocation tokens (that only the authority knows). In the traceability game, the challenger gives to the adversary the group public key and the key needed to trace: in one case the master secret key and in the other one the set of all revocation tokens. After this phase, the two games are similar: the adversary can make signing and corruption queries then outputs a forgery (with a set of revocation tokens in the case of the VLR group signature). Then the three conditions for the adversary to win the game are the same.

Chapter 7

A Lattice-Based Group Signature with Logarithmic Signature Size

Lattices and Group Signatures. While numerous works have been (successfully) harnessing the power of lattices for constructing digital signatures (see Chapter 3, [LM08, GPV08, CHKP10, Lyu09, Boy10, Lyu12], and references therein), only two works addressed the problem of efficiently realizing lattice-based group signatures. The main difficulty to overcome is arguably the scarcity of efficient and expressive non-interactive proof systems for statements involving lattices, in particular for statements on the witnesses of the hard average-case lattice problems. This state of affairs contrasts with the situation in bilinear groups, where powerful non-interactive proof systems are available [GOS06, GS08].

In 2010, Gordon *et al.* [GKV10] described the first group signature based on lattice assumptions using the Gentry *et al.* signature scheme [GPV08] as membership certificate, an adaptation of Regev's encryption scheme [Reg09] to encrypt it, and a zero-knowledge proof technique due to Micciancio and Vadhan [MV03]. While elegant in its design principle, their scheme suffers from signatures and public keys of sizes linear in the number of group members, making it utterly inefficient in comparison with constructions based on bilinear maps [BBS04] or the strong RSA assumption [ACJT00]. Quite recently, Camenisch *et al.* [CNR12] proposed anonymous attribute token systems, which can be seen as generalizations of group signatures. One of their schemes improves upon [GKV10] in that the group public key has constant size[1] and the anonymity property is achieved in a stronger model where the adversary is granted access to a signature opening oracle. Unfortunately, all the constructions of [CNR12] inherit the linear signature size of the Gordon *et al.* construction. Thus far, it remained an open problem to break the linear-size barrier. This is an important challenge considering that, as advocated by Bellare *et* al. [BMW03], one should expect practical group signatures not to entail more than poly-logarithmic complexities in the group sizes.

Our Contributions. In this Chapter, we describe the first lattice-based group signatures featuring sub-linear signature sizes. This is a joint work with F. Laguillaumie, B. Libert and D. Stehlé published in [LLLS13]. If λ and N denote the security parameter and the maximal group size, the public keys and signatures are $\widetilde{O}(\lambda^2 \cdot \log N)$ bit long. Notice that no group signature scheme can provide signatures containing $o(\log N)$ bits (such signatures would be impossible to open), so that the main improvement potential lies in the $\widetilde{O}(\lambda^2)$ factor. These first asymptotically efficient

[1] This can also be achieved with [GKV10] by replacing the public key by a hash thereof, and appending the key to the signature.

(in λ and $\log N$) lattice-based group signatures are a first step towards a practical alternative to the pairing-based counterparts. The security proofs hold in the ROM (as for [GKV10, CNR12]), under the Learning With Error (LWE) and Short Integer Solution (SIS) assumptions. While our basic system only provides anonymity in a relaxed model (like [GKV10]) where the adversary has no signature opening oracle, we show how to upgrade it into a fully anonymous group signature, in the anonymity model of Bellare *et* al. [BMW03]. This is achieved at a minimal cost in that the signature length is only increased by a constant factor. In contrast, Camenisch *et* al. [CNR12, Se. 5.2] achieve full anonymity at the expense of inflating their basic signatures by a factor proportional to the security parameter.

Construction Overview. Our construction is inspired by the general paradigm from [BMW03] consisting in *encrypting* a membership certificate under the authority's public key while providing a *non-interactive proof* that the ciphertext encrypts a valid *certificate* belonging to some group member. Nevertheless, our scheme differs from this paradigm in the sense that it is not the certificate itself which is encrypted. Instead, a temporary certificate, produced at each signature generation, is derived from the initial one and encrypted, with a proof of its validity.

We also depart from the approach of [GKV10] at the very core of the design, i.e., when it comes to provide evidence that the encrypted certificate corresponds to a legitimate group member. Specifically, Gordon *et* al. [GKV10] hide their certificate, which is a GPV signature (described in Section 3.2.3), within a set of $N-1$ (encrypted) GPV pseudo-signatures that satisfy the same verification equation without being short vectors. Here, to avoid the $\mathcal{O}(N)$ factor in the signature size, we take a different approach which is reminiscent of the Boyen-Waters group signature (described in Section 3.2.5). Each group member is assigned a unique ℓ-bit identifier $\mathrm{id} = \mathrm{id}[1]\ldots\mathrm{id}[\ell] \in \{0,1\}^\ell$, where $\ell = \lceil \log_2 N \rceil$. Its certificate is an extension of a Boyen signature [Boy10] consisting in a *full* short basis of a certain lattice (instead of a single vector), which allows the signer to generate *temporary certificates* composed of a pair $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{Z}^m$ of discrete Gaussian vectors such that

$$\mathbf{x}_1^T \cdot \mathbf{A} + \mathbf{x}_2^T \cdot (\mathbf{A}_0 + \sum_{1 \le i \le \ell} \mathrm{id}[i] \cdot \mathbf{A}_i) = \mathbf{0} \bmod q. \tag{7.1}$$

Here, q is a small bit length integer and $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{m \times n}$ are part of the group public key. Our choice of Boyen's signature as membership certificate is justified by it being one of the most efficient known lattice-based signatures proven secure in the standard model, and enjoying a simple verification procedure corresponding to a relation for which we can design a proof of knowledge. A signature proven secure in the standard model allows us to obtain an easy-to-prove relation that does not involve a random oracle. As noted for example in [ACJT00, CL02, CL04], signature schemes outside the ROM make it easier to prove knowledge of a valid message-signature pair in the design of privacy-preserving protocols.

We encrypt $\mathbf{x}_2 \in \mathbb{Z}^m$ as in [GKV10], using a variant of the dual-Regev encryption scheme (recalled in Section 3.1.3): the resulting ciphertext is $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s} + \mathbf{x}_2$, where $\mathbf{B}_0 \in \mathbb{Z}_q^{m \times n}$ is a public matrix and $\mathbf{s}$ is uniform in $\mathbb{Z}_q^n$. Then, for each $i \in \{1, \ldots, \ell\}$, we also compute a proper dual-Regev encryption $\mathbf{c}_i$ of $\mathrm{id}[i] \cdot \mathbf{x}_2$ and generate a non-interactive OR proof that $\mathbf{c}_i$ encrypts either the same vector as $\mathbf{c}_0$ or the $\mathbf{0}$ vector.

It remains to prove that the encrypted vectors $\mathbf{x}_2$ are part of a signature satisfying Eq. (7.1) without giving away the $\mathrm{id}[i]$'s. To this end, we choose the signing matrices $\mathbf{A}_i$ orthogonally to the encrypting matrices $\mathbf{B}_i$, as suggested in [GKV10]. Contrarily to the case of [GKV10], the latter technique does not by itself suffice to guarantee the well-formedness of the $\mathbf{c}_i$'s. Indeed, we also need to prove properties about the noise vectors used in the dual-Regev ciphertexts $\{\mathbf{c}_i\}_{i=1}^\ell$. This

is achieved using a modification of Lyubashevsky's protocol [Lyu08, Lyu12] to prove knowledge of a solution to the Inhomogeneous Short Integer Solution problem (ISIS) (defined in Section 2.1). This modification leads to a Σ-protocol which is zero-knowledge when the transcript is conditioned on the protocol not aborting. As the challenge space of this Σ-protocol is binary, we lowered the abort probability so that we can efficiently apply the Fiat-Shamir heuristic [FS86] to a parallel repetition of the basic protocol. In the traceability proof, the existence of a witness extractor will guarantee that a successful forger will either yield a forgery for Boyen's signature or a short non-zero vector in the kernel of one of the matrices $\{\mathbf{A}_i\}_{i=1}^{\ell}$. In either case, the forger allows the simulator to solve a SIS instance.

In the fully anonymous variant of our scheme, the difficulty is to find a way to open adversarially-chosen signatures. This is achieved by implicitly using a "chosen-ciphertext-secure" variant of the signature encryption technique of Gordon *et* al. [GKV10]. While Camenisch *et* al. [CNR12] proceed in a similar way using Peikert's technique [Pei09], we use a much more economical method borrowed from the Agrawal *et al.* [ABB10a] identity-based cryptosystem. In our basic system, each $\mathbf{c}_i$ is of the form $\mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}[i] \cdot \mathbf{x}_2$, where p is an upper bound on $\mathbf{x}_2$'s coordinates, and can be decrypted using a short basis $\mathbf{S}_i$ such that $\mathbf{S}_i \cdot \mathbf{B}_i = \mathbf{0} \bmod q$. Our fully anonymous system replaces each $\mathbf{B}_i$ by a matrix $\mathbf{B}_{i,\mathsf{VK}}$ that depends on the verification key VK of a one-time signature. In the proof of full anonymity, the reduction will be able to compute a trapdoor for all matrices $\mathbf{B}_{i,\mathsf{VK}}$, except for one specific verification key $\mathsf{VK}^\star$ that will be used in the challenge phase. This will provide the reduction with a backdoor allowing it to open all adversarially-generated signatures.

7.1 An Asymptotically Shorter Lattice-Based Group Signature

At a high level, our key generation is based on the variant of Boyen's lattice signatures described in Section 3.2.5: Boyen's secret and verification keys respectively become our secret and public keys, whereas Boyen's message space is mapped to the users' identity space. There are however several additional twists in Keygen. First, each group member is given a *full* short basis of the public lattice associated to its identity, instead of a single short lattice vector. The reason is that, for anonymity and unlinkability purposes, the user has to generate each group signature using a *fresh* short lattice vector. Second, we sample our public key matrices $(\mathbf{A}_i)_{i \leq \ell}$ orthogonally to publicly known matrices $\mathbf{B}_i$, similarly to the group signature scheme from [GKV10]. These $\mathbf{B}_i$'s will be used to publicly verify the validity of the signatures. They are sampled along with short trapdoor bases, using algorithm $\mathsf{SuperSamp}$ (see Lemma 3.12), which become part of the group signature secret key. These trapdoor bases will be used by the group authority to open signatures.

To anonymously sign M, the user samples a Boyen signature $(\mathbf{x}_1, \mathbf{x}_2)$ with its identity as message, which is a temporary certificate of its group membership. It does so using its full trapdoor matrix for the corresponding lattice. The user then encrypts $\mathbf{x}_2$, in a fashion that resembles [GKV10], using Regev's dual encryption scheme from [GPV08, Se. 7.1] with the $\mathbf{B}_i$'s as encryption public keys. Note that in all cases but one ($\mathbf{c}_0$ at Step 2), the signature is not embedded in the encryption noise as in [GKV10], but as proper plaintext. The rest of the signing procedure consists in proving in zero-knowledge that these are valid ciphertexts and that the underlying plaintexts indeed encode a Boyen signature under the group public key. These ZKPoKs are all based on the interactive proof systems recalled in Sections 6.2.2 and 6.2.3. These was made non-interactive via the Fiat-Shamir heuristic with random oracle $H(\cdot)$ taking values in $\{0,1\}^\lambda$, with $\lambda = \Theta(n)$. The message M is embedded in the application of the Fiat-Shamir transform at Step 6 of the signing algorithm.

The verification algorithm merely consists in verifying all proofs of knowledge concerning the Boyen signature embedded in the plaintexts of the ciphertexts.

Finally, the group manager can open any signature by decrypting the ciphertexts (using the group manager secret key) and then recovering the underlying Boyen signature within the plaintexts: this reveals which public key matrices $\mathbf{A}_i$ have been considered by the signer, and therefore its identity.

The scheme depends on several functions m, q, p, α and σ of the security parameter n and the group size $N(=2^\ell)$. They are set so that all algorithms can be implemented in polynomial time and are correct (Theorem 7.2), and so that the security properties (Theorems 7.3 and 7.8) hold, in the ROM, under the SIS and LWE hardness assumptions for parameters for which these problems enjoy reductions from standard worst-case lattice problems with polynomial approximation factors. More precisely, we require that:

- parameter m is $\Omega(n \log q)$,
- parameter σ is $\Omega(m^{3/2}\sqrt{\ell n \log q}\log m)$ and $\leq n^{\mathcal{O}(1)}$,
- parameter p is $\Omega((\alpha q + \sigma)m^{3/2}n)$,
- parameter α is set so that $\alpha^{-1} \geq \Omega(pm^3 \log m)$ and $\leq n^{\mathcal{O}(1)}$,
- parameter q is prime and $\Omega(\ell + \alpha^{-1}\sqrt{n\ell})$ and $\leq n^{\mathcal{O}(1)}$.

For example, we may set $m = \widetilde{O}(n)$, $\sigma = \widetilde{O}(n^2\sqrt{\ell})$, $p = \widetilde{O}(n^{9/2}\sqrt{\ell})$ as well as $\alpha^{-1} = \widetilde{O}(n^{15/2}\sqrt{\ell})$ and $q = \widetilde{O}(\ell + n^8\sqrt{\ell})$. The scheme is described in Figures 7.1 and 7.2.

$\mathsf{Keygen}(1^n, 1^N)$: Given a security parameter $n > 0$ and the desired number of group members $N = 2^\ell \in \mathsf{poly}(n)$, choose parameters q, m, p, α and σ as specified above and make them public. Choose a hash function $H : \{0,1\}^* \to \{0,1\}^t$, for some $t = \Theta(n)$, which will be modelled as a random oracle in the security proof. Then, proceed as follows.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ (defined in Lemma 3.7) to get $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a short basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$.
2. For $i = 0$ to ℓ, sample $\mathbf{A}_i \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and compute $(\mathbf{B}_i, \mathbf{S}_i') \leftarrow \mathsf{SuperSamp}(\mathbf{A}_i, 0^{n \times n})$ (defined in Lemma 3.12). Then, randomize $\mathbf{S}_i'$ as $\mathbf{S}_i \leftarrow \mathsf{RandBasis}(\mathbf{S}_i', \Omega(\sqrt{mn \log q}\log m))$ (defined in Lemma 3.10).[2]
3. For $j = 0$ to $N - 1$, let $\mathrm{id}_j = \mathrm{id}_j[1] \ldots \mathrm{id}_j[\ell] \in \{0,1\}^\ell$ be the binary representation of id_j and define:
$$\mathbf{A}_{\mathrm{id}_j} = \left[\begin{array}{c} \mathbf{A} \\ \hline \mathbf{A}_0 + \sum_{i=1}^{\ell} \mathrm{id}_j[i]\mathbf{A}_i \end{array}\right] \in \mathbb{Z}_q^{2m \times n}.$$
Then, run $\mathbf{T}'_{\mathrm{id}_j} \leftarrow \mathsf{ExtBasis}(\mathbf{A}_{\mathrm{id}_j}, \mathbf{T}_\mathbf{A})$ to get a short delegated basis $\mathbf{T}'_{\mathrm{id}_j}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j})$. Finally, run $\mathbf{T}_{\mathrm{id}_j} \leftarrow \mathsf{RandBasis}(\mathbf{T}'_{\mathrm{id}_j}, \Omega(m\sqrt{\ell n \log q \log m}))$.[2] The j-th member's private key is $\mathsf{gsk}[j] := \mathbf{T}_{\mathrm{id}_j}$.
4. The group manager's private key is $\mathsf{gmsk} := \{\mathbf{S}_i\}_{i=0}^{\ell}$ and the group public key is defined to be $\mathsf{gpk} := \big(\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_{i=0}^{\ell}\big)$. The algorithm outputs $\big(\mathsf{gpk}, \mathsf{gmsk}, \{\mathsf{gsk}[j]\}_{j=0}^{N-1}\big)$.

Figure 7.1: Our group signature scheme: KeyGen.

Remark 7.1. The disjunction of two relations that can be proved by Σ-protocols can also be proved by a Σ-protocol [CDS94, Dam10].

All steps of the scheme above can be implemented in polynomial-time as a function of the security parameter n, assuming that $q \geq 2$ is prime, $m \geq \Omega(n \log q)$, $\sigma \geq \Omega(m^{3/2}\sqrt{\ell n \log q}\log m)$ (using Lemmas 1.24 and 3.10), and $\alpha q \geq \Omega(1)$ (using Lemma 1.24). Under some mild conditions

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)$: To sign a message $M \in \{0,1\}^*$ using the private key $\mathsf{gsk}[j] = \mathbf{T}_{\mathrm{id}_j}$, proceed as follows.

1. Run $\mathsf{GPVSample}(\mathbf{T}_{\mathrm{id}_j}, \sigma)$ to get $(\mathbf{x}_1 \| \mathbf{x}_2) \in \Lambda_q^{\perp}(\mathbf{A}_{\mathrm{id}_j})$ of norm $\leq \sigma\sqrt{2m}$.
2. Sample $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and encrypt $\mathbf{x}_2 \in \mathbb{Z}_q^m$ as $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$.
3. Sample $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$. For $i = 1$ to ℓ, sample $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$ and compute $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_j[i] \cdot \mathbf{x}_2$, which encrypts $\mathbf{x}_2 \in \mathbb{Z}_q^m$ (resp. $\mathbf{0}$) if $\mathrm{id}_j[i] = 1$ (resp. $\mathrm{id}_j[i] = 0$).
4. Generate a NIZKPoK π_0 of $\mathbf{s}_0$ so that $(\mathbf{B}_0, \mathbf{c}_0, \sqrt{2}\sigma/q; \mathbf{s}_0) \in R_{\mathsf{LWE}}$ (see Section 6.2.2).
5. For $i = 1$ to ℓ, generate a NIZKPoK $\pi_{\mathrm{OR},i}$ of $\mathbf{s}$ and $\mathbf{s}_0$ so that either:
   (i) $((\mathbf{B}_i | \mathbf{B}_0), p^{-1}(\mathbf{c}_i - \mathbf{c}_0), \sqrt{2}\alpha; (\mathbf{s} \| - \mathbf{s}_0)) \in R_{\mathsf{LWE}}$ (the vectors $\mathbf{c}_i$ and $\mathbf{c}_0$ encrypt the same $\mathbf{x}_2$, so that $p^{-1}(\mathbf{c}_i - \mathbf{c}_0)$ is close to the $\mathbb{Z}_q$-span of $(\mathbf{B}_i | \mathbf{B}_0)$);
   (ii) or $(\mathbf{B}_i, p^{-1}\mathbf{c}_i, \alpha; \mathbf{s}) \in R_{\mathsf{LWE}}$ (the vector $\mathbf{c}_i$ encrypts $\mathbf{0}$, so that $p^{-1}\mathbf{c}_i$ is close to the $\mathbb{Z}_q$-span of $\mathbf{B}_i$).

   This can be achieved by OR-ing two proofs for R_{LWE}, and making the resulting protocol non-interactive with the Fiat-Shamir heuristic (see Remark 7.1).
6. For $i = 1$ to ℓ, set $\mathbf{y}_i = \mathrm{id}_j[i]\mathbf{x}_2 \in \mathbb{Z}^m$ and generate a NIZKPoK π_K of $\{\mathbf{e}_i\}_{i=0}^{\ell}, \{\mathbf{y}_i\}_{i=0}^{\ell}, \mathbf{x}_1$ such that,

$$\mathbf{x}_1^T \mathbf{A} + \sum_{i=0}^{\ell} \mathbf{c}_i^T \mathbf{A}_i = \sum_{i=1}^{\ell} \mathbf{e}_i^T (p\mathbf{A}_i) \tag{7.2}$$

$$\mathbf{e}_i^T (p\mathbf{A}_i) + \mathbf{y}_i^T \mathbf{A}_i = \mathbf{c}_i^T \mathbf{A}_i, \quad \text{for } i \in \{1, \ldots, \ell\} \tag{7.3}$$

   with $\|\mathbf{e}_i\|, \|\mathbf{y}_i\|, \|\mathbf{x}_1\| \leq \max(\sigma, \alpha q)\sqrt{m}$ for all i.
   This is achieved using $\mathsf{Prove}_{\mathsf{ISIS}}$ in order to produce a triple $(\mathsf{Comm}_K, \mathsf{Chall}_K, \mathsf{Resp}_K)$, where $\mathsf{Chall}_K = H(M, \mathsf{Comm}_K, \{\mathbf{c}_i\}_{i=0}^{\ell}, \pi_0, \{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell})$.

The signature consists of

$$\Sigma = \left(\{\mathbf{c}_i\}_{i=0}^{\ell}, \pi_0, \{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}, \pi_K\right). \tag{7.4}$$

$\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$: Parse Σ as in (7.4). Then, return 1 if $\pi_0, \{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}, \pi_K$ properly verify. Else, return 0.

$\mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, M, \Sigma)$: Parse gmsk as $\{\mathbf{S}_i\}_{i=0}^{\ell}$ and Σ as in (7.4). Compute $\mathbf{x}_2$ by decrypting $\mathbf{c}_0$ using $\mathbf{S}_0$. For $i = 1$ to ℓ, use $\mathbf{S}_i$ to determine which one of the vectors $p^{-1}\mathbf{c}_i$ and $p^{-1}(\mathbf{c}_i - \mathbf{x}_2)$ is close to the $\mathbb{Z}_q$-span of $\mathbf{B}_i$. Set $\mathrm{id}[i] = 0$ in the former case and $\mathrm{id}[i] = 1$ in the latter. Eventually, output $\mathrm{id} = \mathrm{id}[1] \ldots \mathrm{id}[\ell]$.

Figure 7.2: Sign, Verify and Open.

on the parameters, the scheme above is correct, i.e., the verifier accepts honestly generated signatures, and the group manager successfully opens honestly generated signatures. In particular, multiplying the ciphertexts by the $\mathbf{S}_i$ modulo q should reveal $p \cdot \mathbf{e}_i + \mathrm{id}_j[i] \cdot \mathbf{x}_2$ over the integers, and $\|\mathrm{id}_j[i] \cdot \mathbf{x}_2\|_\infty$ should be smaller than p.

Theorem 7.2. *Let us assume that $q \geq 2$ is prime and that we have $m \geq \Omega(n \log q)$, $\sigma \geq \Omega(m^{3/2}\sqrt{\ell n \log q}\log m)$, $\alpha^{-1} \geq \Omega(pm^{5/2}\log m\sqrt{n \log q})$ as well as $q \geq \Omega(\alpha^{-1}+\sigma m^{5/2}\log m\sqrt{n \log q})$. Then, the group signature scheme above can be implemented in time polynomial in n, is correct, and the bit-size of the generated signatures in $\mathcal{O}(\ell n m \log q)$.*

Proof. Setting $m = \Omega(n \log q)$ allows us to use algorithms TrapGen and SuperSamp from Lemmas 3.7 and 3.12, at Steps 1 and 2 of algorithm Keygen. Also, the rows of the matrix $\mathbf{A}$ sampled at Step 1 span $\mathbb{Z}_q^n$ with probability $\geq 1 - 2^{-\Omega(n)}$. At Steps 2 and 3, the second inputs to the calls to RandBasis are sufficiently large for the assumption of Lemma 3.10 to hold (note that in the second case, it is much larger than needed, but this choice is important for the simulation in the traceability proof). At the end of the execution of Keygen, we have $\|\widetilde{\mathbf{S}_i}\| \leq \mathcal{O}(m \log m\sqrt{n \log q})$ for all $i \in \{0, \ldots, \ell\}$ and $\|\widetilde{\mathbf{T}_{\mathrm{id}_j}}\| \leq \mathcal{O}(m^{3/2}\sqrt{\ell n \log q \log m})$ for all $j \in \{0, \ldots, N-1\}$.

At Step 1 of algorithm Sign, the parameter σ is sufficiently large for applying Lemma 1.24 and obtain a distribution within statistical distance $2^{-\Omega(n)}$ from $D_{\Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j}),\sigma}$. The same holds for all $\mathbf{e}_i$'s of Step 3.

Correctness of algorithm Verify follows from the completeness property of the underlying proof systems. Now, consider algorithm Open. We have $\mathbf{S}_0 \cdot \mathbf{c}_0 = \mathbf{S}_0 \cdot \mathbf{x}_2 \mod q$. But on the other hand $\|\mathbf{S}_0 \cdot \mathbf{x}_2\| \leq \sqrt{m}\|\mathbf{S}_0\|\|\mathbf{x}_2\| \leq m\|\widetilde{\mathbf{S}_0}\|\|\mathbf{x}_2\|$, which is itself $\mathcal{O}(\sigma m^{3/2} n \log m\sqrt{n \log q})$ with probability $\geq 1 - 2^{-\Omega(n)}$, by Lemma 1.36. As q has been set sufficiently large, we obtain that $\mathbf{S}_0 \cdot \mathbf{x}_2$ is known over the integers: Multiplying by $\mathbf{S}_0^{-1}$ over the rationals allows the group manager to recover $\mathbf{x}_2$. The argument is similar for the other $\mathbf{c}_i$'s, except that $\|\mathbf{S}_i \cdot \mathbf{c}_i \bmod q\| \leq \mathcal{O}(p\alpha q m^{3/2} n \log m\sqrt{n \log q})$. Again, α has been set sufficiently small to allow the group manager to recover $p \cdot \mathbf{e}_i + \mathrm{id}_j[i] \cdot \mathbf{x}_2$.

Finally, the total bit-size of all proofs is $\mathcal{O}(\ell n m \log q)$. The same bound holds for the ciphertexts. □

7.2 Security

We now focus on the security of the scheme described in Section 7.1.

7.2.1 Anonymity

Like in [GKV10, BBS04], we use a relaxation of the anonymity definition, called weak anonymity and recalled in Definition 6.7. Analogously to the notion of IND-CPA security for public-key encryption, the adversary does not have access to a signature opening oracle. We show that the two versions (for $b = 0, 1$) of the anonymity security experiment recalled in Section 6.3.2 are indistinguishable under the LWE assumption. We use several intermediate hybrid experiments called $G_b^{(i)}$, and show that each of these experiments is indistinguishable from the next one. At each step, we only change one element of the game (highlighted by an arrow in Figure 7.3), to finally reach the experiment $G^{(4)}$ where the signature scheme does not depend on the identity of the user anymore.

Theorem 7.3. *In the random oracle model, the scheme provides weak anonymity in the sense of Definition 6.7 under the $\mathsf{LWE}_{q,\alpha}$ assumption. Namely, for any PPT adversary $\mathcal{A}$ with advantage ε, there exists an algorithm $\mathcal{B}$ solving the $\mathsf{LWE}_{q,\alpha}$ problem with advantage at most $2^{-\Omega(n)}$ smaller.*

Proof. We define by G_0 the experiment of Definition 6.7 with $b = 0$ and by G_1 the same experiment with $b = 1$. To show the anonymity of the scheme, we prove that G_0 and G_1 are indistinguishable. We use several hybrid experiments named $\mathrm{G}_b^{(1)}$, $\mathrm{G}_b^{(2)}$, $\mathrm{G}_b^{(3)}$ and $G^{(4)}$ (described in Figure 7.3), where b is either 0 or 1.

Lemma 7.4. *For each $b \in \{0,1\}$, G_b and $G_b^{(1)}$ are statistically indistinguishable.*

We only change the way we generate $(\mathbf{x}_1 \| \mathbf{x}_2)$, by using the fact that one way to generate it is to first sample $\mathbf{x}_2$ from $D_{\mathbb{Z}^m,\sigma}$ and then generate $\mathbf{x}_1$ from $D_{\mathbb{Z}^m,\sigma}$ such that $(\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = 0 \bmod q$ (by using the trapdoor $\mathbf{T}_\mathbf{A}$). This change is purely conceptual and the vector $(\mathbf{x}_1 \| \mathbf{x}_2)$ has the same distribution anyway. The two experiments are thus identical from $\mathcal{A}$'s view and $\mathbf{x}_2$ is chosen independently of the signer's identity in the challenge phase.

Lemma 7.5. *For each $b \in \{0,1\}$, $G_b^{(1)}$ and $G_b^{(2)}$ are statistically indistinguishable.*

The differences are simply: Instead of generating the proofs $\{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}$ and π_K using the witnesses, we simulate them (see Section 6.2.2).

Lemma 7.6. *For each $b \in \{0,1\}$, if the $\mathsf{LWE}_{q,\alpha}$ problem is hard, then the experiments $G_b^{(2)}$ and $G_b^{(3)}$ are computationally indistinguishable.*

Proof. This proof uses the same principle as the proof of [GKV10, Claim 1]: We use the adversary $\mathcal{A}$ to construct a PPT algorithm $\mathcal{B}$ for the $\mathsf{LWE}_{q,\alpha}$ problem. We consider an LWE instance $(\mathbf{B}', \mathbf{z}) \in \mathbb{Z}_q^{m\ell \times (n+1)}$ such that $\mathbf{B}' = (\mathbf{B}'_1, \ldots, \mathbf{B}'_\ell)$ and $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_\ell)$ with $\mathbf{B}'_i \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{z}_i \in \mathbb{Z}_q^m$. The component $\mathbf{z}$ is either uniform in $\mathbb{Z}_q^{m\ell}$, or of the form $\mathbf{z} = \mathbf{B}' \cdot \mathbf{s} + \mathbf{e}$ where $\mathbf{e}$ is sampled from $D_{\mathbb{Z}^{m\ell},\alpha q}$ and $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^{n+1})$.

We construct a modified Keygen algorithm using this LWE instance: It generates the matrix $\mathbf{A}$ with a basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$. Instead of generating the $\mathbf{B}_i$'s genuinely, we pick $\mathbf{B}_0$ uniformly in $\mathbb{Z}_q^{m \times n}$ and set $\mathbf{B}_i = \mathbf{B}'_i$ for $1 \le i \le \ell$. For $0 \le i \le \ell$, we compute $(\mathbf{A}_i, \mathbf{T}_i) \leftarrow \mathsf{SuperSamp}(\mathbf{B}_i, \mathbf{0})$. Then, for each $1 \le j \le N-1$, we define $\mathbf{A}_{\mathrm{id}_j}$ as in the original Keygen algorithm, and compute a trapdoor $\mathbf{T}_{\mathrm{id}_j}$ using $\mathbf{T}_\mathbf{A}$. The adversary $\mathcal{A}$ is given gpk and $\{\mathsf{gsk}_j\}_j$. In the challenge phase, it outputs j_0, j_1 and a message M. By [GKV10], this Keygen algorithm and the one in all the experiments are statistically indistinguishable. Then, the signature is created on behalf of the group member j_b. Namely, $\mathcal{B}$ first chooses $\mathbf{x}_2 \leftarrow D_{\mathbb{Z}^m,\sigma}$ and finds $\mathbf{x}_1$ such that $(\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \bmod q$. Then it chooses $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and computes $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$. Third, it computes $\mathbf{c}_i = p \cdot \mathbf{z}_i + \mathrm{id}_{j_b}[i] \cdot x_2$ (with the $\mathbf{z}_i$ of the LWE instance). Then it generates π_0 and simulates the $\pi_{\mathrm{OR},i}$'s and π_K proofs.

We let $\mathcal{D}_{\mathsf{LWE}}$ denote this experiment when $\mathbf{z} = \mathbf{B}' \cdot \mathbf{s} + \mathbf{e}$: This experiment is statistically close to $\mathrm{G}_b^{(2)}$. Then, we let $\mathcal{D}_{rand}$ denote this experiment when $\mathbf{z}$ is uniform: It is statistically close to $\mathrm{G}_b^{(3)}$. As a consequence, if the adversary $\mathcal{A}$ can distinguish between the experiments $\mathrm{G}_b^{(2)}$ and $\mathrm{G}_b^{(3)}$ with some advantage, then we can solve the $\mathsf{LWE}_{q,\alpha}$ problem with advantage at most $2^{-\Omega(n)}$ smaller.

□

Lemma 7.7. *For each $b \in \{0,1\}$, $G_b^{(3)}$ and $G^{(4)}$ are indistinguishable.*

Between these two experiments, we change the first and third steps. In the former, we no longer generate $\mathbf{x}_1$ and, in the latter, $\mathbf{c}_i$ is uniformly sampled in $\mathbb{Z}_q^m$. These changes are purely conceptual. Indeed, in experiment $G_b^{(3)}$, the vector $\mathbf{x}_1$ is not used beyond Step 1. In the same experiment, we also have $\mathbf{c}_i = \mathbf{z}_i + \mathrm{id}_{j_b}[i]$. Since the $\mathbf{z}_i$'s are uniformly sampled in $\mathbb{Z}_q^m$, the $\mathbf{c}_i$'s are also uniformly distributed in $\mathbb{Z}_q^m$. As a consequence, the $\mathbf{c}_i$'s of $G_b^{(3)}$ and the $\mathbf{c}_i$'s of $G^{(4)}$ have

Experiment $\mathbf{G}_b$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  1. $(\mathbf{x}_1 \| \mathbf{x}_2) \leftarrow \mathsf{GPVSample}(\mathbf{T}_{\mathrm{id}_{j_b}}, \sigma)$; we have $(\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$, compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$.
  3. Choose $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, and for $i = 1$ to ℓ, choose $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$ and compute $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_{j_b}[i] \cdot \mathbf{x}_2$.
  4. Generate π_0.
  5. Generate $\{\pi_{\mathrm{OR},i}\}_i$.
  6. Generate π_K.

Experiment $\mathbf{G}_b^{(1)}$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  → 1. Sample $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m, \sigma}$ and, using $\mathbf{T}_{\mathbf{A}}$, sample $\mathbf{x}_1 \hookleftarrow D_{\mathbb{Z}^m, \sigma}$ conditioned on $(\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$, compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$,
  3. Choose $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, and for $i = 1$ to ℓ, choose $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$ and compute $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_{j_b}[i] \cdot \mathbf{x}_2$.
  4. Generate π_0.
  5. Generate $\{\pi_{\mathrm{OR},i}\}_i$.
  6. Generate π_K.

Experiment $\mathbf{G}_b^{(2)}$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  1. Sample $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m, \sigma}$; sample $\mathbf{x}_1 \hookleftarrow D_{\mathbb{Z}^m, \sigma}$, conditioned on $(\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$,
  3. Choose $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, and for $i = 1$ to ℓ, choose $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$ and compute $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_{j_b}[i] \cdot \mathbf{x}_2$.
  4. Generate π_0.
  → 5. Simulate $\{\pi_{\mathrm{OR},i}\}_i$.
  → 6. Simulate π_K.

Experiment $\mathbf{G}_b^{(3)}$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  1. Sample $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m, \sigma}$ Sample $\mathbf{x}_1 \hookleftarrow D_{\mathbb{Z}^m, \sigma}$ conditioned on $(\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \mathbf{A}_{\mathrm{id}_{j_b}} = \mathbf{0} \mod q$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$,
  → 3. For $i = 1$ to ℓ, choose $\mathbf{z}_i \hookleftarrow U(\mathbb{Z}_q^m)$ and compute $\mathbf{c}_i = \mathbf{z}_i + \mathrm{id}_{j_b}[i] \cdot \mathbf{x}_2$.
  4. Generate π_0.
  5. Simulate $\{\pi_{\mathrm{OR},i}\}_i$.
  6. Simulate π_K.

Experiment $\mathbf{G}^{(4)}$

- Run Keygen; give $\mathsf{gpk} = (\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_i)$ and $\mathsf{gsk} = \{\mathbf{T}_{id_j}\}_j$ to $\mathcal{A}$.
- $\mathcal{A}$ outputs j_0, j_1 and a message M.
- The signature of user j_b is computed as follows:
  → 1. Sample $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m, \sigma}$.
  2. Choose $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and compute $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^m$,
  → 3. For $i = 1$ to ℓ, choose $\mathbf{z}_i \hookleftarrow U(\mathbb{Z}_q^m)$ and set $\mathbf{c}_i = \mathbf{z}_i$.
  4. Generate π_0.
  5. Simulate $\{\pi_{\mathrm{OR},i}\}_i$.
  6. Simulate π_K.

Figure 7.3: Experiments G_b, $\mathrm{G}_b^{(1)}$, $\mathrm{G}_b^{(2)}$,$\mathrm{G}_b^{(3)}$ and $\mathrm{G}^{(4)}$.

the same distribution. In $G_b^{(4)}$, we conclude that $\mathcal{A}$'s view is exactly the same as in experiments $G_b^{(3)}$. □

Since the experiment $G^{(4)}$ no longer depends on the bit $b \in \{0,1\}$ that determines the signer's identity, the announced result follows.

7.2.2 Traceability

The proof of traceability relies on the technique of [ABB10a, Boy10] and a refinement from [HW09, MP12], which is used in order to allow for a smaller modulus q.

A difference with the proof of [GKV10] is that we need to rely on the knowledge extractor of a proof of knowledge π_K. We distinguish two cases, depending on whether the extracted witnesses $\{\mathbf{e}_i, \mathbf{y}_i\}_{i=1}^{\ell}$ of relation (7.3) satisfy $\mathbf{y}_i = \mathrm{id}_j[i]\mathbf{x}_2$ for all i or not. The strategy of the reduction and the way it uses its given $\mathsf{SIS}_{m,q,\beta}$ instance will depend on which case is expected to occur.

Theorem 7.8. *Assume that* $q > \log N$, $m \geq \Omega(n \log q)$, $p \geq \Omega((\alpha q + \sigma)m^{3/2}n)$ *and* $\beta \geq \Omega(\sigma m^{5/2}n\sqrt{\log N} + p\alpha q m^{3/2}n)$. *Then for any* PPT *traceability adversary* $\mathcal{A}$ *with success probability* ε, *there exists a* PPT *algorithm* $\mathcal{B}$ *solving* $\mathsf{SIS}_{m,q,\beta}$ *with probability* $\varepsilon'' \geq \frac{\varepsilon'}{2N} \cdot (\frac{\varepsilon'}{q_H} - 2^{-\lambda}) + \frac{\varepsilon'}{2\log N}$, *where* $\varepsilon' = \varepsilon - 2^{-\lambda} - 2^{-\Omega(n)}$ *and* q_H *is the number of queries to the random oracle* $H : \{0,1\}^* \to \{0,1\}^{\lambda}$.

Proof. Let $\mathcal{A}$ be a PPT adversary that can defeat the traceability of the scheme with non-negligible success probability ε in the game of Definition 6.8. We construct a PPT algorithm $\mathcal{B}$ that emulates $\mathcal{A}$'s challenger and attacks $\mathsf{SIS}_{m,q,\beta}$: It takes as input $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m \times n}$ with the task of finding $\mathbf{v} \in \Lambda_q^{\perp}(\bar{\mathbf{A}})$ with $0 < \|\mathbf{v}\| \leq \beta$.

Initialization. Before starting its interaction with $\mathcal{A}$, algorithm $\mathcal{B}$ samples $coin \hookleftarrow U(\{0,1\})$. It also samples $j^\star \hookleftarrow U(\{0,\ldots,N-1\})$, a guess that $\mathcal{A}$'s forgery will open to user $j^\star$. Depending on $coin$, the group public key is prepared in two different ways.

• If $coin = 0$, algorithm $\mathcal{B}$ first calls $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain $\mathbf{C} \in \mathbb{Z}_q^{m \times n}$ and a basis $\mathbf{T_C}$ of $\Lambda_q^{\perp}(\mathbf{C})$ with $\|\widetilde{\mathbf{T_C}}\| \leq \mathcal{O}(\sqrt{n \log q})$. Then, it samples $\ell+1$ matrices $\mathbf{Q}_k \in \mathbb{Z}^{m \times m}$, with each matrix entry sampled independently from $D_{\mathbb{Z},\sqrt{m}}$ (as in [Boy10, Th. 25], with a larger standard deviation to get exponentially small statistical distances later on).. Let $\mathrm{id}_{j^\star} = \mathrm{id}_{j^\star}[1] \ldots \mathrm{id}_{j^\star}[\ell] \in \{0,1\}^{\ell}$ denote the binary expansion of $\mathrm{id}_{j^\star}$. The reduction $\mathcal{B}$ defines the matrices $\{\mathbf{A}_i\}_{i=0}^{\ell}$ as

$$\begin{cases} \mathbf{A}_0 = \mathbf{Q}_0 \cdot \bar{\mathbf{A}} + (\sum_{i=1}^{\ell} \mathrm{id}_{j^\star}[i]) \cdot \mathbf{C} \\ \mathbf{A}_i = \mathbf{Q}_i \cdot \bar{\mathbf{A}} + (-1)^{\mathrm{id}_{j^\star}[i]} \cdot \mathbf{C}, \quad \text{for } i \in \{1,\ldots,\ell\}. \end{cases}$$

It also sets $\mathbf{A} = \bar{\mathbf{A}}$. Next, it runs $\mathsf{SuperSamp}(\mathbf{A}_i, \mathbf{0})$ to obtain $\mathbf{B}_i \in \mathbb{Z}_q^{m \times n}$ along with short bases $\mathbf{S}_i'$ of $\Lambda_q^{\perp}(\mathbf{B}_i)$, and then computes $\mathbf{S}_i \leftarrow \mathsf{RandBasis}(\mathbf{S}_i', \Omega(\sqrt{mn \log q} \log m))$, as in Step 2 of the genuine key generation algorithm. The group public key $\mathsf{gpk} = \big(\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_{i=0}^{\ell}\big)$ is finally given to $\mathcal{A}$.

We note that, for each $j \neq j^\star$, we have $\mathbf{A}_{\mathrm{id}_j}$ equals to

$$\left[\frac{\bar{\mathbf{A}}}{\mathbf{A}_0 + \sum_{i=1}^{\ell} \mathrm{id}_j[i]\mathbf{A}_i}\right] = \left[\frac{\bar{\mathbf{A}}}{(\mathbf{Q}_0 + \sum_{i=1}^{\ell} \mathrm{id}_j[i]\mathbf{Q}_i) \cdot \bar{\mathbf{A}} + (\sum_{i=1}^{\ell} \mathrm{id}_{j^\star}[i] + (-1)^{\mathrm{id}_{j^\star}[i]}\mathrm{id}_j[i]) \cdot \mathbf{C}}\right]$$
$$= \left[\frac{\bar{\mathbf{A}}}{(\mathbf{Q}_0 + \sum_{i=1}^{\ell} \mathrm{id}_j[i]\mathbf{Q}_i) \cdot \bar{\mathbf{A}} + h_{\mathrm{id}_j} \cdot \mathbf{C}}\right]$$

where $h_{\mathrm{id}_j} \in \{1, \ldots, \ell\}$ stands for the Hamming distance between the identifiers id_j and $\mathrm{id}_{j^\star}$. Since $q > \ell$, we have $h_{\mathrm{id}_j} \neq 0 \bmod q$ whenever $\mathrm{id}_j \neq \mathrm{id}_{j^\star}$, so that algorithm $\mathcal{B}$ is able to compute (see [ABB10a, Se. 4.2], using the basis $\mathbf{T_C}$ of $\Lambda_q^\perp(\mathbf{C})$ and the refined $\mathsf{GPVSample}$ of Lemma 1.24) a basis $\mathbf{T}'_{\mathrm{id}_j}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j})$ with $\|\widetilde{\mathbf{T}'_{\mathrm{id}_j}}\| \leq \Omega(m\sqrt{\ell n \log q})$. Then algorithm $\mathcal{B}$ runs $\mathbf{T}_{\mathrm{id}_j} \leftarrow \mathsf{RandBasis}(\mathbf{T}'_{\mathrm{id}_j}, \Omega(m\sqrt{\ell n \log q \log m}))$. Algorithm $\mathcal{B}$ is thus able to compute a trapdoor $\mathbf{T}_{\mathrm{id}_j}$ for each $j \neq j^\star$. In contrast, algorithm $\mathcal{B}$ lacks a trapdoor for $\mathbf{A}_{\mathrm{id}_{j^\star}}$ as the latter only depends on $\mathbf{A}$ and $\{\mathbf{Q}_k\}_{k=0}^{\ell}$.

Observe that since the rows of the $\mathbf{Q}_k$'s are sampled from $D_{\mathbb{Z}^m,\sqrt{m}}$, the matrices $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell$ are within statistical distance $2^{-\Omega(m)}$ of $U(\mathbb{Z}_q^{m\times n})$ (this is a consequence of [GPV08, Le. 5.2]). Further, by Lemma 3.10, the distribution of the $\mathbf{T}_{\mathrm{id}_j}$'s generated by $\mathcal{B}$ is statistically close to that of the real scheme.

• If $coin = 1$, algorithm $\mathcal{B}$ samples $i^\star \hookleftarrow U(\{1, \ldots, \ell\})$ and embeds its $\mathsf{SIS}_{m,q,\beta}$ instance in the matrix $\mathbf{A}_{i^\star}$ that will be part of gpk. It calls $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ and a basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$ with $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$. Next, it independently samples $\mathbf{A}_j \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ for $j \neq i^\star \in \{0, \ldots, \ell\}$ and defines $\mathbf{A}_{i^\star} = \bar{\mathbf{A}}$. Then, algorithm $\mathcal{B}$ computes $(\mathbf{B}_i, \mathbf{S}'_i) \leftarrow \mathsf{SuperSamp}(\mathbf{A}_i, \mathbf{0})$ and $\mathbf{S}_i \leftarrow \mathsf{RandBasis}(\mathbf{S}'_i, \Omega(\sqrt{mn \log q} \log m))$, as in Step 2 of Keygen. The group public key $\mathsf{gpk} = \big(\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_{i=0}^{\ell}\big)$, which is distributed as in the real scheme, is given to the adversary $\mathcal{A}$. Since it knows $\mathbf{T_A}$, algorithm $\mathcal{B}$ is able to sample a trapdoor $\mathbf{T}_{\mathrm{id}_j}$ for all users, with exactly the same distribution as in the real scheme.

In either case, $\mathcal{B}$ runs the adversary $\mathcal{A}$ on inputs $\mathsf{gpk} = \big(\mathbf{A}, \{\mathbf{A}_i, \mathbf{B}_i\}_{i=0}^{\ell}\big)$ and $\mathsf{gmsk} = \{\mathbf{S}_i\}_{i=0}^{\ell}$.

Queries. Algorithm $\mathcal{B}$ then starts interacting with $\mathcal{A}$ and handles $\mathcal{A}$'s queries depending on *coin*.

