Privacy-preserving cryptography from pairings and lattices

Fabrice Mouhartem

Numéro National de Thèse : 2018LYSEN060

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
l'École Normale Supérieure de Lyon

École Doctorale N°512
École Doctorale en Informatique et Mathématiques de Lyon

Spécialité de doctorat : Informatique

Soutenue publiquement le 18/10/2018, par :
Fabrice Mouhartem

Privacy-preserving cryptography from pairings and lattices

Cryptographie protégeant la vie privée à base de couplages et de réseaux

---

Devant le jury composé de :

Catalano Dario, Associate Professor, Università di Catania (Italie)	Rapporteur
Pointcheval David, Directeur de Recherche, CNRS et ENS	Rapporteur
Agrawal Shweta, Assistant Professor, I.I.T. Madras (Inde)	Examinatrice
Fouque Pierre-Alain, Professeur, Université Rennes 1	Examinateur
Gaborit Philippe, Professeur, Université de Limoges	Examinateur
Ràfols Carla, Post-Doctorante, Univestitat Pompeu Fabra (Espagne)	Examinatrice
Libert Benoît, Directeur de Recherche, CNRS et ENS de Lyon	Directeur de thèse

À tonton Patou qui m'a fait découvrir un goût pour la recherche,
Et Betty, Maddie, Aïcha qui ont été là pour m'accompagner dans cette aventure…

Résumé

Dans cette thèse, nous étudions les constructions cryptographiques prouvées pour la protection de la vie privée. Pour cela nous nous sommes intéressés aux preuves et arguments à divulgation nulle de connaissance et leurs applications. Un exemple de ces constructions est la signature de groupe. Ce protocole a pour but de permettre à un utilisateur de s'authentifier comme appartenant à un groupe, sans révéler son identité. Afin que les utilisateurs restent responsables de leurs agissements, une autorité indépendante est capable de lever l'anonymat d'un utilisateur en cas de litige. Une telle construction peut ainsi être utilisée, par exemple, dans les systèmes de transport en commun. Un utilisateur qui rentre dans un bus prouve ainsi son appartenance aux utilisateurs possédant un abonnement valide, sans révéler qui il est, et évitant ainsi que la société de transport ne le trace. En revanche, en cas d'incident sur le réseau, la société peut faire appel à la police pour lever l'anonymat des usagers présents au moment de l'incident. Nous avons proposé deux constructions de ces signatures de groupe, prouvées sûres sous des hypothèses simples dans le monde des couplages et des réseaux euclidiens. Dans la continuité de ces travaux, nous avons aussi proposé la première construction de chiffrement de groupe (l'équivalent de la signature de groupe pour le chiffrement) à base de réseaux euclidiens. Finalement, ces travaux nous ont amenés à la construction d'un schéma de transfert inconscient adaptatif avec contrôle d'accès à base de réseaux euclidiens. Ces constructions à base de réseaux ont été rendues possibles par des améliorations successives de l'expressivité du protocole de Stern, qui reposait initialement sur la difficulté du problème du décodage de syndrome.

Abstract

In this thesis, we study provably secure privacy-preserving cryptographic constructions. We focus on zero-knowledge proofs and their applications. Group signatures are an example of such constructions. This primitive allows users to sign messages on behalf of a group (which they formerly joined), while remaining anonymous inside this group. Additionally, users remain accountable for their actions as another independent authority, a judge, is empowered with a secret information to lift the anonymity of any given signature. This construction has applications in anonymous access control, such as public transportations. Whenever someone enters a public transportation, he signs a timestamp. Doing this proves that he belongs to the group of people with a valid subscription. In case of problem, the transportation company hands the record of suspicious signatures to the police, which is able to un-anonymize them. We propose two constructions of group signatures for dynamically growing groups. The first is based on pairing-related assumptions and is fairly practical. The second construction is proven secure under lattice assumptions for the sake of not putting all eggs in the same basket. Following the same spirit, we also propose two constructions for privacy-preserving cryptography. The first one is a group encryption scheme, which is the encryption analogue of group signatures. Here, the goal is to hide the recipient of a ciphertext who belongs to a group, while proving some properties on the message, like the absence of malwares. The second is an adaptive oblivious transfer protocol, which allows a user to anonymously query an encrypted database, while keeping the unrequested messages hidden. These constructions were made possible through a series of work improving the expressiveness of Stern's protocol, which was originally based on the syndrome decoding problem.

Remerciements/Acknowledgements

Je tiens tout d'abord à remercier mon directeur de thèse Benoît Libert, qui m'a accompagné durant toute la durée de ma thèse et m'a gratifié de sa présence et de sa disponibilité tout au long de cette aventure. J'ai particulièrement apprécié sa rigueur critique et son esprit scientifique. Je le remercie aussi de son support lors des phases de soumissions, des possibles *rebuttals* qui s'en suivaient et lors des présentations en conférences. Ses enseignements et ses conseils m'ont aidé à avoir une vision plus claire de la recherche et me permettront désormais de prendre mon envol dans le vaste monde.