• If $coin = 0$, it aborts in the event that $\mathcal{A}$ queries the unavailable secret key $\mathsf{gsk}[j^\star]$. When $\mathcal{A}$ queries a secret key $\mathsf{gsk}[j]$ for $j \neq j^\star$, algorithm $\mathcal{B}$ reveals the short basis $\mathbf{T}_{\mathrm{id}_j}$ that was computed in the initialization phase. When it comes to answer signing queries, algorithm $\mathcal{B}$ faithfully runs the signing algorithm whenever the involved user j differs from $j^\star$. As for signing queries involving the expected target user $j^\star$, the reduction $\mathcal{B}$ samples $\mathbf{s}_0, \mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ and $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ for each $i \in \{1, \ldots, \ell\}$. It then computes $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2$ as well as $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}_i + \mathrm{id}_{j^\star}[i]\mathbf{x}_2$ for each $i \in \{1, \ldots, \ell\}$. The proof π_0 is then generated using the actual witness $\mathbf{x}_2$ whereas the other non-interactive proofs $\{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}$ and π_K are simulated (exactly as in experiment $G_b^{(2)}$ in the proof of anonymity). By the statistical zero-knowledge property of the simulator, the signature Σ will be statistically indistinguishable from a genuine signature.

• If $coin = 1$, algorithm $\mathcal{B}$ knows $\mathbf{T_A}$ and can answer $\mathcal{A}$'s queries by running the real signing algorithm or returning the queried secret keys $\mathsf{gsk}[j]$ (all of which are available).

Regardless of the value of *coin*, queries to the random oracle H are handled by returning a uniformly chosen value in $\{0,1\}^\lambda$. For each $\kappa \leq q_H$, we let r_κ denote the answer to the κ-th H-query. Of course, if the adversary makes a given query more than once, then $\mathcal{B}$ consistently returns the previously defined value.

Forgery. When $\mathcal{A}$ terminates, it outputs a signature $\Sigma^\star = \big(\{\mathbf{c}_i^\star\}_{i=0}^{\ell}, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^{\ell}, \pi_K^\star\big)$ on some message $M^\star$ with probability $\geq \varepsilon - 2^{-\Omega(n)}$. If we parse $\pi_K^\star$ as $(\mathsf{Comm}_K^\star, \mathsf{Chall}_K^\star, \mathsf{Resp}_K^\star)$, with overwhelming probability, the adversary $\mathcal{A}$ must have queried H on the following input: $(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^{\ell}, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^{\ell})$. Indeed, otherwise, the probability to have the equality $\mathsf{Chall}_K^\star = H(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^{\ell}, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^{\ell})$ is at most $2^{-\lambda}$. With probability $\geq \varepsilon' := \varepsilon - 2^{-\lambda} - 2^{-\Omega(n)}$, the tuple $(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^{\ell}, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^{\ell})$ thus coincides with the $\kappa^\star$-th hash query for some $\kappa^\star \leq q_H$.

At this stage, the reduction $\mathcal{B}$ runs a second execution of the adversary $\mathcal{A}$ with the *same* random tape and input as in the original execution. All queries are answered as previously with only one difference in the treatment of random oracle queries. Namely, the first $\kappa^\star - 1$ hash queries – which are identical to those of the first execution since $\mathcal{A}$ is run with the same random tape as before – receive the same answers $r_1, \ldots, r_{\kappa^\star - 1}$ as in the initial run. This implies that the $\kappa^\star$-th query will involve the tuple $(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell)$ as in the first execution. However, from the $\kappa^\star$-th query onwards, $\mathcal{A}$ obtains fresh random oracle values $r'_{\kappa^\star}, \ldots, r'_{q_H}$ which depart from the sequence of answers in the first execution. The General Forking Lemma of [BN06] implies that, with probability $\geq \varepsilon'(\varepsilon'/q_H - 2^{-\lambda})$, $\mathcal{A}$'s forgery also involves $(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell)$ in the second run and we also have $r'_{\kappa^\star} \neq r_{\kappa^\star}$. In this case, using $\mathsf{Extract}$, algorithm $\mathcal{B}$ can obtain vectors $\mathbf{e}_1, \ldots, \mathbf{e}_\ell, \mathbf{x}_1, \mathbf{y}_1, \ldots, \mathbf{y}_\ell \in \mathbb{Z}^m$ satisfying

$$\mathbf{x}_1^T\mathbf{A} + \sum_{i=0}^{\ell} \mathbf{c}_i^T\mathbf{A}_i = \sum_{i=1}^{\ell} \mathbf{e}_i^T(p\mathbf{A}_i) \quad \text{and} \quad \mathbf{e}_i^T\big(p\mathbf{A}_i\big) + \mathbf{y}_i^T\mathbf{A}_i = \mathbf{c}_i^T\mathbf{A}_i \qquad \text{for } i \in \{1, \ldots, \ell\} \tag{7.5}$$

with $\|\mathbf{e}_i\|, \|\mathbf{y}_i\|, \|\mathbf{x}_1\| \leq \mathcal{O}((\alpha q + \sigma)m^{3/2}n)$ for all $i \in \{1, \ldots, \ell\}$ (see Section 6.2.2).

The reduction $\mathcal{B}$ then opens one of the two forgeries using $\{\mathbf{S}_i\}_{i=0}^\ell$ (note that both signatures necessarily open to the same identity id). At this point, $\mathcal{B}$ aborts and reports failure if the opening algorithm does not point to user $j^\star$. However, with probability $\geq 1/N$, $\mathcal{B}$'s initial choice for $j^\star$ turns out to be correct and the opening algorithm reveals $\mathrm{id}_{j^\star}$.

We now assume that $\Sigma^\star$ indeed traces to user $j^\star$. We let $\mathbf{x}_2 \in \mathbb{Z}^m$ denote the vector obtained by decrypting $\mathbf{c}_0^\star$ using $\mathbf{S}_0$. Algorithm $\mathcal{B}$ considers the following two situations:

• If $\mathbf{y}_i = \mathrm{id}_{j^\star}[i]\mathbf{x}_2$ for all $i \in \{1, \ldots, \ell\}$, then $\mathcal{B}$ aborts if $coin = 1$ and continues if $coin = 0$. The relations (7.5) and the fact that $\mathbf{c}_0^\star$ is of the form $\mathbf{c}_0^\star = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2 \mod q$ with $\mathbf{B}_0^T \cdot \mathbf{A}_0 = \mathbf{0} \mod q$ imply that (modulo q):

$$\begin{aligned} \mathbf{0} = \mathbf{x}_1^T\mathbf{A} + \mathbf{c}_0^{\star T}\mathbf{A}_0 + \sum_{i=1}^{\ell} \mathrm{id}_{j^\star}[i]\mathbf{x}_2^T \cdot \mathbf{A}_i &= (\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \left[\frac{\mathbf{A}}{\mathbf{A}_0 + \sum_{i=1}^{\ell} \mathrm{id}_{j^\star}[i]\mathbf{A}_i}\right] \\ &= (\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \left[\frac{\bar{\mathbf{A}}}{(\mathbf{Q}_0 + \sum_{i=1}^{\ell} \mathrm{id}_{j^\star}[i]\mathbf{Q}_i) \cdot \bar{\mathbf{A}}}\right], \end{aligned}$$

by construction of the matrices $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell$. It comes that $\mathbf{v}^T = \mathbf{x}_1^T + \mathbf{x}_2^T \cdot \big(\mathbf{Q}_0 + \sum_{i=1}^{\ell} \mathrm{id}_{j^\star}[i]\mathbf{Q}_i\big) \in \Lambda^\perp(\bar{\mathbf{A}})$. The same analysis as in [Boy10] shows that $0 < \|\mathbf{v}\| \leq \mathcal{O}((\alpha q + \sigma)m^{5/2}n\sqrt{\ell})$ holds with probability $1 - 2^{-\Omega(m)}$.

• If there exists $i \in \{1, \ldots, \ell\}$ such that $\mathbf{y}_i \neq \mathrm{id}_{j^\star}[i]\mathbf{x}_2$, then $\mathcal{B}$ aborts if $coin = 0$ and continues if $coin = 1$. The non-interactive proofs π_0 and $\pi_{\mathrm{OR},i}$ imply that $\mathbf{c}_i = \mathbf{B}_i \cdot \mathbf{s} + p\mathbf{e}'_i + \mathrm{id}_{j^\star}[i]\mathbf{x}_2 \mod q$ for some $\mathbf{s}_0, \mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{x}_2, \mathbf{e}'_i \in \mathbb{Z}^m$ such that $\|\mathbf{x}_2\| \leq \mathcal{O}(\sigma m^{3/2}n)$ and $\|\mathbf{e}'_i\| \leq \mathcal{O}(\alpha q m^{3/2}n)$. If we multiply $\mathbf{c}_i^T$ by $\mathbf{A}_i$, we find

$$\mathbf{c}_i^T\mathbf{A}_i = p\mathbf{e}_i'^T \cdot \mathbf{A}_i + \mathrm{id}_{j^\star}[i]\mathbf{x}_2^T \cdot \mathbf{A}_i.$$

By subtracting the latter equation from the second equation of (7.5), we find (still modulo q):

$$\big(p(\mathbf{e}_i^T - \mathbf{e}_i'^T) + (\mathbf{y}_i^T - \mathrm{id}_{j^\star}[i]\mathbf{x}_2^T)\big) \cdot \mathbf{A}_i = \mathbf{0}.$$

If $p(\mathbf{e}_i - \mathbf{e}'_i) + (\mathbf{y}_i - \mathrm{id}_{j^\star}[i]\mathbf{x}_2) \neq \mathbf{0}$, it is a non-zero vector in $\Lambda^\perp(\mathbf{A}_i)$ of norm $\leq \mathcal{O}((\sigma + p\alpha q)m^{3/2}n)$. Given that we have $\mathbf{A}_i = \bar{\mathbf{A}}$ with probability $1/\ell$, then $i = i^\star$, we solved the given SIS instance with the same probability. Finally, if $p(\mathbf{e}_i - \mathbf{e}'_i) + (\mathbf{y}_i - \mathrm{id}_{j^\star}[i]\mathbf{x}_2) = \mathbf{0}$, the relative norms of the

vectors $\mathbf{e}_i, \mathbf{e}'_i, \mathbf{y}_i, \mathbf{x}_2$ with respect to p imply $\mathbf{e}_i = \mathbf{e}'_i$ and $\mathbf{y}_i = \mathrm{id}_{j^\star}[i]\mathbf{x}_2$ (over the integers), which is in contradiction with $\mathbf{y}_i \neq \mathrm{id}_{j^\star}[i]\mathbf{x}_2$.

The lower bound on $\mathcal{B}$'s advantage is obtained by combining the probability of obtaining a successful forking, the fact that $\mathcal{B}$'s choice for $j^\star \in U(\{0,\ldots,N-1\})$ is independent of $\mathcal{A}$'s view when $coin = 0$ and the observation that $\mathcal{B}$'s choice for $coin$ is also independent of $\mathcal{A}$'s view. □

7.3 A variant with full (CCA-)anonymity

7.3.1 Description

We modify our basic group signature scheme to reach the strongest anonymity level (Definition 6.7), in which the attacker is authorized to query an opening oracle. This implies the simulation of an oracle which opens adversarially-chosen signatures in the proof of anonymity. To this end, we replace each $\mathbf{B}_i$ from our previous scheme by a matrix $\mathbf{B}_{i,\mathsf{VK}}$ that depends on the verification key VK of a strongly unforgeable one-time signature. The reduction will be able to compute a trapdoor for all these matrices, except for one specific verification key $\mathsf{VK}^\star$ that will be used in the challenge phase. This will provide the reduction with a backdoor allowing it to open all adversarially-generated signatures.

It is assumed that the one-time verification keys VK belong to $\mathbb{Z}_q^n$ (note that this condition can always be enforced by hashing VK). Following Agrawal *et* al. [ABB10a], we rely on a full-rank difference function $H_{vk} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ such that, for any two distinct $\mathbf{u}, \mathbf{v} \in \mathbb{Z}_q^n$, the difference $H_{vk}(\mathbf{u}) - H_{vk}(\mathbf{v})$ is a full rank matrix.

The scheme is described in Figures 7.4 and 7.5. In the rest of the section we prove the following theorems.

7.3.2 Full anonymity

Theorem 7.9. *In the random oracle model, the scheme provides full anonymity in the ROM if the* $\mathsf{LWE}_{q,\alpha}$ *assumption holds and if the one-time signature is strongly unforgeable.*

We now prove the full anonymity of the scheme in an attack game which is exactly the one of Definition 6.7 with the difference that the adversary is granted access to a signature opening oracle. Namely, before and after the challenge phase, the latter oracle can be invoked for adversarially-chosen signatures as long as these do not coincide with the challenge signature $\Sigma^\star$. The proof of Theorem 7.9 relies on the all-but-one simulation technique [BB04] in the same way as in the Agrawal-Boneh-Boyen IBE [ABB10a].

Proof. Like the proof of Theorem 7.3, the proof proceeds via a sequence of hybrid experiments. For each i, we define W_i to be the event that experiment $G_i^{(b)}$ outputs 1.

Experiment $G_0^{(b)}$. This experiment if the real attack game. Namely, the challenger performs the setup of the system by following the specification of the Keygen algorithm. The adversary $\mathcal{A}$ is given gpk and $\{\mathsf{gsk}[j]\}_{j=0}^{N-1}$ at the beginning of the game. All opening queries are answered faithfully, by returning the uncovered identity $\mathrm{id} \in \{0,1\}^\ell$. At the challenge phase, the adversary chooses a message M as well as indexes $j_0, j_1 \in \{0,\ldots,N-1\}$ and obtains a challenge $\Sigma^\star = \big(\mathsf{VK}^\star, \{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell \pi_K^\star, sig^\star\big) \leftarrow \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j_b], M)$. The experiment ends with the adversary $\mathcal{A}$ outputting a bit $b' \in \{0,1\}$. At this point, the experiment returns 1 if $b' = b$ and 0 otherwise. The probability $\Pr[W_0]$ is thus the probability to have $b' = b$.

Experiment $G_1^{(b)}$. We make a simple conceptual change to the generation of the challenge signature $\Sigma^\star$. Namely, instead of sampling $(\mathbf{x}_1 \| \mathbf{x}_2) \in \mathbb{Z}^{3m}$ in $\Lambda^\perp(\mathbf{A}_{\mathrm{id}})$, Experiment $G_1^{(b)}$ first

$\mathsf{Keygen}(1^n, 1^N)$: Given a security parameter $n > 0$ and the desired number of members $N = 2^\ell \in \mathsf{poly}(n)$, choose parameters q, m, p, α, σ as in Section 7.1 and make them public. Choose a hash function $H : \{0,1\}^* \to \{0,1\}^t$ for some $t = \Theta(n)$, that will be modelled as a random oracle, and a one-time signature $\Pi^{\mathrm{ots}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ (Section 6.1.1). Then, proceed as follows.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$.
2. For $i = 0$ to ℓ, repeat the following steps.
   a. Choose uniformly random matrices $\mathbf{A}_{i,1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1} \in \mathbb{Z}_q^{m \times n}$.
   b. Sample $\mathbf{A}_{i,2}$ uniformly such that $\mathbf{B}_{i,1}^T \cdot \mathbf{A}_{i,2} = \mathbf{0} \bmod q$. Define
   $$\mathbf{A}_i = \left[\frac{\mathbf{A}_{i,1}}{\mathbf{A}_{i,2}}\right] \in \mathbb{Z}_q^{2m \times n}.$$
   c. Run $(\mathbf{B}_{i,-1}, \mathbf{S}_i') \leftarrow \mathsf{SuperSamp}(\mathbf{A}_{i,1}, -\mathbf{A}_{i,2}^T \cdot \mathbf{B}_{i,0})$ to obtain $\mathbf{B}_{i,-1} \in \mathbb{Z}_q^{m \times n}$ such that $\mathbf{B}_{i,-1}^T \cdot \mathbf{A}_{i,1} + \mathbf{B}_{i,0}^T \cdot \mathbf{A}_{i,2} = \mathbf{0} \bmod q$.
   d. Compute a re-randomized trapdoor $\mathbf{S}_i \leftarrow \mathsf{RandBasis}(\mathbf{S}_i', \Omega(\sqrt{mn \log q} \log m))^3$ for $\mathbf{B}_{i,-1}$.

   For any string VK, if the matrix $H_{vk}(\mathsf{VK})$ is used to define
   $$\mathbf{B}_{i,\mathsf{VK}} = \left[\frac{\mathbf{B}_{i,-1}}{\mathbf{B}_{i,0} + \mathbf{B}_{i,1} H_{vk}(\mathsf{VK})}\right] \in \mathbb{Z}_q^{2m \times n},$$
   we have $\mathbf{B}_{i,\mathsf{VK}}^T \cdot \mathbf{A}_i = \mathbf{0} \bmod q$ for all i.
3. For $j = 0$ to $N - 1$, let $\mathrm{id}_j = \mathrm{id}_j[1] \ldots \mathrm{id}_j[\ell] \in \{0,1\}^\ell$ be the binary representation of id_j and define:
   $$\mathbf{A}_{\mathrm{id}_j} = \left[\frac{\mathbf{A}}{\mathbf{A}_0 + \sum_{i=1}^{\ell} \mathrm{id}_j[i] \mathbf{A}_i}\right] \in \mathbb{Z}_q^{3m \times n}.$$
   Then run $\mathbf{T}_{\mathrm{id}_j}' \leftarrow \mathsf{ExtBasis}(\mathbf{T_A}, \mathbf{A}_{\mathrm{id}_j})$ to get a short delegated basis $\mathbf{T}_{\mathrm{id}_j}'$ of $\Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j})$. Finally, run $\mathbf{T}_{\mathrm{id}_j} \leftarrow \mathsf{RandBasis}(\mathbf{T}_{\mathrm{id}_j}', \Omega(m\sqrt{\ell n \log q \log m}))$ and define $\mathsf{gsk}[j] := \mathbf{T}_{\mathrm{id}_j}$.
4. Finally, define $\mathsf{gpk} := \big(\mathbf{A}, \{\mathbf{A}_i, (\mathbf{B}_{i,-1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1})\}_{i=0}^{\ell}, \Pi^{\mathrm{ots}}\big)$ and $\mathsf{gmsk} := \{\mathbf{S}_i\}_{i=0}^{\ell}$. The algorithm outputs $\big(\mathsf{gpk}, \mathsf{gmsk}, \{\mathsf{gsk}[j]\}_{j=0}^{N-1}\big)$.

Figure 7.4: KeyGen.

samples $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and uses the trapdoor $\mathbf{T_A}$ to compute $\mathbf{x}_1 \in D_{\mathbb{Z}^m,\sigma}$ such that $(\mathbf{x}_1 \| \mathbf{x}_2)^T \cdot \mathbf{A}_{\mathrm{id}} = 0 \bmod q$. This change is purely conceptual since the vector $(\mathbf{x}_1 \| \mathbf{x}_2)$ has the same distribution either way. Clearly, it holds that $\Pr[W_1] = \Pr[W_2]$.

Experiment $G_2^{(b)}$. We introduce a slight modification w.r.t. Experiment $G_1^{(b)}$. At the outset of the game, the challenger generates a one-time signature key pair $(\mathsf{VK}^\star, \mathsf{SK}^\star) \leftarrow \mathcal{G}(1^n)$. If $\mathcal{A}$ queries the opening oracle with a valid signature $\Sigma = \big(\mathsf{VK}, \{\mathbf{c}_i\}_{i=0}^{\ell}, \pi_0, \{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell} \pi_K, sig\big)$ such that $\mathsf{VK} = \mathsf{VK}^\star$, the experiment halts and outputs a random bit. The assumed strong security of the one-time signature implies that Experiment $G_2^{(b)}$ cannot depart from Experiment $G_1^{(b)}$. Indeed, if a valid opening query is made after the challenge phase, the adversary is able to break the strong unforgeability of the one-time signature (the proof is straightforward and omitted). Moreover, before the challenge phase, the one-time verification key $\mathsf{VK}^\star$ is independent of $\mathcal{A}$'s view. As long as no one-time verification key is produced by the one-time key generation algorithm

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[j], M)$: To sign a message $M \in \{0,1\}^*$ using the private key $\mathsf{gsk}[j] = \mathbf{T}_{\mathrm{id}_j}$, generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^n)$ for Π^{ots} and proceed as follows.

1. Run $\mathsf{GPVSample}(\mathbf{T}_{\mathrm{id}_j}, \sigma)$ to get $(\mathbf{x}_1 \| \mathbf{x}_2) \in \Lambda_q^{\perp}(\mathbf{A}_{\mathrm{id}_j})$ of norm $\leq \sigma\sqrt{3m}$.
2. Sample $\mathbf{s}_0 \hookleftarrow U(\mathbb{Z}_q^n)$ and encrypt $\mathbf{x}_2 \in \mathbb{Z}^{2m}$ as $\mathbf{c}_0 = \mathbf{B}_{0,\mathsf{VK}} \cdot \mathbf{s}_0 + \mathbf{x}_2 \in \mathbb{Z}_q^{2m}$.
3. Sample $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$. For $i = 1$ to ℓ, sample $\mathbf{e}_i \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$ and a random matrix $\mathbf{R}_i \in \mathbb{Z}^{m \times m}$ whose columns are sampled from $D_{\mathbb{Z}^m, \sigma}$. Then, compute:
$$\mathbf{c}_i = \mathbf{B}_{i,\mathsf{VK}} \cdot \mathbf{s} + p \cdot (\mathbf{e}_i \| (\mathbf{e}_i^T \cdot \mathbf{R}_i)^T) + \mathrm{id}_j[i] \cdot \mathbf{x}_2,$$
which encrypts $\mathbf{x}_2 \in \mathbb{Z}^{2m}$ (resp. $\mathbf{0}^{2m}$) if $\mathrm{id}_j[i] = 1$ (resp. $\mathrm{id}_j[i] = 0$).
4. Generate a NIZKPoK π_0 of $\mathbf{s}_0$ so that $(\mathbf{B}_0, \mathbf{c}_0, \sqrt{2}\sigma/q; \mathbf{s}_0) \in R_{\mathsf{LWE}}$.
5. For $i = 1$ to ℓ, generate a NIZKPoK $\pi_{\mathrm{OR},i}$ of $\mathbf{s}$ and $\mathbf{s}_0$ so that either:
   (i) $((\mathbf{B}_{i,\mathsf{VK}} | \mathbf{B}_{0,\mathsf{VK}}), p^{-1}(\mathbf{c}_i - \mathbf{c}_0), \sqrt{2}\alpha; (\mathbf{s} \| -\mathbf{s}_0)) \in R_{\mathsf{LWE}}$ (the vectors $\mathbf{c}_i$ and $\mathbf{c}_0$ encrypt the same $\mathbf{x}_2$, so that the vector $p^{-1}(\mathbf{c}_i - \mathbf{c}_0)$ is close to the $\mathbb{Z}_q$-span of $(\mathbf{B}_{i,\mathsf{VK}} | \mathbf{B}_{0,\mathsf{VK}})$);
   (ii) or $(\mathbf{B}_{i,\mathsf{VK}}, p^{-1}\mathbf{c}_i, \alpha; \mathbf{s}) \in R_{\mathsf{LWE}}$ (the vector $\mathbf{c}_i$ encrypts $\mathbf{0}$, so that $p^{-1}\mathbf{c}_i$ is close to the $\mathbb{Z}_q$-span of $\mathbf{B}_{i,\mathsf{VK}}$).
6. For $i = 1$ to ℓ, set $\mathbf{y}_i = \mathrm{id}_j[i]\mathbf{x}_2 \in \mathbb{Z}^{2m}$ and generate a NIZKPoK π_K of $\{\mathbf{e}_i\}_{i=1}^{\ell}$, $\{\mathbf{y}_i\}_{i=1}^{\ell}$, $\mathbf{x}_1$ such that:
$$\mathbf{x}_1^T\mathbf{A} + \sum_{i=0}^{\ell} \mathbf{c}_i^T\mathbf{A}_i = \sum_{i=1}^{\ell} \mathbf{e}_i^T(p \cdot \mathbf{A}_i) \text{ and } \mathbf{e}_i^T(p \cdot \mathbf{A}_i) + \mathbf{y}_i^T\mathbf{A}_i = \mathbf{c}_i^T\mathbf{A}_i \quad \text{for } i \in \{1, \ldots, \ell\},$$
with $\|\mathbf{e}_i\|, \|\mathbf{y}_i\|, \|\mathbf{x}_1\| \leq \max(\sigma, \alpha q) \cdot \sqrt{2m}$. This is achieved using $\mathsf{Prove}_{\mathsf{ISIS}}$, giving a triple $(\mathsf{Comm}_K, \mathsf{Chall}_K, \mathsf{Resp}_K)$, where $\mathsf{Chall}_K = H(M, \mathsf{Comm}_K, \{\mathbf{c}_i\}_{i=0}^{\ell}, \pi_0, \{\pi_{\mathrm{OR},i}\}_{i=i}^{\ell})$.
7. Compute $sig = \mathcal{S}(\mathsf{SK}, \{\mathbf{c}_i\}_{i=0}^{\ell}, \pi_0, \{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}, \pi_K)$.

The signature consists of
$$\Sigma = \big(\mathsf{VK}, \{\mathbf{c}_i\}_{i=0}^{\ell}, \pi_0, \{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}, \pi_K, sig\big). \tag{7.6}$$

$\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$: Parse the signature Σ as in (7.6). Then, return 1 in the event that $\mathcal{V}(\mathsf{VK}, sig, \{\mathbf{c}_i\}_{i=0}^{\ell}, \pi_0, \{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}, \pi_K) = 1$. and if all proofs $\pi_0, \{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}, \pi_K$ properly verify. Otherwise, return 0.

$\mathsf{Open}(\mathsf{gpk}, \mathsf{gmsk}, M, \Sigma)$: Parse gmsk as $\{\mathbf{S}_i\}_{i=0}^{\ell}$ and Σ as in (7.6). For $i = 0$ to ℓ, compute a trapdoor $\mathbf{S}_{i,\mathsf{VK}} \leftarrow \mathsf{ExtBasis}(\mathbf{S}_i, \mathbf{B}_{i,\mathsf{VK}})$ for $\mathbf{B}_{i,\mathsf{VK}}$. Using the delegated basis $\mathbf{S}_{0,\mathsf{VK}} \in \mathbb{Z}^{2m \times 2m}$ (for which we have $\mathbf{S}_{0,\mathsf{VK}} \cdot \mathbf{B}_{0,\mathsf{VK}} = \mathbf{0} \bmod q$), compute $\mathbf{x}_2$ by decrypting $\mathbf{c}_0$. Then, using $\mathbf{S}_{i,\mathsf{VK}} \in \mathbb{Z}^{2m \times 2m}$, determine which vector among $p^{-1}\mathbf{c}_i \bmod q$ and $p^{-1}(\mathbf{c}_i - \mathbf{x}_2) \bmod q$ is close to the $\mathbb{Z}_q$-span of $\mathbf{B}_{i,\mathsf{VK}}$. Set $\mathrm{id}[i] = 0$ in the former case and $\mathrm{id}[i] = 1$ in the latter. Eventually, output $\mathrm{id} = \mathrm{id}[1] \ldots \mathrm{id}[\ell]$.

Figure 7.5: Sign, Verify and Open.

with too high probability (which is implied by the strong unforgeability property), the chance of $\mathsf{VK}^\star$ to show up in a valid pre-challenge opening query is negligible. There thus exists a PPT forger $\mathcal{B}^{\text{ots}}$ against the one-time signature for which $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\text{suf-ots}}(\mathcal{B}^{\text{ots}})$. In the following, we henceforth assume that no opening query involves $\mathsf{VK}^\star$.

Experiment $G_3^{(b)}$. We bring a first modification to the generation of the group public key gpk in the setup phase. Namely, for each $i \in \{0, \ldots, \ell\}$, the experiment first runs $(\mathbf{B}_{i,1}, \mathbf{T}_{i,1}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ to obtain a matrix $\mathbf{B}_{i,1} \in \mathbb{Z}_q^{m \times n}$ with a short basis $\mathbf{T}_{i,1} \in \mathbb{Z}^{m \times m}$. Note that the distribution of $\mathbf{B}_{i,1}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{m \times n}$. Next, the experiment sets $\mathbf{B}_{i,0} = \mathbf{R}_i \cdot \mathbf{B}_{i,-1} - \mathbf{B}_{i,1} \cdot H_{vk}(\mathsf{VK}^\star)$, where $\mathbf{R}_i \in \mathbb{Z}^{m \times m}$ is a matrix whose rows are vectors sampled from the distribution $D_{\mathbb{Z}^m,\sigma}$. The result of [GPV08, Lemma 5.2] implies that matrices $\{\mathbf{B}_{i,0}\}_{i=0}^{\ell}$ will be statistically close to the uniformly distributed matrices produced by the real key generation algorithm. We can write $|\Pr[W_3] - \Pr[W_2]| \in \mathsf{negl}(1^n)$.

Experiment $G_4^{(b)}$. In this experiment, we modify the signature opening oracle in the following way. Recall that, due to the modification introduced in Experiment $G_2^{(b)}$, each opening query involves a signature $\Sigma = \big(\mathsf{VK}, \{\mathbf{c}_i\}_{i=0}^{\ell}, \pi_0, \{\pi_{\text{OR},i}\}_{i=1}^{\ell} \pi_K, sig\big)$ for which $\mathsf{VK} \neq \mathsf{VK}^\star$ unless the one-time signature is not strongly unforgeable. For this reason, each matrix $\mathbf{B}_{i,\mathsf{VK}}$ can be written as

$$\mathbf{B}_{i,\mathsf{VK}} = \left[\frac{\mathbf{B}_{i,-1}}{\mathbf{B}_{i,0} + \mathbf{B}_{i,1} H_{vk}(\mathsf{VK})}\right] = \left[\frac{\mathbf{B}_{i,-1}}{\mathbf{R}_i \cdot \mathbf{B}_{i,-1} + \mathbf{B}_{i,1} \cdot \big(H_{vk}(\mathsf{VK}) - H_{vk}(\mathsf{VK}^\star)\big)}\right],$$

where $H_{vk}(\mathsf{VK}) - H_{vk}(\mathsf{VK}^\star)$ is a non-singular $n \times n$ matrix over $\mathbb{Z}_q$. This implies that the trapdoor $\mathbf{T}_{i,1} \in \mathbb{Z}^{m \times m}$ of $\mathbf{B}_{i,1}$ – which was defined in Experiment $G_3^{(b)}$ – can be used to generate a short basis for the lattice $\Lambda^{\perp}(\mathbf{B}_{i,\mathsf{VK}})$ as in step 2 of the $\mathsf{SampleRight}$ algorithm of [ABB10a, Section 4.2]. The obtained short basis $\mathbf{T}_{i,\mathsf{VK}} \in \mathbb{Z}^{2m \times 2m}$ satisfies $\mathbf{T}_{i,\mathsf{VK}} \in \mathbb{Z}^{2m \times 2m} \cdot \mathbf{B}_{i,\mathsf{VK}} = 0 \bmod q$ and it can be used exactly in the same way as the delegated bases $\mathbf{S}_{i,\mathsf{VK}}$ of the actual opening algorithm to identify the signer. This modification is thus purely conceptual and we thus have $\Pr[W_4] = \Pr[W_3]$. We remark that, in this experiment, the trapdoors $\{\mathbf{S}_i\}_{i=0}^{\ell}$ of matrices $\{\mathbf{B}_{i,-1}\}_{i=0}^{\ell}$ are not used any longer.

Experiment $G_5^{(b)}$. This experiment is identical to Experiment $G_4^{(b)}$ but we slightly modify the setup phase in step c of the key generation algorithm. Recall that Experiment $G_4^{(b)}$ generates $(\mathbf{B}_{i,-1}, \mathbf{S}_i') \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{A}_{i,1}, -\mathbf{A}_{i,2}^T \cdot \mathbf{B}_{i,0})$ so as to obtain a matrix $\mathbf{B}_{i,-1} \in \mathbb{Z}_q^{m \times n}$ satisfying the equality

$$\mathbf{B}_{i,-1}^T \cdot \mathbf{A}_{i,1} + \mathbf{B}_{i,0}^T \cdot \mathbf{A}_{i,2} = 0 \bmod q \tag{7.7}$$

at step c of Keygen. In contrast, Experiment $G_5^{(b)}$ proceeds by generating $(\mathbf{B}_{i,1}, \mathbf{T}_{i,1}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ and choosing $\mathbf{B}_{i,-1}, \mathbf{B}_{i,0}$ uniformly in $\mathbb{Z}_q^{m \times n}$. Then, it generates

$$(\mathbf{A}_{i,1}, \mathbf{T}_i') \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{B}_{i,-1}, -\mathbf{B}_{i,0}^T \cdot \mathbf{A}_{i,2}),$$

which satisfies (7.7). The same arguments as in [GKV10, Lemma 5] imply that the set $\{\mathbf{B}_{i,-1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1}, \mathbf{A}_i\}_{i=0}^{\ell}$ have a distribution which is negligibly far apart from their distribution in Experiment $G_4^{(b)}$.

The setup phase is completed by using $\mathbf{T}_{\mathbf{A}}$ to compute group member's private keys $\{\mathsf{gsk}[j]\}_{j=0}^{N-1}$. Since $\mathcal{A}$'s view is not noticeably affected by this modification, we have $|\Pr[W_5] - \Pr[W_4]| \in \mathsf{negl}(1^n)$.

Experiment $G_6^{(b)}$. Here, we modify the generation of the challenge signature $\Sigma^\star$ as follows. At step 5 of the signing algorithm, instead of computing the NIZK proofs $\{\pi_{\text{OR},i}^\star\}_{i=1}^{\ell}$ using the

actual witnesses, the experiment generates a simulated non-interactive proof by programming the random oracle. The statistical zero-knowledge property of the Micciancio-Vadhan proof system [MV03] guarantees that the distribution of $\{\pi^\star_{\mathrm{OR},i}\}_{i=1}^{\ell}$ remains statistically unchanged (note that $\{\pi^\star_{\mathrm{OR},i}\}_{i=1}^{\ell}$ are simulated proofs for true statements). Therefore it comes that $|\Pr[W_6]-\Pr[W_5]| \in \mathsf{negl}(1^n)$. Note that $\mathsf{negl}(1^n)$ incorporates the small probability that the NIZK simulator fails because it accidentally has to program the random oracle on an input where it was previously defined.

Experiment $G_7^{(b)}$. In this experiment, we bring a new modification to the generation of $\Sigma^\star$. The real proof of knowledge $\pi_K^\star$ is replaced by a simulated proof which is obtained by programming the random oracle H at step 6 of the signing algorithm. Similarly to the previous transition, we can write $|\Pr[W_7]-\Pr[W_6]| \in \mathsf{negl}(1^n)$, where $\mathsf{negl}(1^n)$ encompasses the tiny probability that the NIZK simulator fails.

Experiment $G_8^{(b)}$. We introduce yet another change in the generation of $\Sigma^\star$. For each $i \in \{1,\ldots,\ell\}$, instead of computing $\mathbf{c}_i^\star = \mathbf{B}_{i,\mathsf{VK}^\star}\cdot\mathbf{s} + p\cdot e_i + \mathrm{id}[j_b]\mathbf{x}_2$, where $\mathbf{x}_2 \in \mathbb{Z}^{2m}$ is the vector encrypted by $\mathbf{c}_0$, the experiment sets $\mathbf{c}_i^\star = \mathbf{z}_i + \mathrm{id}[j_b]\cdot\mathbf{x}_2$ for a randomly drawn $\mathbf{z}_i \hookleftarrow U(\mathbb{Z}_q^{2m})$. Under the $\mathsf{LWE}_{q,\alpha}$ assumption, we argue that this change should not significantly affect $\mathcal{A}$'s view. Concretely, assuming that an adversary can distinguish Experiment $G_8^{(b)}$ from Experiment $G_7^{(b)}$, we can build a distinguisher $\mathcal{B}^{\mathrm{lwe}}$ for the $\mathsf{LWE}_{q,\alpha}$. The latter distinguisher is described in the proof of Lemma 7.10 for completeness. For this reason, we find $|\Pr[W_8]-\Pr[W_7]| \leq \mathbf{Adv}^{\mathsf{LWE}_{q,\alpha}}(\mathcal{B}^{\mathrm{lwe}})$.

Experiment $G_9^{(b)}$. As a final change in the generation of $\Sigma^\star$, we choose $\mathbf{c}_i^\star$ at random in $U(\mathbb{Z}_q^{2m})$ for $i \in \{1,\ldots,\ell\}$. This is just a conceptual change since $\{\mathbf{c}_i^\star\}_{i=1}^{\ell}$ have exactly the same distribution as in Experiment $G_8^{(b)}$. This implies $\Pr[W_9] = \Pr[W_8]$. Moreover, in Experiment $G_9^{(b)}$, it is obvious that $\Pr[W_9] = 1/2$ since $\Sigma^\star$ is completely independent of the random bit $b \in_R \{0,1\}$.

To conclude the proof, we prove the indistinguishability of Experiment $G_8^{(b)}$ and Experiment $G_7^{(b)}$.

Lemma 7.10. *Under the* $\mathsf{LWE}_{q,\alpha}$ *assumption, no* PPT *adversary can distinguish Experiment* $G_8^{(b)}$ *and Experiment* $G_7^{(b)}$.

Proof. Towards a contradiction, suppose that an adversary $\mathcal{A}$ can tell the two experiments apart with non-negligible advantage. We build the following LWE distinguisher $\mathcal{B}^{\mathrm{lwe}}$. It takes as input a $\mathsf{LWE}_{q,\alpha}$ instance $\{(\mathbf{B}'_i,\mathbf{z}_i)\}_{i=1}^{\ell}$, where $\mathbf{B}'_i \in \mathbb{Z}_q^{m\times n}$ and $\mathbf{z}_i \in \mathbb{Z}_q^m$ for each $i \in \{1,\ldots,\ell\}$. Each component $\mathbf{z}_i$ is either uniform in $\mathbb{Z}_q^m$ or of the form $\mathbf{z}_i = \mathbf{B}'_i\cdot\mathbf{s}+\mathbf{e}_i$, where $\mathbf{e}_i$ is sampled from $D_{\mathbb{Z}^m,\alpha q}$.

In order to prepare the group public key gpk, algorithm $\mathcal{B}^{\mathrm{lwe}}$ defines $\mathbf{B}_{i,-1} = \mathbf{B}'_i$ for $i = 1$ to ℓ. For each $i \in \{1,\ldots,\ell\}$, it also generates $\mathbf{B}_{i,1}$ by running $(\mathbf{B}_{i,1},\mathbf{T}_{i,1}) \leftarrow \mathsf{TrapGen}(1^n,1^m,q)$ and also sets $\mathbf{B}_{i,0} = \mathbf{R}_i\cdot\mathbf{B}_{i,-1} - \mathbf{B}_{i,1}H_{vk}(\mathsf{VK}^\star)$ as in Experiment $G_7^{(b)}$. By doing so, $\mathcal{B}^{\mathrm{lwe}}$ is able to answer all signature opening queries using the trapdoor $\mathbf{T}_{i,1}$ of $\mathbf{B}_{i,1}$ unless the failure event introduced in Experiment $G_2^{(b)}$ occurs.

During the challenge phase, $\mathcal{B}^{\mathrm{lwe}}$ samples $\mathbf{x}_2$ in $D_{\mathbb{Z}^{2m},\sigma}$ and defines

$$\mathbf{c}_i^\star = \left\lfloor \frac{p\cdot\mathbf{z}_i}{\mathbf{R}_i\cdot(p\cdot\mathbf{z}_i)} \right\rceil + \mathrm{id}[j_b]\cdot\mathbf{x}_2, \text{ for } i \in \{1,\ldots,\ell\},$$

while $\mathbf{c}_0^\star$ is obtained by faithfully encrypting $\mathbf{x}_2$. The proof $\pi_0^\star$ is generated as a real proof whereas $\{\pi^\star_{\mathrm{OR},i}\}_{i=1}^{\ell}$ and $\pi_K^\star$ are obtained from their respective NIZK simulators.

After the challenge phase, $\mathcal{A}$ is granted further access to the opening oracle and its opening

queries are handled as in the first phase. At the end of the experiment, $\mathcal{A}$ outputs a random bit b' and $\mathcal{B}^{\text{lwe}}$ outputs 1 if and only if $b' = b$.

We note that each $\mathbf{B}_{i,\mathsf{VK}^\star}$ is such that $\mathbf{B}_{i,\mathsf{VK}^\star} = \left[\frac{\mathbf{B}_{i,-1}}{\mathbf{R}_i \cdot \mathbf{B}_{i,-1}}\right]$ for $i \in \{1, \ldots, \ell\}$. If each $\mathbf{z}_i$ is such that $\mathbf{z}_i = \mathbf{B}'_i \cdot \mathbf{s} + \mathbf{e}_i$, where $\mathbf{e}_i \in D_{\mathbb{Z}^m, \alpha q}$, then $\{\mathbf{c}_i^\star\}_{i=1}^\ell$ are distributed as in Experiment $G_7^{(b)}$. Indeed, the matrices $\{\mathbf{R}_i\}_{i=1}^\ell$ introduced in Experiment $G_3^{(b)}$ are statistically independent of $\mathcal{A}$'s view until the challenge phase because the product $\mathbf{R}_i \cdot \mathbf{B}_{i,-1}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{m \times n}$. In this case, the reduction $\mathcal{B}^{\text{lwe}}$ is running Experiment $G_7^{(b)}$ with $\mathcal{A}$. Now, if each $\mathbf{z}_i$ is uniform in $\mathbb{Z}_q^m$, we are clearly in Experiment $G_8^{(b)}$. □

This concludes the proof of Theorem 7.9. □

7.3.3 Traceability

Theorem 7.11. *Assuming that $q > \log N$, the scheme is fully traceable in the ROM under the $\mathsf{SIS}_{m,q,\beta}$ assumption. More precisely, for any PPT traceability adversary $\mathcal{A}$ with success probability ε, there exists an algorithm $\mathcal{B}$ solving the $\mathsf{SIS}_{m,q,\beta}$ problem with probability at least*

$$\frac{1}{2N} \cdot \left(\varepsilon - \frac{1}{2^\lambda}\right) \cdot \left(\frac{\varepsilon - 1/2^\lambda}{q_H} - \frac{1}{2^\lambda}\right),$$

where q_H is the number of queries to $H : \{0,1\}^* \to \{0,1\}^\lambda$.

The traceability property is proved in the same way as in the proof of Theorem 7.8.

Proof. For the sake of contradiction, let us assume that a traceability adversary $\mathcal{A}$ has non-negligible success probability ε in the model of Definition 6.8. In the random oracle model, we build an algorithm $\mathcal{B}$ that solves a given $\mathsf{SIS}_{2m,q,\beta}$ instance with non-negligible probability. Algorithm $\mathcal{B}$ receives as input a matrix $\hat{\mathbf{A}} \in \mathbb{Z}_q^{2m \times n}$ and has to find a vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $\Lambda_q^\perp(\hat{\mathbf{A}})$ such that $0 < \|\mathbf{v}\| \leq \beta$. Let $\bar{\mathbf{A}} \in \mathbb{Z}_q^{m \times n}$ be the matrix consisting of the first m rows of $\hat{\mathbf{A}}$.

Initialization. As in the proof of Theorem 7.8, algorithm $\mathcal{B}$ first flips a fair coin $coin \hookleftarrow U(\{0,1\})$ that will determine its strategy and the way to set up the group public key. If $coin = 0$, algorithm $\mathcal{B}$ will try to find a non-zero short vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$ and pad it with zeroes to obtain a short non-zero vector in $\Lambda_q^\perp(\hat{\mathbf{A}})$. If $coin = 1$, $\mathcal{B}$ will embed the entire input matrix $\hat{\mathbf{A}}$ in one of the $\{\mathbf{A}_i\}_{i=1}^\ell$.

• If $coin = 0$, algorithm $\mathcal{B}$ first runs $\mathsf{TrapGen}(1^n, 1^{2m}, q)$ to generate $\mathbf{C} \in \mathbb{Z}_q^{2m \times n}$ with a basis $\mathbf{T_C} \in \mathbb{Z}^{2m \times 2m}$ of $\Lambda_q^\perp(\mathbf{C})$ with $\|\widetilde{\mathbf{T_C}}\| \leq \mathcal{O}(\sqrt{n \log q})$. Next, $\mathcal{B}$ samples a collection of $\ell + 1$ matrices $\mathbf{Q}_0, \ldots, \mathbf{Q}_\ell \in \mathbb{Z}^{2m \times m}$, where each matrix entry sampled independently in $D_{\mathbb{Z}, \omega(\sqrt{\log n})}$. Then, $\mathcal{B}$ draws $j^\star \hookleftarrow U(\{0, \ldots, N-1\})$, hoping that user $j^\star$ will be the one whose identity $\mathrm{id}_{j^\star} = \mathrm{id}_{j^\star}[1] \ldots \mathrm{id}_{j^\star}[\ell] \in \{0,1\}^\ell$ will be uncovered by the opening algorithm for $\mathcal{A}$'s forgery at the end of the game. Also, $\mathcal{B}$ defines $\mathbf{A}_0 = \mathbf{Q}_0 \cdot \bar{\mathbf{A}} + (\sum_{i=1}^\ell \mathrm{id}_{j^\star}[i]) \cdot \mathbf{C}$ and $\mathbf{A}_i = \mathbf{Q}_i \cdot \bar{\mathbf{A}} + (-1)^{\mathrm{id}_{j^\star}[i]} \cdot \mathbf{C}$ for each $i \in [1, \ell]$. It also sets $\mathbf{A} = \bar{\mathbf{A}}$.

Then, for each $i \in \{0, \ldots, \ell\}$, $\mathcal{B}$ chooses $\mathbf{B}_{i,0} \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ and parses the matrix $\mathbf{A}_i \in \mathbb{Z}_q^{2m \times n}$ as $\mathbf{A}_i^T = [\mathbf{A}_{i,1}^T \mid \mathbf{A}_{i,2}^T]$, where $\mathbf{A}_{i,1}, \mathbf{A}_{i,2} \in \mathbb{Z}_q^{m \times n}$. Then, it runs $(\mathbf{B}_{i,1}, \mathbf{T}_{\mathbf{B}_{i,1}}) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{A}_{i,2}, \mathbf{0})$ to obtain a matrix $\mathbf{B}_{i,1} \in \mathbb{Z}_q^{m \times n}$ such that $\mathbf{A}_{i,2}^T \cdot \mathbf{B}_{i,1} = 0 \bmod q$. It erases $\mathbf{T}_{\mathbf{B}_{i,1}}$, that will not be needed, and generates $(\mathbf{B}_{i,-1}, \mathbf{S}'_i) \leftarrow \mathsf{SuperSamp}(1^n, 1^m, q, \mathbf{A}_{i,1}, -\mathbf{B}_{i,0}^T \cdot \mathbf{A}_{i,2})$ which will satisfy

$$\mathbf{B}_{i,-1}^T \cdot \mathbf{A}_{i,1} + \mathbf{B}_{i,0}^T \cdot \mathbf{A}_{i,2} = 0 \bmod q,$$

as desired. Finally, $\mathcal{B}$ re-randomizes each $\mathbf{S}'_i$ as $\mathbf{S}_i \leftarrow \mathsf{RandBasis}(\mathbf{S}'_i)$ for $i = 0$ to ℓ. We observe that $\mathcal{B}$ notably departs from the real key generation algorithm in that $\mathbf{B}_{i,1}$ is generated from $\mathbf{A}_{i,2}$ (whereas Keygen proceeds the other way around at step 2) using $\mathsf{SuperSamp}$. However, by Lemma 4 in [GKV10], the distribution of the resulting matrices is statistically the same either way.

The group public key $\mathsf{gpk} = \big(\mathbf{A}, \{\mathbf{A}_i, (\mathbf{B}_{i,-1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1})\}_{i=0}^{\ell}\big)$ is finally given to $\mathcal{A}$. As in the proof of Theorem 7.8, for each $j \neq j^\star$, we have

$$\mathbf{A}_{\mathrm{id}_j} = \left[\frac{\bar{\mathbf{A}}}{\mathbf{A}_0 + \sum_{i=1}^{\ell} \mathrm{id}_j[i]\mathbf{A}_i}\right] = \left[\frac{\bar{\mathbf{A}}}{(\mathbf{Q}_0 + \sum_{i=1}^{\ell} \mathrm{id}_j[i]\mathbf{Q}_i)\cdot \bar{\mathbf{A}} + h_{\mathrm{id}_j}\cdot \mathbf{C}}\right] \in \mathbb{Z}_q^{2m\times n},$$

where $h_{\mathrm{id}_j} \in \{1, \ldots, \ell\}$ denotes the Hamming distance between id_j and $\mathrm{id}_{j^\star}$. As in the proof of Theorem 7.8, for each identifier $\mathrm{id}_j \neq \mathrm{id}_{j^\star}$, $\mathcal{B}$ is able to compute a basis $\mathbf{T}'_{\mathrm{id}_j}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathrm{id}_j})$ with $\|\widetilde{\mathbf{T}'_{\mathrm{id}_j}}\| \leq \omega(\sqrt{2mn\log q \log n})$ from the basis $\mathbf{T}_{\mathbf{C}}$ of $\Lambda_q^\perp(\mathbf{C})$. The obtained bases $\{\mathbf{T}'_{\mathrm{id}_j}\}_{\mathrm{id}_j \neq \mathrm{id}_{j^\star}}$ are then re-rerandomized as $\mathbf{T}_{\mathrm{id}_j} \leftarrow \mathsf{RandBasis}(\mathbf{T}'_{\mathrm{id}_j}, \omega(\sqrt{2mn\log q}\log n))$. However, the reduction $\mathcal{B}$ is unable to compute a trapdoor for the matrix $\mathbf{A}_{\mathrm{id}_{j^\star}}$ corresponding to the expected target group member $j^\star$. Fortunately, $\mathcal{B}$ can derive a trapdoor $\mathbf{T}_{\mathrm{id}_j}$ for each $j \neq j^\star$.

Since the rows of each $\mathbf{Q}_k$ are sampled from $D_{\mathbb{Z}^m, \omega(\sqrt{\log n})}$, the matrices $\mathbf{A}_0, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{2m\times n}$ have a distribution which is statistically close to that of independent and uniformly random matrices over $\mathbb{Z}_q^{2m\times n}$, which are also statistically independent of $\mathbf{A}$. Also, by Lemma 3.10, the distribution of $\{\mathbf{T}_{\mathrm{id}_j}\}_{j\neq j^\star}$ is statistically close to that of the real system.

• If $coin = 1$, the reduction $\mathcal{B}$ chooses $i^\star \hookleftarrow U(\{1, \ldots, \ell\})$ and defines $\hat{\mathbf{A}}$ to be the matrix $\mathbf{A}_{i^\star} \in \mathbb{Z}_q^{2m\times n}$ that will be part of gpk. It runs $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ with a basis $\mathbf{T}_{\mathbf{A}}$ of $\Lambda_q^\perp(\mathbf{A})$ such that $\|\widetilde{\mathbf{T}_{\mathbf{A}}}\| \leq \mathcal{O}(\sqrt{n\log q})$. Next, it independently samples $\mathbf{A}_0, \ldots, \mathbf{A}_{i^\star-1}, \mathbf{A}_{i^\star+1}, \ldots, \mathbf{A}_\ell \hookleftarrow U(\mathbb{Z}_q^{2m\times n})$ and sets $\mathbf{A}_{i^\star} = \hat{\mathbf{A}}$. Finally, $\mathcal{B}$ computes $\{(\mathbf{B}_{i,-1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1})\}_{i=0}^{\ell}$ in the same way as in the case $coin = 0$. As in the previous case, $\mathcal{B}$ thus knows a trapdoor $\mathbf{S}_i$ for $\mathbf{B}_{i,-1}$ for each $i \in \{0, \ldots, \ell\}$. The group public key $\mathsf{gpk} = \big(\mathbf{A}, \{\mathbf{A}_i, (\mathbf{B}_{i,-1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1})\}_{i=0}^{\ell}\big)$, which is distributed (statistically) as in the real system, is given as input to $\mathcal{A}$. Using $\mathbf{T}_{\mathbf{A}}$, the reduction $\mathcal{B}$ is able to compute a delegated basis $\mathbf{T}_{\mathrm{id}_j}$ for all users $j \in \{0, \ldots, N-1\}$ exactly as in the real scheme.

Regardless of the value of $coin \in \{0, 1\}$, the adversary $\mathcal{A}$ is run on input of $\mathsf{gmsk} := \{\mathbf{S}_i\}_{i=0}^{\ell}$ and $\mathsf{gpk} := \big(\mathbf{A}, \{\mathbf{A}_i, (\mathbf{B}_{i,-1}, \mathbf{B}_{i,0}, \mathbf{B}_{i,1})\}_{i=0}^{\ell},\ H,\ \Pi^{\mathrm{ots}},\ p\big)$.

Queries. Algorithm $\mathcal{B}$ starts interacting with adversary $\mathcal{A}$ whose queries are handled in a way that depends on $coin \in \{0, 1\}$.

• If $coin = 0$, $\mathcal{B}$ aborts if $\mathcal{A}$ ever queries the private key $\mathsf{gsk}[j^\star]$ of user $j^\star$. When $\mathcal{A}$ queries a private key $\mathsf{gsk}[j]$ for $j \in \{0, \ldots, N-1\}\backslash\{j^\star\}$, $\mathcal{B}$ reveals the previously computed short basis $\mathbf{T}_{\mathrm{id}_j}$. When $\mathcal{A}$ queries the signing oracle, $\mathcal{B}$ faithfully runs the signing algorithm whenever the involved user j is not $j^\star$. For each signing query involving the expected target user $j^\star$, $\mathcal{B}$ samples $\mathbf{x}_2 \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and $\mathbf{s}_0, \mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$. Then, it computes $\mathbf{c}_0 = \mathbf{B}_0 \cdot \mathbf{s}_0 + \mathbf{x}_2$ as well as $\mathbf{c}_i = \mathbf{B}_{i,\mathsf{VK}} \cdot \mathbf{s} + p \cdot [\mathbf{e}_i | \mathbf{e}_i \cdot \mathbf{R}_i] + \mathrm{id}_{j^\star}[i^\star] \cdot \mathbf{x}_2$ for each $i \in \{1, \ldots, \ell\}$. The proof π_0 is computed as a real proof (i.e., using the witness $\mathbf{x}_2$), whereas all other non-interactive proofs $\{\pi_{\mathrm{OR},i}\}_{i=1}^{\ell}$ and π_K are simulated using the appropriate NIZK simulator, by programming the random oracle. Since the simulator is statistically zero-knowledge, the resulting signature Σ will be statistically indistinguishable from a real signature.

• If $coin = 1$, $\mathcal{B}$ has all private keys $\{\mathsf{gsk}[j]\}_{j=0}^{N-1}$ at disposal since it knows $\mathbf{T}_{\mathbf{A}}$. It can thus perfectly answer $\mathcal{A}$'s queries by running the actual signing algorithm or returning the queried private keys $\mathsf{gsk}[j]$.

For each $coin \in \{0,1\}$, queries to the random oracle H are handled by returning a uniformly chosen value in $\{0,1\}^\lambda$. For each $\kappa \in \{1,\ldots,q_H\}$, r_κ will stand for the answer to the κ-th H-query. As usual, if a given random oracle query occurs more than once, $\mathcal{B}$ responds by returning the previously defined value.

Forgery. Eventually, $\mathcal{A}$ outputs a signature $\Sigma^\star = \left(\{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell, \pi_K^\star\right)$ on some message $M^\star$ with probability ε. If we parse the proof $\pi_K^\star$ as $(\mathsf{Comm}_K^\star, \mathsf{Chall}_K^\star, \mathsf{Resp}_K^\star)$, w.h.p., $\mathcal{A}$ must have queried H on the input $(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell)$. Otherwise, the probability to have $\mathsf{Chall}_K^\star = H(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell)$ is at most $1/2^\lambda$. With probability $\varepsilon - 1/2^\lambda$, the tuple $(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell)$ was the input of the κ-th random oracle query for some $\kappa^\star \in \{1,\ldots,q_H\}$.

Now, $\mathcal{B}$ starts a second execution of the adversary $\mathcal{A}$ with the *same* random tape and input as in the first run. All queries are answered as in the latter with a difference in the treatment of random oracle queries. Namely, the first $\kappa^\star - 1$ hash queries – which are necessarily the same as in the first execution because $\mathcal{A}$'s random tape has not changed – receive the same answers $r_1,\ldots,r_{\kappa^\star-1}$ as in the first run. Consequently, the $\kappa^\star$-th query will involve the tuple $(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell)$ as in the first execution. However, a forking occurs as, from this point forward, $\mathcal{A}$ obtains fresh random oracle values $r'_{\kappa^\star},\ldots,r'_{q_H}$ which are independent of the subsequence of answers in the first execution. The General Forking Lemma of Bellare and Neven [BN06] implies that, with probability at least $\left(\varepsilon - \frac{1}{2^\lambda}\right)\left(\frac{\varepsilon - 1/2^\lambda}{q_H} - \frac{1}{2^\lambda}\right)$, it holds that: (1) $\mathcal{A}$'s forgery also pertains to $(M^\star, \mathsf{Comm}_K^\star, \{\mathbf{c}_i^\star\}_{i=0}^\ell, \pi_0^\star, \{\pi_{\mathrm{OR},i}^\star\}_{i=1}^\ell)$ in the second run; (2) we also have $r'_{\kappa^\star} \neq r_{\kappa^\star}$. Hence, using the knowledge extractor of the proof of knowledge $\pi_K^\star$, $\mathcal{B}$ extracts vectors $\mathbf{e}_1,\ldots,\mathbf{e}_\ell \in D_{\mathbb{Z}^{2m},\alpha q}$ and $\mathbf{x}_1 \in \mathbb{Z}^m$, $\mathbf{y}_1,\ldots,\mathbf{y}_\ell \in \mathbb{Z}^{2m}$ satisfying

$$\mathbf{x}_1^T\mathbf{A} + \sum_{i=0}^{\ell} \mathbf{c}_i^T\mathbf{A}_i = \sum_{i=1}^{\ell} \mathbf{e}_i^T\left(p\cdot\mathbf{A}_i\right) \quad \text{and} \quad \mathbf{e}_i^T\left(p\cdot\mathbf{A}_i\right) + \mathbf{y}_i^T\mathbf{A}_i = \mathbf{c}_i^T\mathbf{A}_i, \text{ for } i \in \{1,\ldots,\ell\} \tag{7.8}$$

with $||\mathbf{x}_1|| \leq \sigma\sqrt{m}$ and $||\mathbf{y}_i|| \leq \sigma\sqrt{2m}$ for each $i \in \{1,\ldots,\ell\}$.

The reduction $\mathcal{B}$ then opens either of the two forgeries using $\{\mathbf{S}_i\}_{i=0}^\ell$ (note that both signatures necessarily open to the same identity id as they involve the same $\{\mathbf{c}_i^\star\}_{i=1}^\ell$). At this point, $\mathcal{B}$ aborts and declares failure if the opening does not unveil user $j^\star$'s identity. Still, with probability at least $1/N$, $\mathcal{B}$'s was fortunate in its random choice for $j^\star$ and the opening algorithm reveals $\mathrm{id}_{j^\star}$.

If this desirable event occurs, $\mathcal{B}$ considers the following situations.

- If $\mathbf{y}_i = \mathrm{id}_{j^\star}[i]\cdot\mathbf{x}_2$ for each $i \in \{1,\ldots,\ell\}$, where $\mathbf{x}_2 \in \mathbb{Z}^{2m}$ is the vector encrypted by $\mathbf{c}_0^\star$, $\mathcal{B}$ aborts if $coin = 1$. Otherwise, relations (7.8) guarantee that

$$\begin{aligned}
\mathbf{x}_1^T\cdot\mathbf{A} + \mathbf{c}_0^T\cdot\mathbf{A}_0 + \sum_{i=1}^{\ell}\mathrm{id}_{j^\star}[i]\cdot\mathbf{x}_2^T\cdot\mathbf{A}_i &= \mathbf{x}_1^T\cdot\mathbf{A} + \mathbf{x}_2^T\cdot\mathbf{A}_0 + \sum_{i=1}^{\ell}\mathrm{id}_{j^\star}[i]\cdot\mathbf{x}_2^T\cdot\mathbf{A}_i \\
&= (\mathbf{x}_1||\mathbf{x}_2)^T\cdot\left[\frac{\mathbf{A}}{\mathbf{A}_0 + \sum_{i=1}^{\ell}\mathrm{id}_{j^\star}[i]\cdot\mathbf{A}_i}\right] \\
&= (\mathbf{x}_1||\mathbf{x}_2)^T\cdot\left[\frac{\bar{\mathbf{A}}}{(\mathbf{Q}_0 + \sum_{i=1}^{\ell}\mathrm{id}_{j^\star}[i]\mathbf{Q}_i)\cdot\bar{\mathbf{A}}}\right] \\
&= 0 \bmod q,
\end{aligned}$$

where the first equality follows from the fact that $\mathbf{B}_0^T\cdot\mathbf{A}_0 = 0 \bmod q$ and $\mathbf{c}_0^\star$ is of the form $\mathbf{c}_0^\star = \mathbf{B}_0\cdot\mathbf{s}_0 + \mathbf{x}_2$. This implies that $\mathbf{v} = \mathbf{x}_1 + \mathbf{x}_2\cdot\left(\mathbf{Q}_0 + \sum_{i=1}^{\ell}\mathrm{id}_{j^\star}[i]\mathbf{Q}_i\right)$ is a vector of $\Lambda^\perp(\bar{\mathbf{A}})$.

A similar analysis to [Boy10] shows that $\mathbf{v}$ is both short and non-zero with overwhelming probability. As a consequence, $\mathcal{B}$ outputs $(\mathbf{x}_2 \| 0^m)$ which is a short non-zero vector such that $(\mathbf{x}_2 \| 0^m)^T \cdot \hat{\mathbf{A}} = 0 \bmod q$.

- If there exists $i \in \{1, \ldots, \ell\}$ such that $\mathbf{y}_i \neq \mathrm{id}_{j^\star}[i] \cdot \mathbf{x}_2$, where $\mathbf{x}_2 \in \mathbb{Z}^{2m}$ is the vector obtained by decrypting $\mathbf{c}_0^\star$ using $\mathbf{S}_0$, then $\mathcal{B}$ aborts if $coin = 0$. Otherwise, the non-interactive proofs $\{\pi^\star_{\mathrm{OR},i}\}_i$ imply that $\mathbf{c}_i^\star = \mathbf{B}_i \cdot \mathbf{s} + p \cdot \mathbf{e}'_i + \mathrm{id}_{j^\star}[i] \cdot \mathbf{x}_2$ for some $\mathbf{x}_2, \mathbf{e}'_1, \ldots, \mathbf{e}'_\ell \in \mathbb{Z}^{2m}$ and $\mathbf{s}_0, \mathbf{s} \in \mathbb{Z}_q^n$. By multiplying the latter expression of $\mathbf{c}_i^{\star T}$ by $\mathbf{A}_i$, we find

$$\mathbf{c}_i^T \cdot \mathbf{A}_i \quad = \quad p \cdot (\mathbf{e}_i'^T \cdot \mathbf{A}_i) + \mathrm{id}_{j^\star}[i] \cdot \mathbf{x}_2^T \mathbf{A}_i.$$

Subtracting the latter equation from the second equation of (7.8), we find

$$\big(p \cdot (\mathbf{e}_i^T - \mathbf{e}_i'^T) + (\mathbf{y}_i^T - \mathrm{id}_{j^\star}[i] \cdot \mathbf{x}_2^T)\big) \cdot \mathbf{A}_i \quad = \quad 0 \bmod q.$$

If $p \cdot (\mathbf{e}_i^T - \mathbf{e}_i'^T) + (\mathbf{y}_i^T - \mathrm{id}_{j^\star}[i] \cdot \mathbf{x}_2^T) \neq 0 \bmod q$, it is a short non-zero vector in $\Lambda^\perp(\mathbf{A}_i)$. Given that $\mathbf{A}_i = \hat{\mathbf{A}}$ with probability $1/\ell$, we solved the given SIS instance with the same probability. Finally, if

$$p \cdot (\mathbf{e}_i^T - \mathbf{e}_i'^T) + (\mathbf{y}_i^T - \mathrm{id}_{j^\star}[i] \cdot \mathbf{x}_2^T) = 0 \bmod q,$$

the relative lengths of vectors $\mathbf{e}_i, \mathbf{e}'_i, \mathbf{y}_i, \mathbf{x}_2$ with respect to p implies $\mathbf{e}_i = \mathbf{e}'_i$ and $\mathbf{y}_i = \mathrm{id}_{j^\star}[i] \cdot \mathbf{x}_2$, which contradicts the assumption that $\mathbf{y}_i \neq \mathrm{id}_{j^\star}[i] \cdot \mathbf{x}_2$.

The lower bound on the reduction's probability of success is assessed exactly in the same way as in the proof of Theorem 7.8. □

Chapter 8

A Lattice-Based Group Signature with Verifier-Local Revocation

In this chapter, which corresponds to a joint work with S. Ling, K. Nguyen and H. Wang published in [LLNW14], we introduce a group signature with verifier-local revocation from lattice assumptions (defined in Section 6.4). In comparison with known lattice-based group signatures, while the schemes from [GKV10], [CNR12] and Chapter 7 follow the CPA-*anonymity* and CCA-*anonymity* notions from [BBS04, BMW03], our construction satisfies the (weaker) notion of *selfless-anonymity* for VLR group signatures from [BS04]. Nevertheless, our scheme has several advantages over the contemporary counterparts:

- Functionality: Our scheme is the first lattice-based group signature that supports membership revocation. This is a desirable functionality for any group signature scheme.
- Simplicity: Our scheme is conceptually simple. The signature is basically an all-in-one proof of knowledge, made non-interactive using Fiat-Shamir paradigm [FS86]. Moreover, the scheme departs from the traditional paradigm, and is free of LWE-based encryptions.
- Efficiency: For a security parameter λ and for a group of N members, the group public key and the signature have bit-sizes $\widetilde{\mathcal{O}}(\lambda^2) \cdot \log N$ and $\widetilde{\mathcal{O}}(\lambda) \cdot \log N$, respectively. This result is comparable to that of Chapter 7, and is a noticeable improvement over those of [GKV10] and [CNR12].
- Security assumption: Our scheme is proved to be secure (in the random oracle model) based on the worst-case hardness of approximating the Shortest Independent Vectors Problem, for general lattices of dimension n, to within a factor $\gamma = \widetilde{\mathcal{O}}(n^{1.5})$.

Overview of Our Techniques. The main building block of our VLR group signature scheme is an interactive protocol allowing a prover to convince the verifier that he is a certified group member (i.e., he possesses a valid secret signing key), and that he has not been revoked (i.e., his "revocation token" is not in the verifier's blacklist). The protocol is repeated many times to make the soundness error negligibly small, and then is converted to a signature scheme via the Fiat-Shamir heuristic. Roughly speaking, in the random oracle model, the traceability and anonymity of the resulting group signature are based on the facts that the underlying protocol is a proof of knowledge, and that it can be simulated.

We consider a group of $N = 2^\ell$ users, where each user is identified by a string $d \in \{0,1\}^\ell$ denoting the binary representation of his index in the group. Let n, m, β, and $q \geq 2$ be integers (to be determined later). Our scheme operates within the structure of a *Bonsai tree* of hard random lattices [CHKP10] and described in Chapter 3, namely, a matrix $\mathbf{A}$ as described in Figure 8.1 and a vector $\mathbf{u} \in \mathbb{Z}_q^n$. Initially, the group user with identity $d = d[1] \dots d[\ell] \in \{0,1\}^\ell$ is issued a

$$\mathbf{A} = \left[\begin{array}{c} \mathbf{A}_0 \\ \hline \mathbf{A}_1^0 \\ \hline \mathbf{A}_1^1 \\ \hline \vdots \\ \hline \mathbf{A}_\ell^0 \\ \hline \mathbf{A}_\ell^1 \end{array}\right] \in \mathbb{Z}_q^{(2\ell+1)m \times n}, \mathbf{A}_d = \left[\begin{array}{c} \mathbf{A}_0 \\ \hline \mathbf{A}_1^d[1] \\ \hline \vdots \\ \hline \mathbf{A}_\ell^d[\ell] \end{array}\right] \in \mathbb{Z}_q^{(\ell+1)m \times n}.$$

Figure 8.1: Matrices $\mathbf{A}$ and $\mathbf{A}_d$.

Bonsai signature of his identity, see Section 3.2.4, that is a small vector $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$, such that $\|\mathbf{z}\|_\infty \leq \beta$ and $\mathbf{z}^T \mathbf{A}_d = \mathbf{u}^T \bmod q$, where $\mathbf{A}_d$, defined in Figure 8.1, is a subtree defined by d. In other words, $\mathbf{z}$ is a solution to the Inhomogeneous Small Integer Solution (ISIS, see Section 2.1 of Chapter 2) instance $(\mathbf{A}_d, \mathbf{u})$. To prove that he is a certified group member without leaking $\mathbf{z}$, the user can perform a proof of knowledge (e.g., [MV03, Lyu08, LNSW13]) to convince the verifier that he knows such a vector $\mathbf{z}$ in zero-knowledge.

At this stage, one can obtain a secure identity-based identification scheme (as shown in [Rüc10a]), but it is insufficient for our purposes: to achieve anonymity, the group user also has to *hide* his identity d, and hence the matrix $\mathbf{A}_d$ should not be explicitly given. This raises an interesting question: If the verifier does not know $\mathbf{A}_d$, how could he be convinced that $\mathbf{z}^T \cdot \mathbf{A}_d = \mathbf{u}^T \bmod q$? To address this issue, we introduce the following extension: we add ℓ suitable *zero-blocks* of size m to vector $\mathbf{z}$ to obtain an extended vector $\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \mathbb{Z}^{(2\ell+1)m}$, where the added zero-blocks are $\mathbf{x}_1^{1-d[1]}, \dots, \mathbf{x}_\ell^{1-d[\ell]}$. We then have $\|\mathbf{x}\|_\infty \leq \beta$, and $\mathbf{x}^T \cdot \mathbf{A} = \mathbf{u}^T \bmod q$. Namely $\mathbf{x}$ is a solution to the ISIS instance given by the *whole* Bonsai tree, with an additional condition: for each $i \in \{1, \dots, \ell\}$, one of the two blocks $\mathbf{x}_i^0, \mathbf{x}_i^1$ must be zero, where the arrangement of the zero-blocks is determined by d. To prove in zero-knowledge the possession of such a vector $\mathbf{x}$, we adapt the "Stern Extension" proof system from [LNSW13], where the user identity d is hidden by a *"one-time pad"* technique. This technique is as follows. In each round of the protocol, the user samples a fresh uniformly random $e \in \{0,1\}^\ell$ and permutes the blocks of $\mathbf{x}$ to obtain the permuted vector $\mathbf{v}$, whose zero-blocks are arranged according to $d \oplus e$ (where $\oplus$ denotes the bit XOR operation). Depending on the verifier's challenge, the user later will either reveal e, or reveal $d \oplus e$ and show that $\mathbf{v}$ has the correct shape determined by $d \oplus e$. Since $d \oplus e$ is uniformly random over $\{0,1\}^\ell$, the user identity d is completely hidden. As a result, the user can anonymously prove his group membership.

We now briefly review our revocation mechanism. For each group user's secret key $\mathbf{x}$, consider the first block $\mathbf{x}_0$ that corresponds to the "root" $\mathbf{A}_0$ of the Bonsai tree, and let his revocation token be $\mathbf{x}_0^T \cdot \mathbf{A}_0 \bmod q \in \mathbb{Z}_q^n$. We choose suitable parameters, and sample $\mathbf{x}_0$ from a proper distribution, so that the token is statistically close to uniform over $\mathbb{Z}_q^n$. At a high level, our revocation mechanism works as follows. The user is asked to sample a uniformly random vector $\mathbf{r}_0 \in \mathbb{Z}_q^m$, and to compute a commitment $\mathbf{c}_0$ using a (lattice-based) statistically hiding and computationally binding string commitment scheme COM, for which the value $\mathbf{r}_0^T \cdot \mathbf{A}_0 \bmod q$ is part of the committed string. Depending on the verifier's challenge, the user will either reveal $\mathbf{r}_0$

or reveal $\mathbf{x}_0 + \mathbf{r}_0$. In the former case, the verifier can check for honest computation of $\mathbf{c}_0$, while in the latter case, he can perform the revocation check using a list of tokens of revoked users $RL = \{\{\mathbf{u}_i\}_i\} \subset \mathbb{Z}_q^n$, as follows:

$$\forall\, \mathbf{u}_i \in RL,\ \text{check that } \mathbf{c}_0 \neq \mathsf{COM}\big((\mathbf{x}_0 + \mathbf{r}_0)^T \cdot \mathbf{A}_0 - \mathbf{u}_i^T \bmod q\big).$$

Assuming that the user has been revoked, i.e., there exists i such that $\mathbf{x}_0^T \cdot \mathbf{A}_0 \bmod q = \mathbf{u}_i^T$. If he follows the protocol, then $\mathsf{COM}\big((\mathbf{x}_0 + \mathbf{r}_0)^T \cdot \mathbf{A}_0 - \mathbf{u}_i^T \bmod q\big) = \mathsf{COM}(\mathbf{r}_0^T \cdot \mathbf{A}_0 \bmod q) = \mathbf{c}_0$, and thus, he gets rejected. If there is a false acceptance, then we can use it to break the computational binding property of COM. On the other hand, the probability of false rejection is negligibly small, since COM is statistically regular.

Putting everything together, we obtain a lattice-based VLR group signature that has several nice features, as mentioned earlier. In the process, we exploit the rich structure of the Bonsai tree [CHKP10], and the versatility of the "Stern Extension" proof system [LNSW13]. We also employ a special "one-time pad" technique, and a novel revocation mechanism.

8.1 Preparation

We now describe the parameters and some specific constructions that will be used in our scheme.

8.1.1 Parameters

Our group signature scheme involves two main parameters: a security parameter n and a maximum expected number of group users $N = 2^\ell \in \mathsf{poly}(n)$. Given n, we fix the other scheme parameters as in Table 8.1.

Parameter	**Value** or **Asymptotic bound**
Modulus q	$\omega(n^2 \log n)$
Dimension m	$\geq 2n \log q$
Gaussian parameter σ	$\omega(\sqrt{n \log q \log n})$
Integer norm bound β	$\lceil \sigma \cdot \log m \rceil$
Number of 'decompositions' p	$\lfloor \log \beta \rfloor + 1$
Sequence of integers $\beta_1, \beta_2, \beta_3, \ldots, \beta_p$	$\beta_1 = \lceil \beta/2 \rceil; \beta_2 = \lceil (\beta - \beta_1)/2 \rceil$ $\beta_3 = \lceil (\beta - \beta_1 - \beta_2)/2 \rceil; \ldots; \beta_p = 1$
Number of protocol repetitions t	$\omega(\log n)$

Table 8.1: Parameters of our VLR group signature scheme. The sequence $\beta_1, \beta_2, \ldots, \beta_p$ satisfies $\sum_{j=1}^{p} \beta_j = \beta$, and every integer in the interval $[-\beta, \beta]$ can be efficiently expressed as a subset sum of elements in the set $\{\pm\beta_1, \pm\beta_2, \ldots, \pm\beta_p\}$.

8.1.2 Some Specific Sets

We now define some specific sets of vectors and permutations that will be extensively used throughout this work. First, we denote by B_{3m} the set of all vectors in $\{-1, 0, 1\}^{3m}$ having

exactly m coordinates -1; m coordinates 0; and m coordinates 1. Given a binary string $d = d[1] \dots d[\ell] \in \{0,1\}^\ell$, we define two sets:

- $\mathsf{Secret}_\beta(d)$: The set of all vectors $\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \mathbb{Z}^{(2\ell+1)m}$ consisting of $2\ell+1$ *blocks* of size m, such that $\|\mathbf{x}\|_\infty \leq \beta$, and the following ℓ blocks are *zero-blocks* $\mathbf{0}^m$: $\mathbf{x}_1^{1-d[1]}, \dots, \mathbf{x}_\ell^{1-d[\ell]}$.
- $\mathsf{SecretExt}(d)$: The set of all vectors $\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \{-1,0,1\}^{(2\ell+1)3m}$ consisting of $2\ell+1$ *blocks* of size $3m$, such that the $\ell+1$ blocks $\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \dots, \mathbf{x}_\ell^{d[\ell]}$ are elements of B_{3m}, and the remaining ℓ blocks $\mathbf{x}_1^{1-d[1]}, \dots, \mathbf{x}_\ell^{1-d[\ell]}$ are *zero-blocks* $\mathbf{0}^{3m}$.

Given a vector $\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \mathbb{Z}^{(2\ell+1)3m}$ consisting of $2\ell+1$ blocks of size $3m$, we define two sets of permutations of $\mathbf{x}$:

- The set $\mathcal{S}$ of all permutations that keep the arrangement of the blocks. Specifically, if $\pi \in \mathcal{S}$, then
$$\pi(\mathbf{x}) = \left(\tau_0(\mathbf{x}_0) \| \tau_1^0(\mathbf{x}_1^0) \| \tau_1^1(\mathbf{x}_1^1) \| \dots \| \tau_\ell^0(\mathbf{x}_\ell^0) \| \tau_\ell^1(\mathbf{x}_\ell^1)\right),$$
where $\tau_0, \tau_1^0, \tau_1^1, \dots, \tau_\ell^0, \tau_\ell^1$ are certain permutations of $3m$ elements.
- The set $\mathcal{T} = \{T_e \,|\, e \in \{0,1\}^\ell\}$, where for $e = e[1] \dots e[\ell]$, $T_e \in \mathcal{T}$ *rearranges* the blocks as follows:
$$T_e(\mathbf{x}) = \left(\mathbf{x}_0 \| \mathbf{x}_1^{e[1]} \| \mathbf{x}_1^{1-e[1]} \| \dots \| \mathbf{x}_\ell^{e[\ell]} \| \mathbf{x}_\ell^{1-e[\ell]}\right).$$

In particular, given $d, e \in \{0,1\}^\ell$, $\pi \in \mathcal{S}$, and $\mathbf{x} \in \mathbb{Z}^{(2\ell+1)3m}$, it can be checked that:

$$\mathbf{x} \in \mathsf{SecretExt}(d) \Leftrightarrow \pi(\mathbf{x}) \in \mathsf{SecretExt}(d) \Leftrightarrow T_e \circ \pi(\mathbf{x}) \in \mathsf{SecretExt}(d \oplus e). \tag{8.1}$$

8.1.3 The Decomposition - Extension Technique

Ling et al. [LNSW13] proposed a Stern-type zero-knowledge proof of knowledge for the $\mathrm{ISIS}^\infty_{q,m,\beta}$ problem that enjoys a strong security guarantee: the best way to break their protocol is to solve the underlying ISIS problem. They achieve this feature by using a versatile Decomposition-Extension framework. Adapting their technique, we construct the following procedures:

Elementary Decomposition. On input a vector $\mathbf{v} = (v_1, v_2, \dots, v_m) \in \mathbb{Z}^m$ such that $\|\mathbf{v}\|_\infty \leq \beta$, the procedure EleDec outputs $p = \lfloor \log \beta \rfloor + 1$ vectors $\widetilde{\mathbf{w}}_1, \dots, \widetilde{\mathbf{w}}_p \in \{-1,0,1\}^m$, such that $\sum_{j=1}^p \beta_j \cdot \widetilde{\mathbf{w}}_j = \mathbf{v}$. This procedure works as follows:

1. For each $i \in \{1, \dots, m\}$, express v_i as $v_i = \beta_1 \cdot v_{i,1} + \beta_2 \cdot v_{i,2} + \dots + \beta_p \cdot v_{i,p}$, where $\forall j \in \{1, \dots, p\} : v_{i,j} \in \{-1,0,1\}$. It was noted in [LNSW13] that for $\beta_1, \beta_2, \dots, \beta_p$ given in Table 8.1, this step can easily be done.
2. For each $j \in \{1, \dots, p\}$, let $\widetilde{\mathbf{w}}_j := (v_{1,j}, v_{2,j}, \dots, v_{m,j}) \in \{-1,0,1\}^m$. Output $\widetilde{\mathbf{w}}_1, \dots, \widetilde{\mathbf{w}}_p$.

Elementary Extension. On input a vector $\widetilde{\mathbf{w}} \in \{-1,0,1\}^m$, the procedure EleExt extends $\widetilde{\mathbf{w}}$ to a vector $\mathbf{w} \in \mathsf{B}_{3m}$. This procedure works as follows:

1. Let $\lambda^{(-1)}$, $\lambda^{(0)}$ and $\lambda^{(1)}$ be the numbers of coordinates of $\widetilde{\mathbf{w}}$ that equal to -1, 0, and 1 respectively.
2. Pick a random vector $\widehat{\mathbf{w}} \in \{-1,0,1\}^{2m}$ that has exactly $(m - \lambda^{(-1)})$ coordinates -1, $(m - \lambda^{(0)})$ coordinates 0, and $(m - \lambda^{(1)})$ coordinates 1. Output $\mathbf{w} = \left(\widetilde{\mathbf{w}} \| \widehat{\mathbf{w}}\right) \in \mathsf{B}_{3m}$.

Witness Decomposition and Extensions. On input $\mathbf{x} \in \mathsf{Secret}_\beta(d)$ for some $d = d[1] \dots d[\ell] \in \{0,1\}^\ell$, the procedure $\mathsf{WitnessDE}$ outputs p vectors $\mathbf{z}_1, \dots \mathbf{z}_p \in \mathsf{SecretExt}(d)$. This procedure works as follows:

1. Write $\mathbf{x}$ as the concatenation of $2\ell+1$ blocks of size m, namely: $\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right)$.

2. Run EleDec on each of the $\ell+1$ blocks $\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \dots, \mathbf{x}_\ell^{d[\ell]}$ to obtained $(\ell+1)p$ decomposed vectors. Then run EleExt on each of the decomposed vectors to obtain $(\ell+1)p$ vectors in B_{3m}, denoted respectively by $\{\mathbf{w}_{0,j}\}_{j=1}^p, \{\mathbf{w}_{1,j}^{d[1]}\}_{j=1}^p, \dots, \{\mathbf{w}_{\ell,j}^{d[\ell]}\}_{j=1}^p$.

3. Create ℓp *zero-vectors* of dimension $3m$, and denote them by $\{\mathbf{w}_{1,j}^{1-d[1]}\}_{j=1}^p, \dots, \{\mathbf{w}_{\ell,j}^{1-d[\ell]}\}_{j=1}^p$.

4. For each $j \in \{1, \dots, p\}$, let $\mathbf{z}_j = \left(\mathbf{w}_{0,j} \| \mathbf{w}_{1,j}^0 \| \mathbf{w}_{1,j}^1 \| \dots \| \mathbf{w}_{\ell,j}^0 \| \mathbf{w}_{\ell,j}^1\right)$. Output $\mathbf{z}_1, \dots, \mathbf{z}_p \in \mathsf{SecretExt}(d)$.

Matrix Extension. On input matrix $\mathbf{A} \in \mathbb{Z}_q^{(2\ell+1)m \times n}$, the following procedure $\mathsf{MatrixExt}$ outputs matrix $\mathbf{A}^* \in \mathbb{Z}_q^{(2\ell+1)3m \times n}$:

1. Write $\mathbf{A}$ as the concatenation of $2\ell+1$ component-matrices in $\mathbb{Z}_q^{m \times n}$.

2. Append $2m$ *zero-columns* to each of the component-matrices, then output the extended matrix $\mathbf{A}^*$.

In particular, let $\{\mathbf{z}_j\}_{j=1}^p \leftarrow \mathsf{WitnessDE}(\mathbf{x})$ and $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$ then we have $\mathbf{x}^T \cdot \mathbf{A} = (\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j)^T \cdot \mathbf{A}^*$. We illustrate our Decomposition-Extension technique in Figure 8.2, where the first bit of d is 1 and its last bit is 0. After performing Decomposition-Extension, one has that $\mathbf{z}_j \in \mathsf{SecretExt}(d)$ for all $j \in \{1, \dots, p\}$, and $\left(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j\right)^T \cdot \mathbf{A}^* = \mathbf{x}^\mathbf{T} \cdot \mathbf{A} = \mathbf{u}^T \bmod q$.

Therefore, in the protocol in Section 8.2, in order to prove that $\mathbf{x} \in \mathsf{Secret}_\beta(d)$ for some $d \in \{0,1\}^\ell$, and $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$, one can instead prove that:

$$(\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j)^T \cdot \mathbf{A}^* = \mathbf{u}^T \bmod q \quad \text{and} \quad \forall j \in \{1, \dots, p\}, \pi \in \mathcal{S}, e \in \{0,1\}^\ell : T_e \circ \pi(\mathbf{z}_j) \in \mathsf{SecretExt}(d \oplus e),$$

where the latter relation follows from the fact that $\mathbf{z}_j \in \mathsf{SecretExt}(d)$ for all $j \in \{1, \dots, p\}$, and from Equation (8.1).

8.2 The Underlying interactive protocol

We recall that the main building block of our VLR group signature scheme is an interactive protocol that allows the prover to convince the verifier that he is a certified group member (i.e., he has a valid secret key), and that he has not been revoked (i.e., his revocation token is not in the verifier's list RL). In Section 8.3, the protocol is repeated $t = \omega(\log n)$ times to make the soundness error negligibly small, and then is transform to a signature scheme via Fiat-Shamir heuristic. The interactive protocol is summarized as follows:

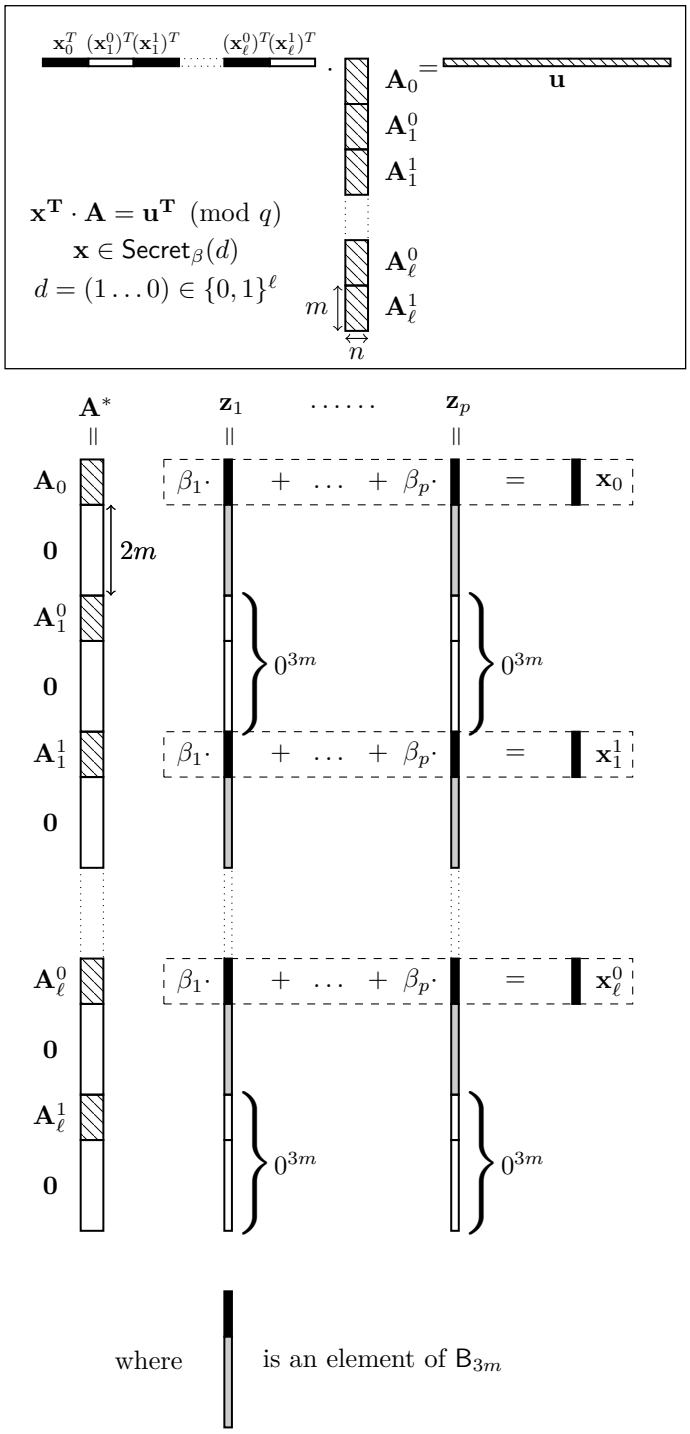

Figure 8.2: An illustration of our Decomposition-Extension technique.

- The public parameters are

$$\mathbf{A} = \left[\begin{array}{c} \mathbf{A}_0 \\ \hline \mathbf{A}_1^0 \\ \hline \mathbf{A}_1^1 \\ \hline \vdots \\ \hline \mathbf{A}_\ell^0 \\ \hline \mathbf{A}_\ell^1 \end{array}\right] \in \mathbb{Z}_q^{(2\ell+1)m \times n}. \text{ and } \mathbf{u} \in \mathbb{Z}_q^n.$$

- The prover's witness is a $\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \mathsf{Secret}_\beta(d)$ for some $d \in \{0,1\}^\ell$. The verifier's additional input is a set $RL = \left\{\{\mathbf{u}_i\}_i\right\} \subset \mathbb{Z}_q^n$, whose cardinality is at most $N-1$.

- The prover's goal is to convince the verifier in that:

  1. $\mathbf{x}^T \cdot \mathbf{A} = \mathbf{u}^T \bmod q$ and $\mathbf{x} \in \mathsf{Secret}_\beta(d)$, while keeping d *secret*.
  2. $\mathbf{x}_0^T \cdot \mathbf{A}_0 \bmod q \notin RL$.

8.2.1 Description of the Protocol

Let COM be the KTX commitment scheme [KTX08]. Let $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$. Prior to the interaction, the prover applies the Decomposition-Extension technique on his witness: Let $\mathbf{z}_1, \ldots, \mathbf{z}_p \leftarrow \mathsf{WitnessDE}(\mathbf{x})$. The protocol follows Stern's approach for three-pass zero-knowledge identification schemes [Ste96], for which we employ an additional commitment $\mathbf{c}_0$ to enable the revocation mechanism. It is described in Figure 8.3.

8.2.2 Witness Extraction

The following lemma says that in our protocol, one can extract a satisfying witness under specific conditions.

Lemma 8.1. *Assume that for a given commitment* CMT*, there exist* 3 valid *responses* $\mathrm{RSP}^{(1)}$, $\mathrm{RSP}^{(2)}$*, and* $\mathrm{RSP}^{(3)}$ *corresponding to all* 3 *possible values of the challenge* Ch*. If* COM *is a computationally binding commitment scheme, then one can efficiently extract a vector* $\mathbf{y} = \left(\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1\right) \in \mathbb{Z}^{(2\ell+1)m}$ *satisfying* $\mathbf{y}^T \cdot \mathbf{A} = \mathbf{u}^T \bmod q$*,* $\mathbf{y} \in \mathsf{Secret}_\beta(d)$ *for some* $d \in \{0,1\}^\ell$*, and* $\mathbf{y}_0^T \cdot \mathbf{A}_0 \bmod q \notin RL$*.*

Proof. Let $\mathrm{CMT} = \left(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\right) \in (\mathbb{Z}_q^n)^4$, and let $\mathrm{RSP}^{(1)}$, $\mathrm{RSP}^{(2)}$, $\mathrm{RSP}^{(3)}$ as in (8.3), (8.4), and (8.5), respectively. Since all 3 responses satisfy the verification conditions, the followings are true:

$$\begin{cases} \forall j \in \{1,\ldots,p\} : \ \mathbf{v}_j \in \mathsf{SecretExt}(d_1); \ \ \mathbf{c}_0 = \mathsf{COM}(d_3, \ \{\psi_j\}_{j=1}^p, \ \left(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_{j,0}\right)^T \cdot \mathbf{A}_0 \bmod q); \\ \forall \mathbf{u}_i \in RL : \ \mathbf{c}_0 \neq \mathsf{COM}\left(d_2, \ \{\phi_j\}_{j=1}^p, \ \left(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\right)^T \cdot \mathbf{A}_0 - \mathbf{u}_i^T \bmod q\right); \\ \mathbf{c}_1 = \mathrm{COM}\Big(d_2, \ \{\phi_j\}_{j=1}^p, \ \left(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j\right)^T \cdot \mathbf{A}^* - \mathbf{u}^T\Big) = \mathrm{COM}\Big(d_3, \ \{\psi_j\}_{j=1}^p, \ \left(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_j\right)^T \cdot \mathbf{A}^*\Big); \\ \mathbf{c}_2 = \mathrm{COM}\left(\{\mathbf{w}_j\}_{j=1}^p\right) = \mathrm{COM}\Big(\{\mathrm{T}_{d_3} \circ \psi_j(\mathbf{h}_j)\}_{j=1}^p\Big); \\ \mathbf{c}_3 = \mathrm{COM}\left(\{\mathbf{v}_j + \mathbf{w}_j\}_{j=1}^p\right) = \mathrm{COM}\Big(\{\mathrm{T}_{d_2} \circ \phi_j(\mathbf{s}_j)\}_{j=1}^p\Big). \end{cases}$$

1. **Commitment:** The prover samples a string $e \hookleftarrow U(\{0,1\}^{\ell})$, p permutations $\pi_1, \ldots, \pi_p \hookleftarrow U(\mathcal{S})$, and p vectors $\mathbf{r}_1, \ldots, \mathbf{r}_p \hookleftarrow U(\mathbb{Z}_q^{(2\ell+1)\cdot 3m})$. For each $j \in \{1, \ldots, p\}$, let $\mathbf{r}_{j,0} = \mathsf{Parse}(\mathbf{r}_j, 1, m)$. Then it sends the commitment $\mathrm{CMT} = \big(\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3\big) \in (\mathbb{Z}_q^n)^4$ to the verifier, where

$$\begin{cases} \mathbf{c}_0 = \mathsf{COM}\big(e,\ \{\pi_j\}_{j=1}^p,\ \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big)^T \cdot \mathbf{A}_0 \bmod q\big), \\ \mathbf{c}_1 = \mathsf{COM}\big(e,\ \{\pi_j\}_{j=1}^p,\ \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j\big)^T \cdot \mathbf{A}^* \bmod q\big), \\ \mathbf{c}_2 = \mathsf{COM}\big(\{\mathrm{T}_e \circ \pi_j(\mathbf{r}_j)\}_{j=1}^p\big), \\ \mathbf{c}_3 = \mathsf{COM}\big(\{\mathrm{T}_e \circ \pi_j(\mathbf{z}_j + \mathbf{r}_j)\}_{j=1}^p\big). \end{cases} \tag{8.2}$$

2. **Challenge:** The verifier sends a challenge $Ch \hookleftarrow U(\{1,2,3\})$ to the prover.