I would also like to thank Dario Catalano and David Pointcheval, who accepted to review this thesis during the summer and granted me with their useful remarks. I also thank the rest of the committee: Shweta Agrawal, who also invited me for a visit in the nice city of Chennai and taught me to be cautious with monkeys; Pierre-Alain Fouque, who also accepted to be part of my half-way thesis committee; Philippe Gaborit and our discussions at *La-Londe-lès-Maures*; and Carla Ràfols, who I met in several conferences and for the organization of the insightful Cost-Iacr School on Randomness in Cryptography.

I also want to thank my coauthors for their ideas and our useful discussions: Benoît Libert, San Ling, Khoa Nguyen, Thomas Peters, Huaxiong Wang and Moti Yung. Without you, my research would have be less productive!

Ma famille a aussi tenu une place importante jusque l'acheminement de cette thèse et durant son déroulement : Maman, Maddie, Aïcha, Hachim, Patrick, Solange, Sylvain, Lilye, Mamie, James, Fabio, Rudy, Joanna.

For inviting me and their warm reception, I would like to express my gratitude to Shweta Agrawal from IIT Madras; Adeline Langlois and Pierre-Alain Fouque from Rennes; Satya Lokam from Microsoft Bangalore; Abderrahmane Nitaj and Brigitte Vallée from Caen; and Ali El Kaafarani from Oxford. It was the opportunity for me to discover other research environments and our interesting discussions.

Without having to move so far, I also want to thank the AriC team in Lyon and especially the crypto part with its dynamic that is very supportive to research: Damien, Benoît, Fabien, Alexandre, Alice, Alice, Alonso, Benjamin, Chen, Chitchanok, Elena, Gottfried, Ida, Jiangtao, Jie, Junqing, Laurent, Marie, Miruna, Radu, Sanjay, Shi, Somindu and Weiqiang. I especially want to thank to Damien Stehlé, who introduced me to cryptography with his course in master, and helped me going further in this topic by introducing me to Fré

Vercauteren from KU Leuven who was my first contact with research in cryptography. Where the boundaries are blurry, I would like to thank Guillaume Hanrot, who was there to guide me in solving the administrative conundrum and also for showing his interest for my research. From the rest of the permanent members of AriC team, I want to thank Nicolas, Claude-Pierre, Vincent, Nicolas, Jean-Michel, Nathalie, Bruno and Gilles. And the ephemeral members: Anastasia, Valentina, Laurent, Paula, Silviu, Sébastien, Steve, Vincent, Antoine, Serge, Florent (who is not fully AriC). I also want to thank all my officemates, who kindly accepts to share an office with me: Valentina, Vincent, Jingming, Laurent, Radu and Chitchanok. And the other people from the LIP laboratory with whom I had the chance to discuss: Michaël, Éric, Stephan, Nathalie, Natacha, Nicolas, Sebastián, Matthieu, Timothée, Nam, Étienne, William, Alexandre, Pierre C., Patrick, Denis, Colin, Damien P., Russ, Valeria, Simon C., Aurore, Ievgeniia, Marc, Pierre P., Laureline, Christian, Marie, Chiraz, Kadiatou, Évelyne, Catherine, Nelly, Joris, Dominique, Simon D., Jean-Christophe, Loris, Bora, Anne, Yves, Jean-Yves, Frédéric, Loïc, Gilles, Bertrand, Li, Valentin, Changjiang, Daniel, Issam, Radu, Hélène, Semen, Jérôme, Pédro, Violaine, Vincent L., Arnaud, Alexandre, Marcos, David, Christophe, Laure, Matthieu, and everyone I may have forgotten. It was my pleasure to discuss coffee with you!

I also want to thanks the young researchers I met during conferences and gdr-im meetings, who help me understanding the social aspect of research, especially during the “journées C2”, but not only: Thomas Prest, Pierre Karpman, Thomas Peters, Pierrick Méaux, Yann Rotella, Alain Passelègue, Ilaria Chillotti, Dahmun Goudarzi, Geoffroy Couteau, Fabrice Ben Hammouda--Guichoux, Florian Bourse, Pauline Bert, Baptiste Lambin, Clothilde Jean-goudoux, Claire Delaplace, Cécile Pierrot, Léo Ducas…

Je tenais aussi à remercier mes amis, qui ont partagé aussi partagé l’expérience de la thèse : Valentin Gledel, François Pirot, bientôt Baptiste Rozière, Camilo Broc, peut-être Hadrien Titeux, Lucien Derainne, Thomas Bertin, Armael Guéneau, quasi-Antoine Pouille, Henri Derycke, Marion Blancher.