3. **Response:** Depending on the challenge, the prover computes the response RSP differently:

   - Case $Ch = 1$: $\forall\, j \in \{1, \ldots, p\}$, let $\mathbf{v}_j = \mathrm{T}_e \circ \pi_j(\mathbf{z}_j)$, $\mathbf{w}_j = \mathrm{T}_e \circ \pi_j(\mathbf{r}_j)$, $d_1 = d \oplus e$, and set:
   $$\mathrm{RSP} = \big(d_1,\ \{\mathbf{v}_j\}_{j=1}^p,\ \{\mathbf{w}_j\}_{j=1}^p\big). \tag{8.3}$$
   - Case $Ch = 2$: $\forall\, j \in \{1, \ldots, p\}$, let $\phi_j = \pi_j$, $\mathbf{s}_j = \mathbf{z}_j + \mathbf{r}_j$, $d_2 = e$, and set:
   $$\mathrm{RSP} = \big(d_2,\ \{\phi_j\}_{j=1}^p,\ \{\mathbf{s}_j\}_{j=1}^p\big). \tag{8.4}$$
   - Case $Ch = 3$: $\forall\, j \in \{1, \ldots, p\}$, let $\psi_j = \pi_j$, $\mathbf{h}_j = \mathbf{r}_j$, $d_3 = e$, and set:
   $$\mathrm{RSP} = \big(d_3,\ \{\psi_j\}_{j=1}^p,\ \{\mathbf{h}_j\}_{j=1}^p\big). \tag{8.5}$$

Verification: Receiving the response RSP, the verifier proceeds as follows:

- Case $Ch = 1$: Parse RSP as in (8.3). Check that $\forall j \in \{1, \ldots, p\}$: $\mathbf{v}_j \in \mathsf{SecretExt}(d_1)$, and that:
$$\mathbf{c}_2 = \mathsf{COM}\big(\{\mathbf{w}_j\}_{j=1}^p\big) \text{ and } \mathbf{c}_3 = \mathsf{COM}\big(\{\mathbf{v}_j + \mathbf{w}_j\}_{j=1}^p\big).$$
- Case $Ch = 2$: Parse RSP as in (8.4). $\forall j \in \{1, \ldots, p\}$, let $\mathbf{s}_{j,0} = \mathsf{Parse}(\mathbf{s}_j, 1, m)$. Check that:
$$\begin{cases} \forall \mathbf{u}_i \in RL :\ \mathbf{c}_0 \neq \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big)^T \cdot \mathbf{A}_0 - \mathbf{u}_i^T \bmod q\big) \\ \mathbf{c}_1 = \mathsf{COM}\big(d_2,\ \{\phi_j\}_{j=1}^p,\ \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j\big)^T \cdot \mathbf{A}^* - \mathbf{u}^T \bmod q\big);\ \ \mathbf{c}_3 = \mathsf{COM}\big(\{\mathrm{T}_{d_2} \circ \phi_j(\mathbf{s}_j)\}_{j=1}^p\big). \end{cases}$$
- Case $Ch = 3$: Parse RSP as in (8.5). $\forall j \in \{1, \ldots, p\}$, let $\mathbf{h}_{j,0} = \mathsf{Parse}(\mathbf{h}_j, 1, m)$. Check that:
$$\begin{cases} \mathbf{c}_0 = \mathsf{COM}(d_3,\ \{\psi_j\}_{j=1}^p,\ \big(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_{j,0}\big)^T \cdot \mathbf{A}_0 \bmod q\big) \\ \mathbf{c}_1 = \mathsf{COM}\big(d_3,\ \{\psi_j\}_{j=1}^p,\ \big(\sum_{j=1}^p \beta_j \cdot \mathbf{h}_j\big)^T \cdot \mathbf{A}^* \bmod q\big);\ \ \mathbf{c}_2 = \mathsf{COM}\big(\{\mathrm{T}_{d_3} \circ \psi_j(\mathbf{h}_j)\}_{j=1}^p\big). \end{cases}$$

The verifier outputs Valid if and only if all the conditions hold. Otherwise, he outputs $\mathsf{Invalid}$.

Figure 8.3: Our protocol.

Since COM is computationally binding, one can deduce that $d_2 = d_3$, $\phi_j = \psi_j$ for all $j \in \{1, \ldots, p\}$, and that:

$$\begin{cases} \big(\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_{j,0} - \mathbf{h}_{j,0})\big)^T \cdot \mathbf{A}_0 \notin RL, \\ \forall j \in \{1, \ldots, p\} : \ \mathbf{w}_j = \mathrm{T}_{d_2} \circ \phi_j(\mathbf{h}_j) \text{ and } \ \mathbf{v}_j + \mathbf{w}_j = \mathrm{T}_{d_2} \circ \phi_j(\mathbf{s}_j), \\ \big(\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_j - \mathbf{h}_j)\big)^T \cdot \mathbf{A}^* = \mathbf{u}^T \bmod q. \end{cases}$$

For each $j \in \{1, \ldots, p\}$, let $\mathbf{y}'_j = \mathbf{s}_j - \mathbf{h}_j$, then $\mathrm{T}_{d_2} \circ \phi_j(\mathbf{y}'_j) = \mathrm{T}_{d_2} \circ \phi_j(\mathbf{s}_j) - \mathrm{T}_{d_2} \circ \phi_j(\mathbf{h}_j) = \mathbf{v}_j \in \mathsf{SecretExt}(d_1)$. It then follows that $\phi_j(\mathbf{y}'_j) \in \mathsf{SecretExt}(d_1 \oplus d_2)$. Let $d = d_1 \oplus d_2$, then $\mathbf{y}'_j \in \mathsf{SecretExt}(d)$ for all $j \in \{1, \ldots, p\}$, since the permutation $\phi_j \in \mathcal{S}$ preserves the arrangements of the blocks of $\mathbf{y}'_j$. Now let $\mathbf{y}' = \sum_{j=1}^{p} \beta_j \cdot \mathbf{y}'_j \in \mathbb{Z}_q^{(2\ell+1)3m}$, and let $\mathbf{y} \in \mathbb{Z}^{(2\ell+1)m}$ be the vector obtained from $\mathbf{y}'$ by removing the last $2m$ coordinates in each $3m$-block. We note that $\|\mathbf{y}\|_\infty \le \|\mathbf{y}'\|_\infty \le \sum_{j=1}^{p} \beta_j \cdot \|\mathbf{y}_j\|_\infty = \sum_{j=1}^{p} \beta_j \cdot 1 = \beta$. Moreover, as $\mathbf{y}'_j \in \mathsf{SecretExt}(d)$ for all $j \in \{1, \ldots, p\}$, we have that $\mathbf{y} \in \mathsf{Secret}_\beta(d)$.

Let $\mathbf{y} = \big(\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1\big)$, then the blocks $\mathbf{y}_1^{1-d[1]}, \ldots, \mathbf{y}_\ell^{1-d[\ell]}$ are zero-blocks $\mathbf{0}^m$. Furthermore, we have that:

$$\mathbf{y}_0^T \cdot \mathbf{A}_0 = \big(\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_{j,0} - \mathbf{h}_{j,0})\big)^T \cdot \mathbf{A}_0 \notin RL.$$

Finally, by construction, we have: $\mathbf{y}^T \cdot \mathbf{A} = \mathbf{y}'^T \cdot \mathbf{A}^* = \sum_{j=1}^{p} \beta_j \cdot \mathbf{y}_j^T \cdot \mathbf{A}^* = \big(\sum_{j=1}^{p} \beta_j \cdot (\mathbf{s}_j - \mathbf{h}_j)\big)^T \cdot \mathbf{A}^* = \mathbf{u}^T \bmod q$. Therefore, we have obtained a vector $\mathbf{y}$ satisfying all the conditions stated in the lemma. □

8.3 The VLR group signature scheme

In this section we describe our lattice-based VLR group signature scheme and we prove that the scheme satisfies the requirements defined in Section 6.4: correctness, selfless-anonymity and traceability.

8.3.1 Description

We describe the scheme in Figures 8.4 and 8.5.

Remark 8.2. We have some observations on the behaviour of the above key generation algorithm:

- By Theorem 3.7, the distribution of matrix $\mathbf{A}_0$ generated by $\mathsf{TrapGen}(n, m, q)$ is statistically close to uniform over $\mathbb{Z}_q^{m \times n}$. Thus, the distribution of gpk output by $\mathsf{KeyGen}(n, N)$ is statistically close to uniform over $\mathbb{Z}_q^{(2\ell+1)m \times n} \times \mathbb{Z}_q^n$. We note that the pair $(\mathbf{A}, \mathbf{u})$ resembles the Bonsai tree structure [CHKP10], where $\mathbf{A}_0$ is the "root" of the tree.
- In Step (3a), each coordinate of vector $\mathbf{x}^{(d)}$ is either 0 or distributed according to the distribution $D_{\mathbb{Z},\sigma}$ (see Theorem 1.24 regarding the output distribution of algorithm $\mathsf{GPVSample}$). By setting $\beta = \lceil \sigma \cdot \log m \rceil$, we ensure that $\|\mathbf{x}^{(d)}\|_\infty \le \beta$ with overwhelming probability (see Lemma 1.36). Thus, the event that Step (3a) needs to be repeated only occurs with negligible probability.
- The secret key $\mathbf{x}^{(d)}$ of group user with index d satisfies $(\mathbf{x}^{(d)})^T \cdot \mathbf{A} = \mathbf{u}^T \bmod q$, and $\mathbf{x}^{(d)} \in \mathsf{Secret}_\beta(d)$.

$\mathsf{Keygen}(1^n, 1^N)$: Given a security parameter $n > 0$ and N the expected numbers of group members, proceed as follows.

1. Run $\mathsf{TrapGen}(n, m, q)$ (defined in Lemma 3.7) to get $\mathbf{A}_0 \in \mathbb{Z}_q^{m \times n}$ and trapdoor $\mathbf{R}$.
2. Sample $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$, and $\mathbf{A}_i^b \hookleftarrow U(\mathbb{Z}_q^{m \times n})$ for all $b \in \{0,1\}$ and $i \in \{1, \ldots, \ell\}$. Then define the matrix
$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 \\ \hline \mathbf{A}_1^0 \\ \hline \mathbf{A}_1^1 \\ \hline \vdots \\ \hline \mathbf{A}_\ell^0 \\ \hline \mathbf{A}_\ell^1 \end{bmatrix} \in \mathbb{Z}_q^{(2\ell+1)m \times n}.$$
3. For group user with index $d \in \{0, 1, \ldots, N-1\}$, let $d[1] \ldots d[\ell] \in \{0,1\}^\ell$ denote the binary representation of d, and do the following:
   a) Sample vectors $\mathbf{x}_1^{d[1]}, \ldots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m, \sigma}$. Compute $\mathbf{z}^T = \sum_{i=1}^{\ell} (\mathbf{x}_i^{d[i]})^T \cdot \mathbf{A}_i^{d[i]} \bmod q$, and sample $\mathbf{x}_0 \in \mathbb{Z}^m$ with $\mathbf{x}_0 \hookleftarrow \mathsf{GPVSample}\big(\mathbf{R}, \mathbf{A}_0, \mathbf{u} - \mathbf{z}, \sigma\big)$ (defined in Theorem 1.24). Let $\mathbf{x}_1^{1-d[1]}, \ldots, \mathbf{x}_\ell^{1-d[\ell]}$ be zero-vectors $\mathbf{0}^m$, and define $\mathbf{x}^{(d)} = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}$. If $\|\mathbf{x}^{(d)}\|_\infty \leq \beta$ then go to step (3b); else, repeat step (3a).
   b) Let $\mathsf{gsk}[d] = \mathbf{x}^{(d)}$ and $\mathsf{grt}[d] = \mathbf{x}_0^T \cdot \mathbf{A}_0 \in \mathbb{Z}_q^n$.
4. Finally, the algorithm outputs $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$, where
$$\mathsf{gpk} = (\mathbf{A}, \mathbf{u}); \mathsf{gsk} = \big(\{\mathsf{gsk}[d]\}_{d=0}^{N-1}\big); \mathsf{grt} = \big(\{\mathsf{grt}[d]\}_{d=0}^{N-1}\big).$$

Figure 8.4: KeyGen algorithm of our VLR signature scheme.

- By Lemma 3.9, the distribution of each user revocation token $\mathsf{grt}[d]$ is statistically close to uniform over $\mathbb{Z}_q^n$. The trivial requirement is that the revocation tokens of two different group users must be different. In the very rare event of conflict (i.e., there exist $d_1, d_2 \in \{0, \ldots, N-1\}$ such that $d_2 > d_1$ and $\mathsf{grt}[d_1] = \mathsf{grt}[d_2]$), the algorithm simply re-samples the key and token for user with index d_2.

8.3.2 Analysis of the scheme

We now analyse this scheme.

Efficiency and Correctness. The parameters in Table 8.1 are set so that all of the algorithms in the VLR group signature in Section 8.3.1 can be implemented in polynomial time. Asymptotically, the group public key has bit-size $\ell \cdot \widetilde{\mathcal{O}}(n^2) = \log N \cdot \widetilde{\mathcal{O}}(n^2)$, while the group signatures have bit-size $\ell \cdot \widetilde{\mathcal{O}}(n) = \log N \cdot \widetilde{\mathcal{O}}(n)$. The revocation check, i.e., the check against $\mathbf{c}_0^{(k)}$ in the case $Ch^{(k)} = 2$, runs in linear time in the number of revoked users, as it seems unavoidable for secure VLR group signature schemes.

Theorem 8.3. *Our VLR group signature scheme is correct with overwhelming probability.*

$\mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M)$: Let $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$ be a hash function, modelled as a random oracle. Given $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, to sign a message $M \in \{0,1\}^*$ using the secret key $\mathsf{gsk}[d] = \mathbf{x} \in \mathsf{Secret}_\beta(d)$, performs the following steps:

1. Generate a proof that the user is a certified group members and that he has not been revoked. This is done by repeating $t = \omega(\log n)$ times the basic protocol from Section 8.2 with public parameter $(\mathbf{A}, \mathbf{u})$ and prover's witness $\mathbf{x}$, and then making it non-interactive with the Fiat-Shamir heuristic as a triple $\big(\{\mathrm{CMT}^{(k)}\}_{k=1}^t,\ \mathrm{CH},\ \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big)$, where
$$\mathrm{CH} = \big(\{Ch^{(k)}\}_{k=1}^t\big) = \mathcal{H}\big(M,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big) \in \{1,2,3\}^t.$$
2. Output the group signature:
$$\Sigma = \big(M,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t,\ \{Ch^{(k)}\}_{k=1}^t,\ \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big). \tag{8.6}$$

$\mathsf{Verify}(\mathsf{gpk}, RL, , M, \Sigma)$: On input $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, a set of tokens $RL = \big\{\{\mathbf{u}_i\}_i\big\} \subset \mathbb{Z}_q^n$ whose cardinality is at most $N-1$, a message $M \in \{0,1\}^*$, and a purported group signature Σ on M, performs the following steps:

1. Parse the signature Σ as in (8.6).
2. Check if $\big(\{Ch^{(k)}\}_{k=1}^t\big) = \mathcal{H}\big(M,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big)$.
3. For $k = 1$ to t, run the verification of the protocol from Section 8.2 to check the validity of $\mathrm{RSP}^{(k)}$ with respect to $\mathrm{CMT}^{(k)}$ and $Ch^{(k)}$. If any of the verification conditions does not hold, then output $\mathsf{Invalid}$ and terminate.
4. Output Valid.

Figure 8.5: Sign, Verify and Open of our VLR group signature.

Proof. We have to prove that for all $\mathsf{gpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$, $\mathsf{gsk} = (\{\mathsf{gsk}[d]\}_{d=0}^{N-1})$, $\mathsf{grt} = (\{\mathsf{grt}[d]\}_{d=0}^{N-1})$ outputted by $\mathsf{KeyGen}(n, N)$, all $d \in \{0, 1, \ldots, N-1\}$, and all $M \in \{0,1\}^*$, we have:

$$\mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M), M) = \mathsf{Valid} \Leftrightarrow \mathsf{grt}[d] \notin RL.$$

1. We first prove that: $\mathsf{grt}[d] \notin RL \Rightarrow \mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M), M) = \mathsf{Valid}$.

   Suppose that $\mathsf{grt}[d] \notin RL$. We will show that, for each $k \in [t]$, all the checks performed by the verification algorithm hold true, except for negligible probability. For simplicity, we will not consider the trivial checks for correct computations, e.g., the case $Ch^{(k)} = 3$.

   a) If $Ch^{(k)} = 1$: The crucial point is to check whether $\forall j \in \{1, \ldots, p\} : \mathbf{v}_j^{(k)} \in \mathsf{SecretExt}(d_1^{(k)})$. Note that if $\mathbf{x} = \mathsf{gsk}[d]$ is outputted by $\mathsf{KeyGen}(n, N)$ then $\mathbf{x} \in \mathsf{Secret}_\beta(d)$, and thus, all the vectors $\mathbf{z}_1, \ldots, \mathbf{z}_p$ outputted by the procedure $\mathsf{WitnessDE}(\mathbf{x})$ belong to the set $\mathsf{SecretExt}(d)$. It then follows from the special properties of the permutation sets $\mathcal{S}$ and $\mathcal{T}$ that $\forall j \in \{1, \ldots, p\} : \mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j) \in \mathsf{SecretExt}(d \oplus e^{(k)})$. Finally, it is worth to recall that $\forall j \in \{1, \ldots, p\} : \mathbf{v}_j^{(k)} = \mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j)$, and that $d_1^{(k)} = d \oplus e^{(k)}$.

   b) If $Ch^{(k)} = 2$: There are two crucial checks:

      i. Check if $\forall \mathbf{u}_i \in RL : \mathbf{c}_0 \neq \mathsf{COM}\big(d_2, \{\phi_j\}_{j=1}^p, \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big)^T \cdot \mathbf{A}_0 - \mathbf{u}_i^T \bmod q\big)$. For each i, let $\alpha_i^T = \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big)^T \cdot \mathbf{A}_0 - \mathbf{u}_i^T \in \mathbb{Z}_q^n$. Meanwhile, $\mathbf{c}_0^{(k)} =$

$\mathsf{COM}(d_2, \{\phi_j\}_{j=1}^p, \alpha)$, where $\alpha^T = \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big)^T \cdot \mathbf{A}_0 = \alpha_i^T + \mathbf{u}_i^T - \mathsf{grt}[d]$. Since $\mathsf{grt}[d] \notin RL$, we have $\mathsf{grt}[d] \neq \mathbf{u}_i^T$ for all i, and thus, $\alpha \neq \alpha_i$. Moreover, over the randomness of all algorithms, the distributions of $\mathsf{COM}(d_2, \{\phi_j\}_{j=1}^p, \alpha)$ and $\mathsf{COM}(d_2, \{\phi_j\}_{j=1}^p, \alpha_i)$ are statistically close to uniform over $\mathbb{Z}_q^n$ (this follows from the statistically hiding property of COM). Hence, we have $\mathsf{COM}(d_2, \{\phi_j\}_{j=1}^p, \alpha) \neq \mathsf{COM}(d_2, \{\phi_j\}_{j=1}^p, \alpha_i)$ with overwhelming probability.

ii. Check if $\big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j^{(k)}\big)^T \cdot \mathbf{A}^* - \mathbf{u}^T = \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(k)}\big)^T \cdot \mathbf{A}^*$. This is true, because

$$\begin{aligned}
\big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j^{(k)}\big)^T \cdot \mathbf{A}^* &= \sum_{j=1}^p \beta_j \cdot \big(\mathbf{z}_j + \mathbf{r}_j^{(k)}\big)^T \cdot \mathbf{A}^* \\
&= \big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j\big)^T \cdot \mathbf{A}^* + \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(k)}\big)^T \cdot \mathbf{A}^* \\
&= \mathbf{u}^T + \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(k)}\big)^T \cdot \mathbf{A}^*,
\end{aligned}$$

where the last equation follows from the fact that $\big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j\big)^T \cdot \mathbf{A}^* = \mathbf{x}^T \cdot \mathbf{A} = \mathbf{u}^T \bmod q$.

Therefore, the verification algorithm outputs Valid with overwhelming probability, over the randomness of all algorithms.

2. We then prove that: $\mathsf{Verify}(\mathsf{gpk}, RL, \mathsf{Sign}(\mathsf{gpk}, \mathsf{gsk}[d], M), M) = \mathsf{Valid} \Rightarrow \mathsf{grt}[d] \notin RL$.

Assume by contradiction that $\mathsf{grt}[d] = \mathbf{x}_0^T \cdot \mathbf{A}_0 \bmod q \in RL$, and fix any $k \in \{1, \ldots, t\}$. Note that in the signing algorithm, we construct $\mathbf{c}_0^{(k)}$ so that:

$$\mathbf{c}_0^{(k)} = \mathsf{COM}\big(d_2, \{\phi_j\}_{j=1}^p, \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big)^T \cdot \mathbf{A}_0 \bmod q\big)$$

On the other hand, since the verification algorithm outputs Valid, the following requirement must satisfy (in the case $Ch^{(k)} = 2$):

$$\mathbf{c}_0^{(k)} \neq \mathsf{COM}\big(d_2, \{\phi_j\}_{j=1}^p, \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}\big)^T \cdot \mathbf{A}_0 - \mathbf{u}_i^T \bmod q\big)$$

As we have $\mathbf{s}_{j,0}^{(k)} = \mathbf{z}_{j,0} + \mathbf{r}_{j,0}^{(k)}$ and $\mathbf{x}_0^T \cdot \mathbf{A}_0 = \big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_{j,0}\big)^T \cdot \mathbf{A}_0$, we have that $\big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}^{(k)}\big)^T \cdot \mathbf{A}_0 = \big(\sum_{j=1}^p \beta_j \cdot \mathbf{s}_{j,0}^{(k)}\big)^T \cdot \mathbf{A}_0 - \mathbf{x}_0^T \cdot \mathbf{A}_0 \bmod q$. Thus, we obtain a contradiction. Namely, it must be true that $\mathsf{grt}[d] \notin RL$. This concludes the proof.

□

Selfless-Anonymity. We now prove that our VLR group signature scheme is selfless-anonymous.

Theorem 8.4. *If* COM *is a statistically hiding string commitment scheme, then the VLR group signature scheme in Section 8.3.1 is selfless-anonymous in the random oracle model.*

Proof. We define two hybrid games G_0 and G_1. Game G_0 is the original selfless-anonymity game (see Chapter 6.4.3). In game G_1, we make the distribution of the challenger's output independent of the bit $b \in \{0,1\}$. We then prove that these two games are statistically indistinguishable. Since the adversary's advantage in game G_1 is 0, this implies the selfless-anonymity of our scheme.

Game G_0:

1. Run $\mathsf{KeyGen}(n, N)$ to obtain

$$\mathsf{gpk} = (\mathbf{A}, \mathbf{u}); \mathsf{gsk} = \big(\{\mathsf{gsk}[d]\}_{d=0}^{N-1}\big); \mathsf{grt} = \big(\{\mathsf{grt}[d]\}_{d=0}^{N-1}\big).$$

   Set $RL := \emptyset$, $\mathsf{Corrupted} := \emptyset$, and give gpk to the adversary $\mathcal{A}$.

2. If $\mathcal{A}$ queries the signature on any message M by user of index d, return $\Sigma = \mathsf{Sign}\big(\mathsf{gpk}, \mathsf{gsk}[d], M\big)$. If $\mathcal{A}$ queries the corruption of user of index d, set $\mathsf{Corrupted} := \mathsf{Corrupted} \cup \{d\}$, and return $\mathsf{gsk}[d]$. If $\mathcal{A}$ queries the revocation of user d, set $RL := RL \cup \{\mathsf{grt}[d]\}$, and return $\mathsf{grt}[d]$.

3. $\mathcal{A}$ outputs a message M^* and d_0, d_1 such that $d_b \notin \mathsf{Corrupted}$ and $\mathsf{grt}[d_b] \notin RL$ for each $b \in \{0,1\}$.

4. Pick a bit $b \hookleftarrow U(\{0,1\})$, generate a valid signature

$$\Sigma = \mathsf{Sign}\big(\mathsf{gpk}, \mathsf{gsk}[d_b], M^*\big) = \big(M^*, \{\mathrm{CMT}^{(k)}\}_{k=1}^{t}, \{Ch^{(k)}\}_{k=1}^{t}, \{\mathrm{RSP}^{(k)}\}_{k=1}^{t}\big),$$

   and return Σ to $\mathcal{A}$.

5. $\mathcal{A}$ can still make queries as before, but it is not allowed to ask for $\mathsf{gsk}[d_b]$ or $\mathsf{grt}[d_b]$, for each $b \in \{0,1\}$.

6. Finally $\mathcal{A}$ outputs a bit b'.

Game G_1:
In this game, we make the following modification with respect to **Game** G_0: In Step 4, instead of generating a legitimate signature, we simulate the signature generation. Our simulation algorithm is such that:

- **Input:** The group public key $\mathsf{gpk} = \big(\mathbf{A}, \mathbf{u}\big)$ obtained from Step 1, the set of user revocation tokens RL obtained at the end of Step 2, and the message M^* obtained from Step 3.
- **Output:** A *valid* group signature Σ^* for message M^* under gpk and RL. Moreover, Σ^* is *independent* of the bit b, and it is statistically indistinguishable from the legitimate signature Σ in game G_0.

Let $\mathbf{A} = \big[(\mathbf{A}_0)^T | (\mathbf{A}_1^0)^T | (\mathbf{A}_1^1)^T | \ldots | (\mathbf{A}_\ell^0)^T | (\mathbf{A}_\ell^1)^T\big]^T$ and $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}(\mathbf{A})$. The simulation algorithm does the following:

1. For each $k \in \{1, \ldots, t\}$, pick a "fake" challenge $\overline{Ch}^{(k)} \hookleftarrow U(\{1,2,3\}$, that is a "prediction" of what the real challenge will *not* be. Then pick a real challenge $Ch^{(k)} \hookleftarrow U(\{1,2,3\}) \setminus \{\overline{Ch}^{(k)}\}$. It turns out that $Ch^{(k)}$ is uniformly distributed in $\{1,2,3\}$, which satisfies the requirement on the output of the random oracle $\mathcal{H}$. Then prepare $\mathrm{CMT}^{(k)}$, and the response $\mathrm{RSP}^{(k)}$ to $(\mathrm{CMT}^{(k)}, Ch^{(k)})$ as follows:

   a) **Case** $\overline{Ch}^{(k)} = 1$:

i. Use linear algebra to compute $\mathbf{z} \in \mathbb{Z}_q^{(2\ell+1)3m}$ such that $\mathbf{z}^T \cdot \mathbf{A}^* = \mathbf{u}^T \bmod q$. Let $\mathbf{g}_0 = \mathsf{Parse}(\mathbf{z}, 1, m)$. If $\mathbf{g}_0^T \cdot \mathbf{A}_0 \in RL$ then repeat this step. Otherwise, compute $\mathbf{z}_1^{(k)}, \ldots, \mathbf{z}_p^{(k)} \in \mathbb{Z}_q^{(2\ell+1)3m}$ such that $\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j^{(k)} = \mathbf{z} \bmod q$.

ii. Sample $e^{(k)} \hookleftarrow U(\{0,1\}^\ell)$, and for all $j \in \{1, \ldots, p\}$, sample $\pi_j^{(k)} \hookleftarrow U(\mathcal{S})$ and $\mathbf{r}_j^{(k)} \hookleftarrow U(\mathbb{Z}_q^{(2\ell+1)\cdot 3m})$, and let $\mathbf{r}_{j,0}^{(k)} = \mathsf{Parse}(\mathbf{r}_j^{(k)}, 1, m)$.

iii. Compute $\mathrm{CMT}^{(k)} = \left(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\right) \in (\mathbb{Z}_q^n)^4$ as in (8.2), from Section 8.2.

iv. If $Ch^{(k)} = 2$, then set

$$\mathrm{RSP}^{(k)} = \left(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^p, \{\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}\}_{j=1}^p\right). \tag{8.7}$$

If $Ch^{(k)} = 3$, then set

$$\mathrm{RSP}^{(k)} = \left(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^p, \{\mathbf{r}_j^{(k)}\}_{j=1}^p\right). \tag{8.8}$$

b) **Case** $\overline{Ch}^{(k)} = 2$:

i. Sample $d^{(k)}, e^{(k)} \hookleftarrow U(\{0,1\}^\ell)$. For all $j \in \{1, \ldots, p\}$, sample $\pi_j^{(k)} \hookleftarrow U(\mathcal{S})$, and $\mathbf{r}_j^{(k)} \hookleftarrow U(\mathbb{Z}_q^{(2\ell+1)\cdot 3m})$, and $\mathbf{z}_j^{(k)} \hookleftarrow U(\mathsf{SecretExt}(d^{(k)}))$. Let $\mathbf{r}_{j,0}^{(k)} = \mathsf{Parse}(\mathbf{r}_j^{(k)}, 1, m)$.

ii. Compute $\mathrm{CMT}^{(k)} = \left(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\right) \in (\mathbb{Z}_q^n)^4$ as in (8.2), from Section 8.2.

iii. If $Ch^{(k)} = 1$, then set

$$\mathrm{RSP}^{(k)} = \left(d^{(k)} \oplus e^{(k)}, \left\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)})\right\}_{j=1}^p \left\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\right\}_{j=1}^p\right). \tag{8.9}$$

If $Ch^{(k)} = 3$, then set

$$\mathrm{RSP}^{(k)} = \left(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^p,\ \{\mathbf{r}_j^{(k)}\}_{j=1}^p\right). \tag{8.10}$$

c) **Case** $\overline{Ch}^{(k)} = 3$:

i. Sample $d^{(k)}, e^{(k)} \hookleftarrow U(\{0,1\}^\ell)$. For all $j \in \{1, \ldots, p\}$ sample $\pi_j^{(k)} \hookleftarrow U(\mathcal{S})$ and $\mathbf{r}_j^{(k)} \hookleftarrow U(\mathbb{Z}_q^{(2\ell+1)\cdot 3m})$, and let $\mathbf{r}_{j,0}^{(k)} = \mathsf{Parse}(\mathbf{r}_j^{(k)}, 1, m)$.

ii. For all $j \in \{1, \ldots, p\}$, sample $\mathbf{z}_j^{(k)} \hookleftarrow U(\mathsf{SecretExt}(d^{(k)}))$, and let $\mathbf{z}_{j,0}^{(k)} = \mathsf{Parse}(\mathbf{z}_j^{(k)}, 1, m)$. If $(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_{j,0}^{(k)})^T \cdot \mathbf{A}_0 \in RL$, then repeat this step.

iii. Compute $\mathrm{CMT}^{(k)} = \left(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\right) \in (\mathbb{Z}_q^n)^4$, where $\mathbf{c}_0^{(k)}, \mathbf{c}_2^{(k)}$ and $\mathbf{c}_3^{(k)}$ are as in (8.2), from Section 8.2, while $\mathbf{c}_1^{(k)}$ is computed as follows:

$$\mathbf{c}_1^{(k)} = \mathsf{COM}\Big(e^{(k)}, \{\pi_j^{(k)}\}_{j=1}^p, \sum_{j=1}^p \beta_j \cdot (\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)})^T \cdot \mathbf{A}^* - \mathbf{u}^T\Big).$$

iv. If $Ch^{(k)} = 1$, then set

$$RSP^{(k)} = \left(d^{(k)} \oplus e^{(k)}, \left\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)})\right\}_{j=1}^p, \left\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\right\}_{j=1}^p\right). \tag{8.11}$$

If $Ch^{(k)} = 2$, then set

$$RSP^{(k)} = \left(e^{(k)},\ \{\pi_j^{(k)}\}_{j=1}^p,\ \{\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}\}_{j=1}^p\right). \tag{8.12}$$

2. Program the random oracle: $\mathcal{H}\big(M^*, \{\mathrm{CMT}^{(k)}\}_{k=1}^t\big) = \big(\{Ch^{(k)}\}_{k=1}^t\big)$.

3. Output the simulated signature $\Sigma^* = \big(M^*, \{\mathrm{CMT}^{(k)}\}_{k=1}^t, \{Ch^{(k)}\}_{k=1}^t, \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big)$.

We have the following observations on the above construction:

- For every $k \in \{1, \ldots, t\}$, the distribution of $\mathrm{CMT}^{(k)}$ is statistically close to uniform over $(\mathbb{Z}_q^n)^4$. This follows from the statistically hiding property of COM.
- The distribution of $\big(\{Ch^{(k)}\}_{k=1}^t\big)$ is uniform over $\{1,2,3\}^t$.
- For every $k \in \{1, \ldots, t\}$:
  1. If $Ch^{(k)} = 1$, the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)}$ *and* $\mathrm{RSP}^{(k)}$ is either (1(b)ii) *and* (8.9), or (1(c)iii) *and* (8.11).
  2. If $Ch^{(k)} = 2$, the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)}$ *and* $\mathrm{RSP}^{(k)}$ is either (1(a)iii) *and* (8.7), or (1(c)iii) *and* (8.12).
  3. If $Ch^{(k)} = 3$, the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)}$ *and* $\mathrm{RSP}^{(k)}$ is either (1(a)iii) *and* (8.8), or (1(b)ii) *and* (8.10).

  We remark that, in every case, $\mathrm{RSP}^{(k)}$ is intentionally designed to be a valid "response" to $\mathrm{CMT}^{(k)}$ and $Ch^{(k)}$, and to be statistically close to that produced by Step (4) in Game G_0.

These observations imply that Σ^* is a valid group signature, i.e., $\mathsf{Verify}\big((\mathbf{A}, \mathbf{u}), RL, \Sigma^*, M^*\big) = \mathsf{Valid}$, and that Σ^* is statistically indistinguishable from the legitimate signature Σ produced by Game G_0 (for a more detailed analysis, see Lemma 8.5). It then follows that Game G_0 and Game G_1 are statistically indistinguishable. Moreover, Σ^* is independent of the bit $b \in \{0,1\}$, thus, the adversary's advantage in Game G_1 is 0. As a result, the adversary's advantage in Game G_0 is negligible. In other words, our VLR group signature is selfless-anonymous. □

Lemma 8.5. *The signature Σ^* outputted by Game G_1 is a valid signature, and is statistically indistinguishable from the legitimate signature Σ produced by Game G_0.*

Proof. Let

$$\Sigma^* = \big(M^*,\ \{\mathrm{CMT}^{(k)}\}_{k=1}^t, \{Ch^{(k)}\}_{k=1}^t, \{\mathrm{RSP}^{(k)}\}_{k=1}^t\big)$$

be the signature outputted by Game G_1. First of all, we observe that:

- For every $k \in \{1, \ldots, t\}$, the distribution of $\mathrm{CMT}^{(k)}$ is statistically close to uniform over $(\mathbb{Z}_q^n)^4$. This follows from the statistical regularity property of $f_{\mathbf{B}}$ and the statistically hiding property of COM.
- The distribution of $\big(Ch^{(1)}, \ldots, Ch^{(t)}\big)$ is uniform over $\{1,2,3\}^t$.

Therefore, the distributions of $\{\mathrm{CMT}^{(k)}\}_{k=1}^t$ and $\{Ch^{(k)}\}_{k=1}^t$ are statistically close to those of the legitimate signature Σ. We now will show that for every $k \in \{1, \ldots, t\}$, $\mathrm{RSP}^{(k)}$ is statistically close to that of the legitimate signature, and it is valid 'response' to $\mathrm{CMT}^{(k)}$ and $Ch^{(k)}$. Indeed, for each $k \in [t]$, we have:

1. If $Ch^{(k)} = 1$, then the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)} = \big(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\big)$ and $\mathrm{RSP}^{(k)}$ is one of the following two cases:

a)

$$\begin{cases} \mathbf{c}_0^{(k)} = \mathsf{COM}\big(d_2, \{\phi_j\}_{j=1}^p, \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big)^T \cdot \mathbf{A}_0\big), \\ \mathbf{c}_1^{(k)} = \mathrm{COM}\big(e^{(k)}, \{\pi_j^{(k)}\}_{j=1}^p, \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j^{(k)}\big)^T \cdot \mathbf{A}^*\big), \\ \mathbf{c}_2^{(k)} = \mathrm{COM}\big(\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\}_{j=1}^p\big), \\ \mathbf{c}_3^{(k)} = \mathrm{COM}\big(\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)})\}_{j=1}^p\big), \end{cases} \tag{8.13}$$

and

$$\mathrm{RSP}^{(k)} = \big(d^{(k)} \oplus e^{(k)}, \big\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)})\big\}_{j=1}^p, \big\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\big\}_{j=1}^p\big). \tag{8.14}$$

For all $j \in \{1, \ldots, p\}$, since $\mathbf{z}_j^{(k)} \in \mathsf{SecretExt}(d^{(k)})$, it follows from (8.1) that $\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)}) \in \mathsf{SecretExt}(d^{(k)} \oplus e^{(k)})$. Thus $\mathrm{RSP}^{(k)}$ satisfies the verification conditions for the case $Ch^{(k)} = 1$ (since the checks with respect to $\mathbf{c}_2^{(k)}$ and $\mathbf{c}_3^{(k)}$ obviously hold true). Note that by construction, $d^{(k)} \oplus e^{(k)}$ is uniform in $\{0,1\}^\ell$; $\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)})$ is uniform in $\mathsf{SecretExt}(d^{(k)} \oplus e^{(k)})$; and $\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})$ is uniform in $\mathbb{Z}_q^{(2\ell+1)3m}$. Therefore, the distribution of $\mathrm{RSP}^{(k)}$ is identical to that of the legitimate signature.

b)

$$\begin{cases} \mathbf{c}_0^{(k)} = \mathsf{COM}\big(d_2, \{\phi_j\}_{j=1}^p, \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{j,0}\big)^T \cdot \mathbf{A}_0\big), \\ \mathbf{c}_1^{(k)} = \mathrm{COM}_{\mathbf{B}}\big(e^{(k)}, \{\pi_j^{(k)}\}_{j=1}^p, \sum_{j=1}^p \beta_j \cdot (\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)})^T \cdot \mathbf{A}^* - \mathbf{u}^T\big), \\ \mathbf{c}_2^{(k)} = \mathrm{COM}\big(\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{r}_j^{(k)})\}_{j=1}^p\big), \\ \mathbf{c}_3^{(k)} = \mathrm{COM}\big(\{\mathrm{T}_{e^{(k)}} \circ \pi_j^{(k)}(\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)})\}_{j=1}^p\big), \end{cases} \tag{8.15}$$

and $\mathrm{RSP}^{(k)}$ is computed as in (8.14). The analysis for this case is similar to the above one.

2. If $Ch^{(k)} = 2$, then the view of $\mathcal{A}$ on $\mathrm{CMT}^{(k)} = \big(\mathbf{c}_0^{(k)}, \mathbf{c}_1^{(k)}, \mathbf{c}_2^{(k)}, \mathbf{c}_3^{(k)}\big)$ and $\mathrm{RSP}^{(k)}$ is one of the following two cases:

   a) $\mathrm{CMT}^{(k)}$ is computed as in (8.13), and $\mathrm{RSP}^{(k)} = \big(e^{(k)}, \{\pi_j^{(k)}\}_{j=1}^p, \{\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}\}_{j=1}^p\big)$. Observe that:

   - By construction, we have $\mathbf{g}_0^T \cdot \mathbf{A}_0 \notin RL$. The correctness of the VLR group signature then implies that: the revocation check with respect to $\mathbf{c}_0^{(k)}$ holds true with overwhelming probability.
   - By construction, we have $\big(\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j^{(k)}\big)^T \cdot \mathbf{A}^* = \mathbf{u}^T \bmod q$. This implies that the check with respect to $\mathbf{c}_1^{(k)}$ holds true.
   - The check with respect to $\mathbf{c}_3^{(k)}$ obviously hold true.

   Hence $\mathrm{RSP}^{(k)}$ satisfies the verification conditions for the case $Ch^{(k)} = 2$. Moreover, $\mathrm{RSP}^{(k)}$ is uniform over $\{0,1\}^\ell \times \mathcal{S}^p \times \big(\mathbb{Z}_q^{(2\ell+1)3m}\big)^p$, and thus, is identically distributed with that of the legitimate signature.

   b) $\mathrm{CMT}^{(k)}$ is computed as in (8.15), and $\mathrm{RSP}^{(k)} = \big(e^{(k)}, \{\pi_j^{(k)}\}_{j=1}^p, \{\mathbf{z}_j^{(k)} + \mathbf{r}_j^{(k)}\}_{j=1}^p\big)$. As above, the distribution of $\mathrm{RSP}^{(k)}$ is the same as in the legitimate signature. Moreover:

- Since we have $(\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_{j,0}^{(k)})^T \cdot \mathbf{A}_0 \notin RL$, the revocation check with respect to $\mathbf{c}_0^{(k)}$ holds true with overwhelming probability.
- We remark that we do not have $\left(\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j^{(k)}\right)^T \cdot \mathbf{A}^* = \mathbf{u}^T \bmod q$, but we construct $\mathbf{c}_1^{(k)}$ so that the check with respect to it holds true.
- The check with respect to $\mathbf{c}_3^{(k)}$ obviously hold true.

3. If $Ch^{(k)} = 3$, then in any of the two views of the adversary, the verification checks with respect to $\mathbf{c}_1^{(k)}$, and $\mathbf{c}_2^{(k)}$ are checks for correct computations, and thus, they hold true. Moreover, the distribution of $\mathrm{RSP}^{(k)}$ is uniform over $\{0,1\}^{\ell} \times \mathcal{S}^p \times \left(\mathbb{Z}_q^{(2\ell+1)3m}\right)^p$, as in the legitimate signature.

Hence, we have shown that the simulated signature Σ^* produced by game G_1 is a valid signature of M^* under gpk and RL, and it is statistically close to the legitimate signature Σ produced by game G_0. □

Traceability. We now prove that, in the random oracle model, our VLR group signature scheme is traceable if the $\mathrm{SIS}^{\infty}_{q,(\ell+1)\cdot m,2\beta}$ problem is hard.

Theorem 8.6. *If there is a traceability adversary $\mathcal{A}$ with success probability ϵ and running time T, then there is an algorithm $\mathcal{F}$ that solves the $SIS^{\infty}_{q,(\ell+1)\cdot m,2\beta}$ problem with success probability $\epsilon' > \left(1-(7/9)^t\right) \cdot \frac{1}{2N}$, and running time $T' = 32 \cdot T \cdot q_{\mathcal{H}}/(\epsilon - 3^{-t}) + \mathsf{poly}(n, N)$, where $q_{\mathcal{H}}$ is the number of queries to the random oracle $\mathcal{H} : \{0,1\}^* \to \{1,2,3\}^t$.*

The results of Theorem 2.3 and Theorem 8.6 imply that our scheme is traceable in the random oracle model, based on the worst-case hardness of the SIVP_{γ} problem (in the ℓ_2 norm), with $\gamma = 2\beta \cdot \widetilde{\mathcal{O}}(n) = \widetilde{\mathcal{O}}(n^{1.5})$.

Proof. First, suppose that adversary $\mathcal{A}$ can break the computational binding property of the commitment scheme COM with non-negligible probability. As mentioned earlier (see Section 6.1.2), we can use $\mathcal{A}$ to solve the $\mathrm{SIS}^{\infty}_{q,(\ell+1)\cdot m,2\beta}$ problem. Therefore, without loss of generality, we assume that COM is computationally binding.

We construct a PPT algorithm $\mathcal{F}$ solving the $\mathrm{SIS}^{\infty}_{q,(\ell+1)\cdot m,2\beta}$ problem with non-negligible probability, which works as follows:

$\mathsf{Challenge}$: Algorithm $\mathcal{F}$ is given a uniformly random matrix

$$\mathbf{C} = \left[\begin{array}{c} \mathbf{C}_0 \\ \hline \mathbf{C}_1 \\ \hline \vdots \\ \hline \mathbf{C}_{\ell} \end{array}\right] \in \mathbb{Z}_q^{(\ell+1)m \times n}..$$

It wins the challenge if it can produce a non-zero vector $\mathbf{x} \in \mathbb{Z}^{(\ell+1)\cdot m}$ such that $\|\mathbf{x}\|_{\infty} \leq 2\beta$ and $\mathbf{x}^T \cdot \mathbf{C} = \mathbf{0} \bmod q$.

Setup: $\mathcal{F}$ performs the following steps:

1. Sample vector $\mathbf{z} = \left(\mathbf{z}_0 \| \mathbf{z}_1 \| \ldots \| \mathbf{z}_{\ell}\right) \in \mathbb{Z}^{(\ell+1)\cdot m}$, where each coordinate of $\mathbf{z}$ is sampled from $D_{\mathbb{Z},\sigma}$. If $\|\mathbf{z}\|_{\infty} > \beta$, then repeat the sampling. Otherwise, compute $\mathbf{u}^T = \mathbf{z}^T \cdot \mathbf{C} \bmod q$.

2. Run $\mathsf{TrapGen}(n, m, q)$ algorithm ℓ times, and let the outputs be $\left((\mathbf{F}_1, \mathbf{R}_1), (\mathbf{F}_2, \mathbf{R}_2), \ldots, (\mathbf{F}_{\ell}, \mathbf{R}_{\ell})\right)$.

3. Pick a target index $d^* = d^*[1] \dots d^*[\ell] \hookleftarrow U(\{0,1\}^\ell$, and define

$$\mathbf{A} = \left[\begin{array}{c} \mathbf{A}_0 \\ \hline \mathbf{A}_1^0 \\ \hline \mathbf{A}_1^1 \\ \hline \vdots \\ \hline \mathbf{A}_\ell^0 \\ \hline \mathbf{A}_\ell^1 \end{array}\right] \in \mathbb{Z}_q^{(2\ell+1)m \times n},$$

where $\mathbf{A}_0 = \mathbf{C}_0$, and for each $i \in \{1, \dots, \ell\}$: $\mathbf{A}_i^{d^*[i]} = \mathbf{C}_i$ and $\mathbf{A}_i^{1-d^*[i]} = \mathbf{F}_i$.

4. Define the secret key and revocation token for user d^* as follows:
   - $\mathsf{gsk}[d^*] = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)\cdot m}$, where $\mathbf{x}_0 = \mathbf{z}_0$, $\forall i \in \{1, \dots, \ell\}$: $\mathbf{x}_i^{d^*[i]} = \mathbf{z}_i$ and $\mathbf{x}_i^{1-d^*[i]} = \mathbf{0}^m$,
   - $\mathsf{grt}[d^*] = \mathbf{x}_0^T \cdot \mathbf{A}_0 \bmod q \in \mathbb{Z}_q^n$.

5. Generate the secret key and the revocation token for each user $d \neq d^*$, where $d = d[1] \dots d[\ell]$, as follows:
   - Let $d[b]$ $(1 \leq b \leq \ell)$ be the first bit from the left where $d[b] \neq d^*[b]$. Since $d \neq d^*$, such b must exist. It follows that $\mathbf{A}_b^{d[b]} = \mathbf{A}_b^{1-d^*[b]} = \mathbf{F}_b$.
   - Sample ℓ vectors $\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \dots, \mathbf{x}_{b-1}^{d[b-1]}, \mathbf{x}_{b+1}^{d[b+1]}, \dots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, and let
   $$(\mathbf{t}^{(d)})^T = \mathbf{u}^T - \big(\mathbf{x}_0^T \cdot \mathbf{A}_0 + (\sum_{i \in [\ell], i \neq b} \mathbf{x}_i^{d[i]})^T \cdot \mathbf{A}_i^{d[i]}\big) \bmod q.$$
   - Sample $\mathbf{x}_b^{d[b]} \hookleftarrow \mathsf{GPVSample}(\mathbf{R}_b, \mathbf{F}_b, \mathbf{t}^{(d)}, \sigma)$.
   - For each $i \in \{1, \dots, \ell\}$, let $\mathbf{x}_i^{1-d[i]} = \mathbf{0}^m$, then let $\mathbf{x}^{(d)} = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)\cdot m}$. If the very rare event that $\|\mathbf{x}^{(d)}\|_\infty > \beta$ happens, then repeat the sampling. Otherwise, set $\mathsf{gsk}[d] = \mathbf{x}^{(d)}$ and $\mathsf{grt}[d] = \mathbf{x}_0^T \cdot \mathbf{A}_0 \bmod q \in \mathbb{Z}_q^n$.

6. Let $\mathsf{gpk} = (\mathbf{A}, \mathbf{u})$, $\mathsf{gsk} = \big(\{\mathsf{gsk}[d]\}_{d=0}^{N-1}\big)$, $\mathsf{grt} = \big(\{\mathsf{grt}[d]\}_{d=0}^{N-1}\big)$. We note that, by construction, the distribution of $(\mathsf{gpk}, \mathsf{gsk}, \mathsf{grt})$ is statistically close to that of the real scheme, and the choice of d^* is hidden from the adversary. Algorithm $\mathcal{F}$ then gives $(\mathsf{gpk}, \mathsf{grt})$ to $\mathcal{A}$.

Queries: Algorithm $\mathcal{F}$ answers the queries of $\mathcal{A}$ as follows:

- **Corruption queries**: The corruption set U is initially set to be empty. If $\mathcal{A}$ queries the secret key of any user $d \in \{0, \dots, N-1\}$, then $\mathcal{F}$ adds d to the corruption set U, and returns $\mathsf{gsk}[d]$.
- **Signatures queries**: If $\mathcal{A}$ queries signature of user d on arbitrary message M, then $\mathcal{F}$ returns $\Sigma = \mathsf{Sign}\big(\mathsf{gpk}, \mathsf{gsk}[d], M\big)$. Queries to the random oracle $\mathcal{H}$ are handled by consistently returning uniformly random values in $\{1,2,3\}^t$. For each $\kappa \leq q_\mathcal{H}$, we let r_κ denote the answer to the κ-th query.

Forgery: Eventually, $\mathcal{A}$ outputs a message M^*, a set of tokens RL^* and a non-trivial forged signature

$$\Sigma^* = \big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t, \{Ch_i\}_{i=1}^t, \{\mathrm{RSP}_i\}_{i=1}^t\big),$$

such that $\mathsf{Verify}(\mathsf{gpk}, RL^*, \Sigma^*, M^*) = \mathsf{valid}$, and the implicit tracing algorithm fails or traces to a user outside of the coalition $U \setminus RL^*$. Now algorithm $\mathcal{F}$ exploits the forgery as follows.

First, one can argue that $\mathcal{A}$ must have queried $\mathcal{H}$ on input $\big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t\big)$, as otherwise, the probability that $(Ch_1, \ldots, Ch_t) = \mathcal{H}\big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t\big)$ is at most 3^{-t}. Therefore, with probability at least $\epsilon - 3^{-t}$, there exists certain $\kappa^* \leq q_{\mathcal{H}}$ such that the κ^*-th oracle queries involves the tuple $\big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t\big)$. Next, $\mathcal{F}$ picks κ^* as the target forking point and replays $\mathcal{A}$ many times with the same random tape and input as in the original run. In each rerun, for the first $\kappa^* - 1$ queries, $\mathcal{A}$ is given the same answers $r_1, \ldots, r_{\kappa^*-1}$ as in the initial run, but from the κ^*-th query onwards, $\mathcal{F}$ replies with fresh random values $r'_{\kappa^*}, \ldots, r'_{q_{\mathcal{H}}} \hookleftarrow U(\{1,2,3\}^t$. The Improved Forking Lemma of Pointcheval and Vaudenay [PV97, Lemma 7] implies that, with probability larger than $1/2$, algorithm $\mathcal{F}$ can obtain a 3-fork involving the tuple $\big(M^*, \{\mathrm{CMT}_i\}_{i=1}^t\big)$ after less than $32 \cdot q_{\mathcal{H}}/(\epsilon - 3^{-t})$ executions of $\mathcal{A}$. Now, let the answers of $\mathcal{F}$ with respect to the 3-fork branches be

$$r_{\kappa^*}^{(1)} = (Ch_1^{(1)}, \ldots, Ch_t^{(1)}); r_{\kappa^*}^{(2)} = (Ch_1^{(2)}, \ldots, Ch_t^{(2)}); r_{\kappa^*}^{(3)} = (Ch_1^{(3)}, \ldots, Ch_t^{(3)}).$$

A simple calculation shows that: $\Pr\big[\exists i \in \{1, \ldots, t\} : \{Ch_i^{(1)}, Ch_i^{(2)}, Ch_i^{(3)}\} = \{1,2,3\}\big] = 1 - (7/9)^t$. Conditioned on the existence of such index i, one parses the 3 forgeries corresponding to the fork branches to obtain $\big(\mathrm{RSP}_i^{(1)}, \mathrm{RSP}_i^{(2)}, \mathrm{RSP}_i^{(3)}\big)$. They turn out to be 3 *valid* responses with respect to 3 different challenges for the same commitment CMT_i. Since COM is assumed to be computationally-binding, we can apply Lemma 8.1 to extract a vector $\mathbf{y} = \big(\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m}$ satisfying $\mathbf{y}^T \cdot \mathbf{A} = \mathbf{u}^T \bmod q$, $\mathbf{y}_0^T \cdot \mathbf{A}_0 \bmod q \notin RL^*$, and $\mathbf{y} \in \mathsf{Secret}_\beta(d)$ for some $d \in \{0,1\}^\ell$. Now consider two cases:

- If $d \neq d^*$, which happens with probability at most $\frac{N-1}{N}$, then algorithm $\mathcal{F}$ declares Fail and aborts.
- If $d = d^*$, then let $\mathbf{y}^* = \big(\mathbf{y}_0 \| \mathbf{y}_1^{d^*[1]} \| \ldots \| \mathbf{y}_\ell^{d^*[\ell]}\big) \in \mathbb{Z}^{(\ell+1)m}$, obtained by removing the zero-blocks $\mathbf{y}_1^{1-d^*[1]}, \ldots, \mathbf{y}_\ell^{1-d^*[\ell]}$ from $\mathbf{y}$. Note that, by construction, one has $(\mathbf{y}^*)^T \cdot \mathbf{C} = \mathbf{y}^T \cdot \mathbf{A} = \mathbf{u}^T = \mathbf{z}^T \cdot \mathbf{C} \bmod q$.

We will show that, over the randomness of all algorithms, $\mathbf{y}^* \neq \mathbf{z}$ with overwhelming probability. Recall that Σ^* is a valid signature such that the implicit tracing algorithm either fails or outputs an index $e \notin U \setminus RL^*$.

- If the tracing algorithm fails, then, in particular, one has $\mathsf{Verify}(\mathsf{gpk}, \mathsf{grt}[d^*], \Sigma^*, M^*) = \mathsf{Valid}$. It follows from the correctness of the VLR group signature that $\mathbf{y}_0^T \cdot \mathbf{A}_0 \neq \mathsf{grt}[d^*] = \mathbf{z}_0^T \cdot \mathbf{A}_0$. This implies that $\mathbf{y}_0 \neq \mathbf{z}_0$, and thus $\mathbf{y}^* \neq \mathbf{z}$.
- If the tracing algorithm outputs $e \notin U \setminus RL^*$, namely the following two facts simultaneously hold true:

  $$\mathsf{Verify}(\mathsf{gpk}, \mathsf{grt}[e], \Sigma^*, M^*) = \mathsf{Invalid} \text{ and } \mathsf{Verify}(\mathsf{gpk}, RL^*, \Sigma^*, M^*) = \mathsf{Valid}.$$

  This leads to $\mathsf{grt}[e] \notin RL^*$, and hence $e \notin U$. Furthermore, the correctness of the revocation check and the computational binding property of COM imply that $\mathbf{y}_0^T \cdot \mathbf{A}_0 \bmod q = \mathsf{grt}[e]$. Now consider 2 cases:

  1. If $\mathcal{A}$ has never requested the secret key $\mathsf{gsk}[d^*]$, then $\mathbf{z}$ is unknown to $\mathcal{A}$. In this case, because $\mathbf{z}$ has large min-entropy given $\mathbf{u}$ (see Lemma 1.35), we have $\mathbf{z} \neq \mathbf{y}^*$ with overwhelming probability.

2. If the adversary $\mathcal{A}$ has requested the secret key $\mathsf{gsk}[d^*]$ in the $\mathsf{Queries}$ phase, then $d^* \in U$. In particular, it must be true that $d^* \neq e$ (because $e \notin U$) , and thus $\mathsf{grt}[d^*] \neq \mathsf{grt}[e]$. In other words, we have $\mathbf{y}_0^T \cdot \mathbf{A}_0 \neq \mathbf{z}_0^T \cdot \mathbf{A}_0 \bmod q$. This leads to $\mathbf{y}^* \neq \mathbf{z}$.

Now let $\mathbf{x} = \mathbf{z} - \mathbf{y}^* \in \mathbb{Z}^{(\ell+1)m}$, then $\mathbf{x} \neq \mathbf{0}$; $\mathbf{x}^T \cdot \mathbf{C} = \mathbf{0} \bmod q$; and $\|\mathbf{x}\|_\infty \leq \|\mathbf{z}\|_\infty + \|\mathbf{y}\|_\infty \leq \beta + \beta = 2\beta$. Algorithm $\mathcal{F}$ finally outputs the vector $\mathbf{x}$, which is a valid solution to the given $\mathrm{SIS}^\infty_{q,(\ell+1)\cdot m,2\beta}$ instance.

We observe that the probability that $\mathcal{F}$ does not abort is at least $1/N$, and conditioned on not aborting, it can solve the $\mathrm{SIS}^\infty_{q,(\ell+1)\cdot m,2\beta}$ problem with probability larger than $1/2 \cdot \left(1 - (7/9)^t\right)$ in time

$$T \cdot 32 \cdot q_{\mathcal{H}}/(\epsilon - 3^{-t}) + \mathsf{poly}(n, N).$$

This completes the proof. □

Part Four

Cryptographic Constructions: Multilinear Maps

Bilinear maps and multilinear maps have a lot of cryptographic applications, see [Jou00, SOK00, BF03] and [BS03, RS09, PTT10, Rot13], respectively. But unlike bilinear maps, built with pairings on elliptic curves, the construction of cryptographic multilinear maps was an open problem for several years. In [BS03], Boneh and Silverberg studied the interest of such maps, and gave two applications: multipartite Diffie-Hellman key exchange and very efficient broadcast encryption. But they conjectured that multilinear maps will probably "come from outside the realm of algebraic geometry."

The GGH Graded Encoding Scheme [GGH13a], based on ideal lattices, is the first plausible approximation to a cryptographic multilinear map. Unfortunately, using the security analysis in [GGH13a], the scheme requires very large parameters to provide security for its underlying "encoding re-randomization" process. Our main contributions are to formalize, simplify, in Chapter 9, and improve, in Chapter 10, the efficiency and the security analysis of the re-randomization process in the GGH construction. This results in a new construction that we call GGHLite, published in a joint work with Damien Stehlé and Ron Steinfeld [LSS14]. In particular, we first lower the size of a standard deviation parameter of the re-randomization process of [GGH13a] from exponential to polynomial in the security parameter. This first improvement is obtained via a finer security analysis of the "drowning" step of re-randomization, in which we apply the *Rényi divergence* instead of the conventional *statistical distance* as a measure of distance between distributions. Our second improvement is to reduce the number of randomizers needed from $\Omega(n \log n)$ to 2, where n is the dimension of the underlying ideal lattices. These two contributions allow us to decrease the bit size of the public parameters from $O(\lambda^5 \log \lambda)$ for the GGH scheme to $O(\lambda \log^2 \lambda)$ in GGHLite, with respect to the security parameter λ (for a constant multilinearity parameter κ).

In Chapter 9, we recall the Garg et al. scheme from [GGH13a], and its related hard problems. We then discuss the re-randomization step of the scheme and explain what should be expected from it, in terms of security. This security requirement is unclear in [GGH13a] and [AGHS13]. We formulate it precisely. In Chapter 10, we give our two main contributions: the reduction of the re-randomization drowning ratio from exponential to polynomial and the new leftover hash lemma over the ring $R = \mathbb{Z}[x]/(x^n + 1)$. Then we describe our GGHLite scheme and we compare the asymptotic parameters of GGHLite with those of the original GGH scheme. Finally, we show how to adapt the N-party non interactive Diffie-Hellman key exchange, such that our security result on GGHLite applies.

Chapter 9

The GGH Graded Encoding Scheme and the Security of its Rerandomization Procedure

Boneh and Silverberg [BS03] defined a *cryptographic κ-multilinear map* e as a map from $G_1 \times \ldots \times G_\kappa$ to G_T, all cyclic groups of order p, which enjoys three main properties: first, for any elements $g_i \in G_i$ for $i \leq \kappa$, $j \leq \kappa$ and $\alpha \in \mathbb{Z}_p$, we have $e(g_1, \ldots, \alpha \cdot g_j, \ldots, g_\kappa) = \alpha \cdot e(g_1, \ldots, g_\kappa)$; second, the map e is non-degenerate, i.e., if the g_i's are generators of their respective G_i's then $e(g_1, \ldots, g_\kappa)$ generates G_T; and third, there is no efficient algorithm to compute discrete logarithms in any of the G_i's. Multilinear maps have a lot of cryptographic applications [BS03, RS09, PTT10, Rot13], but the construction of cryptographic multilinear maps was an open problem for several years. In 2013, Garg, Gentry and Halevi [GGH13a] introduced the first "approximate" multilinear maps construction, based on ideal lattices, and the powerful notion of *graded encoding scheme*. Based on their work, Coron, Lepoint and Tibouchi [CLT13] recently described an alternative construction of graded encoding scheme.

In this chapter, we first describe the GGH multilinear maps construction from [GGH13a], then its underlying computational problems, and the strong re-randomization security requirement from [GGH13a]. Then we introduce our canonical computational problems and formulate our precise security goal for re-randomization with respect to the canonical problems. This security requirement is unclear in [GGH13a] and [AGHS13]. We formulate it precisely. Finally we give the implicit re-randomization security reduction of the GGH scheme.

9.1 Graded encoding scheme

In this section, we give the definition of a graded encoding scheme, and describe the adaptation of the application of [BS03]: the N-party non interactive Diffie-Hellman key exchange. Then we recall the hardness assumption associated to the security of this scheme.

9.1.1 Definition

We first define a graded encoding scheme.

Definition 9.1 ([GGH13a, Definition 2])**.** A κ-graded encoding system consists of a ring R and a system of sets $S = \{S_i^{(\alpha)} \subset \{0,1\}^* : \alpha \in R, 0 \leq i \leq \kappa\}$, with the following properties:

1. For every fixed index i, the sets $\{S_i^{(\alpha)} : \alpha \in R\}$ are disjoint.

2. There are an associative binary operation $+$ and a self-inverse unary operation $-$ such that for every $\alpha_1, \alpha_2 \in R$, every index $i \leq \kappa$ and every $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$, it holds that:

$$u_1 + u_2 \in S_i^{(\alpha_1+\alpha_2)} \qquad \text{and} \qquad -u_1 \in S_i^{(-\alpha_1)},$$

where $\alpha_1 + \alpha_2$ and $-\alpha_1$ are addition and negation in R.

3. There is an associative binary operation $\times$ such that for every $\alpha_1, \alpha_2 \in R$, every i_1, i_2 such that $i_1 + i_2 \leq \kappa$, and every $u_1 \in S_{i_1}^{(\alpha_1)}$ and $u_2 \in S_{i_2}^{(\alpha_2)}$, it hold that:

$$u_1 \times u_2 \in S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)},$$

where $\alpha_1 \cdot \alpha_2$ is multiplication in R and $i_1 + i_2$ is an integer addition.

A graded encoding scheme uses the notion of *encoding level*: the plaintext is a level-0 encoding, from the level-0 encoding one can construct level-i encoding of the same element until κ, where κ is called the multilinearity parameter. But given a level-i encoding, one cannot come back and find a level-j encoding for $j < i$ for the same element. The encodings are both additively and multiplicatively homomorphic, up to a limited number of operations. More precisely, a product of i level-1 encodings is a level-i encoding. One can multiply any number of encodings up to κ, instead of exactly κ in the ideal multilinear maps of [BS03]. We now construct it with the following procedures.

$\mathsf{InstGen}(1^\lambda, 1^\kappa) \to (\mathsf{params}, \mathbf{p}_{zt})$. This algorithm takes λ and κ as inputs and outputs $(\mathsf{params}, \mathbf{p}_{zt})$, where params is a description of the graded encoding system as above, and $\mathbf{p}_{zt}$ is a zero-testing parameter at level κ.

$\mathsf{Samp}(\mathsf{params}) \to a$. The ring sampler algorithm takes as input the parameters params and outputs a "level-0 encoding" $a \in S_0^{(\alpha)}$ for a nearly uniform element $\alpha \in R$.

$\mathsf{Enc}_i(\mathsf{params}, a) \to u$. The encoding algorithm takes as inputs the parameters params, a level i and a level-0 encoding $a \in S_0^{(\alpha)}$ of an element $\alpha \in R$. It outputs the level-i encoding $u \in S_i^{(\alpha)}$ for the same α.

$\mathsf{Add}(\mathsf{params}, i, u_1, u_2) \to u$. The addition algorithm takes as inputs the parameters params, a level i, and two level-i encodings $u_1 \in S_i^{(\alpha_1)}$ and $u_2 \in S_i^{(\alpha_2)}$. It outputs a level-i encoding $u_1 + u_2 \in S_i^{(\alpha_1+\alpha_2)}$.

$\mathsf{Neg}(\mathsf{params}, i, u_1) \to u$. The negation algorithm takes as inputs the parameters params, a level i, and a level-i encoding $u_1 \in S_i^{(\alpha_1)}$. It outputs a level-i encoding $-u_1 \in S_i^{(-\alpha_1)}$.

$\mathsf{Mult}(\mathsf{params}, i_1, i_2, u_1, u_2) \to u$. The multiplication algorithm takes as inputs the parameters params, two levels i_1 and i_2 such that $i_1 + i_2 \leq \kappa$, and a level-i_1 (resp. i_2) encoding $u_1 \in S_{i_1}^{(\alpha_1)}$ and $u_2 \in S_{i_2}^{(\alpha_2)}$. It outputs a level-$(i_1 + i_2)$ encoding $u_1 \times u_2 \in S_{i_1+i_2}^{(\alpha_1 \cdot \alpha_2)}$.

$\mathsf{isZero}(\mathsf{params}, \mathbf{p}_{zt}, u) \to \{0, 1\}$. The zero-test algorithm takes as inputs the parameters params, the zero-testing parameter $\mathbf{p}_{zt}$ and a level-κ encodings $u \in S_\kappa^{(\alpha)}$. It outputs 1 for every $u \in S_\kappa^{(0)}$, and 0 otherwise, except with negligible probability:

$$\Pr_{\alpha \in R}\left[\exists u \in S_\kappa^{(\alpha)} \text{ s.t. } \mathsf{isZero}(\mathsf{params}, \mathbf{p}_{zt}, u) = 1\right] = \text{negligible}(\lambda).$$

$\mathsf{Ext}(\mathsf{params}, \mathbf{p}_{zt}, u) \to s$. The extraction algorithm takes as inputs the parameters params, the zero-testing parameter $\mathbf{p}_{zt}$ and a level-κ encodings $u \in S_\kappa^{(\alpha)}$. It outputs s such that:

1. For a randomly chosen $a \leftarrow \mathsf{Samp}(\mathsf{params})$, and two encodings of a: $u_1 \leftarrow \mathsf{Enc}_\kappa(\mathsf{params}, a)$ and $u_2 \leftarrow \mathsf{Enc}_\kappa(\mathsf{params}, a)$ then:

$$\Pr\left[\mathsf{Ext}(\mathsf{params}, \mathbf{p}_{zt}, u_1) = \mathsf{Ext}(\mathsf{params}, \mathbf{p}_{zt}, u_2)\right] \geq 1 - \text{negligible}(\lambda).$$

2. The distribution $\{\mathsf{Ext}(\mathsf{params}, \mathbf{p}_{zt}, u) : a \leftarrow \mathsf{Samp}(\mathsf{params}), u \leftarrow \mathsf{Enc}_\kappa(\mathsf{params}, a)\}$ is nearly uniform over $\{0,1\}^\lambda$.

The zero-testing procedure allows to test if a level-κ encoding is an encoding of 0, and also to test if two encodings $u_1, u_2 \in S_\kappa$ encode the same element. This definition allows false positives for this procedure (with negligible probability) but no false negatives. The extraction procedure extracts a "canonical" representation of a ring element from their level-κ encoding. It is also a probabilistic procedure which may fail with negligible probability.

9.1.2 One-round N-party Diffie-Hellman key exchange

We recall the construction given by [GGH13a] to adapt the N-party Diffie-Hellman key exchange using an encoding scheme with $\kappa = N - 1$. The principle of the key exchange is that each party shares some public parameters and starts by sampling a secret key. Then each party publishs a public element (computed with its secret key), and given all the public elements and his secret, each party must be able to compute a shared secret key. The consistency requirement is that all parties must generate the same shared secret key.

- **Setup** $\mathsf{Setup}(1^\lambda, 1^N)$: Given security parameter λ and number of parties N, run $\mathsf{InstGen}(1^\lambda, 1^{N-1})$ for the graded encoding scheme to get (par, p_{zt}) and output protocol public parameters (par, p_{zt}).
- **Publish** $\mathsf{Publish}(\text{par}, p_{zt}, i)$: The ith party runs the level-0 encoding sampler to generate a random secret key $e_i = \mathsf{Samp}(\text{par})$, and publishes a corresponding level-1 public key $u_i = \mathsf{enc}_1(\text{par}, e_i)$.
- **KeyGen** $\mathsf{KeyGen}(\text{par}, p_{zt}, j, e_j, \{u_i\}_{i \neq j})$: The jth party computes a level-$(N-1)$ encoding $v_j = e_j \cdot \prod_{i \neq j} u_i$ of the product $\prod_i e_i$, and computes the key $K_j = \mathsf{ext}(\text{par}, p_{zt}, v_j)$.

Figure 9.1: The N-party Diffie-Hellman key exchange protocol.

The consistency requirement follows from the agreement property of the extraction procedure. As proven in [GGH13a], the security follows from the randomness property of the extraction procedure and from the GDDH assumption that we will now define.

9.1.3 Hardness assumption: GDDH

The hardness assumption of this scheme is associated to an extension of the discrete logarithm and the Decisional Diffie-Hellman (DDH) assumptions in multilinear group. The security game asks to distinguish between a level-κ encoding of a random element, and a level-κ encoding of the product of $\kappa + 1$ elements, knowing the level-1 encoding of all these elements. Note that with all

the level-1 encodings, it is easy to compute a "level-$\kappa+1$" encoding of the product by multiplying them, but hard to find a level-κ encoding.

Given the security game of Figure 9.2, the Graded Decisional Diffie-Hellman (GDDH) assumption is that the two following distributions are computationally indistinguishable:

$$\mathcal{D}_{GDDH} = \{(\mathsf{params}, p_{zt}, \{u_i\}_i, u^*)\} \text{ and } \mathcal{D}_{Rand} = \{(\mathsf{params}, p_{zt}, \{u_i\}_i, \widehat{u})\}.$$

Given parameters λ, κ, proceed as follows:

1. $(\mathsf{params}, \mathbf{p}_{zt}) \leftarrow \mathsf{InstGen}(1^\lambda, 1^\kappa)$.
2. For $i = 0, \ldots, \kappa$:
   - Choose $a_i \leftarrow \mathsf{Samp}(\mathsf{params})$,
   - Set $u_i \leftarrow \mathsf{Enc}(\mathsf{params}, 1, a_i)$.
3. Set $a^* = \left[\prod_{i=0}^{\kappa} a_i\right]_q$,
4. Set $\widehat{a} \leftarrow \mathsf{Samp}(\mathsf{params})$,
5. Set $u^* \leftarrow \mathsf{Enc}(\mathsf{params}, \kappa, a^*)$.
6. Set $\widehat{u} \leftarrow \mathsf{Enc}(\mathsf{params}, \kappa, \widehat{a})$.

Figure 9.2: Graded Decisional Diffie Hellman security game.

9.2 The GGH scheme

9.2.1 Description of the scheme

We recall the GGH scheme in Figure 9.3.

If we come back to the definition of cryptographic multilinear maps, the authors of [GGH13a] notice that $\alpha \cdot g_i$ can be viewed as an "encoding" of the "plaintext" $\alpha \in \mathbb{Z}_q$. They consider the polynomial rings $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = R/qR$ (replacing the exponent space $\mathbb{Z}_p$). They generate a small secret $g \in R$ and let $\mathcal{I} = (g)$ be the principal ideal over R generated by g. They also sample a uniform $z \in R_q$ which stays secret. The "plaintext" is an element of $R/\mathcal{I}$, and is encoded via a division by z in R_q: to encode a coset of $R/\mathcal{I}$, return $[c/z]_q$, where c is an arbitrary small coset representative. In practice, as g is hidden, they give another public parameter y, which is an encoding of 1, and the encoding of the coset is computed as $[e \cdot y]_q$, where e is a small coset representative (possibly different from c). But, as opposed to multilinear maps, their graded encoding scheme uses the notion of *encoding level*: the plaintext e is a level-0 encoding, the encoding $[c/z]_q$ is a level-1 encoding, and at level i, an encoding of $e + \mathcal{I}$ is given by $[c/z^i]_q = [e \cdot y^i]_q$. For $0 \leq 1 \leq \kappa$, the sets of Definition 9.1 are such that:

$$S_i^{(e)} = \{c/z^i \in R_q : c \in e + \mathcal{I}, \|c\| \leq q^{1/8}\},$$

To ensure the security of the cryptographic constructions, the second main difference with multilinear maps is the randomization of the encodings. The principle is as follows:

- First some level-1 encodings of 0, called $\{x_j = [b_j/z]_q\}_{j \leq m_r}$, are given as part of the public parameters;

- **Instance generation** $\mathsf{InstGen}(1^\lambda, 1^\kappa)$: Given security parameter λ and multilinearity parameter κ, determine scheme parameters n, q, m_r, σ, σ', $\ell_{g^{-1}}$, ℓ, based on the scheme analysis. Then proceed as follows:
  - Sample $g \hookleftarrow D_{R,\sigma}$ until $\|g^{-1}\| \leq \ell_{g^{-1}}$ and $\mathcal{I} = (g)$ is a prime ideal. Define encoding domain $R_g = R/(g)$.
  - Sample $z \hookleftarrow U(R_q)$.
  - Sample a level-1 encoding of 1: set $y = [a \cdot z^{-1}]_q$ with $a \hookleftarrow D_{1+\mathcal{I},\sigma'}$.
  - For $k \leq \kappa$, sample m_r level-k encodings of 0: set $x_j^{(k)} = [b_j^{(k)} \cdot z^{-k}]_q$ with $b_j^{(k)} \hookleftarrow D_{\mathcal{I},\sigma'}$ for all $j \leq m_r$.
    (Note that $a = 1 + gr_y$ and $b_j^{(k)} = gr_j^{(k)}$ for some $r_y, r_j^{(k)} \in R$.)
  - Sample $h \hookleftarrow D_{R,\sqrt{q}}$ and define the zero-testing parameter $p_{zt} = [\frac{h}{g} z^\kappa]_q \in R_q$.
  - Return public parameters $\mathrm{par} = (n, q, y, \{x_j^{(k)}\}_{j \leq m_r, k \leq \kappa})$ and p_{zt}.
- **Level-0 sampler** $\mathsf{samp}(\mathrm{par})$: Sample $e \hookleftarrow D_{R,\sigma'}$ and return e.
  (Note that $e = e_L + ge_H$ for some unique coset representative $e_L \in \mathcal{P}_g$, and some $e_H \in R$.)
- **Level-k encoding** $\mathsf{enc}_k(\mathrm{par}, e)$: Given level-0 encoding $e \in R$ and parameters par:
  - Encode e at level k: Compute $u' = [e \cdot y^k]_q$.
  - Re-randomize: Sample $\rho_j \hookleftarrow \chi_k$ for $j \leq m_r$ and return $u = [u' + \sum_{j=1}^{m_r} \rho_j x_j^{(k)}]_q$.
    (Note that $u' = [c'/z^k]_q$ with $c' \in e_L + \mathcal{I}$ and $u = [(c' + \sum_j \rho_j b_j^{(k)})/z^k]_q$.)
- **Adding encodings** add: Given level-k encodings $u_1 = [c_1/z^k]_q$ and $u_2 = [c_2/z^k]_q$:
  - Return $u = [u_1 + u_2]_q$, a level-k encoding of $[c_1 + c_2]_g$.
- **Multiplying encodings** mult: Given level-k_1 encoding $u_1 = [c_1/z^{k_1}]_q$ and a level-k_2 encoding $u_2 = [c_2/z^{k_2}]_q$:
  - Return $u = [u_1 \cdot u_2]_q$, a level-$(k_1 + k_2)$ encoding of $[c_1 \cdot c_2]_g$.
- **Zero testing at level** κ $\mathsf{isZero}(\mathrm{par}, p_{zt}, u)$: Given a level-$\kappa$ encoding $u = [c/z^\kappa]_q$, return 1 if $\|[p_{zt} u]_q\|_\infty < q^{3/4}$ and 0 else.
  (Note that $[p_{zt} \cdot u]_q = [hc/g]_q$.)
- **Extraction at level** κ $\mathsf{ext}(\mathrm{par}, p_{zt}, u)$: Given a level-$\kappa$ encoding $u = [c/z^\kappa]_q$, return $v = \mathrm{MSB}_\ell([p_{zt} \cdot u]_q)$.
  (Note that if $c = [c]_g + gr$ for some $r \in R$, then $v = MSB_\ell(\frac{h}{g}([c]_g + gr)) = MSB_\ell(\frac{h}{g}[c]_g + hr)$, which is equal to $MSB_\ell(\frac{h}{g}[c]_g)$, with probability $1 - \lambda^{-\omega(1)}$.)

Figure 9.3: The GGH graded encoding scheme.

- Then, to randomize a level-1 encoding $u' = [e \cdot y]_q$, one outputs:

$$u = [u' + \sum_j \rho_j x_j]_q = [c/z]_q,$$

with $c = c' + \sum_j \rho_j b_j$, where the ρ_j's are sampled from a discrete Gaussian distribution over $\mathbb{Z}$ with deviation parameter σ^*.

In Figure 9.3, we present this scheme in a slightly more general form than [GGH13a]: we leave as a parameter the distribution χ_k of the re-randomization coefficients ρ_j for a level-k encoding (for any $k \leq \kappa$). In the original GGH scheme, we have $\chi_k = D_{\mathbb{Z},\sigma_k^*}$ for some σ_k^*'s, i.e., the ρ_j's are integers sampled from a discrete Gaussian distribution.

The aim of isZero is to test whether the input $u = [c/z^\kappa]_q$ is a level-κ encoding of 0 or not, i.e., whether $c = g \cdot r$ for some $r \in R$. The following conditions (explained in Section 9.2.2) ensure correctness of isZero, when $\chi_k = D_{\mathbb{Z},\sigma_k^*}$ (for all $k \leq \kappa$): the first one implies that false negatives do not exist (if u is level-κ encoding of 0, then isZero(u) returns 1), whereas the second one implies that false positives occur with negligible probability.

$$q > \max((n\ell_{g^{-1}})^8, ((m_r+1) \cdot n^{1.5}\sigma_1^*\sigma')^{8\kappa}) \quad (9.1)$$

$$q > (2n\sigma)^4. \quad (9.2)$$

The aim of ext is to extract a quantity from its input $u = [c/z^\kappa]_q$ that depends only on the encoded value $[c]_g$, but not on the randomizers. To avoid trivial solutions, one requires that this extracted value has min-entropy $\geq 2\lambda$ (if that is the case, then one can obtain a uniform distribution on $\{0,1\}^\lambda$, using a strong randomness extractor). The following two inequalities (also explained in Section 9.2.2) guarantee these properties, when $\chi_k = D_{\mathbb{Z},\sigma_k^*}$ (for all k). The first one implies that $\varepsilon_{ext} = \Pr[\mathsf{ext}(u) \neq \mathsf{ext}(u')]$ is negligible, when u and u' encode the same value $[c]_g$, whereas the second one provides large min-entropy.

$$1/4 \ln q - \ln(\frac{2n}{\varepsilon_{ext}}) \geq \ell \geq \ln(8n\sigma). \quad (9.3)$$

9.2.2 Correctness analysis of the scheme

We now explain how to derive these correctness conditions. For this, we need the following result.

Lemma 9.2 (Adapted from [GGH13a, Lemma 4]). *Let $g \in R$ such that $\mathcal{I} = (g)$ is a prime ideal in R, let $c \in R$ with $\|c\| < q^{1/8}$ and $h \in R$ with $\|h\| < \sqrt{n}q^{1/2}$ and $c, h \notin \mathcal{I}$ and $q > (2tn\sigma)^4$ for some $t \geq 1$. Then $\|[h \cdot c/g]_q\|_\infty > t \cdot q^{3/4}$.*

Correctness of zero-testing. To satisfy the "no false negatives" zero-testing condition, we need $\|p_{zt}u\|_\infty < q^{3/4}$ for all valid level-κ encodings $u = [c/z^\kappa]_q \in S_\kappa^{(0)}$ of zero. As $p_{zt} = [\frac{h}{g}z^k]_q$, we have

$$\|[p_{zt}u]_q\|_\infty = \|[h \cdot \frac{c}{g}]_q\|_\infty = \|hc/g\|_\infty \leq \|h\| \cdot \|c\| \cdot \|g^{-1}\|\sqrt{n}.$$

To satisfy $\|p_{zt}u\|_\infty < q^{3/4}$, it therefore suffices to have $\|c\| \leq q^{1/8}$ and $\|h\| \cdot q^{1/8} \cdot \ell_{g^{-1}} \cdot \sqrt{n} < q^{3/4}$.

- As $\|c\| = \|\prod_{i=1}^\kappa u_i\| \leq \sqrt{n}^{\kappa-1} \cdot (\max_i \|u_i\|)^\kappa$ with $\|u_i\| = \|e_i + \sum_j \rho_j b_j^{(1)}\|$, we use the fact that $\|e_i\| \leq \sigma'\sqrt{n}$, $\|\rho_j\| \leq \sigma_1^*\sqrt{n}$ (by Lemma 1.36), then:

$$\|u_i\| \leq \|e_i\| + m_r \max_j |\rho_j| \cdot \|b_j^{(1)}\| \leq (m_r+1) \cdot n\sigma_1^*\sigma'.$$

As a consequence, then condition $\|c\| \leq q^{1/8}$ is satisfied if $q > ((m_r+1) \cdot n^{1.5}\sigma_1^*\sigma')^{8\kappa}$.

- As $\|h\| \leq \sqrt{n}q^{1/2}$, the condition $\|h\| \cdot q^{1/8} \cdot \ell_{g^{-1}} \cdot \sqrt{n} < q^{3/4}$ is satisfied if $q > (\ell_{g^{-1}} n)^8$

Then these two conditions are satisfied and $\|p_{zt}u\|_\infty < q^{3/4}$ if:

$$q > \max\left((n\ell_{g^{-1}})^8, ((m_r+1) \cdot n^{1.5}\sigma_1^*\sigma')^{8\kappa}\right). \tag{9.4}$$

To satisfy the "negligible probability false positives" zero-testing condition, we need $\|p_{zt}u\|_\infty > q^{3/4}$, for any level-$\kappa$ encoding $u = [c/z^\kappa]_q \in S_\kappa^{(e_L)}$ of $e_L \in R_g$, except with negligible probability $\varepsilon_{zt} = \lambda^{-\omega(1)}$ over the uniformly random choice of $e_L \in R_g$. By Lemma 9.2 with $t = 1$, the fact that I is prime and that $\|c\| < q^{1/8}$, it follows that $\|p_{zt}u\|_\infty > q^{3/4}$ for any encoding of a non-zero $e_L \notin I$ (and hence $\varepsilon_{zt} = \Pr[e_L = 0] = 1/|R_g| = O(2^{-n})$), assuming the condition

$$q > (2n\sigma)^4. \tag{9.5}$$

We have $h \notin \mathcal{I}$, except with probability $O(1/|R/\mathcal{I}|)$ over the choice of h, by Lemma 1.33, when $q = \omega(n\sigma)^2$. Note that thanks to the remark just after Lemma 10.1, we have $|R/I| \geq \sigma_n(\mathrm{rot}(g))^n \geq (\frac{1}{\sqrt{n}\cdot\|g^{-1}\|})^n$. Now, by the InstGen rejection test, we have $\|g^{-1}\| \leq \ell_{g^{-1}}$. Condition (10.1) finally implies that $|R/I| \geq 2^n$ when $n \geq 8$.

Correctness of extraction. To satisfy the extraction min-entropy condition, we need that the min-entropy of $[p_{zt}u]_q$ is $\geq 2\lambda$. Indeed, any two level-κ encodings $u = [(e_L + gr)/z^\kappa]_q$ and $u' = [(e'_L + gr')/z^\kappa]_q$ of different elements $e_L \neq e'_L \in R_g$ have different extracted elements $\mathrm{MSB}_\ell(p_{zt}u) \neq \mathrm{MSB}_\ell(p_{zt}u')$ as long as:

$$\|[p_{zt}u]_q - [p_{zt}u']_q\|_\infty = \|[p_{zt}(u-u')]_q\|_\infty > 2^{L-\ell+1}.$$

If that condition is satisfied, then the min-entropy is $\log_2|R/\mathcal{I}|$. As $|R/I| \geq 2^n$ for $n \geq 8$ (see above), we have $\log_2|R/\mathcal{I}| \geq n \geq 2\lambda$. We now prove that the condition $\|[p_{zt}(u-u')]_q\|_\infty > 2^{L-\ell+1}$ is satisfied. Since $u - u'$ is an encoding of a non-zero element $e_L - e'_L \in R_g$ this follows, similarly to the zero-testing correctness above, from Lemma 9.2 with t satisfying $tq^{3/4} > 2^{L-\ell+1}$. The latter holds with $t = q^{1/4}2^{-\ell+2}$. The condition $t > 1$ is satisfied by the upper bound (9.7) on ℓ below, while the condition $q > (2tn\sigma)^4$ is satisfied by the lower bound

$$\ell > \ln(8n\sigma). \tag{9.6}$$

To satisfy the "negligible failure probability" extraction condition, we need:

$$\mathrm{MSB}_\ell(p_{zt}u) = \mathrm{MSB}_\ell(p_{zt}u'),$$

for any two level-κ encodings $u = [(e_L + gr)/z^\kappa]_q$ and $u' = [(e_L + gr')/z^\kappa]_q$ of the same element $e_L \in R_g$, except with negligible probability ε_{ext} over the uniformly random choice of $e_L \in R_g$. Since $[p_{zt}u]_q = [he_L/g]_q + hr$ and $[p_{zt}u']_q = [he_L/g]_q + hr'$ with $\|hr\|_\infty, \|hr'\|_\infty < q^{3/4}$, we can only have $\mathrm{MSB}_\ell(p_{zt}u) \neq \mathrm{MSB}_\ell(p_{zt}u')$ if he_L/g falls within infinity distance $< q^{3/4}$ of a multiple of $2^{L-\ell+1}$, where $L = \lfloor \log q \rfloor$. Under the heuristic assumption that each coefficient of $[he_L/g]_q$ is uniformly random in $\mathbb{Z}_q$ over the choice of e_L (this heuristic assumption is reasonable from the point of view of entropy; indeed, by the min-entropy condition above, the entropy of $[he_L/g]_q \in R_q$ over the choice of e_L uniformly in $R/\mathcal{I}$, is at least n bit, and this exceeds $\log_2 q$ because of the lattice rule of thumb security requirement $n = \Omega(\lambda \log q)$ in Eq. (10.12)), we have by a union bound over all n coefficients that this "bad" event occurs with probability:

$$p \leq \frac{2nq^{3/4}}{2^{L-\ell+1}}.$$

To make this probability $\leq \varepsilon_{ext}$, it suffices to take

$$\ell \leq \frac{1}{4}\ln q - \ln(\frac{2n}{\varepsilon_{ext}}). \tag{9.7}$$

9.3 Security of the GGH scheme

We now describe the related hard problems and the security requirement for this scheme.

9.3.1 The GDDH, GCDH and Ext-GCDH problems

The computational problems that are required to be hard for the GGH scheme depend on the application. Here we recall the definitions of the Graded Decisional and Computational Diffie-Hellman (GDDH and GCDH) problems from [GGH13a]. We introduce another natural variant that we call the Extraction Graded Computational Diffie-Hellman (Ext-GCDH), in which the goal is to compute the extracted string of a Diffie-Hellman encoding.

We first define the GGH security experiment in Figure 9.4.

Given parameters $\lambda, n, q, m_r, \kappa, \sigma'$, proceed as follows:

1. Run $\mathsf{InstGen}(1^n, 1^\kappa)$ to get
   $\mathrm{par} = (n, q, y, \{x_j^{(k)}\}_{j,k})$ and p_{zt}.
2. For $i = 0, \ldots, \kappa$:
   - Sample $e_i \hookleftarrow D_{R,\sigma'}$, $f_i \hookleftarrow D_{R,\sigma'}$,
   - Set $u_i = [e_i \cdot y + \sum_j \rho_{ij} x_j]_q$
     with $\rho_{ij} \hookleftarrow \chi_1$ for all j.
3. Set $u^* = [\prod_{i=1}^{\kappa} u_i]_q$.
4. Set $v_C = [e_0 u^*]_q$.
5. Sample $\rho_j \hookleftarrow \chi_\kappa$ for all j,
   set $v_D = [e_0 u^* + \sum_j \rho_j x_j^{(\kappa)}]_q$.
6. Set $v_R = [f_0 u^* + \sum_j \rho_j x_j^{(\kappa)}]_q$.

Figure 9.4: The GGH security experiment.

Definition 9.3 (GCDH/Ext-GCDH/GDDH)**.** The problems GCDH, Ext-GCDH and GDDH are defined as follows with respect to experiment of Figure 9.4:[1]

- **κ-graded CDH problem (GCDH):** On inputs par, p_{zt} and the u_i's of Step 2, output a level-κ encoding of $\prod_{i\geq 0} e_i + \mathcal{I}$, i.e., $w \in R_q$ such that $\|[p_{zt}(v_C - w)]_q\| \leq q^{3/4}$.
- **Extraction κ-graded CDH problem (Ext-GCDH):** On inputs par, p_{zt} and the u_i's of Step 2, output the extracted string for a level-κ encoding of $\prod_{i\geq 0} e_i + \mathcal{I}$, i.e., $w = \mathsf{ext}(\mathrm{par}, p_{zt}, v_C) = \mathrm{MSB}_\ell([p_{zt} \cdot v_C]_q)$.
- **κ-graded DDH problem (GDDH):** Distinguish between v_D and v_R, i.e., between the distributions $\mathcal{D}_{DDH} = \{\mathrm{par}, p_{zt}, (u_i)_{0\leq i\leq\kappa}, v_D\}$ and $\mathcal{D}_R = \{\mathrm{par}, p_{zt}, (u_i)_{0\leq i\leq\kappa}, v_R\}$.

[1] Note that we use a slightly different process from [GGH13a], by adding a re-randomization to the element v_D. Without it, there exists a "division attack" against GDDH.

Ext-GCDH is at least as hard as GDDH: given v_x with $x \in \{\mathrm{DDH}, \mathrm{R}\}$, use the Ext-GCDH oracle to compute $w = \mathsf{ext}(\mathrm{par}, p_{zt}, v_C)$.

9.3.2 The GGH re-randomization security requirement

The encoding re-randomization step in the GGH scheme is necessary for the hardness of the problems above. In [GGH13a], Garg et al. imposed the informal requirement that the re-randomization process "erases" the structure of the input encoding, while preserving the encoded coset. In setting parameters, they interpreted this requirement in the following natural way.

Definition 9.4 (Strong re-randomization security requirement)**.** We let:

* $u' = [c'/z^k]_q$, with $c' = e_L + gr'$ be a fixed level-k encoding of $e_L \in R_g$,
* $u = [u' + \sum_j \rho_j x_k^{(j)}]_q = [c/z^k]_q$ with $c = e_L + gr$ and $r = r' + \sum_j \rho_j r_j^{(k)}$ be the re-randomized encoding, with $\rho_j \hookleftarrow \chi_k$ for $j \le m_r$.
* $D_u^{(k)}(e_L, r')$ denote the distribution of u (over the randomness of ρ_j's), parameterized by (e_L, r'),
* $D_{\mathrm{can}}^{(k)}(e_L)$ denote some canonical distribution, parameterized by e_L, that is independent of r'.

Then we say that the *strong* re-randomization security requirement is satisfied at level k with respect to $D_{\mathrm{can}}^{(k)}(e_L)$ and encoding norm $\gamma^{(k)}$ if

$$\Delta(D_u^{(k)}(e_L, r'), D_{\mathrm{can}}^{(k)}(e_L)) \le 2^{-\lambda}$$

for any $u' = [c'/z^k]_q$ with $\|c'\| \le \gamma^{(k)}$.

The authors of [GGH13a] argued that with $\chi_k = D_{\mathbb{Z},\sigma_k^*}$ (for $k \le \kappa$) and a "drowning ratio" $\sigma_k^*/\|r'\|$ exponential in security parameter λ, the distribution $D_u^{(k)}(e_L, r')$ is within negligible statistical distance to the canonical distribution $D_{\mathrm{can}}^{(k)}(e_L) = [D_{\mathcal{I}+e_L,\sigma_k^*(B^{(k)})^T} \cdot z^{-k}]_q$. This requirement may be stronger than needed. Accordingly, we now clarify the desired goal.

9.3.3 Our security goal: canonical assumptions

We formalize a re-randomization security goal to capture a security guarantee against "statistical correlation" attacks on GCDH/Ext-GCDH/GDDH. We define *canonical variants* cGCDH/Ext-cGCDH/cGDDH of GCDH/Ext-GCDH/GDDH, using Figure 9.6.

The main difference with Figure 9.5 is that the encodings $u_i = [c_i/z]_q$ of the hidden elements e_i, are sampled from a canonical distribution $D_{\mathrm{can}}^{(1)}(e_i)$, parameterized by e_i, whose statistical parameters are independent of the encoded coset e_i, so that it is "by construction" immune against statistical correlation attacks. In particular, in the canonical distribution $D_{\mathrm{can}}^{(1)}(e_i)$ that we use, c_i is sampled from a discrete Gaussian distribution $D_{\mathcal{I}+e_i,\sigma_1^*(B^{(1)})^T}$ (over the choice of the randomization, for a fixed e_i), whose statistical parameters such as center (namely 0) and deviation matrix $\sigma_1^*(B^{(1)})^T$ are independent of e_i. The only dependence this distribution has on the encoded element e_i is via its support $\mathcal{I} + e_i$.

We believe the canonical problems are cleaner and more natural than the non-canonical variants, since they decouple the re-randomization aspect from the rest of the computational problem. As a further simplification, the canonical variants also have their level-0 elements e_i distributed uniformly on R_g (rather than as reductions mod $\mathcal{I}$ of Gaussian samples).

Definition 9.5 (cGCDH/Ext-cGCDH/cGDDH)**.** The canonical problems cGCDH, Ext-cGCDH and cGDDH are defined as follows with respect to the experiment of Figure 9.6 and canonical encoding distribution $D_{can}^{(k)}(e)$ (parameterized by encoding level k and encoded element e):

Given parameters $\lambda, n, q, m_r, \kappa, \sigma'$, proceed as follows:

1. Run $\mathsf{InstGen}(1^n, 1^\kappa)$ to get $\text{par} = (n, q, y, \{x_j^{(k)}\}_{j,k})$ and p_{zt}.
2. For $i = 0, \ldots, \kappa$:
   - Sample $e_i \hookleftarrow D_{R,\sigma'}$, $f_i \hookleftarrow D_{R,\sigma'}$,
   - Set $u_i = [e_i \cdot y + \sum_j \rho_{ij} x_j]_q$ with $\rho_{ij} \hookleftarrow \chi_1$ for all j.
3. Set $u^* = [\prod_{i=1}^{\kappa} u_i]_q$.
4. Set $v_C = [e_0 u^*]_q$.
5. Sample $\rho_j \hookleftarrow \chi_\kappa$ for all j, set $v_D = [e_0 u^* + \sum_j \rho_j x_j^{(\kappa)}]_q$.
6. Set $v_R = [f_0 u^* + \sum_j \rho_j x_j^{(\kappa)}]_q$.

Figure 9.5: The GGH security experiment.

Given parameters $\lambda, n, q, m_r, \kappa, (\sigma_k^*)_{k \le \kappa}$, proceed as follows:

1. Run $\mathsf{InstGen}(1^n, 1^\kappa)$ to get $\text{par} = (n, q, y, \{x_j^{(k)}\}_{j,k})$ and p_{zt}. Write $x_j^{(k)} = [b_j^{(k)} z^{-k}]_q$ and $B^{(k)} = [b_1^{(k)}, \cdots, b_{m_r}^{(k)}] \in \mathcal{I}^{m_r}$.
2. For $i = 0, \ldots, \kappa$:
   - Sample $e_i \hookleftarrow U(R_g)$, $f_i \hookleftarrow U(R_g)$,
   - Set $u_i = [c_i z^{-1}]_q \hookleftarrow D_{can}^{(1)}(e_i)$ with $c_i \hookleftarrow D_{\mathcal{I}+e_i, \sigma_1^*(B^{(1)})^T}$.
3. Set $u^* = [\prod_{i=1}^{\kappa} u_i]_q$.
4. Set $v_C = [e_0 u^*]_q$.
5. Set $v_D = [c_D \cdot z^{-\kappa}]_q \hookleftarrow D_{can}^{(\kappa)}(\prod_{i=0}^{\kappa} e_i)$, with $c_D \hookleftarrow D_{\mathcal{I}+\prod_{i=0}^{\kappa} e_i, \sigma_\kappa^*(B^{(\kappa)})^T}$.
6. Set $v_R = [c_R \cdot z^{-\kappa}]_q \hookleftarrow D_{can}^{(\kappa)}(f_0 \prod_{i=1}^{\kappa} e_i)$, with $c_R \hookleftarrow D_{\mathcal{I}+f_0 \prod_{i=1}^{\kappa} e_i, \sigma_\kappa^*(B^{(\kappa)})^T}$.

Figure 9.6: The canonical security experiment.

- **cGCDH:** On inputs par, p_{zt} and the u_i's, output $w \in R_q$ such that $\|[p_{zt}(v_C - w)]_q\| \le q^{3/4}$.
- **Ext-cGCDH:** On inputs par, p_{zt} and the u_i's, output: $w = \mathsf{ext}(\text{par}, p_{zt}, v_C) = \text{MSB}_\ell([p_{zt} \cdot v_C]_q)$.
- **cGDDH:** Distinguish between

$$\mathcal{D}_{DDH} = \{\text{par}, p_{zt}, (u_i)_{0 \le i \le \kappa}, v_D\} \text{ and } \mathcal{D}_R = \{\text{par}, p_{zt}, (u_i)_{0 \le i \le \kappa}, v_R\}.$$

Remark 9.6. One could consider alternative definitions of natural canonical encoding distributions besides the one we adopt here. For instance, our results in the Chapter 10 can also be adapted to hold for the canonical distribution $D_{\text{can}}^{(1)}(e_i)$ of $u_i = [c_i/z]_q$ in which c_i is sampled from $D_{\mathcal{I}+e_i, \sigma_1^*(B^{(1)})^T, e_i}$. In this alternative, although the center of c_i's distribution depends on e_i, the distribution of the randomizer r in the representation $c_i = e_i + g \cdot r$, is independent of e_i. Our results can also be adapted to apply to this variant of the problem.

Given the canonical problems on whose hardness we wish to rely, our security goal for re-randomization with respect to the GCDH (resp. Ext-GCDH/GDDH) problems can now be easily formulated: hardness of the latter should be implied by hardness of the former.

Definition 9.7 (Re-randomization security goal)**.** We say that the re-randomization security goal is satisfied with respect to GCDH (resp. Ext-GCDH/GDDH) if any adversary against GCDH (resp. Ext-GCDH/ GDDH) with run-time $T = O(2^\lambda)$ and advantage $\varepsilon = \Omega(2^{-\lambda})$ can

be used to construct an adversary against cGCDH (resp. Ext-cGCDH/cGDDH) with run-time $T' = \text{poly}(T, \lambda)$ and advantage $\varepsilon' = \Omega(\text{poly}(\varepsilon, \lambda))$.

9.3.4 Review of GGH re-randomization security reduction

To set the background for our result, we now show that Definition 9.4 implies that our security goal is reached: We review the re-randomization security reduction from the non-canonical problems to their canonical variant, that is implicit in the work of Garg et al (GGH) [GGH13a]. For simplicity, we explain it for the case of Ext-GCDH, although it holds similarly for the other variants GCDH and GDDH.

First step. The first step is to show that re-randomization security goal in Definition 9.7 is satisfied if the Strong Re-randomization requirement in Definition 9.4 is satisfied. Let $\mathcal{A}$ denote the (T, ε) adversary against problem Ext-GCDH, we define the following games:

Game_1: In this game, $e_i \hookleftarrow D_{R,\sigma'}$, $u'_i = [e_i \cdot y]_q = [(e_{i,L} + gr'_i)/z]_q$, and $u_i = [u'_i + \sum_j \rho_{ij} x_j]_q$ where $\rho_{ij} \hookleftarrow D_{R,\sigma_1^*}$, for $i \in \{0, \ldots, \kappa\}$ and $j \in \{1, \ldots m_r\}$.

Game_3: In this game, $e_i \hookleftarrow D_{R,\sigma'}$ and $u_i = [(e_{i,L} + gr_i)/z]_q$ with $e_{i,L} = [e_i]_g$ and $gr_i \hookleftarrow D^{(1)}_{\text{can}} = D_{\mathcal{I},\sigma_1^*(B^{(1)})^T}$.

Note that the only difference between the two games is the distribution of the randomizer r_i in both games: in Game_1, we have $r_i = r'_i + \sum_j \rho_{ij} r_j^{(1)}$, which has the distribution D_r in Definition 9.4 (over the randomness of ρ_{ij}), while in Game_3, we have r_i sampled from the canonical distribution $D^{(1)}_{\text{can}}$. Hence, by the strong re-randomization requirement in Definition 9.4, the statistical distance between the r_i's in the two games is $\leq 2^{-\lambda}$. Therefore, we have that the statistical distance between the distributions of the view of $\mathcal{A}$ in the two games is at most $(\kappa + 1) \cdot 2^{-\lambda}$.

Finally, we define:

Game_4: This game denotes the Ext-cGCDH game.

The only difference between Game_3 and Game_4 is the distribution of $e_{i,L}$: in Game_3, we have $e_{i,L} = [e_i]_g$ with e_i sampled from $D_{R,\sigma'}$, whereas in Game_4 we have $e_{i,L}$ sampled uniformly from R_g. By Lemma 1.33, if $\sigma' \geq \eta_{\varepsilon_e}(\mathcal{I})$, then the statistical distance between the distributions of $e_{i,L}$ in both games is $\leq 2\varepsilon_e$, so that the statistical distance between the view of $\mathcal{A}$ in both games is $O(\kappa \cdot \varepsilon_e)$. By Lemma 1.27, the latter condition is satisfied if

$$\sigma' = \|g\| \cdot \Omega\left(\sqrt{\log(n\varepsilon_e^{-1})}\right) \geq \sigma\sqrt{n} \cdot \Omega\left(\sqrt{\log(n\varepsilon_e^{-1})}\right). \tag{9.8}$$

Second step. The second step is to show that the strong re-randomization requirement in Definition 9.4 is satisfied, i.e., that the distribution $D^{(1)}_{\text{can}}$ of r_i in Game_3 is statistically close to the distribution of r_i in Game_1. To do so, consider the intermediate game Game_2,

Game_2: In this game, the distribution of the term $\sum_j \rho_{ij} r_j^{(1)}$ is replaced by $D^{(1)}_{\text{can}}$, so that $r_i = r'_i + w$, where $w \hookleftarrow D^{(1)}_{\text{can}}$.

There are now two changes to analyze:

- For the change from Game_1 to Game_2, the authors of [GGH13a] apply a discrete Gaussian variant of the leftover hash Lemma from [AGHS13] (see Theorem 10.7 in Section 10.2.1) to show that $\Delta(\sum_j \rho_{ij} r_j^{(1)} : \rho_{ij} \hookleftarrow D_{\mathbb{Z},\sigma_1^*}; D_{\mathcal{I},\sigma_1^*(B^{(1)})^T}) \leq 2\varepsilon_\rho$ if $m_r = \Omega(n \log n)$ and $\sigma_1^* = \Omega(mn^2 \log(1/\varepsilon_\rho))$.
- For the change from Game_2 to Game_3, the authors of [GGH13a] argue (informally) that if the randomizer deviation parameter σ_1^* is sufficiently large to "drown" the offset $r_i' \in \mathcal{I}$ by an exponential ratio, i.e., if $\sigma^*/\|r_i'\| \geq 2^\lambda$, then the statistical distance between $r_i' + D_{\mathcal{I},\sigma_1^*(B^{(1)})^T}$ and $D_{\mathcal{I},\sigma_1^*(B^{(1)})^T}$ is $O(\|r_i'\|/\sigma^*) \leq O(2^{-\lambda})$.

Overall, the statistical distance between the view of $\mathcal{A}$ in Game_1 and Game_4 is $\Delta(\mathsf{Game}_1, \mathsf{Game}_4) = O(\kappa \cdot (\varepsilon_\rho + \|r_i'\|/\sigma^* + \varepsilon_e))$. Therefore, algorithm $\mathcal{A}$ solves Ext-cGCDH with run-time $T' = T$ and success probability

$$\varepsilon' \geq \varepsilon - O(\kappa \cdot (\varepsilon_\rho + \|r_i'\|/\sigma^* + \varepsilon_e)), \tag{9.9}$$

so that the re-randomization security goal of Definition 9.7 is satisfied if

$$\|r_i'\|/\sigma^*, \varepsilon_\rho, \varepsilon_e \;=\; O(\kappa^{-1} \cdot 2^{-\lambda}), \tag{9.10}$$

and $m_r = \Omega(n \log n)$ and $\sigma' = \|g\| \cdot \Omega\left(\sqrt{\log(n\varepsilon_e^{-1})}\right)$.

Our main contribution in the next Chapter is to improve the above analysis, and show how to satisfy the security goal with much better parameters, namely $\|r_i'\|/\sigma^*, \varepsilon_\rho, \varepsilon_e = O(\kappa^{-1})$.

Chapter 10

GGHLite: More Efficient Multilinear Maps from Ideal Lattices

This chapter is a joint work with Damien Stehlé and Ron Steinfeld, published in [LSS14], where we study and improve the GGH graded encoding scheme [GGH13a]. We first recall the high level description of the GGH scheme, fully described in Chapter 9. In the polynomial rings $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = R/qR$ (replacing the exponent space $\mathbb{Z}_p$), they generate a small secret $g \in R$ and let $\mathcal{I} = (g)$ be the principal ideal over R generated by g. They also sample a uniform $z \in R_q$ which stays secret. The "plaintext" is a coset of $R/\mathcal{I}$, and is encoded by $[c/z]_q$, where c is an arbitrary small coset representative. In practice, as g is hidden, they give another public parameter y, which is an encoding of 1, and the encoding of the coset is computed as $[e \cdot y]_q$, where e is a small coset representative (possibly different from c). Their graded encoding scheme uses the notion of *encoding level*: the plaintext e is a level-0 encoding, the encoding $[c/z]_q$ is a level-1 encoding, and at level i, an encoding of $e + \mathcal{I}$ is given by $[c/z^i]_q = [e \cdot y^i]_q$. These encodings are both additively and multiplicatively homomorphic, up to a limited number of operations. More precisely, a product of i level-1 encodings is a level-i encoding. One can multiply any number of encodings up to κ, instead of exactly κ in multilinear maps (the parameter κ is called the multilinearity parameter).

The authors of [GGH13a] introduced new hardness assumptions: the Graded Decisional Diffie-Hellman (GDDH) and its computational variant (GCDH). These are natural analogues of the Diffie-Hellman problems from group-based cryptography. To ensure their hardness, and hence the security of the cryptographic constructions, the second main difference with multilinear maps is the randomization of the encodings described in Section 9.2. The principle is as follows: first some level-1 encodings of 0, called $\{x_j\}_{j \leq m_r}$, are given in the public parameters; then, to randomize a level-1 encoding u', one outputs $u = [u' + \sum_j \rho_j x_j]_q$ where the ρ_j's are sampled from a discrete Gaussian distribution over $\mathbb{Z}$ with deviation parameter σ^*. Without this re-randomization, the encoding u' of e allows e to be efficiently recovered using $u = [u'y^{-1}]_q$. Adding the re-randomization step prevents this division attack, but the statistical properties of the distribution of the re-randomized encoding u remain correlated to some extent with the original encoding u'. In Chapter 9, we formalized the re-randomization security goal in the GGH construction, that is implicit in the work of [GGH13a]. A primary security goal of re-randomization is to guarantee security of the GDDH problem against statistical correlation attacks. Accordingly, we formulate a security goal that captures this security guarantee, by introducing a canonical variant of GDDH, called cGDDH. In this variant, the encodings of some elements are sampled from a canonical distribution whose statistical properties are independent of the encoded elements. Consequently,

the canonical problems are by construction not subject to "statistical correlation" attacks. Our re-randomization security goal is formulated as the existence of an efficient computational reduction from the canonical problems to their corresponding non-canonical variants.

As we saw in Chapter 9, in [GGH13a] the authors use a "drowning step" to solve this problem. This technique, also called "smudging," was previously used in other applications [AJLA+12, Gen09, ASP12, BPR12]. Generally, "drowning" consists in hiding a secret vector $\mathbf{s} \in \mathbb{Z}^n$ by adding a sufficiently large random noise $\mathbf{e} \in \mathbb{Z}^n$ to it, so that the distribution of $\mathbf{s} + \mathbf{e}$ becomes "almost independent" of $\mathbf{s}$. In all of the above applications, to achieve a security level 2^λ (where λ denotes the security parameter), the security analysis requires "almost independent" to be interpreted as "within statistical distance $2^{-\lambda}$ from a distribution that is independent of $\mathbf{s}$." In turn, this requirement implies the need for "exponential drowning," i.e., the ratio $\gamma = \|\mathbf{e}\|/\|\mathbf{s}\|$ between the magnitude of the noise and the magnitude of secret needs to be $2^{\Omega(\lambda)}$. Exponential drowning imposes a severe penalty on the efficiency of these schemes, as their security is related to γ-approximation lattice problems, whose complexity decreases exponentially with $\log \gamma$. As a result, the schemes require a lattice dimension n at least quadratic in λ and key length at least cubic in λ. In summary, the GGH re-randomization step, necessary for its security, is also a primary factor in its inefficiency.

Our contributions. Our first main improvement to the GGH scheme relies on a new security analysis of the drowning step in the GGH re-randomization algorithm. We show that our re-randomization security goal can be satisfied *without* "exponential drowning," thus removing the main efficiency bottleneck. Namely, our analysis provides a re-randomization at security level 2^λ while allowing the use of a re-randomization deviation parameter σ^* that only drowns the norm of the randomness offset $r' \in \mathcal{I}$ (from the original encoding to be re-randomized) by a *polynomial* (or even constant) drowning ratio $\gamma = \lambda^{O(1)}$ (rather than $\gamma = 2^{\Omega(\lambda)}$, as needed in the analysis of [GGH13a]). However, our analysis only works for the search variant of the Graded Diffie-Hellman problem. Fortunately, we show that the application of the GGH scheme – the N-party Key Agreement [GGH13a] – can be modified to rely on this computational assumption (in the random oracle model).

Our second main improvement of the re-randomization process is to decrease m_r, the number of encodings of 0 needed, from $\Omega(n \log n)$ to 2. We achieve this result by presenting a new discrete Gaussian Leftover Hash Lemma (LHL) over algebraic rings. In [GGH13a], the authors apply the discrete Gaussian LHL from [AGHS13] to show that the distribution of the sum $\sum_{j \le m_r} \rho_j b_j$ is close to a discrete Gaussian on the ideal $\mathcal{I}$ (where $x_j = [b_j/z]_q$). Our improvement consists in sampling the randomizers ρ_j as elements of the full n-dimensional ring R, rather than just from $\mathbb{Z}$. Since each randomizer now has n times more entropy than before, one may hope to obtain a similar LHL result as in [AGHS13] while reducing m_r by a factor $\approx n$. However, as the designers of the GGH scheme notice in [GGH13a, Section 6.4], the proof techniques from [AGHS13] do not seem to immediately carry over to our "algebraic ring" LHL setting. Our new LHL over rings resolves this problem.

The two contributions above allow us to decrease the bit size of the public parameters from $O(\kappa^3 \lambda^5 \log(\kappa\lambda))$ for the GGH scheme to $O(\kappa^3 \lambda \log^2(\kappa\lambda))$ for GGHLite, for multilinearity factor κ and security level 2^λ for the graded Diffie-Hellman problem.

Technical overview. Our first main result is to reduce the size of the parameter σ^* in the re-randomization process. Technically, our improved analysis of drowning is obtained by using the *Rényi divergence* (RD), see definition and properties in Section 1.1, to replace the conventional statistical distance (SD) as a measure of distribution closeness. The Rényi divergence was already exploited in a different context in [LPR13, Claim 5.11], to show the hardness of Ring-LWE.

Here, we use the Rényi divergence to decrease the amount of drowning, by bounding the Rényi divergence between a discrete Gaussian distribution and its offset. This suffices for relating the hardness of the search problems using these encoding distributions, even though the statistical distance between the distributions is non-negligible. The technique does not seem to easily extend to the decision problems, as Rényi divergence induces a multiplicative relationship between success probabilities, rather than an additive relationship as statistical distance does.

Our second main result is a new LHL over the ring R. We now briefly explain this result and its proof. For a fixed $X = [x_1, x_2] \in R^2$, with each x_i sampled from $D_{R,s}$, our goal is to study the distribution $\widetilde{\mathcal{E}}_{X,s} = x_1 \cdot D_{R,s} + x_2 \cdot D_{R,s}$. In particular, we prove that $\widetilde{\mathcal{E}}_{X,s}$ is statistically close to $D_{\mathbb{Z}^n, sX^T}$. For this, we adapt the proof of the LHL in [AGHS13], recalled in Section 10.2.1: we follow a similar series of steps, but the proofs of these steps differ technically, as we exploit the ring structure.

We first show that $X \cdot R^2 = R$, except with some constant probability < 1. For this, we adapt a result from [SS13] on the probability that two Gaussian samples of R are coprime. Note that in contrast to the LHL over $\mathbb{Z}$ in [AGHS13], in our setting the probability that $X \cdot R^2 \neq R$ is non-negligible. This is unavoidable with the ring $R = \mathbb{Z}[x]/(x^n + 1)$, since each random element of R falls in the ideal $(x + 1)$ with probability $\approx 1/2$, both x_1 and x_2 (and hence the ideal they generate) get "stuck" in $(x + 1)$ with probability $\approx 1/4$. However, the probability of this bad event is bounded away from 1 by a constant and thus we only need a constant number of trials on average with random X's to obtain a good X by rejection.

Then, we define the orthogonal R-module $A_X = \{\mathbf{v} \in R^2 : X \cdot \mathbf{v} = 0\}$, and apply a directly adapted variant of [AGHS13, Lemma 10] to show that if the parameter s is larger than the smoothing parameter $\eta_\varepsilon(A_X)$ (with A_X viewed as an integral lattice), then the statistical distance between $\widetilde{\mathcal{E}}_{X,s}$ and the ellipsoidal Gaussian $D_{\mathbb{Z}^n, sX^T}$ is bounded by 2ε. We finally show that this condition on the smoothing parameter of A_X holds. For this, we observe that the Minkowski minima of the lattice A_X are equal, due to the R-module structure of A_X. This allows us to bound the last minimum from above using Minkowski's second theorem. A similar approach was previously used (e.g., in [LM06]) to bound the smoothing parameter of ideal lattices.

Roadmap. The rest of this Chapter is organized as follows. In Section 10.1, we study the Rényi divergence as an alternative to the statistical distance in order to improve the security analysis of re-randomization "drowning" step. Section 10.2.2 contains our second main improvement to the re-randomization process: the algebraic ring variant of the discrete Gaussian leftover hash lemma from [AGHS13]. In Section 10.3, we show how to combine the results from the previous two sections to obtain our improved construction GGHLite. Section 10.4 compares the asymptotic parameters of GGHLite with those of the original GGH scheme. Finally, in Section 10.5, we show how to adapt some applications of multilinear maps to rely on the hardness of the Ext-GCDH problem, to which our security result for GGHLite applies.

10.1 Polynomial drowning via Rényi divergence

In this section, we present our first result towards our improvement of the GGH scheme re-randomization.

10.1.1 Preliminaries

To use our improved drowning lemma in Section 10.1, we need a lower bound on the least singular value $\sigma_n(\mathrm{rot}(b))$ of the matrix $\mathrm{rot}(b) \in \mathbb{Z}^{n \times n}$ corresponding to the map $x \mapsto b \cdot x$ over R, for a Gaussian distributed $b \hookleftarrow D_{I,\sigma}$. We also let $b[j] = b(\zeta^{2j+1})$ denote the jth complex embedding

of b, where $\zeta \in \mathbb{C}$ is a primitive $2n$th root of unity. We define $T_2(b) = (\sum_j |b[j]|^2)^{1/2}$. Recall that we have $T_2(b)^2 = n\|b\|^2$ (see, e.g., [SS13]). In the proof of [SS13, Lemma 4.1], a probabilistic lower bound on $\min_{j\in[n]} |b[j]|$ is obtained for a Gaussian distributed b. Since

$$\begin{aligned}\sigma_n(b)^2 &= \min_{u\in K,\|u\|=1} \|u\cdot b\|^2 = \frac{1}{n}\min_{u\in K, T_2(u)^2=n} \sum_{j\in[n]} |u[j]|^2\cdot|b[j]|^2 \\ &= \min_{j\in[n]} |b[j]|^2 = \frac{1}{\max_{j\in[n]}|b[j]^{-1}|^2} \\ &\le \frac{1}{\frac{1}{n}\sum_{j\in[n]}|b[j]^{-1}|^2} = \frac{1}{\|b^{-1}\|^2},\end{aligned}$$

we can immediately adapt it to get the following.

Lemma 10.1 (Adapted from [SS13, Lemma 4.1])**.** *Let $R = \mathbb{Z}^n[x]/(x^n+1)$ for n a power of 2. For any ideal $I \subseteq R$, $\delta \in (0,1)$, $t \ge \sqrt{2\pi}$ and $\sigma \ge \frac{t}{\sqrt{2\pi}}\cdot\eta_\delta(I)$, we have:*

$$\Pr_{b\hookleftarrow D_{I,\sigma}}\left[\|b^{-1}\| \ge \frac{t}{\sigma\sqrt{n/2}}\right] \le \Pr_{b\hookleftarrow D_{I,\sigma}}\left[\sigma_n(b) \le \frac{\sigma\sqrt{n/2}}{t}\right] \le \frac{1+\delta}{1-\delta}\frac{n\sqrt{2\pi e}}{t}.$$

We can also obtain a lower bound $\sigma_n(b)^2 \ge \frac{1}{n}\cdot\|b^{-1}\|^{-2}$ by replacing the last line in the equations above Lemma 10.1 by $\ge \frac{1}{\sum_{j\in[n]}|b[j]^{-1}|^2} = \frac{1}{n\cdot\|b^{-1}\|^2}$.

10.1.2 Intuition

We show that one may reduce the re-randomization "drowning" ratio $\sigma_k^*/\|r'\|$ from exponential to polynomial in the security parameter λ. Although the statistical distance between the re-randomized encoding distribution D_1 (essentially a discrete Gaussian with an added offset vector r') and the desired canonical encoding distribution D_2 (a discrete Gaussian without an added offset vector) is then non-negligible, we show that these encoding distributions are still sufficiently close with respect to an alternative closeness measure to the statistical distance, in the sense that switching between them preserves the success probability of any search problem adversary receiving these encodings as input, up to a polynomial transformation. This allows us to show that our re-randomization goal is satisfied for the search problems GCDH and Ext-GCDH.

Technically, the closeness measure we study is the *Rényi divergence*, see definition in Section 1.1, $R(D_1\|D_2)$ between the distributions D_1 and D_2, defined as the expected value of $D_1(r)/D_2(r)$ over the randomness of r sampled from D_1 (for brevity we will call $R(D_1\|D_2)$ the Rényi divergence between D_1 and D_2). Intuitively, the Rényi divergence is an alternative to statistical distance as measure of distribution closeness, where we replace the *difference* between the distributions in statistical distance, by the *ratio* of the distributions in Rényi divergence. Accordingly, one may hope Rényi divergence to have analogous properties to statistical distance, where addition in the property of statistical distance is replaced by multiplication in the analogous property of Rényi divergence. Remarkably, this holds true in some sense, and we explore some of this below. In particular, a very important property of the statistical distance is *probability preservation*: for any two distributions D_1, D_2 on space X, and any event $E \subseteq X$, we have $D_2(E) \ge D_1(E) - \Delta(D_1, D_2)$. Lyubashevsky et al. [LPR13] observed an analogous property of the Rényi divergence that follows roughly the above intuition: $D_2(E) \ge D_1(E)^2/R(D_1\|D_2)$. The latter property implies that as long as $R(D_1\|D_2)$ is bounded as $\mathrm{poly}(\lambda)$, any event E of non-negligible probability $D_1(E)$ under D_1 will also have non-negligible probability $D_2(E)$ under D_2. We show that for our discrete Gaussian distributions D_1, D_2 above, we have $R(D_1\|D_2) = O(\mathrm{poly}(\lambda))$, if $\sigma_k^*/\|r'\| = \Omega(\mathrm{poly}(\lambda))$, as required for our re-randomization security goal.

10.1.3 The Rényi divergence between a discrete Gaussian and its offset

For our re-randomization application, we are interested in the Rényi divergence between two discrete Gaussians with the same deviation matrix S, that differ by some fixed offset vector d. The following result shows that their Rényi divergence is $O(1)$ if $\sigma_n(S)/\|d\| = \Omega(1)$.

Lemma 10.2. *For any n-dimensional lattice $\Lambda \subseteq \mathbb{R}^n$ and rank n matrix $S \in \mathbb{R}^{m\times n}$ (with $m \geq n$), let P be the distribution $D_{\Lambda,S,w}$ and Q be the distribution $D_{\Lambda,S,z}$ for some fixed $w, z \in \mathbb{R}^n$. If $w, z \in \Lambda$, let $\varepsilon = 0$. Otherwise, fix $\varepsilon \in (0,1)$ and assume that $\sigma_n(S) \geq \eta_\varepsilon(\Lambda)$. Then:*

$$\begin{aligned} R(P\|Q) &\in \left[\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^2, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2\right] \cdot \exp(2\pi\|S^{-T}(w-z)\|^2) \\ &\subseteq \left[\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^2, \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^2\right] \cdot \exp\left(\frac{2\pi\|w-z\|^2}{\sigma_n(S)^2}\right). \end{aligned}$$

Proof. By definition,

$$P(x) = \frac{\exp(-\pi\|(S^T)^\dagger(x-w)\|^2)}{\sum_{x\in\Lambda}\exp(-\pi\|(S^T)^\dagger(x-w)\|^2)} \quad \text{and} \quad Q(x) = \frac{\exp(-\pi\|(S^T)^\dagger(x-z)\|^2)}{\sum_{x\in\Lambda}\exp(-\pi\|(S^T)^\dagger(x-z)\|^2)}.$$

We have:

$$\begin{aligned} R(P\|Q) &= \sum_{x\in\Lambda}\frac{P(x)^2}{Q(x)} \\ &= \frac{\sum_{y\in\Lambda}\exp(-\pi\|(S^T)^\dagger(y-z)\|^2)}{(\sum_{y\in\Lambda}\exp(-\pi\|(S^T)^\dagger(y-w)\|^2))^2} \cdot \sum_{x\in\Lambda}\exp(-2\pi\|(S^T)^\dagger(x-w)\|^2 + \pi\|(S^T)^\dagger(x-z)\|^2). \end{aligned}$$

Defining $c = 2w - z$, we have that:

$$2\|(S^T)^\dagger(x-w)\|^2 - \|(S^T)^\dagger(x-z)\|^2 = \|(S^T)^\dagger(x-c)\|^2 - 2\|(S^T)^\dagger(w-z)\|^2.$$

Hence,

$$R(P\|Q) = \exp(2\pi\|(S^T)^\dagger(w-z)\|^2) \cdot \frac{\sum_{x\in\Lambda}\exp(-\pi\|(S^T)^\dagger(x-c)\|^2) \cdot \sum_{y\in\Lambda}\exp(-\pi\|(S^T)^\dagger(y-z)\|^2)}{(\sum_{y\in\Lambda}\exp(-\pi\|(S^T)^\dagger(y-w)\|^2))^2}.$$

Notice that for any $z \in \Lambda$, we have $\sum_{x\in\Lambda}\exp(-\pi\|(S^T)^\dagger(x-z)\|^2) = \sum_{x\in\Lambda}\exp(-\pi\|(S^T)^\dagger x\|^2)$. From this, we conclude that if $w, z \in \Lambda$, then $c \in \Lambda$ and hence the sums in the quotient above cancel out, and we get $R(P\|Q) = \exp(2\pi\|(S^T)^\dagger(w-z)\|^2)$. In general, for any $y, z \in \mathbb{R}^n$, we have

$$\sum_{y\in\Lambda}\exp(-\pi\sigma_1((S^T)^\dagger)^2 \cdot \|y-z\|^2) \leq \sum_{y\in\Lambda}\exp(-\pi\|(S^T)^\dagger \cdot (y-z)\|^2) \leq \sum_{y\in\Lambda}\exp(-\pi\sigma_n((S^T)^\dagger)^2 \cdot \|y-z\|^2),$$

using the fact that $\sigma_n((S^T)^\dagger) \cdot \|y-z\| \leq \|(S^T)^\dagger \cdot (y-z)\| \leq \sigma_1((S^T)^\dagger) \cdot \|y-z\|$. But

$$\begin{aligned} \sum_{y\in\Lambda}\exp(-\pi\sigma_1((S^T)^\dagger)^2 \cdot \|y-z\|^2) &= \rho_{1/\sigma_1((S^T)^\dagger),z}(\Lambda) = \rho_{\sigma_n(S),z}(\Lambda) \\ \sum_{y\in\Lambda}\exp(-\pi\sigma_n((S^T)^\dagger)^2 \cdot \|y-z\|^2) &= \rho_{1/\sigma_n((S^T)^\dagger),z}(\Lambda) = \rho_{\sigma_1(S),z}(\Lambda). \end{aligned}$$

Using the assumption $\sigma_1(S) \geq \sigma_n(S) \geq \eta_\varepsilon(\Lambda)$ and Lemma 1.32, it follows that $\rho_{\sigma_1(S),z}(\Lambda)$ and $\rho_{\sigma_n(S),z}(\Lambda)$ are both in the interval $[1-\varepsilon, 1+\varepsilon] \cdot (\det \Lambda)^{-1}$. From the above inequality, we get that $\sum_{y\in\Lambda} \exp(-\pi\|(S^T)^\dagger \cdot (y-z)\|^2)$ is also in this interval. Applying this to the sums in the expression for $R(P\|Q)$ gives the claimed interval for $R(P\|Q)$.

The claimed inequality follows from $\|(S^T)^\dagger z\|^2 \leq \sigma_1((S^T)^\dagger)^2 \cdot \|z\|^2$ and $\sigma_1((S^T)^\dagger) = 1/\sigma_n(S)$. □

10.2 A discrete Gaussian leftover hash lemma over R

In this section, we present our second main result for improving the GGH scheme re-randomization algorithm. Recall that the GGH algorithm re-randomizes a level-k encoding u' into $u = [u' + \sum_{j=1}^{m_r} \rho_j x_j^{(k)}]_q$, where the ρ_j's are sampled from $\chi_k = D_{\mathbb{Z},\sigma_k^*}$ and $x_j^{(k)} = [b_j^{(k)}/z^k]_q = [g r_j^{(k)}/z^k]_q$. To show that the distribution of $\sum_{j=1}^{m_r} \rho_j b_j^{(k)}$ is close to a discrete Gaussian over $\mathcal{I}$, they then apply the discrete Gaussian LHL from [AGHS13, Theorem 3], using $m_r = \Omega(n \log n)$ fixed elements $b_j^{(k)} \in \mathcal{I}$ that are published obliviously as randomizers "inside" the public zero-encodings $x_j^{(k)}$. We show that it suffices to sample 2 randomizers as elements of the full n-dimensional ring R, rather than just from $\mathbb{Z}$, i.e., we set $\chi_k = D_{R,\sigma_k^*}$. We first review the results of [AGHS13], as our proof follows the same high-level steps.

10.2.1 Discrete gaussian leftover hash lemma

We now review the main result of [AGHS13]. For $X \in \mathbb{Z}^{n\times m}$ and $s > 0$, the authors define the distribution $\mathcal{E}_{X,s} = X \cdot D_{\mathbb{Z}^m,s}$ as the distribution induced by sampling an integer vector $\mathbf{v}$ from a discrete spherical Gaussian with parameter s and outputting $\mathbf{y} = X \cdot \mathbf{v}$,

$$\mathcal{E}_{X,s} = \{X \cdot \mathbf{v} : \mathbf{v} \hookleftarrow D_{\mathbb{Z}^m,s}\}.$$

They show that with overwhelming probability over the choice of X, the distribution $\mathcal{E}_{X,s}$ is statistically close to a discrete Gaussian distribution. This result is used to study the distribution of the randomization of an encoding in the GGH scheme.

Theorem 10.3 ([AGHS13, Theorem 2]). *For ε negligible in n, let $S \in \mathbb{R}^{n\times n}$ be a matrix such that $s_n = \sigma_n(S) \geq 18K\eta_\varepsilon(\mathbb{Z}^n)$ (for some universal constant $K > 0$), and set $s_1 = \sigma_1(S)$ and $w = s_1/s_n$. Also let m, s be parameters such that $m \geq 10n\log(8(mn)^{1.5}s_1 w)$ and $s' \geq 4mnw\ln(1/\varepsilon)$.*

Then, when choosing the columns of an n-by-m matrix X from the ellipsoid Gaussian over $\mathbb{Z}^n$, $X \hookleftarrow (D_{\mathbb{Z}^n,S})^m$, we have with all but probability $2^{-O(m)}$ over the choice of X, that the statistical distance between $\mathcal{E}_{X,s}$ and the ellipsoid Gaussian $D_{\mathbb{Z}^n,sX^T}$ is bounded by 2ε.

Note that this result has been recently improved in [AR13], but this improvement is independent from the one we obtain in the next Chapter. In [AR13], the authors keep the same distribution $\mathcal{E}_{X,s}$, but obtain weaker conditions under which the result holds. We recall the proof line of [AGHS13], as we will modify it in Chapter 10. In [AGHS13], the proof of this theorem proceeds by the following three lemmata.

Lemma 10.4 ([AGHS13, Lemma 9]). *With parameters as above, when drawing the columns of an n-by-m matrix X independently at random from $D_{\mathbb{Z}^n,S}$, we get $X \cdot \mathbb{Z}^n = \mathbb{Z}^n$ with all but probability $2^{-O(m)}$.*

Let $A = A(X) = \{\mathbf{v} \in \mathbb{Z}^m : X \cdot \mathbf{v} = 0\}$ be the $(m-n)$-dimensional lattice in $\mathbb{Z}^n$ orthogonal to all the rows of X. If the smoothing parameter of A is small, then $\mathcal{E}_{X,s}$ and $D_{\mathbb{Z}^n,sX^T}$ must be close.

Lemma 10.5 ([AGHS13, Lemma 10]). *Fix X and A as above. If $s \geq \eta_\varepsilon(A)$, then for any point $\mathbf{z} \in \mathbb{Z}^n$, the probability mass assigned to $\mathbf{z}$ by $\mathcal{E}_{X,s}$ differs from that assigned by $D_{\mathbb{Z}^n, sX^T}$ by at most a factor of $(1-\varepsilon)/(1+\varepsilon)$, namely*

$$\mathcal{E}_{X,s}(\mathbf{z}) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right] \cdot D_{\mathbb{Z}^n, sX^T}(\mathbf{z})$$

In particular, if $\varepsilon < 1/3$ then the statistical distance between $\mathcal{E}_{X,s}$ and $D_{\mathbb{Z}^n, sX^T}$ is at most 2ε.

Finally, the authors of [AGHS13] show that the smoothing parameter of A is indeed small.

Lemma 10.6 ([AGHS13, Corollary 3]). *With the parameters above, the smoothing parameter of A satisfies $\eta_\varepsilon(A) \leq 4mnw \ln(1/\varepsilon)$, except with probability $2^{-O(m)}$.*

The following also holds for general lattices.

Theorem 10.7 ([AGHS13, Theorem 3]). *Let $\Lambda \subseteq \mathbb{R}^n$ be a full-rank lattice and B a matrix whose columns form a basis of Λ. Also let $M \in \mathbb{R}^{n \times n}$ be a full-rank matrix, and denote $S = M(B^T)^{-1}$, $s_1 = \sigma_1(S)$, $s_n = \sigma_n(S)$, and $w = s_1/s_n$. Finally, let ε be negligible in n and m, s be parameters such that $m \geq 10n \log(8(mn)^{1.5} s_1 w)$ and $s \geq 4mnw \ln(1/\varepsilon)$. If $s \geq \eta_\varepsilon(\mathbb{Z}^n)$, then when choosing the columns of an n-by-m matrix X from the ellipsoid Gaussian over Λ, $X \hookleftarrow (D_{\Lambda,M})^m$, we have with all but probability $2^{-O(m)}$ over the choice of X, that the statistical distance between $\mathcal{E}_{X,s}$ and the ellipsoid Gaussian D_{Λ, sX^T} is bounded by 2ε.*

10.2.2 Our new leftover hash lemma

For a fixed $X = (x_1, x_2) \in R^2$, we define the distribution $\widetilde{\mathcal{E}}_{X,s} = x_1 D_{R,s} + x_2 D_{R,s}$ as the distribution induced by sampling $\mathbf{u} = (u_1, u_2) \in R^2$ from a discrete spherical Gaussian with parameter s, and outputting $y = x_1 u_1 + x_2 u_2$. We prove the following result on $\widetilde{\mathcal{E}}_{X,s}$.

Theorem 10.8. *Let $R = \mathbb{Z}[x]/(x^n + 1)$ with n a power of 2 and $\mathcal{I} = (g) \subseteq R$, for some $g \in R$. Fix $\varepsilon \in (0, 1/2)$, $X = (x_1, x_2) \in \mathcal{I} \times \mathcal{I}$ and $s > 0$ satisfying the conditions*

- ***Column span:*** *$X \cdot R^2 = \mathcal{I}$.*
- ***Smoothing:*** *$s \geq \max(\|g^{-1}x_1\|_\infty, \|g^{-1}x_2\|_\infty) \cdot n \cdot \sqrt{2\log(2n(1+1/\varepsilon))/\pi}$.*

Then, for all $x \in \mathcal{I}$ we have $\widetilde{\mathcal{E}}_{X,s}(x) = cf(x) \cdot D_{\mathcal{I}, sX^T}(x)$, for some constant c and function f with values in $[\frac{1-\varepsilon}{1+\varepsilon}, 1]$. In particular, we have

$$\Delta(\widetilde{\mathcal{E}}_{X,s}, D_{\mathcal{I}, sX^T}) \leq 2\varepsilon \quad \text{and} \quad \max(R_\infty(\widetilde{\mathcal{E}}_{X,s} \| D_{\mathcal{I}, sX^T}), R_\infty(D_{\mathcal{I}, sX^T} \| \widetilde{\mathcal{E}}_{X,s})) \leq 1 + 4\varepsilon.$$

Finally, if $s' \cdot \sigma_n(g^{-1}) \geq 7n^{1.5} \ln^{1.5}(n)$,[1] $x_1, x_2 \hookleftarrow D_{\mathcal{I}, s'}$ and n grows to infinity, then the first condition holds with probability $\Omega(1)$.

We prove this result for $g = 1$, and then we generalize to general g. First, we consider the column span condition.

Lemma 10.9 (Adapted from [SS13, Lemma 4.2 and Lemma 4.4]). *Let $S \in \mathbb{R}^{n \times n}$, and $\sigma_n(S) \geq 7n^{1.5} \ln^{1.5}(n)$. For n going to infinity, we have $\Pr_{x_1, x_2 \hookleftarrow D_{R,S}}[X \cdot R^2 = R] \geq \Omega(1)$.*

[1] By abuse of notation, we identify $g^{-1} \in K$ with the linear map over $\mathbb{Q}^n$ obtained by applying the polynomial-to-coefficient-vector mapping to the map $r \mapsto g^{-1}r$.

Let $A_X \subseteq \{(v_1, v_2) \in R^2 : x_1 v_1 + x_2 v_2 = 0\}$ be the 1-dimensional R-module of vectors orthogonal to X. We view A_X as an n-dimensional lattice in $\mathbb{Z}^{2n}$, via the polynomial-to-coefficient-vector mapping.

Lemma 10.10 (Adapted from [AGHS13, Lemma 10])**.** *Fix X such that $X \cdot R^2 = R$ and A_X as above. If $s \geq \eta_\varepsilon(A_X)$, then $\widetilde{\mathcal{E}}_{X,s}(z) = cf(z) \cdot D_{\mathbb{Z}^n, sX^T}(z)$ for any $z \in R$, for some constant c and function f with values in $[\frac{1-\varepsilon}{1+\varepsilon}, 1]$.[2] In particular, we have*

$$\Delta(\widetilde{\mathcal{E}}_{X,s}, D_{\mathbb{Z}^n, sX^T}) \leq \frac{\varepsilon}{1-\varepsilon} \quad \text{and} \quad \max(R_\infty(\widetilde{\mathcal{E}}_{X,s} \| D_{\mathbb{Z}^n, sX^T}), R_\infty(D_{\mathbb{Z}^n, sX^T} \| \widetilde{\mathcal{E}}_{X,s})) \leq \frac{1+\varepsilon}{1-\varepsilon}.$$

We now study the quantity $\eta_\varepsilon(A_X)$. First, we show that all successive Minkowski minima of A_X are equal. This property is inherited from the "equal minima property" of ideal lattices in R.

Lemma 10.11. *Let X and A_X be as above. Then $\lambda_1(A_X) = \cdots = \lambda_n(A_X)$.*

Proof. We observe that A_X is closed under scalar multiplication by an arbitrary element $w \in R$, i.e., if $\mathbf{v} = (v_1, v_2) \in A_X$ then $w \cdot \mathbf{v} = (w \cdot v_1, w \cdot v_2) \in A_X$. In particular, let $\mathbf{v} \in A_X$ be a vector of norm $\|\mathbf{v}\| = \lambda_1(A_X)$. For $i = 0, \ldots, n-1$, let $e_i(x) = x^i \in R$. Then the n vectors $(e_0 \cdot \mathbf{v}, \ldots, e_{n-1} \cdot \mathbf{v})$ are in A_X, and all have the same norm $\lambda_1(A_X)$, because $\|e_j \cdot v_i\| = \|v_i\|$ for all i, j. Further, these n vectors are linearly independent over $\mathbb{Q}$: let i be such that $v_i \neq 0$ (which must exist since $\mathbf{v} \neq \mathbf{0}$); the vectors $(e_0 \cdot v_i, \ldots, e_{n-1} \cdot v_i)$ are linearly independent over $\mathbb{Q}$, because the fraction field K of R is a field (it they were not linearly independent over $\mathbb{Q}$, we would have $(\sum_j \alpha_j e_j) \cdot v_i = 0$ for some non-zero $\alpha = \sum_j \alpha_j e_j \in K$). It follows that $\lambda_1(A_X) = \cdots = \lambda_n(A_X) = \|\mathbf{v}\|$. □

Lemma 10.12. *Let X and A_X be as above. Then we have $\eta_\varepsilon(A_X) \leq \max(\|x_1\|_\infty, \|x_2\|_\infty) \cdot n \cdot \sqrt{2\log(2n(1+1/\varepsilon))/\pi}$.*

Proof. We first use Lemma 10.11 and Minkowski's second theorem (see Lemma 1.9) on the lattice A_X:

$$\lambda_n(A_X) \;=\; \big(\textstyle\prod_{1 \leq i \leq n} \lambda_i(A_X)\big)^{1/n} \;\leq\; \sqrt{n} \cdot \big(\det(A_X)\big)^{1/n}.$$

Now, observe that $A_X = L_X^\perp$, where $L_X = R \cdot X = \{(r \cdot x_1, r \cdot x_2) : r \in R\}$ is viewed as a sublattice of $\mathbb{Z}^{2n}$. We have, by Lemma 1.10, that $\det(A_X) \leq \det(L_X) \leq \|X\|^n$, where the latter inequality follows from the Hadamard inequality, with $\|X\| = \sqrt{\|x_1\|^2 + \|x_2\|^2} \leq \max(\|x_1\|_\infty, \|x_2\|_\infty) \cdot \sqrt{2n}$. As a consequence $\lambda_n(A_X) \leq \max(\|x_1\|_\infty, \|x_2\|_\infty) \cdot \sqrt{2}n$. By Lemma 1.27, we have $\eta_\varepsilon(A_X) \leq \sqrt{\ln(2n(1+1/\varepsilon))/\pi} \cdot \lambda_n(A_X)$, which completes the proof. □

Combining the above lemmas, we get Theorem 10.8 for $g = 1$. The general case is proved as follows. The injective map $M_g : y \mapsto g \cdot y$ on R takes the distribution $\widetilde{\mathcal{E}}_{\overline{X},s}$ with $\overline{X} = g^{-1} \cdot X$ to the distribution $\widetilde{\mathcal{E}}_{X,s}$, while it takes $D_{R, s\overline{X}^T}$ to $D_{\mathcal{I}, sX^T}$, with $\mathcal{I} = (g)$. The conditions $X \cdot R^2 = \mathcal{I}$ and $\overline{X} \cdot R^2 = R$ are equivalent. The smoothing condition is satisfied for $\overline{X}$ by the choice of s. Thus we can apply Theorem 10.8 with $g = 1$ to $\widetilde{\mathcal{E}}_{\overline{X},s}$, and conclude by applying the mapping M_g to get the general case of Theorem 10.8. For the very last statement of Theorem 10.8, it suffices to observe that $D_{\mathcal{I},\beta} = g \cdot D_{R, s'(g^{-1})^T}$.[3] □

[2] The normalization constant c was omitted in [AGHS13].

[3] With the same abuse of notation as in the previous footnote, for the term $(g^{-1})^T$.

10.3 Our improved GGH grading scheme: GGHLite

We are now ready to describe our simpler and more efficient variant of the GGH grading scheme, that we call GGHLite. The scheme is summarized in Figure 10.1. The modifications from the original GGH scheme consist in:

- Using $m_r = 2$ re-randomization elements x_1, x_2 in the public key, sampling the randomizers ρ_1, ρ_2 from a discrete Gaussian D_{R,σ_1^*} over the whole ring R (rather than from $\mathbb{Z}$), applying our algebraic ring variant of the LHL from Section 10.2.2.

- Saving an exponential factor $\approx 2^\lambda$ in the re-randomization parameter σ_1^* by applying the Rényi divergence bounds from Section 10.1.

In terms of re-randomization security requirement, we relax the strong SD-based requirement on the original GGH scheme to the following weaker RD-based requirement on GGHLite.

Definition 10.13 (Weak re-randomization security requirement)**.** Using the notations of Definition 9.4, we say that the *weak* re-randomization security requirement is satisfied at level k with respect to $D_{\text{can}}^{(k)}(e_L)$ and encoding norm $\gamma^{(k)}$ if $R(D_u^{(k)}(e_L, r')\|D_{\text{can}}^{(k)}(e_L)) = O(\text{poly}(\lambda))$ for any $u' = [c'/z^k]_q$ such that $\|c'\| \leq \gamma^{(k)}$.

We summarize GGHLite in Figure 10.1, which only shows the algorithms differing from those in the GGH scheme of Figure 9.3.

- **Instance generation** $\mathsf{InstGen}(1^\lambda, 1^\kappa)$: Given security parameter λ and multilinearity parameter κ, determine scheme parameters n, q, $m_r = 2$, σ, σ', $\ell_{g^{-1}}$, ℓ_b and ℓ based on the scheme analysis. Then proceed as follows:
  - Sample $g \hookleftarrow D_{R,\sigma}$ until $\|g^{-1}\| \leq \ell_{g^{-1}}$ and $\mathcal{I} = (g)$ is a prime ideal and $\|g\| \leq \sqrt{n} \cdot \sigma$.
  - Sample $z \hookleftarrow U(R_q)$.
  - Sample a level-1 encoding of 1: $y = [a \cdot z^{-1}]_q$ with $a \hookleftarrow D_{1+\mathcal{I},\sigma'}$.
  - For $k \leq \kappa$:
    - Sample $B^{(k)} = (b_1^{(k)}, b_2^{(k)})$ from $(D_{\mathcal{I},\sigma'})^2$. If $(b_1^{(k)}, b_2^{(k)}) \neq \mathcal{I}$, or $\sigma_n(\text{rot}(B^{(k)})) < \ell_b$ or $\|B^{(k)}\| > \sqrt{n} \cdot \sigma'$, then re-sample.
    - Define level-k encodings of 0: $x_1^{(k)} = [b_1^{(k)} \cdot z^{-k}]_q$, $x_2^{(k)} = [b_2^{(k)} \cdot z^{-k}]_q$.
  - Sample $h \hookleftarrow D_{R,\sqrt{q}}$ and define the zero-testing parameter $p_{zt} = [\frac{h}{g} z^\kappa]_q \in R_q$.
  - Return public parameters $\text{par} = (n, q, y, \{(x_1^{(k)}, x_2^{(k)})\}_{k \leq \kappa})$ and p_{zt}.
- **Level-k encoding** $\mathsf{enc}_k(\text{par}, e)$: Given level-0 encoding $e \in R$ and parameters par:
  - Encode e at level k: Compute $u' = [e \cdot y^k]_q$.
  - Return $u = [u' + \rho_1 \cdot x_1^{(k)} + \rho_2 \cdot x_2^{(k)}]_q$, with $\rho_1, \rho_2 \hookleftarrow D_{R,\sigma_k^*}$.

Figure 10.1: The new algorithms of our GGHLite scheme.

Choice of σ, $\ell_{g^{-1}}$ and σ', ℓ_b. The upper bound $\ell_{g^{-1}}$ on $\|g^{-1}\|$ in the rejection test of InstGen can be chosen as small as possible while keeping the rejection probability p_g bounded from 1. According to Lemma 10.1 and Lemma 1.27 with $t = 2\sqrt{2\pi e}np_g^{-1}$ and $\delta = 1/3$, one can choose

$$\ell_{g^{-1}} = 4\sqrt{\pi en}/(p_g\sigma) \quad \text{and} \quad \sigma \geq 4\pi n\sqrt{e\ln(8n)/\pi}/p_g, \tag{10.1}$$

to achieve $p_g < 1$. Note that the same choices apply to the GGH scheme: here we have a rigorous bound on p_g instead of the heuristic arguments for estimating in $\|g^{-1}\|$ in [GGH13a]; however, as in [GGH13a], we do not have a rigorous bound on the probability that $\mathcal{I}$ is prime conditioned on this choice.

Let p_b be the rejection probability for the lower bound ℓ_b on $\sigma_n(B^{(k)})$ in the rejection test of InstGen. To keep p_b away from 1, we use that $\sigma_n(B^{(k)})^2 = \min_{u\in K,\|u\|=1}\sum_{i=1,2}\|u\cdot b_i^{(k)}\|^2 \geq \sum_{i=1,2}\sigma_n(b_i^{(k)})^2$. Applying Lemma 10.1 with $t = 2\sqrt{2\pi e}np_b^{-1}$ and $\delta = 1/3$, we get that $\sigma_n(b_i^{(k)}) > \frac{p_b}{8\sqrt{\pi en}}\cdot\sigma'$, except with probability $\leq p_b$ for $i \in \{1,2\}$ if $\sigma' \geq \frac{t}{\sqrt{2\pi}}\eta_{1/3}(\mathcal{I})$, where $\eta_{1/3}(\mathcal{I}) \leq \sqrt{\ln(8n)/\pi}\cdot\|g\|$ by Lemma 1.27. Therefore, we can choose

$$\ell_b = \frac{p_b}{2\sqrt{\pi en}}\cdot\sigma' \quad \text{and} \quad \sigma' \geq 2n^{1.5}\sigma\sqrt{e\ln(8n)/\pi}/p_b. \tag{10.2}$$

We also need to bound the probability p'_b of the first rejection test $(b_1^{(k)}, b_2^{(k)}) \neq \mathcal{I}$. This is bounded by some constant < 1 by Theorem 10.8, but it requires the assumption $\sigma'\cdot\sigma_n(g^{-1}) \geq 7n^{1.5}\ln^{1.5}(n)$. To use Theorem 10.8 to obtain a rigorous bound on p'_b, we can satisfy the assumption as follows. Using the lower bound $\sigma_n(g^{-1}) \geq \frac{1}{\sqrt{n}\|g\|}$ from the remark after Lemma 10.1, and using the rejection condition $\|g\| \leq \sqrt{n}\cdot\sigma$, we have $\sigma_n(g^{-1}) \geq \frac{1}{n\sigma}$, so the Theorem 10.8 assumption is satisfied by setting

$$\sigma' \geq 7n^{2.5}\ln^{1.5}(n)\cdot\sigma. \tag{10.3}$$

Zero-testing and extraction correctness. The correctness conditions for zero-testing and correctness remain the same as conditions (9.2), (9.3) for the original GGH scheme. The only modification needed is for condition (9.1), because in GGHLite, $m_r = 2$ and $\rho_j \in R$ so $\|\rho_j b_j^{(1)}\| \leq \sqrt{n}\|\rho_j\|\|b_j^{(1)}\|$. Accordingly, condition (9.1) is replaced by:

$$q > \max\left((n\ell_{g^{-1}})^8, (3\cdot n^{1.5}\sigma^*\sigma')^{8\kappa}\right). \tag{10.4}$$

Security. We state our improved re-randomization security reduction for GGHLite, that works with much smaller parameters than GGH. To our knowledge, it is the first security proof in which the Rényi divergence is used to replace the statistical distance in a sequence of games, using the Rényi divergence properties from Section 10.1 to combine the bounds on changes between games. This allows us to gain the benefits of Rényi divergence over statistical distance, for both the drowning and smoothing aspects. Namely, with $\varepsilon_d, \varepsilon_\rho, \varepsilon_e$ in Theorem 10.14 set as large as $O(\log\lambda/\kappa)$, our weak security requirement of Definition 10.13 is satisfied (the Rényi divergence between real and canonical encoding distributions is bounded by the quantity $R = \text{poly}(\lambda)$ in Theorem 10.14), and our re-randomization goal for Ext-GCDH is achieved (whereas the strong requirement of Definition 9.4 is not satisfied).

Theorem 10.14 (Security of GGHLite)**.** *Let $\varepsilon_d, \varepsilon_\rho, \varepsilon_e \in (0, 1/2)$ and $\kappa \leq 2^n$. Suppose that the following conditions are satisfied for* GGHLite*:*

- ***LHL Smoothing:***

$$\sigma_1^* \geq n^{1.5}\cdot\ell_{g^{-1}}\cdot\sigma'\cdot\sqrt{2\log(4n\cdot\varepsilon_\rho^{-1})/\pi}. \tag{10.5}$$

- ***Offset "Drowning:"***

$$\sigma_1^* \geq n^{1.5} \cdot (\sigma')^2 \cdot \sqrt{8\pi\varepsilon_d^{-1}}/\ell_b. \tag{10.6}$$

- samp ***Uniformity Smoothing:***

$$\sigma' \geq \sigma \cdot \sqrt{n \ln(4n \cdot \varepsilon_e^{-1})/\pi}. \tag{10.7}$$

Then, if $\mathcal{A}$ is an adversary against the (non-canonical) Ext-GCDH problem for GGHLite *with runtime T and advantage ε, then $\mathcal{A}$ is also an adversary against the canonical problem Ext-cGCDH for* GGHLite *with $T' = T$ and advantage*

$$\varepsilon' \geq (\varepsilon - O(\kappa \cdot 2^{-n}))^2/R \quad \textit{with} \quad R = 2^{O(\kappa \cdot (\varepsilon_d + \varepsilon_\rho + \varepsilon_e + 2^{-n}))}. \tag{10.8}$$

In particular, there exist ε_d, ε_e, ε_ρ bounded as $O(\log \lambda/\kappa)$ such that the re-randomization security goal in Definition 9.7 is satisfied by GGHLite *with respect to problem Ext-GCDH.*

Proof. We consider a sequence of games $\mathsf{Game}_0, \ldots, \mathsf{Game}_5$, where the distributions of the view of $\mathcal{A}$ differ among the games as follows:

- Game_0: The Ext-GCDH experiment, where $y = [az^{-1}]_q$ with $a = 1 + gr_y \hookleftarrow D_{1+\mathcal{I},\sigma'}$ and $\mathcal{I} = \langle g \rangle$, $u_i = [(e_{i,L} + \sum_j \rho_{ij} b_j^{(1)} + c_i) \cdot z^{-1}]_q$ for $i \in \{0, \ldots, \kappa\}$, $e_{i,L} = [e_i]_g$, $e_i = e_{i,L} + g e_{i,H} \hookleftarrow D_{R,\sigma'}$, and $c_i = g(e_{i,L} r_y + e_{i,H}) + g^2 r_y e_{i,H}$.

- Game_1: Modification of Game_0 in which e_i (for $i \in \{0, \ldots, \kappa\}$) and a are sampled from the truncated tail Gaussians $D^t_{R,\sigma'}$ and $D^t_{1+\mathcal{I},\sigma'}$ (instead of the untruncated Gaussians $D_{R,\sigma'}$ and $D_{1+\mathcal{I},\sigma'}$ respectively).

- Game_2: Modification of Game_1 in which the distribution of the re-randomization term $\sum_j \rho_{ij} b_j^{(1)}$ is replaced by the canonical distribution $D_{\mathcal{I},\sigma_1^*(B^{(1)})^T}$, so $u_i = [(e_{i,L} + w_i + c_i) \cdot z^{-1}]_q$, with $w_i \hookleftarrow D_{\mathcal{I},\sigma_1^*(B^{(1)})^T}$ for $0 \leq i \leq \kappa$.

- Game_3: Modification of Game_2 in which offset vector c_i in the randomization of encoding u_i is removed and replaced by $-e_{i,L}$, so that $u_i = [(e_{i,L} + w_i) \cdot z^{-1}]_q$, where $w_i \hookleftarrow D_{\mathcal{I},\sigma_1^*(B^{(1)})^T, -e_{i,L}}$ for $0 \leq i \leq \kappa$ (note that $e_{i,L} + w_i$ is distributed as $D_{\mathcal{I}+e_{i,L},\sigma_1^*(B^{(1)})^T}$ over the randomness of w_i).

- Game_4: Modification of Game_3 in which e_i is sampled from $D_{R,\sigma'}$ (instead of sampling e_i from the truncated tail Gaussian $D^t_{R,\sigma'}$), for $0 \leq i \leq \kappa$, and a is sampled from $D_{1+\mathcal{I},\sigma'}$ (instead of $D^t_{1+\mathcal{I},\sigma'}$).

- Game_5: The Ext-cGCDH experiment, which can be obtained as a modification of Game_4 in which $e_{i,L}$ is sampled uniformly from R_g, instead of being computed from e_i as $e_{i,L} = [e_i]_g$.

For $i = 0, \ldots, 5$, let V_i denote the distribution of the view of A in Game_i, and let E denote the event that A outputs the correct Ext-GCDH solution. By the probability preservation property of Rényi divergence from Lemma 1.4, we have that the advantage of A against Ext-cGCDH is $V_5(E) \geq V_1(E)^2/R(V_1 \| V_5)$ and from the probability preservation property of the statistical distance, the latter is $\geq (\varepsilon - \Delta(V_0, V_1))^2/R(V_1 \| V_5)$.

To complete the proof, it thus remains to show that $\Delta(V_0, V_1) = O(\kappa \cdot 2^{-n})$ and $R(V_1 \| V_5) \leq R$, with R defined in the theorem statement. Using two applications of the weak triangle inequality and

one application of the R_∞ triangle inequality from Lemma 1.4, we get $R(V_1\|V_5) \le R_\infty(V_1\|V_2)^2 \cdot R(V_2\|V_5)$, $R(V_2\|V_5) \le R(V_2\|V_3) \cdot R_\infty(V_3\|V_5)$ and finally

$$R(V_1\|V_5) \le R_\infty(V_1\|V_2)^2 \cdot R(V_2\|V_3) \cdot R_\infty(V_3\|V_4) \cdot R_\infty(V_4\|V_5).$$

We now bound each factor in turn:

- To bound $\Delta(V_0, V_1)$, we use the fact that Game_0 and Game_1 differ only if the norm of one of the sampled e_i (for $i \in \{0, \ldots, \kappa\}$) or a exceeds $2\sqrt{n} \cdot \sigma'$. By Lemma 1.40, since $\sigma' \ge \eta_{1/2}(\mathcal{I})$ (which follows from the samp uniformity smoothing condition, as shown below), this event occurs with probability at most 2^{-n+2} for each of these $\kappa + 2$ Gaussian samples. By the union bound, it thus follows that

$$\Delta(V_0, V_1) \le (\kappa + 2) \cdot 2^{-n+2} = O(\kappa \cdot 2^{-n}).$$

- To bound $R_\infty(V_1\|V_2)^2$, we apply our LHL over R (Theorem 10.8) to conclude that, for each $i \in [\kappa + 1]$, $R_\infty(D(\sum_j \rho_{ij} b_j^{(1)})\|D_{\mathcal{I},\sigma_1^*(B^{(1)})^T}) \le 1 + 4\varepsilon_\rho \le \exp(4\varepsilon_\rho)$ if $\varepsilon_\rho \le 1/2$, $\sigma_1^* \ge \|g^{-1}B^{(1)}\|_\infty n\sqrt{2\log(4n \cdot \varepsilon_\rho^{-1})/\pi}$, and $B^{(1)} \cdot R^2 = \mathcal{I}$. The last condition on $B^{(1)}$ holds by the rejection test of the $\mathsf{InstGen}$ algorithm of $\mathsf{GGHLite}$. The condition on σ_1^* holds by the assumed LHL Smoothing condition and the bound $\|g^{-1} \cdot B^{(1)}\|_\infty \le \|g^{-1}\| \cdot \|B^{(1)}\| \le \ell_{g^{-1}} \cdot \sigma' \cdot \sqrt{n}$, from the rejection tests of the $\mathsf{InstGen}$ algorithm. Using the multiplicativity property over $i \in [\kappa + 1]$, and data processing inequality for R_∞, we conclude that

$$R_\infty(V_1\|V_2)^2 \le \exp(8 \cdot (\kappa + 1) \cdot \varepsilon_\rho).$$

- To bound $R(V_2\|V_3)$, let $D_{1,i} = D_{\mathcal{I},\sigma_1^*(B^{(1)})^T} + c_i = D_{\mathcal{I},\sigma_1^*(B^{(1)})^T,c_i}$ (using $\mathcal{I} + c_i = \mathcal{I}$, since $c_i \in \mathcal{I}$) and $D_{2,i} = D_{\mathcal{I},\sigma_1^*(B^{(1)})^T,-e_{i,L}}$ for $i \in [\kappa+1]$. From the offset drowning condition, we have $\sigma_1^* \cdot \ell_b \ge \sigma'$, and using the samp uniformity smoothing condition, we have $\sigma' \ge \eta_{\varepsilon_e}(\mathcal{I})$, where we have used the bound $\eta_{\varepsilon_e}(\mathcal{I}) \le \sqrt{\frac{\ln(2n(1+1/\varepsilon_e))}{\pi}} \cdot \lambda_n(\mathcal{I})$ from Lemma 1.27, and the fact that $\lambda_n(\mathcal{I}) = \lambda_1(\mathcal{I}) \le \|g\| \le \sqrt{n} \cdot \sigma$. We conclude that $\sigma_n(\sigma_1^*(B^{(1)})^T) \ge \sigma_1^* \cdot \ell_b \ge \eta_{\varepsilon_e}(\mathcal{I})$. Therefore, we can apply our offset Gaussian divergence bound (Lemma 10.2) for each i (with $w = c_i$ and $z = -e_{i,L}$) to get that, conditioned on a fixed value of offset c_i and encoded element $e_{i,L}$ (as well as fixed g, $B^{(1)}$ and a), we have $R(D_{1,i}\|D_{2,i}) \le (\frac{1+\varepsilon_e}{1-\varepsilon_e})^2 \cdot \exp(2\pi\|c_i + e_{i,L}\|^2/(\sigma_1^*\sigma_n(B^{(1)}))^2) \le \exp(2\pi\|c_i + e_{i,L}\|^2/(\sigma_1^*\ell_b)^2 + 8\varepsilon_e)$ using $(\frac{1+\varepsilon_e}{1-\varepsilon_e})^2 \le \exp(8\varepsilon_e)$ for $\varepsilon_e < 1/2$. We also have $\|c_i + e_{i,L}\| = \|e_i \cdot a\| \le \sqrt{n} \cdot \|e_i\| \cdot \|a\| \le n^{1.5} \cdot (\sigma')^2$, using the bounds $\|e_i\| \le \sqrt{n} \cdot \sigma'$, $\|a\| \le 2\sqrt{n}\sigma'$. Therefore, we get $R(D_{1,i}\|D_{2,i}) \le \exp(\varepsilon_d + 8\varepsilon_e)$ using the "Offset Drowning" condition. Using the multiplicativity property over $i \in [\kappa + 1]$, and data processing property of R, we conclude that

$$R(V_2\|V_3) \le \exp((\kappa + 1) \cdot (\varepsilon_d + 8\varepsilon_e)).$$

- To bound $R_\infty(V_3\|V_4)$, we recall that for each $i \in [\kappa + 1]$, the distribution $D^t_{R,\sigma'}$ of e_i in Game_3 is obtained by rejecting and resampling from $D_{R,\sigma}$ if the rejection test $\|e_i\| > \sqrt{n}\sigma'$ is satisfied. It follows that $D^t_{R,\sigma'}(x) = \frac{1}{1-p_{rej}} \cdot D_{R,\sigma'}(x)$ for all x in the support of $D^t_{R,\sigma'}$, where p_{rej} is the probability that a sample $D_{R,\sigma'}$ is rejected, and hence that $R_\infty(D^t_{R,\sigma'}\|D_{R,\sigma'}) = \frac{1}{1-p_{rej}}$. By the discrete Gaussian tail bound Lemma 1.40, we have $p_{rej} \le 2^{-n+2}$ if $\sigma' \ge \eta_{1/2}(R)$, and the latter condition is satisfied by the choice of σ'. It follows that $R_\infty(D^t_{R,\sigma'}\|D_{R,\sigma'}) \le 1 + 2^{-n+3}$. Applying a similar argument to the distribution of a using $\sigma' \ge \eta_{1/2}(\mathcal{I})$, we have

$R_\infty(D^t_{1+\mathcal{I},\sigma'} \| D_{1+\mathcal{I},\sigma'}) \leq 1 + 2^{-n+3}$ and hence by the multiplicativity and data processing properties of the Rényi divergence:

$$R_\infty(V_3 \| V_4) \leq (1 + 2^{-n+3})^{\kappa+2} \leq \exp((\kappa+2) \cdot 2^{-n+3}).$$

- To bound $R_\infty(V_4 \| V_5)$, let D_e denote the distribution of $[e_i]_g$ over the randomness of e_i sampled from $D_{R,\sigma'}$. We apply smoothing Lemma 1.33. to get that $R_\infty(U(R_g) \| D_e) \leq \frac{1+\varepsilon_e}{1-\varepsilon_e}$ if $\sigma' \geq \eta_{\varepsilon_e}(I)$. The latter condition holds as shown above. Using the multiplicativity and data processing properties of Rényi divergence from Lemma 1.4, over $i = 0, \ldots, \kappa$, we conclude that for $\varepsilon_e \leq 1/2$:
$$R_\infty(V_4 \| V_5) \leq \left(\frac{1+\varepsilon_e}{1-\varepsilon_e}\right)^{\kappa+1} \leq \exp((\kappa+1) \cdot 4\varepsilon_e).$$

Combining the above bounds gives the claimed bound. For the last statement, it suffices to observe that $\varepsilon' = \Omega(\varepsilon^2/\mathrm{poly}(\lambda))$ if $\kappa \cdot \max(\varepsilon_d, \varepsilon_\rho, \varepsilon_e) = O(\log \lambda)$.

□

10.3.1 Canonical re-randomization algorithm cenc

In Remark 2 of [GGH13a], the authors of the original GGH scheme define a canonicalizing encoding algorithm cenc that allows for certain applications (like the ABE scheme in [GGH+13c]) to use the encoding re-randomization multiple times. We can define such a canonical re-randomization algorithm for our GGHLite in a similar way.

Algorithm $\mathsf{cenc}_l(\mathrm{par}, k, u')$ takes a level-k encoding u' of some element $e \in R_g$ with $k \leq \kappa$ and returns a re-randomized level-k encoding u of e. The parameter l indicates the "re-randomization depth," i.e., the number of times that cenc has been applied, and determines the re-randomization noise level.

Alternative "pairwise closeness" re-randomization security requirement. For applications such as the ABE scheme in [GGH+13c], it is required that, for any two given level-k encodings $u'_1 = [c_1/z^k]_q, u'_2 = [c_2/z^k]_q$ of the same element e, the pair of distributions $D(u_1), D(u_2)$ of $u_1 = \mathsf{cenc}_l(\mathrm{par}, k, u'_1)$ and $u_2 = \mathsf{cenc}_l(\mathrm{par}, k, u'_2)$, respectively (over the randomness of cenc), are "close." This "pairwise closeness" requirement for re-randomized encodings is an alternative to the "closeness to a canonical distribution" requirement for re-randomized encodings in Definition 9.4 and Definition 10.13. In the case of the strong SD-based "closeness" requirement in Definition 9.4, we have, from the triangle inequality property of statistical distance, that the "closeness to a canonical distribution" requirement of Definition 9.4 implies the "pairwise closeness" requirement. However, due to the lack of such a general triangle inequality property for the Rényi divergence, such an implication does not immediately hold for our weak RD-based "closeness" requirements. Nevertheless, our improved re-randomization analysis of GGHLite above can be carried over to establish the weak "pairwise closeness" requirement as well.

In the following, we define our weak RD-based "pairwise closeness" re-randomization requirement.

Definition 10.15 (Weak pairwise-closeness re-randomization property of cenc)**.** Fix a κ-graded encoding scheme S, and an instance par of this scheme for security parameter λ. For $k \leq \kappa$ and $l \leq L$, let $S_{(k,l)}$ denote a set of "admissible" level-k input encodings at re-randomization depth l. Let cenc_l denote a re-randomization probabilistic algorithm that takes as input (par, k, u') with u' a level-k encoding of some level-0 element e_L, and returns a re-randomized level-k encoding u of e_L. Then we say that cenc satisfies the *weak pairwise closeness re-randomization* property for S with Rényi divergence bound R and admissible input encoding sets $\{S_{(k,l)}\}_{k\in[\kappa],l\in[L]}$ if, for any $k \in [\kappa]$, $l \in [L]$ and two level-k encodings $u'_1, u'_2 \in S_{(k,l)}$ of the same level 0 element e_L, we have

$R(D(u_1)||D(u_2)) \leq R = O(\text{poly}(\lambda))$, where $D(u_i)$ denotes the distribution (over the randomness of cenc) of the re-randomized encoding $u_i = \mathsf{cenc}_l(\text{par}, k, u_i')$ for $i \in \{1, 2\}$.

Next, we show that our requirement above is satisfied for GGHLite by a canonical re-randomization algorithm cenc with a similar choice of parameters as in Theorem 10.14. The proof is very similar to the proof of Theorem 10.14. The main difference is the direct "jump" in the RD-based analysis between the pair of encoding distributions $D(u_1), D(u_2)$ to avoid going through an intermediate canonical distribution, which would require applying a "strong" triangle inequality for the Rényi divergence.

Lemma 10.16 (Weak Pairwise-closeness Re-randomization for GGHLite)**.** *Let $\varepsilon_d, \varepsilon_\rho, \varepsilon_e \in (0, 1/2)$ and $\kappa \leq 2^n$. For $k \leq \kappa$ and $l \in [L]$, let $\mathsf{cenc}_l(\text{par}, k, u')$ denote the canonicalizing encoding algorithm for GGHLite that takes a level-k encoding $u' = [c'/z^k]_q$ with $\|c'\| \leq \gamma_{k,l}$, and returns a re-randomized encoding $u = [u' + \rho_1 \cdot x_1^{(k)} + \rho_2 \cdot x_2^{(k)}]_q$ with $\rho_1, \rho_2 \hookleftarrow D_{R,\sigma^*_{k,l}}$, for some admissible input encoding norm bound $\gamma_{k,l}$. Suppose that the following conditions hold:*

- ***LHL Smoothing:***

$$\sigma^*_{k,l} \geq n^{1.5} \cdot \ell_{g^{-1}} \cdot \sigma' \cdot \sqrt{2\log(4n \cdot \varepsilon_\rho^{-1})/\pi}. \tag{10.9}$$

- ***Offset "Drowning:"***

$$\sigma^*_{k,l} \geq (\sqrt{8\pi\varepsilon_d^{-1}}/\ell_b) \cdot \gamma_{k,l}. \tag{10.10}$$

Then cenc_l satisfies the weak pairwise-closeness re-randomization property for GGHLite with Rényi divergence bound

$$R = \exp(12\varepsilon_\rho + \varepsilon_d), \tag{10.11}$$

and admissible input encoding sets $S_{k,l} = \{u' = [c'/z^k]_q : \|c'\| \leq \gamma_{k,l}\}$.

Proof. We fix an instance $\text{par} = (n, q, y, \{(x_1^{(k)}, x_2^{(k)})\}_{k \leq \kappa})$ and p_{zt} of GGHLite, with $x_1^{(k)} = [b_1^{(k)}/z^k]_q$, $x_2^{(k)} = [b_2^{(k)}/z^k]_q$, and $y = [a/z]_q$ with $a = 1 + gr_y$, and two level-k encodings $u_i' = [c_i'/z^k]_q$ in $S_{k,l}$, i.e. with $\|c_i'\| \leq \gamma_{k,l}$, of the same level 0 element e_L, so that $c_i' = e_L + c_i \in R$ with $c_i \in \mathcal{I}$ for $i \in \{1, 2\}$. We consider the following sequence of games, where in each game, a re-randomized level-k encoding u of e_L is sampled, but the distribution of u differs among the games as follows:

- Game_0: In this game, we define u as the re-randomization of u_1', i.e. $u = \mathsf{cenc}_l(\text{par}, k, u_1') = [(e_L + c_1 + w)/z^k]_q$, where $w = \rho_1 \cdot b_1^{(k)} + \rho_2 \cdot b_2^{(k)} \in R$ and $\rho_i \hookleftarrow D_{R,\sigma^*_{k,l}}$ for $i \in \{1, 2\}$.

- Game_1: Modification of Game_1 in which the distribution of the re-randomization term w is replaced by the distribution $D_{\mathcal{I},\sigma^*_{k,l}(B^{(k)})^T}$, i.e. $u = [(e_L + c_1 + w)/z^k]_q$ with $w \hookleftarrow D_{\mathcal{I},\sigma^*_{k,l}(B^{(k)})^T}$.

- Game_2: Modification of Game_2 in which the randomization offset term $c_1 \in \mathcal{I}$ is replaced by offset term $c_2 \in \mathcal{I}$, i.e. $u = [(e_L + c_2 + w)/z^k]_q$ with $w \hookleftarrow D_{\mathcal{I},\sigma^*_{k,l}(B^{(k)})^T}$.

- Game_3: Modification of Game_2 which "undoes" the modification introduced in Game_1, i.e. in this game we have $u = [(e_L + c_2 + w)/z^k]_q$, where $w = \rho_1 \cdot b_1^{(k)} + \rho_2 \cdot b_2^{(k)} \in R$ and $\rho_i \hookleftarrow D_{R,\sigma^*_{k,l}}$ for $i \in \{1, 2\}$. Observe that in this game, u has exactly the distribution of a re-randomization of u_2', i.e. $u = \mathsf{cenc}_l(\text{par}, k, u_2')$.

For $i = 0, \ldots, 3$, let $D(u)_i$ denote the distribution of u in Game_i. To prove the lemma, it thus suffices to show that $R(D(u)_0 \| D(u)_3) \leq R$, with R defined in the lemma statement. Applying both of the weak triangle inequalities from Lemma 1.4, we get

$$R(D(u)_0 \| D(u)_3) \leq R_\infty(D(u)_0 \| D(u)_1)^2 \cdot R(D(u)_1 \| D(u)_2) \cdot R_\infty(D(u)_2 \| D(u)_3).$$

We now bound each factor in turn:

- To bound $R_\infty(D(u)_0 \| D(u)_1)^2$, we apply our LHL over R (Theorem 10.8) to conclude that $R_\infty(D(u)_0 \| D(u)_1) \leq 1 + 4\varepsilon_\rho$ if $\sigma^*_{k,l} \geq \|g^{-1}B^{(1)}\|_\infty n\sqrt{2\log(4n \cdot \varepsilon_\rho^{-1})/\pi}$, and $B^{(1)} \cdot R^2 = \mathcal{I}$. The last condition on $B^{(1)}$ holds by the rejection test of the $\mathsf{InstGen}$ algorithm of $\mathsf{GGHLite}$. The condition on $\sigma^*_{k,l}$ holds by the assumed LHL Smoothing condition and the bound $\|g^{-1} \cdot B^{(k)}\|_\infty \leq \|g^{-1}\| \cdot \|B^{(k)}\| \leq \ell_{g^{-1}} \cdot \sigma' \cdot \sqrt{n}$, from the rejection tests of the $\mathsf{InstGen}$ algorithm. Using the data processing inequality for R_∞, we conclude that

$$R_\infty(D(u)_0 \| D(u)_1)^2 \leq \exp(8\varepsilon_\rho).$$

- To bound $R(D(u)_1 \| D(u)_2)$, notice that for $i \in \{1, 2\}$, using the fact that $c_i \in \mathcal{I}$, the distribution of $c_i + w$ in Game_i is $D_i \stackrel{\text{def}}{=} D_{\mathcal{I}, \sigma^*_{k,l}(B^{(k)})^T, c_i}$. Applying our offset Gaussian divergence bound (Lemma 10.2) (with $w = c_1, z = c_2$) gives $R(D(u)_1 \| D(u)_2) \leq \exp(2\pi\|c_1 - c_2\|^2/(\sigma^*_{k,l}\sigma_n(B^{(k)}))^2)$. The latter is upper bounded by $\exp(\varepsilon_d)$ if $(\sigma^*_{k,l})^2 \geq \frac{2\pi\|c_1 - c_2\|^2}{\varepsilon_d \cdot \sigma_n(B^{(k)}))^2}$. This last condition is satisfied by the offset drowning condition, using $\|c_1 - c_2\| = \|c'_1 - c'_2\| \leq 2\gamma_{k,l}$ and the acceptance condition $\sigma_n(B^{(k)}) \geq \ell_b$ of the $\mathsf{InstGen}$ algorithm. We conclude that

$$R(D(u)_1 \| D(u)_2) \leq \exp(\varepsilon_d).$$

- To bound $R_\infty(D(u)_2 \| D(u)_3)$, we apply the LHL over R (Theorem 10.8) with the same argument as used to bound $R_\infty(D(u)_0 \| D(u)_1)$, except that this time, the order of the arguments to R_∞ is reversed. Since the R_∞ upper bound of Theorem 10.8 holds regardless of the order, we conclude that

$$R_\infty(D(u)_2 \| D(u)_3) \leq \exp(4\varepsilon_\rho).$$

Combining the above bounds gives the claimed bound R.

□

10.3.2 Eliminating z: an NTRU variant of GGHLite

In this section, we introduce a simplified variant of the GGH/$\mathsf{GGHLite}$ scheme that eliminates the parameter z, and yet preserves the security of the GDDH/GCDH problems. We call our variant the NTRU variant, since it involves publishing "NTRU-like" quotients $pk_i^{(k)} = [x_i^{(k)}/y^k]_q = [b_i^{(k)}/a^k]_q$ instead of the separate GGH parameters $x_i^{(k)}, y$, thus cancelling out the parameter z, and replacing it effectively by a. Similarly, level-k encodings in this construction also correspond to $\mathsf{GGHLite}$ encodings divided by y^k, i.e., have the form $u = [(e \cdot a^k + \rho_1 b_1^{(k)} + \rho_2 b_2^{(k)})/a^k]_q = [e + \rho_1 pk_1^{(k)} + \rho_2 pk_2^{(k)}]_q$. The zero testing parameter is accordingly modified to $p_{zt} = \frac{h}{g}a^k$. The latter encoding resembles an NTRU ciphertext for e with respect to public keys $pk_1^{(k)}, pk_2^{(k)}$, although in NTRU we have only one public key, whereas here we have two public keys. The fact that public parameters and encodings can be efficiently translated from $\mathsf{GGHLite}$ to the NTRU variant by taking quotients in R_q, implies that the security of the NTRU variant is at least as hard as $\mathsf{GGHLite}$. Details of the scheme are summarized in Figure 10.2.

- **Instance Generation** $\mathsf{InstGen}(1^\lambda, 1^\kappa)$: Given security parameter λ and multilinearity parameter κ, determine scheme parameters n, q, $m_r = 2$, σ, σ', $\ell_{g^{-1}}$, ℓ_b,ℓ. Let $R = \mathbb{Z}[x]/(x^n + 1)$ and $R_q = R/qR = \mathbb{Z}_q[x]/(x^n + 1)$. Do the following:
  - Sample $g \hookleftarrow D^t_{R,\sigma}$. If (1) $\|g^{-1}\| > \ell_{g^{-1}}$ or (2) (g) is not a prime ideal, resample g, else define ideal $\mathcal{I} = (g)$.
  - Sample $a \hookleftarrow D^t_{1+I,\sigma'}$ (note that $a = 1 + gr_y$ for some $r_y \in R$).
  - For $k \in [\kappa]$:
    - Sample $B^{(k)} = (b_1^{(k)}, b_2^{(k)})$ from $(D^t_{I,\sigma'})^2$. If: (1) $(b_1^{(k)}, b_2^{(k)}) \neq \mathcal{I}$, or (2) $\sigma_n(\mathrm{rot}(B^{(k)})) < \ell_b$, resample.
    - Define level-k public keys: $pk_1^{(k)} = [b_1^{(k)} \cdot a^{-k}]_q$, $pk_2^{(k)} = [b_2^{(k)} \cdot a^{-k}]_q$.
  - Sample $h \hookleftarrow D_{R,\sqrt{q}}$ and define the zero-testing parameter: $p_{zt,\kappa} = [\frac{h}{g} a^\kappa]_q \in R_q$.
  - Return public parameters $\mathrm{par} = (n, q, \{(pk_1^{(k)}, pk_2^{(k)})\}_{k\in[\kappa]})$ and p_{zt}.
- **Level-k encoding** $\mathsf{enc}_k(\mathrm{par}, e)$: Given level-0 encoding $e \in R$ and parameters par, return $u = [e + \rho_1 \cdot pk_1^{(k)} + \rho_2 \cdot pk_2^{(k)})]_q$, with $\rho_1, \rho_2 \sim D_{R,\sigma^*_k}$ (note $u = [(c' + \rho_1 b_1^{(k)} + \rho_2 b_2^{(k)})/a^k]_q$, where $c' = e \cdot a^k \in e + I$).

Figure 10.2: The new algorithms of our NTRU variant GGHLite scheme. Other algorithms are the same as in the original GGH scheme.

Security of the construction.. We can define the corresponding problems GCDH^{NTRU}, $\mathrm{ExtGCDH}^{NTRU}$ and GDDH^{NTRU} for this NTRU variant, in the natural way as in Section 9.2, but with respect to experiment of Figure 10.3.

Given parameters $\lambda, n, q, m_r, \kappa, \sigma'$, proceed as follows:

1. Run $\mathsf{InstGen}(1^n, 1^\kappa)$ to get $\mathrm{par} = (n, q, \{pk_j^{(k)}\}_{j,k})$ and p_{zt}.
2. For $i = 0, \ldots, \kappa$:
   - Sample $e_i \hookleftarrow D_{R,\sigma'}$ and $f_i \hookleftarrow D_{R,\sigma'}$,
   - Set $u_i = [e_i + \sum_j \rho_{ij} pk_j]_q$ with $\rho_{ij} \hookleftarrow \chi_1$ for all j.
3. Set $u^* = [\prod_{i=1}^{\kappa} u_i]_q$.
4. Set $v_C = [e_0 u^*]_q$.
5. Sample $\rho_j \hookleftarrow \chi_\kappa$ for all j; set $v_D = [e_0 u^* + \sum_j \rho_j pk_j^{(\kappa)}]_q$.
6. Set $v_R = [f_0 u^* + \sum_j \rho_j pk_j^{(\kappa)}]_q$.

Figure 10.3: The GGH^{NTRU} security experiment.

To show that the NTRU variant of the GGH encoding scheme is at least as secure as the GGH scheme, we now provide a formal reduction from GDDH to GDDH^{NTRU} (and similarly for the

other two problems).

Theorem 10.17. *There exists a polynomial time reduction from GDDH (resp. GCDH/ExtGCDH) to GDDH^{NTRU} (resp. GCDH^{NTRU}/ExtGCDH^{NTRU}).*

Proof. For simplicity, we only describe the reduction from GDDH to GDDH^{NTRU}. Let $\{(y, \{x_j\}_j, p_{zt}), u_0, \ldots, u_\kappa, v\}$ be a GDDH instance and let $\mathcal{O}$ be a polynomial-time oracle for solving GDDH^{NTRU}.

- Let $pk_j^{(k)} = [\frac{x_j^{(k)}}{y}]_q$ for $j \in \{1, 2\}$ and $k \in [\kappa]$,
- Let $\hat{p}_{zt} = [p_{zt} \cdot y^\kappa]_q$,
- Let $\hat{v} = [v \cdot y^{-\kappa}]_q$,
- Call the oracle $\mathcal{O}$ on input $\{(\{pk_j^{(k)}\}_{j,k}, \hat{p}_{zt}), [\frac{u_0}{y}]_q, \ldots, [\frac{u_\kappa}{y}]_q, \hat{v}\}$.

We have $u_i = \mathsf{enc}_1(e_i) = [e_i y + \sum_j \rho_j^{(i)} x_j^{(1)}]_q$ for all $i \in [\kappa]$, then let $u_i^{\text{NTRU}} = [\frac{u_i}{y}]_q = [e_i + \sum_j \rho_j^{(i)} \frac{x_j^{(1)}}{y}]_q = [e_i + \sum_j \rho_j^{(i)} pk_j^{(1)}]_q$ is a valid NTRU variant level-1 encoding for e_i. Furthermore, if $v = v_D$, then

$$\hat{v} = [(e_0 \cdot u^* + \sum_j \rho_j \cdot x_j^{(\kappa)}) \cdot y^{-\kappa}]_q = [e_0 \cdot \prod_{i=1}^{\kappa} (\frac{u_i}{y}) + \sum_j \rho_j \cdot \frac{x_j^{(\kappa)}}{y^\kappa}]_q = [e_0 \cdot \prod_{i=1}^{\kappa} u_i^{\text{NTRU}} + \sum_j \rho_j \cdot pk_j^{(\kappa)}]_q,$$

is a valid NTRU variant level-κ encoding of $\prod_i e_i$. Similarly, if $v = v_R$, then $\hat{v}$ is a valid NTRU variant level-κ encoding of $f_0 \prod_{i \geq 1} e_i$, as required.

□

10.4 Parameter settings

In Table 10.1, we summarize asymptotic parameters for GGHLite to achieve 2^λ security for the underlying Ext-GCDH problem, assuming the hardness of the canonical Ext-cGCDH problem, and to satisfy the zero-testing/extraction correctness conditions with error probability $\lambda^{-\omega(1)}$. For simplicity, we assume that $\kappa = \omega(1)$ and $\kappa = O(\text{poly}(\lambda))$. For comparison, we also show the corresponding parameters for GGH. The "Condition" column lists the conditions that determine the corresponding parameter in the case of GGHLite. For security of the canonical Ext-cGCDH problem, we assume (as in [GGH13a]) that the best attack is the one described in [GGH13a, Section 6.3.3], whose complexity is dominated by the cost of solving γ-SVP (the Shortest lattice Vector Problem with approximation factor γ) for the lattice $\mathcal{I}$, with γ set at $\approx q^{3/8}$ to get a sufficiently short multiple of g. By the lattice reduction "rule of thumb," to make this cost 2^λ, we need to set

$$n = \Omega(\lambda \log q). \tag{10.12}$$

When $\kappa = \text{poly}(\log \lambda)$, the dimension n, encoding length $|\mathsf{enc}|$ and public parameters length $|\text{par}|$ in our scheme GGHLite are all asymptotically close to optimal, namely quasi-linear in the security parameter λ, versus quadratic (resp. cubic and quintic) in λ for GGH [GGH13a]. Thus we expect GGHLite's public parameters and encodings to be orders of magnitudes shorter than GGH for typical $\lambda \approx 100$.

Table 10.1: Asymptotic parameters.

Parameter	GGHLite	GGH[GGH13a]	Condition
m_r	2	$\Omega(n \log n)$	LHL: Theorem 10.8
σ	$O(n \log n)$	$O(n \log n)$	Eq. (10.1)
$\ell_{g^{-1}}$	$O(1/\sqrt{n \log n})$	$O(1/\sqrt{n \log n})$	Eq. (10.1)
$\varepsilon_d, \varepsilon_e, \varepsilon_\rho$	$O(\kappa^{-1})$	$O(2^{-\lambda}\kappa^{-1})$	Eq. (10.8)
σ'	$\widetilde{O}(n^{3.5})$	$\widetilde{O}(n^{1.5}\sqrt{\lambda})$	Eq. (10.3)
σ_1^*	$\widetilde{O}(n^{5.5}\sqrt{\kappa})$	$\widetilde{O}(2^\lambda \lambda n^{4.5}\kappa)$	Drown: Eq. (10.6)
ε_{ext}	$O(\lambda^{-\omega(1)})$	$O(\lambda^{-\omega(1)})$	
q	$\widetilde{O}(n^{10.5}\sqrt{\kappa})^{8\kappa}$	$\widetilde{O}(2^\lambda \lambda^{1.5} n^{8.5}\kappa)^{8\kappa}$	Corr.: Eq. (10.4)
n	$O(\kappa\lambda \log \lambda)$	$O(\kappa\lambda^2)$	SVP: Eq. (10.12)
$\lvert\mathsf{enc}\rvert$	$O(\kappa^2 \lambda \log^2 \lambda)$	$O(\kappa^2\lambda^3)$	$O(n \log q)$
$\lvert\mathsf{par}\rvert$	$O(\kappa^3 \lambda \log^2 \lambda)$	$O(\kappa^4\lambda^5 \log \lambda)$	$O(m_r \kappa n \log q)$

10.5 Applications

In previous sections, we have shown that our graded encoding scheme GGHLite can be instantiated much more efficiently than the GGH scheme [GGH13a], but on the other hand, with our efficient choice of parameters for GGHLite, we have only been able prove the hardness of the *search* problem Ext-GCDH (based on the hardness of the corresponding canonical problem) rather than the *decision* problem GDDH used in [GGH13a]. In this section, we show that the hardness of Ext-GCDH is sufficient for important applications of graded encoding schemes, in the random oracle model. In particular, we show that existing protocol based on the hardness of GDDH can be easily modified to make their security based on Ext-GCDH in the random oracle model, while preserving the efficiency of the original protocols, up to a small factor.

10.5.1 Efficient one-round N-party Diffie-Hellman key exchange in the ROM

We show how to adapt the one round N-party key exchange protocol described in [GGH13a, Section 5] and recalled in Section 9.1.2 (originally described by Boneh and Silverman [BS03] in the abstract setting of multilinear maps) to achieve security assuming the hardness the Ext-GCDH problem, rather than the GDDH problem, in the random oracle model. The modification is straightforward: we simply replace the shared key $s = \mathsf{ext}(par, p_{zt}, v)$ in the original protocol, where v is the encoding of the Diffie-Hellman product of the N parties' secrets, by its hash $K = H(\mathsf{ext}(par, p_{zt}, v))$, where $H(.) : \{0,1\}^* \to \{0,1\}^\lambda$ denotes a hash function modelled as a random oracle. Details follow.

Construction. Given a κ-graded encoding scheme with $\kappa = N - 1$ over an encoded element ring $R/\mathcal{I}$ of prime order p, and a hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$, the N-party key exchange protocol is presented in Figure 10.4.

Correctness. We have to show that all the N computed keys $K_1, \ldots, K_N$ are equal except for negligible probability $\lambda^{-\omega(1)}$. In the KeyGen algorithm, each party computes an encoding v_j of the product $e_L = \prod_i e_{i,L}$ in the ring $R/\mathcal{I}$. Since $|R/\mathcal{I}| = \Omega(2^\lambda)$ is prime and the distribution of the $e_{i,L}$'s is within statistical distance $O(2^{-\lambda})$ of uniform on $R/\mathcal{I}$, the product e_L is also within negligible statistical distance $O(2^{-\lambda})$ to a uniformly random element in $R/\mathcal{I}$. Hence by

- **Setup** $\mathsf{Setup}(1^\lambda, 1^N)$: Given security parameter λ and number of parties N, run $\mathsf{InstGen}(1^{2\lambda+1}, 1^{N-1})$ for the graded encoding scheme to get (par, p_{zt}) and output protocol public parameters (par, p_{zt}).
- **Publish** $\mathsf{Publish}(\mathrm{par}, p_{zt}, i)$: The ith party runs the level-0 encoding sampler to generate a random secret key $e_i = \mathsf{Samp}(\mathrm{par})$ (corresponding to encoded element $e_{i,L}$), and publishes a corresponding level-1 public key $u_i = \mathsf{enc}_1(\mathrm{par}, e_i)$.
- **KeyGen** $\mathsf{KeyGen}(\mathrm{par}, p_{zt}, j, e_j, \{u_i\}_{i\neq j})$: The jth party computes a level-$(N-1)$ encoding $v_j = e_j \cdot \prod_{i\neq j} u_i$ of the Diffie-Hellman product $\prod_i e_{i,L}$, and computes the key $K_j = H(s_j)$, where $s_j = \mathsf{ext}(\mathrm{par}, p_{zt}, v_j)$ is the extracted string for v_j.

Figure 10.4: Our modified N-party Diffie-Hellman key exchange protocol.

the extraction correctness property of the encoding scheme, all N extracted strings $\{s_j\}_{i\in[N]}$, and hence also all N computed keys $\{K_j\}_{i\in[N]}$, are equal, except with negligible probability $O(N \cdot \lambda^{-\omega(1)}) = O(\lambda^{-\omega(1)})$ for $N = \lambda^{O(1)}$.

Passive security. We have to show that, given (par, p_{zt}) and the public keys $u_1, \ldots, u_N$, the key (say K_1) is indistinguishable to the adversary $\mathcal{A}$ from a uniformly random string in $\{0,1\}^\lambda$, assuming the hardness of the Ext-GCDH problem and the random oracle model for H. Formally, we define a passive security attack game, in which $\mathcal{A}$ is given (par, p_{zt}), $u_1, \ldots, u_N$, and T_b, for a uniformly random bit $b \in \{0,1\}$, where $T_0 = K_1$ is the real key and $T_1 = R \hookleftarrow U(\{0,1\}^\lambda)$ is an independent uniformly random string, and $\mathcal{A}$ outputs a guess b' for b. We say that $\mathcal{A}$'s advantage is $\varepsilon = 2(\Pr[b' = b] - 1/2)$.

Lemma 10.18. *Let $\mathcal{A}$ denote an attacker, in the random oracle model for H, against the passive security of the N-party Diffie-Hellman key exchange protocol in Figure 10.4, with run-time T and advantage ε, making q_H queries to H. Then there exists an algorithm $\mathcal{A}'$ for the Ext-GCDH problem for the underlying encoding scheme, with run-time $T' = T$ and success probability $\varepsilon' \geq \varepsilon/(2q_H)$.*

Proof. Let Game_1 denote the passive security attack game with $\mathcal{A}$, and let Game_2 denote a modification of Game_1 in which $\mathcal{A}$'s queries to H are answered differently as follows: if the query x is equal to $s_1 = \mathsf{ext}(\mathrm{par}, p_{zt}, e_1 \cdot \prod_{i>1} u_i)$, the query is answered with a uniformly random $K \in \{0,1\}^\lambda$ (instead of $K_1 = H(s_1)$), otherwise, the query is answered with $H(x)$, as in Game_1.

For $i \in \{1,2\}$, let S_i denote the event that $b' = b$ in Game_i, and let E denote the event in Game_1 that $\mathcal{A}$ queries H at s_1. Note that by definition, $\Pr[S_1] = 1/2 + \varepsilon/2$, and we also have $\Pr[S_2] = 1/2$ because in Game_2, T_b is a uniformly random string independent of $\mathcal{A}$'s prior view, regardless of the value of b. On the other hand, since the view of $\mathcal{A}$ is identical in Game_1 and Game_2 until $\mathcal{A}$ queries H at s_1, we have $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[E]$. It follows that $\Pr[E] \geq \varepsilon/2$. Given an input instance $(\mathrm{par}, p_{zt}, \{u_i\}_i)$ of the Ext-GCDH problem, the attacker $\mathcal{A}'$ simply runs $\mathcal{A}$ on input $(\mathrm{par}, p_{zt}, \{u_i\}_i)$ and T_b (with $T_0 = K_1$ chosen uniformly random in $\{0,1\}^\lambda$ – note that $\mathcal{A}'$ does not need to know s_1 to simulate T_0) and simulates Game_1, hoping that the event E occurs. Let $\{x_i\}_{i\in[q_H]}$ denote the queries made by $\mathcal{A}$ to H, When $\mathcal{A}$ finishes, $\mathcal{A}'$ chooses $i \in [q_H]$ uniformly at random and outputs x_i as its guess for $\mathcal{A}$'s query that equals s_1 (note that until $\mathcal{A}$ queries H at s_1, the view of $\mathcal{A}$ is perfectly simulated by $\mathcal{A}'$ as in Game_1, so $\Pr[E]$ is preserved). Conditioned on the event E occurring, we have $x_i = s_1$ with probability $\geq 1/q_H$. Overall, $\mathcal{A}'$ outputs the correct Ext-GCDH solution with probability $\geq 1/q_H \cdot \Pr[E] \geq \varepsilon/(2q_H)$. □

Note that when the protocol attacker $\mathcal{A}$ has run-time $T = 2^\lambda$ (so that also $q_H \leq 2^\lambda$) and advantage $\varepsilon \geq 2^{-\lambda}$, the Ext-GCDH attacker $\mathcal{A}'$ constructed by our security lemma above, has run-time $T' = 2^\lambda$ and advantage $\varepsilon' \geq 2^{-(2\lambda+1)}$, thus contradicting the assumed $2^{2\lambda+1}$-security of the undrlying encoding scheme (it is for this reason that we used a security parameter $\lambda' = 2\lambda + 1$ for the encoding scheme). Consequently, we only lose a constant factor ≈ 2 in relating the security parameter of the encoding scheme to that of the protocol, essentially preserving the efficiency of our encoding scheme in this application.

Conclusion

The Learning With Errors problem (LWE), introduced by Regev [Reg05, Reg09], and the Small Integer Solution problem (SIS), introduced by Ajtai [Ajt96], are fundamental in lattice-based cryptography as most of the recent schemes are based on them. Regev also provided a quantum reduction from a standard lattice problem to prove the hardness of LWE. Peikert [Pei09] provided the first classical reduction, but only for an exponential modulus q. In Chapter 4, we showed that LWE (in dimension n) is at least as hard as standard worst case lattice problems (in dimension $\sqrt{n}$), even with polynomial modulus. We also showed that the hardness of LWE is a function of $n \log q$. In Chapter 5, we studied the hardness of two variants: the Module-SIS and Module-LWE problems, which bridge SIS with Ring-SIS, and LWE with Ring-LWE, respectively. We also showed that the Ring-LWE problem is hard independently of the arithmetic shape of the modulus q. Previously, this problem was only shown to be hard for some specific moduli.

In Chapter 7, we proposed the first lattice-based group signature schemes where the signature and public key sizes are essentially logarithmic in the number of group members (for any fixed security level). The security of our schemes is proved in the random oracle model (ROM) under the SIS and LWE assumptions. In Chapter 8, we introduced the first lattice-based group signature also with logarithmic signature size but enjoying another functionality, verifier local revocation. In the ROM, this scheme is proved to be secure based on the hardness of the SIS problem.

Finally, in Chapters 9 and 10, we studied the GGH Graded Encoding Scheme introduced by Garg, Gentry and Halevi [GGH13a]. The GGH scheme, based on ideal lattices, is the first plausible approximation to a cryptographic multilinear map. The main contributions of our work were to formalize, simplify and improve the security analysis of the "re-randomization process" in the GGH construction. We applied these results in a new construction called GGHLite which is more efficient than the original construction. We first use the Rényi divergence instead of the conventional statistical distance as a measure of distance between distributions in the security reduction to obtain our first improvement. Our second achievement is to construct a scheme with a reduced number of randomizers. These two contributions allow to decrease the bit size of the public parameters from $O(\lambda^5 \log \lambda)$ for the GGH scheme to $O(\lambda \log^2 \lambda)$ in GGHLite, with respect to the security parameter λ (for a constant multilinearity parameter).

I believe that lattice-based cryptography is a good candidate for modern cryptography. We already mentioned the advantages: its simplicity, the existence of well understood security proofs relying on hard problems on lattices, and the fact that it allows to construct basic primitives (such as encryption or signature schemes) and very exciting ones (such as fully homomorphic encryption, or cryptographic multilinear maps), which, at least for now, only exist in lattice-based cryptography. The introduction of the ring variants [Mic02a, Mic07] of the SIS and LWE problems and the adaptation of the corresponding schemes was a first step for all those schemes to be more

efficient in practice. The only problem is that we are not certain of the security of those schemes, as the ring variants of SIS and LWE are only proven secure under the hardness of variants of SVP on ideal lattices [LM06, PR06, LPR10]. Furthermore, even using the ring variants the size of the schemes are still large because of the size of the parameters needed for the security reductions to apply. As a consequence, to be efficient in practice, implementations use parameters smaller than the ones which should be used to keep the security based on worst-case lattice problems (for example in [MR09, GLP12]).

It seems that there is a gap between what we can prove to be secure, and what is implemented in practice, and that the biggest challenge in lattice-based cryptography today is to produce primitives that are both secure and efficient. There already exist some works in this direction, as [LMPR08, MR09, GLP12, PG13, OPG14, PDG14] for practice lattice-based signature, hash function, and encryption schemes. There are many ways to tackle this problem. The first one would be to improve the existing security reductions, to obtain better parameters which could be used in practice, and to work on the reductions of the ring variants of SIS and LWE, maybe to prove their hardness without restrictions on the variant of SVP. Another line of research would be to improve the dependency of the primitives to those parameters, and to work on the implementation of the primitives. In the following, we describe several project for each of these approaches.

The security foundations. Concerning the hardness of the LWE problem, we showed in Chapter 4 that there is a classical reduction from a lattice problem in dimension $\sqrt{n}$ to LWE in dimension n and a polynomial modulus. This quadratic loss in the dimension does not exist in the quantum reduction of Regev [Reg05, Reg09]. The dimension of the lattice is crucial here as the hardness of the lattice problem depends mainly on its dimension. Consequently, a first question that arises is the existence of a reduction which does not have this quadratic loss.

Further, in the same work we proved the classical hardness of Ring-LWE for an exponential modulus under the hardness of problems on general lattices (instead of problems on ideal lattices as in previous reductions). A second question that arises is the existence of a classical reduction that would prove the hardness of the Ring-LWE problem for a polynomial modulus under the hardness of problems in general lattices. We already have some elements to answer this question (in Section 5.2.5): a modulus-switching reduction for this problem allows to reduce Ring-LWE with a modulus q to Ring-LWE with another modulus p (this modulus-switching method is crucial in the proof of classical hardness of LWE). But to be used in this particular case, this reduction requires the first Ring-LWE problem to have a small secret size, and unfortunately it is not proven yet that Ring-LWE remains hard for very small secrets. It would be very interesting to work on this problem, as the Ring variant of LWE is the one on which practical cryptographic constructions are based.

On the other hand, we should continue to study carefully the variants of SVP on ideal lattices on which rely the hardness assumption of the ring variants of SIS and LWE. Ideal lattices are a specific family of Euclidean lattices that correspond to ideals of the ring of integers of a number field. In this particular family there is an additional structure which is used in the schemes to be more efficient, but could also be a problem in term of security. For now, there is no really faster algorithm to solve variants of SVP on ideal lattices, but it would be interesting to see if one can exploit this additional structure to devise more efficient attacks.

In Chapter 10, we use the Rényi divergence in the security reductions instead of the statistical distance. This new technique could be used in other existing constructions, to improve parameters guaranteeing a desired level of security. It is also still open to find a way to use this method for reductions using decisional problems instead of computational ones. In a joint work in progress, with Ron Steinfeld and Damien Stehlé, we suggest a solution to use the Rényi divergence in

reductions between decisional problems. But we still have a condition on the problems, and this solution does not apply to the specific case of GDDH needed in Chapter 10. We also provide alternative reductions for existing problems, such as LWE with uniform noise [DMQ13], which have parameters slightly better than the original reductions and are often simpler. But there still are many reductions which could be improved with this technique, in particular if we manage to remove the condition on the problems to apply our method to decisional problems.

Finally, the security of the N-party Diffie Hellman key exchange using both constructions (GGHLite and [CLT13]) is based on problems (like GDDH for the GGH scheme) which are not well studied. The situation is similar for other variants, such as the Multilinear Jigsaw puzzles, used for the security of obfuscation schemes [GGH+13b]. It would be interesting to either prove the security of these schemes, or to construct a version of this multilinear map with a security based on LWE, or another well studied problem. One way could be to use the similarities between the two existing multilinear map schemes [GGH13a, CLT13] and some existing somewhat homomorphic encryption schemes [Gen09, CCK+13].

Faster primitives and implementations. The two lattice-based group signature schemes that we describe in this thesis are not usable in practice. It is for the same reason as many lattice-based schemes: the parameters for the schemes to be secure are much too high for an efficient implementation. The solution we already mentioned is to use the ring variants of SIS and LWE to construct schemes. It should be interesting to adapt both schemes in the ring setting. These adaptations seem quite direct. The remaining problem concerning efficiency will be the zero-knowledge proofs of knowledge. In both constructions, we have to proceed with parallel repetitions of the zero-knowledge proofs of knowledge needed in the signature (both described in Chapter 6.4.3). In Chapter 7, we need this repetition to lower the rejection probability, whereas in Chapter 8, we need it as the soundness error is a constant. In both cases, this repetition is an obstacle to an efficient variant of group signature schemes. It would also be interesting to implement those ring versions of the lattice-based group signature schemes to compare their efficiency with existing ones [Gro07, AS12], and to determine which size of group we could achieve using those constructions.

Finally, the first step after our result in Chapters 9 and 10 is to implement the GGHLite scheme and to compare the results of the implementation with the only other existing implementation of multilinear maps in [CLT13]. In a work in progress with Martin Albrecht, Catalin Cocis and Fabien Laguillaumie, we work on this implementation of GGHlite and of the N-party Diffie-Hellman key exchange (described in Section 10.5.1). We also work on setting practical and secure parameters, by studying the best known attacks against the scheme.

List of Figures

Bibliography

[ABB10a] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010. viii, xiii, 29, 33, 97, 103, 104, 106, 109

[ABB10b] S. Agrawal, D. Boneh, and X. Boyen. Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In *Proc. of CRYPTO*, volume 6223 of *LNCS*, pages 98–115. Springer, 2010. viii, xiii, 29

[ACJT00] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Proc. of CRYPTO*, volume 1880 of *LNCS*, pages 255–270. Springer, 2000. 83, 84, 95, 96

[ACPS09] B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Proc. of CRYPTO*, volume 5677 of *LNCS*, pages 595–618. Springer, 2009. 26, 59, 74

[AD87] D. Aldous and P. Diaconis. Strong uniform times and finite random walks. *Adv. in Appl. Math.*, 8(1):69–97, 1987. 4

[AD97] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. of STOC*, pages 284–293. ACM, 1997. 47

[AGHS13] S. Agrawal, G. Gentry, S. Halevi, and A. Sahai. Discrete gaussian leftover hash lemma over infinite domains. In *Proc. of ASIACRYPT*, volume 8269 of *LNCS*, pages 97–116. Springer, 2013. xii, 19, 135, 137, 148, 150, 151, 154, 155, 156

[AGV09] A. Akavia, S. Goldwasser, and V. Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In *TCC*, pages 474–495, 2009. 29

[AJLA+12] G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Proc. of EUROCRYPT*, volume 7237 of *LNCS*, pages 483–501. Springer, 2012. 150

[Ajt96] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. of STOC*, pages 99–108. ACM, 1996. viii, ix, x, xiii, xiv, 21, 22, 29, 43, 169

[Ajt99] M. Ajtai. Generating hard instances of the short basis problem. In *Proc. of ICALP*, volume 1644 of *LNCS*, pages 1–9. Springer, 1999. 34

[AKS01] M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proc. of STOC*, pages 601–610. ACM, 2001. ix, xiv

[AP11] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. *TCS*, 48(3):535–553, 2011. 34, 35

[AR13] D. Aggarwal and O. Regev. A note on discrete gaussian combinations of lattice vectors, 2013. 154

[AS12] L. El Aimani and O. Sanders. Efficient group signatures in the standard model. In *ICISC*, volume 7839 of *LNCS*, pages 410–424. Springer, 2012. 171

[ASP12] J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In *Proc. of PKC*, volume 7293 of *LNCS*, pages 334–352. Springer, 2012. 46, 52, 59, 150

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of number. *Math. Ann*, 296:625–635, 1993. 17

[BB04] D. Boneh and X. Boyen. Efficient selective-id secure identity-based encryption without random oracles. In *Proc. of EUROCRYPT*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004. 106

[BBS04] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Proc. of CRYPTO*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004. 83, 84, 92, 95, 100, 115

[BCN+10] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get shorty via group signatures without encryption. In *Proc. of SCN*, pages 381–398, 2010. 81, 84

[BF03] D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003. 135

[BF11] D. Boneh and D. M. Freeman. Linearly homomorphic signatures over binary fields and new tools for lattice-based signatures. In *Proc. of PKC*, volume 6571 of *LNCS*, pages 1–16. Springer, 2011. 59

[BG92] M. Bellare and O. Goldreich. On defining proofs of knowledge. In *Proc. of CRYPTO*, volume 740 of *LNCS*, pages 390–420. Springer, 1992. 85, 86

[BGV11] Z. Brakerski, G. Gentry, and V. Vaikuntanathan. Fully homomorphic encryption without bootstrapping. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:111, 2011. xi, 57, 65

[BGV12] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In *Proc. of ITCS*, pages 309–325, 2012. viii, xiii, 29, 46

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. Classical hardness of learning with errors. In *Proc. of STOC*, pages 575–584. ACM, 2013. xi, xv, 3, 43, 45

[BMW03] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Proc. of EUROCRYPT*, volume 2656 of *LNCS*, pages 614–629, 2003. 83, 84, 89, 91, 92, 95, 96, 115

[BN06] M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *Proc. of ACM-CCS*, pages 390–399. ACM Press, 2006. 105, 113

[Boy10] X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *Proc. of PKC*, volume 6056 of *LNCS*, pages 499–517. Springer, 2010. ix, xiv, 29, 33, 40, 41, 95, 96, 103, 105, 114

[BP91] W. Bosma and M. Pohst. Computations with finitely generated modules over Dedekind rings. In *Proc. of ISSAC*, pages 151–156, 1991. 10, 59

[BPR12] A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In *Proc. of EUROCRYPT*, volume 7237 of *LNCS*, pages 719–737. Springer, 2012. 150

[Bra12] Z. Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In *Proc. of CRYPTO*, volume 7417 of *LNCS*, pages 868–886. Springer, 2012. viii, xiii, 29, 46

[Bri03] E. Brickell. An efficient protocol for anonymously providing assurance of the container of a private key, apr. 2003., 2003. Submitted to the Trusted Computing Group. 83, 84

[BS96] E. Bach and J. Shallit. *Algorithmic number theory. Vol. 1.* Foundations of Computing Series. MIT Press, Cambridge, MA, 1996. 28

[BS99] J. Blömer and J.-P. Seifert. On the complexity of computing short linearly independent vectors and short bases in a lattice. In *Proc. of STOC*, pages 711–720. ACM, 1999. ix, xiv, 11

[BS03] D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography. *Contemporary Mathematics*, 324:71–90, 2003. xi, 135, 137, 138, 166

[BS04] D. Boneh and H. Shacham. Group Signatures with Verifier-local Revocation. In *Proc. of ACM-CCS*, pages 168–177. ACM Press, 2004. 81, 84, 92, 115

[BSZ05] M. Bellare, H. Shi, and C. Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In *Proc. of CT-RSA*, volume 3376 of *LNCS*, pages 136–153. Springer, 2005. 83

[BV96] D. Boneh and R. Venkatesan. Hardness of computing the most significant bits of secret keys in Diffie-Hellman and related schemes. In *Proc. of CRYPTO*, volume 1109 of *LNCS*, pages 129–142. Springer, 1996. 47

[BV11] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *Proc. of FOCS*, pages 97–106, 2011. viii, ix, xiii, xiv, 29, 46

[BW06] X. Boyen and B. Waters. Compact group signatures without random oracles. In *Proc. of EUROCRYPT*, volume 4004 of *LNCS*, pages 427–444. Springer, 2006. 83

[BW07] X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *Proc. of PKC*, volume 4450 of *LNCS*, pages 1–15. Springer, 2007. xi, 83

[CCK+13] J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun. Batch fully homomorphic encryption over the integers. In *Proc. of EUROCRYPT*, volume 7881 of *LNCS*, pages 315–335. Springer, 2013. 171

[CDS94] R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Proc. of CRYPTO*, volume 839 of *LNCS*, pages 174–187. Springer, 1994. 98

[CG04] J. Camenisch and J. Groth. Group signatures: Better efficiency and new theoretical aspects. In *Proc. of SCN*, volume 3352 of *LNCS*, pages 120–133. Springer, 2004. 84

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Proc. of EUROCRYPT*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010. viii, ix, xii, xiii, xiv, xvii, 29, 33, 37, 39, 40, 95, 116, 117, 123

[CL02] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *Proc. of SCN*, volume 2576 of *LNCS*, pages 268–289. Springer, 2002. 83, 84, 96

[CL04] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Proc. of CRYPTO*, volume 3152 of *LNCS*, pages 56–72. Springer, 2004. 83, 96

[CLT13] J.-S. Coron, T. Lepoint, and M. Tibouchi. Practical multilinear maps over the integers. In *Proc. of CRYPTO*, volume 8042 of *LNCS*, pages 476–493. Springer, 2013. viii, xiii, 137, 171

[CNR12] J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *Proc. of SCN*, volume 7485 of *LNCS*, pages 57–75. Springer, 2012. xi, 81, 95, 96, 97, 115

[Coh00] H. Cohen. *Advanced topics in computational number theory*. Springer, 2000. 8, 10, 59

[CvH91] D. Chaum and E. van Heyst. Group signatures. In *Proc. of EUROCRYPT*, volume 547 of *LNCS*, pages 257–265. Springer, 1991. 83

[Dam10] I Damgård. On Σ-protocols. Manuscript, 2010. Available at http://www.daimi.au.dk/~ivan/Sigma.pdf. 98

[DDLL13] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal gaussians. In *Proc. of CRYPTO*, volume 8042 of *LNCS*, pages 40–56. Springer, 2013. viii, xiii, 29

[Dev86] L. Devroye. *Nonuniform random variate generation*. Springer-Verlag, New York, 1986. 14

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22:644–654, 1976. vii, viii, xii, 30

[DMQ13] N. Döttling and J. Müller-Quade. Lossy codes and a new variant of the learning-with-errors problem. In *Proc. of EUROCRYPT*, volume 7881 of *LNCS*, pages 18–34. Springer, 2013. 171

[EH12] T. van Erven and P. Harremoës. Rényi divergence and Kullback-Leibler divergence. *CoRR*, abs/1206.2459, 2012. 4, 5, 6

[FP96] C. Fieker and M. E. Pohst. Lattices over number fields. In *Proc. ANTS*, volume 1122 of *LNCS*, pages 147–157. Springer, 1996. 59

[FS86] A. Fiat and A. Shamir. How to prove yourself – practical solutions to identification and signature problems. In *Proc. of CRYPTO*, volume 263 of *LNCS*, pages 186–194. Springer, 1986. 29, 87, 97, 115

[FS90] U. Feige and A. Shamir. Witness indistinguishable and witness hiding protocols. In *Proc. of STOC*, pages 416–426. ACM, 1990. 86

[FS10] C. Fieker and D. Stehlé. Short bases of lattices over number fields. In *Proc. of ANTS-IX*, volume 6197 of *LNCS*, pages 157–173. Springer, 2010. 59

[Gam84] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proc. of CRYPTO*, volume 196 of *LNCS*, pages 10–18. Springer, 1984. Conference version of [Gam85]. 179

[Gam85] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985. Journal version of [Gam84]. viii, 179

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. of STOC*, pages 169–178. ACM, 2009. viii, xiii, 29, 150, 171

[GGH96] O. Goldreich, S. Goldwasser, and S. Halevi. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996. viii, xiii, 29

[GGH97] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Proc. of CRYPTO*, volume 1294 of *LNCS*, pages 112–131. Springer, 1997. viii, xiii, 29

[GGH13a] S. Garg, C. Gentry, and S. Halevi. Candidate multilinear maps from ideal lattices. In *Proc. of EUROCRYPT*, volume 7881 of *LNCS*, pages 1–17. Springer, 2013. viii, ix, xii, xiii, xiv, xvi, 135, 137, 139, 140, 142, 144, 145, 147, 148, 149, 150, 158, 161, 165, 166, 169, 171

[GGH+13b] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Proc. of FOCS*, pages 40–49, 2013. viii, xiii, 171

[GGH+13c] S. Garg, C. Gentry, S. Halevi, A. Sahai, and B. Waters. Attribute-based encryption for circuits from multilinear maps. In *Proc. of CRYPTO*, volume 8043 of *LNCS*, pages 479–499. Springer, 2013. 161

[GHPS12] C. Gentry, S. Halevi, C. Peikert, and N. P. Smart. Ring switching in BGV-style homomorphic encryption. In *Proc. of SCN*, pages 19–37, 2012. 47

[GKPV10] S. Goldwasser, Y. Tauman Kalai, C. Peikert, and V. Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS*, pages 230–240, 2010. 46

[GKV10] S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Proc. of ASIACRYPT*, volume 6477 of *LNCS*, pages 395–412. Springer, 2010. xi, xvi, 35, 37, 38, 59, 74, 81, 92, 95, 96, 97, 100, 101, 103, 109, 112, 115

[GLM09] Y. H. Gan, C. Ling, and W. H. Mow. Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection. *IEEE Trans. Signal Processing*, 57:2701–2710, 2009. 59

[GLP12] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *Proc. of CHES*, pages 530–547, 2012. viii, xiii, 29, 34, 170

[GMSS99] O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inf. Process. Lett.*, 71(2):55–61, 1999. 10

[GN08] N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *Proc. of EUROCRYPT*, volume 4965 of *LNCS*, pages 31–51. Springer, 2008. ix, xiv, 11

[GOS06] J. Groth, Rafail Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In *Proc. of EUROCRYPT*, volume 4004 of *LNCS*, pages 339–358. Springer, 2006. 95

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proc. of STOC*, pages 197–206. ACM, 2008. viii, ix, xi, xii, xiii, xiv, xvii, 14, 15, 16, 17, 22, 29, 32, 33, 36, 38, 57, 59, 95, 97, 104, 109

[Gro07] J. Groth. Fully anonymous group signatures without random oracles. In *Proc. of ASIACRYPT*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007. xi, 83, 171

[GS08] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Proc. of EUROCRYPT*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008. 83, 95

[GVW13] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *Proc. of STOC*, pages 545–554. ACM, 2013. viii, ix, xii, xiii, xiv, xvii, 29

[HILL99] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999. 6

[HPS98] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *Proc. of ANTS-III*, volume 1423 of *LNCS*, pages 267–288. Springer, 1998. viii, ix, xiii, xiv, xv, 59

[HPS11] G. Hanrot, X. Pujol, and D. Stehlé. Algorithms for the shortest and closest lattice vector problems. In *IWCC*, volume 6639 of *LNCS*, pages 159–190. Springer, 2011. 11

[HR07] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proc. of STOC*, pages 469–477, 2007. 11

[HW09] S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In *Proc. of CRYPTO*, volume 5677 of *LNCS*, pages 654–670. Springer, 2009. 103

[IPWG03] VSC Project IEEE P1556 Working Group. Dedicated short range communications (dsrc), 2003. 83

[Jou00] A. Joux. A one round protocol for tripartite Diffie-Hellman. In *Proc. of ANTS*, volume 1838 of *LNCS*, pages 385–394. Springer, 2000. 135

[KPC+11] E. Kiltz, K. Pietrzak, D. Cash, A. Jain, and D. Venturi. Efficient authentication from hard learning problems. In *Proc. of EUROCRYPT*, volume 6632 of *LNCS*, pages 7–26. Springer, 2011. 59

[KTX08] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Proc. of ASIACRYPT*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008. 85, 121

[KY06] A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. *International Journal of Security and Networks (IJSN)*, 1(1/2):24–45, 2006. 83

[LATV12] A. López-Alt, E. Tromer, and V. Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In *Proc. of STOC*, pages 1219–1234, 2012. 59

[LLL82] A.K. Lenstra, H.W.jun. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982. ix, xiv

[LLLS13] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *Proc. of ASIACRYPT (2)*, volume 8270 of *LNCS*, pages 41–61. Springer, 2013. xi, xvi, 95

[LLNW14] A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-based group signature scheme with verifier-local revocation. In *Proc. of PKC*, volume 8383 of *LNCS*, pages 345–361. Springer, 2014. xi, xvi, 115

[LLS14] F. Laguillaumie, A. Langlois, and D. Stehlé. Chiffrement avancé à partir du problème Learning With Errors. In Sylvain Peyronnet, editor, *Informatique Mathématique une photographie en 2014*, pages 179–225. Presses Universitaires de Perpignan, 2014. xii, xvii, 29

[LM06] V. Lyubashevsky and D. Micciancio. Generalized compact knapsacks are collision resistant. In *Proc. of ICALP (2)*, volume 4052 of *LNCS*, pages 144–155. Springer, 2006. viii, x, xi, xiii, xv, 29, 47, 57, 60, 61, 151, 170

[LM08] V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *Proc. of TCC*, volume 4948 of *LNCS*, pages 37–54. Springer, 2008. viii, xiii, 29, 95

[LMPR08] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen. SWIFFT: A modest proposal for FFT hashing. In *FSE*, volume 5086 of *LNCS*, pages 54–72. Springer, 2008. 170

[LNSW13] S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *Proc. of PKC*, volume 7778 of *LNCS*, pages 107–124. Springer, 2013. 87, 88, 89, 116, 117, 118

[Lov86] L Lovász. *An algorithmic theory of numbers, graphs and convexity*, volume 50 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1986. 16

[LP11] R. Lindner and C. Peikert. Better key sizes (and attacks) for LWE-based encryption. In *CT-RSA*, pages 319–339, 2011. viii, xiii, 29

[LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *Proc. of EUROCRYPT*, LNCS, pages 1–23. Springer, 2010. All result numberings used in the present article correspond to those of the draft of the full version. x, xi, xv, 6, 7, 8, 12, 20, 47, 57, 61, 64, 65, 66, 67, 69, 71, 72, 73, 77, 170

[LPR13] V. Lyubashevsky, C. Peikert, and O. Regev. A toolkit for Ring-LWE cryptography. In *Proc. of EUROCRYPT*, volume 7881 of *LNCS*, pages 35–54. Springer, 2013. 4, 6, 150, 152

[LS] A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. xi, xvi, 43, 57

[LSS14] A. Langlois, D. Stehlé, and R. Steinfeld. Gghlite: More efficient multilinear maps from ideal lattices. In *Proc. of EUROCRYPT*, volume 8441 of *LNCS*, pages 239–256. Springer, 2014. xii, xvii, 135, 149

[LV09] B. Libert and D. Vergnaud. Group signatures with verifier-local revocation and backward unlinkability in the standard model. In *Proc. of CANS*, volume 5888 of *LNCS*, pages 498–517. Springer, 2009. 81, 84

[Lyu08] V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Proc. of PKC*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008. viii, xiii, 29, 87, 97, 116

[Lyu09] V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Proc. of ASIACRYPT*, volume 5912 of *LNCS*, pages 598–616. Springer, 2009. viii, xiii, 29, 95

[Lyu12] V. Lyubashevsky. Lattice signatures without trapdoors. In *Proc. of EUROCRYPT*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012. viii, xiii, 29, 34, 87, 88, 95, 97

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. 1978. Technical report, DSN Progress report 4244, Jet Propulsion Laboratory, Pasadena, Californila. viii

[MG02] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671. Kluwer, 2002. 9

[Mic98] D. Micciancio. The shortest vector in a lattice is hard to approximate to within some constant. In *Proc. of FOCS*, pages 92–98. IEEE Computer Society, 1998. ix, xiv, 11

[Mic02a] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *Proc. of FOCS*, pages 356–365. IEEE, 2002. Conference version of [Mic07]. viii, x, xiii, xv, 29, 169, 182

[Mic02b] D. Micciancio. Improved cryptographic hash functions with worst-case/average-case connection. In *Proc. of STOC*. ACM, 2002. Preliminary version of [Mic04]. 182

[Mic04] D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM*, 34(1):118–169, 2004. Preliminary version in [Mic02b]. 61, 182

[Mic07] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Comput. Complexity*, 16(4):365–411, 2007. Full version of [Mic02a]. xv, 11, 169, 182

[MM11] D. Micciancio and P. Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *CRYPTO*, volume 6841 of *LNCS*, pages 465–484. Springer, 2011. 47

[Mol99] R. A. Mollin. *Algebraic Number Theory*. Chapman and Hall/CRC Press, 1999. 6

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Proc. of EUROCRYPT*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012. 25, 26, 29, 33, 35, 40, 47, 59, 103

[MP13] D. Micciancio and C. Peikert. Hardness of SIS and LWE with small parameters. In *Proc. of CRYPTO (1)*, volume 8042 of *LNCS*, pages 21–39. Springer, 2013. 23, 57

[MR04] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measure. In *Proc. of FOCS*, pages 371–381. IEEE, 2004. Conference version of [MR07]. 17, 22, 61, 63, 183

[MR07] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput*, 37(1):267–302, 2007. Full version of [MR04]. 11, 16, 17, 22, 183

[MR09] D. Micciancio and O. Regev. Lattice-based cryptography. In D.J. Bernstein, J. Buchmann, and E. Dahmen, editors, *Post Quantum Cryptography*, pages 147–191. Springer, 2009. ix, xiv, 11, 79, 170

[MV03] D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Proc. of CRYPTO*, volume 2729 of *LNCS*, pages 282–298. Springer, 2003. 29, 95, 110, 116

[Nap96] H. Napias. A generalization of the LLL-algorithm over Euclidean rings or orders. *J. théorie des nombres de Bordeaux*, 2:387–396, 1996. 59

[NF05] T. Nakanishi and N. Funabiki. Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps. In *Proc. of ASIACRYPT*, volume 3788 of *LNCS*, pages 533–548. Springer, 2005. 81, 84

[NF06] T. Nakanishi and N. Funabiki. A short verifier-local revocation group signature scheme with backward unlinkability. In *IWSEC*, pages 17–32, 2006. 81, 84

[NR06] P. Q. Nguyen and O. Regev. Learning a parallelepiped: Cryptanalysis of ggh and ntru signatures. In *Proc. of EUROCRYPT*, volume 4004 of *LNCS*, pages 271–288. Springer, 2006. Conference version of [NR09]. 183

[NR09] P. Q. Nguyen and O. Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *J. Cryptology*, 22(2):139–160, 2009. Preliminary version in [NR06]. 29, 183

[NS97] P. Q. Nguyen and J. Stern. Merkle-Hellman revisited: A cryptanalysis of the Qu-Vanstone cryptosystem based on group factorizations. In *Proc. of CRYPTO*, volume 1294 of *LNCS*, pages 198–212. Springer, 1997. 9

[NY89] M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *Proc. of STOC*, pages 33–43, 1989. 29

[OPG14] T. Oder, T. Pöppelmann, and T. Güneysu. Beyond ecdsa and rsa: Lattice-based digital signatures on constrained devices. In *Proc. of DAC*, pages 1–6. ACM, 2014. 170

[OPW11] A. O'Neill, C. Peikert, and B. Waters. Bi-deniable public-key encryption. In *Proc. of CRYPTO*, volume 6841 of *LNCS*, pages 525–542. Springer, 2011. 46, 59

[Pat96] J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *Proc. of EUROCRYPT*, volume 1070 of *LNCS*, pages 33–48. Springer, 1996. viii

[PDG14] T. Pöppelmann, L. Ducas, and T. Güneysu. Enhanced lattice-based signatures on reconfigurable hardware. *IACR Cryptology ePrint Archive*, 2014:254, 2014. 170

[Pei08] C. Peikert. Limits on the hardness of lattice problems in ℓ_p norms. *Comput. Complexity*, 2(17):300–351, 2008. 17, 18, 59

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proc. of STOC*, pages 333–342. ACM, 2009. x, xv, 26, 29, 33, 45, 47, 59, 97, 169

[Pei10] C. Peikert. An efficient and parallel gaussian sampler for lattices. In *Proc. of CRYPTO*, volume 6223 of *LNCS*, pages 80–97. Springer, 2010. 14, 20, 74, 76

[PG13] T. Pöppelmann and T. Güneysu. Towards practical lattice-based public-key encryption on reconfigurable hardware. In *Selected Areas in Cryptography*, volume 8282 of *LNCS*, pages 68–85. Springer, 2013. 170

[Pie12] K. Pietrzak. Subspace LWE. In *In Proc. of TCC*, volume 7194 of *LNCS*, pages 548–563. Springer, 2012. 59

[PR06] C. Peikert and A. Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Proc. of TCC*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006. viii, x, xiii, xv, 17, 29, 47, 60, 61, 170

[PR07] C. Peikert and A. Rosen. Lattices that admit logarithmic worst-case to average-case connection factors. In *Proc. of STOC*, pages 478–487. ACM, 2007. 59, 60

[PTT10] C. Papamanthou, R. Tamassia, and N. Triandopoulos. Optimal authenticated data structures with multilinear forms. In *Proc. of Pairing*, pages 246–264, 2010. 135, 137

[PV97] D. Pointcheval and S. Vaudenay. On Provable Security for Digital Signature Algorithms. *Technical Report LIENS-96-17 of the Laboratoire d'Informatique de Ecole Normale Superieure*, 1997. 133

[PV08] C. Peikert and V. Vaikuntanathan. Noninteractive statistical zero-knowledge proofs for lattice problems. In *Proc. of CRYPTO*, volume 5157 of *LNCS*, pages 536–553. Springer, 2008. 85

[PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Proc. of CRYPTO*, volume 5157 of *LNCS*, pages 554–571. Springer, 2008. viii, xiii, 29

[PW08] C. Peikert and B. Waters. Lossy trapdoor functions and their applications. In *Proc. of STOC*, pages 187–196. ACM, 2008. 29

[Ré61] A. Rényi. On measures of entropy and information. In *Proc. of the Fourth Berkeley Symposium on Math. Statistics and Probability*, volume 1, pages 547–561, 1961. 4

[Reg] O. Regev. Lecture notes of *lattices in computer science*, taught at the Computer Science Tel Aviv University. 47

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proc. of STOC*, pages 84–93. ACM, 2005. viii, ix, x, xi, xii, xiii, xiv, xvii, 21, 23, 29, 31, 43, 76, 169, 170, 185

[Reg09] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009. Full version of [Reg05]. x, xiv, 17, 19, 20, 21, 23, 24, 25, 26, 29, 31, 47, 57, 59, 66, 68, 69, 71, 72, 95, 169, 170

[Reg10a] O. Regev. The learning with errors problem, 2010. Invited survey in CCC 2010. 31, 47

[Reg10b] O. Regev. On the complexity of lattice problems with polynomial approximation factors. In Phong Nguyen and Brigitte Vallée, editors, *The LLL Algorithm: Survey and Applications*. Springer-Verlag, New York, 2010. 11

[Rot13] R. Rothblum. On the circular security of bit-encryption. In *Proc. of TCC*, pages 579–598, 2013. 135, 137

[RS09] M. Rückert and D. Schröder. Aggregate and verifiably encrypted signatures from multilinear maps without random oracles. In *Proc. of ISA*, pages 750–759, 2009. 135, 137

[RSA78] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. viii

[Rüc10a] M. Rückert. Adaptively secure identity-based identification from lattices without random oracles. In *Proc. of SCN*, pages 345–362, 2010. 116

[Rüc10b] M. Rückert. Strongly unforgeable signatures and hierarchical identity-based signatures from lattices without random oracles. In *Proc. of PQCrypto*, pages 182–200, 2010. 40

[Sch87] C. P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987. ix, xiv, 11

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. viii, xiii

[SOK00] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. *SCIS*, 2000. 135

[SS11] D. Stehlé and R. Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. In *Proc. of EUROCRYPT*, volume 6632 of *LNCS*, pages 27–47. Springer, 2011. 18, 78, 185

[SS13] D. Stehlé and R. Steinfeld. Making NTRUEncrypt and NTRUSign as secure as standard worst-case problems over ideal lattices. Cryptology ePrint Archive, Report 2013/004, 2013. Full version of [SS11]. ix, xiv, 78, 151, 152, 155

[SSTX09] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *Proc. of ASIACRYPT*, volume 5912 of *LNCS*, pages 617–635. Springer, 2009. x, xv

[Ste96] J. Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996. 121

[vEB81] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, University of Amsterdam, 1981. ix, xiv, 11