Mais aussi les autres, rencontrés aux hasards de la vie : Cécile Bayard, Anne Croizier, Émilie Lapinsonnière, Antony Santos, Bérénice Nguyen, Tito, Hippolyte Catherin, Sylvia Iliescu, Arthur Ouaki, Clément Deslandes, Élodie Aoustin, Emma Chantron, Nathalène, Emmanuel Canaple et sa petite famille (Raïssa et Jana), Grégory Sénan Dossa, Thomas Gérard, Cendrine Dallet, Tibo, Annie & Camille, Antoine Mouche, Benoît Delboven, Adriana Teoli, Pierre-Adrien Collet, Marie-Morgane Paumard, Grégoire, Pilou Alzieu, Max, Marion Balanza, Aliette Cheptitski, et tous ceux que j’ai oublié.

Sans oublier le M.F.P.P.[1] que j’ai découvert peu avant ma thèse et qui m’a accueilli les bras ouverts. Pour toutes les rencontres que j’ai pu y faire avec des personnes d’horizons très différents : Aurèle, Jean-Jacques, François, Patrice, Raymonde, Valérie, Valérie, Valérie, Sandry, Claudine, Pitof, Tudor, Claudine, Solveig, Éric, Naomiki, Chantal, Michel, Martine, Christiane, Andrée, André, Marie-Noelle, Robert, Viviane…

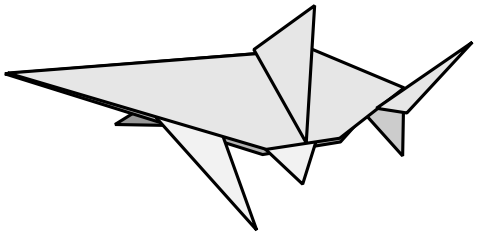
Mais l’origami ne s’arrête pas là, et pour ces autres origamiste : Pierre, Nicolas, Blandine, Vincent, Charlotte, Valentin, Marine Minier, Laetitia Rodet, Josiane, Marie-Odile…

Enfin un dernier merci à tous ces profs de maths que j’ai eu de la sixième à la spé, qui ont sû me donner goût à cette matière, à leurs manières très variées.

[1] Mouvement Français des Plieurs de Papier

Et enfin ceux que je ne connais pas mais qui m'ont aidé au quotidien, directement ou indirectement : ergodis et la communauté bépo pour la disposition du clavier, les développeurs de GNU/Linux et tout ce qui l'entoure, les aficionados des tuppervims, LaTeX et la chasse aux overfull hboxes, les touches "`hjkl`" pour se déplacer, les gestionnaires de fenêtres par pavage (feu wmii et i3), la communauté wikipédia et ses dérives, l'orthographe d'avant 1990 et mes fautes de frappes, les travailleurs des champs de café, les vignerons indépendants et tous les cochons sacrifiés pour mon saucisson hebdomadaire, les gens qui utilisent encore irc en 2018, git et j'en oublie sûrement beaucoup derrière toutes mes vilaines habitudes.

Et enfin un grand merci à la reprographie de l'É.N.S. de Lyon qui a accepté d'imprimer ce manuscrit le jour même, malgré ma tendance à suivre la méthode la Rache.



Contents

List of Symbols

General Notations

TM	Turing Machine
PPT	Probabilistic Polynomial Time
ϵ	empty word
$\mathbf{A}$	bold uppercase letters represent matrices
$\mathbf{b}$	bold lowercase letters represent column vectors
$\widetilde{\mathbf{A}}$	Gram-Schmidt orthogonalization of matrix $\mathbf{A}$
$\mathbf{A}^T, \mathbf{u}^T$	the transpose of a matrix or a vector respectively
$\mathbf{I}_n$	the n dimension identity matrix in $\mathbb{R}^{n\times n}$
$\mathcal{U}(S)$	If S is a finite set, $\mathcal{U}(S)$ denotes the uniform distribution over S
$\mathrm{Supp}(D)$	If D is a probability distribution, $\mathrm{Supp}(D)$ denotes the support of D
$\Pr[E]$	Probability that an event E occurs
$D \approx_s D'$	The distribution D is statistically close to the distribution D'

Usual sets

$\mathbb{Q}$	the set of rational numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{Z}$	the set of relative integers
$\mathbb{Z}_q$	the field $\mathbb{Z}_{/q\mathbb{Z}}$, with q prime
$\mathbb{F}_2$	the field $\mathbb{Z}_{/2\mathbb{Z}}$
$\mathbb{S}^d$	the set of vectors of dimension d in the set $\mathbb{S}$
$\mathbb{S}^{n\times m}$	the set of matrices with n rows and m columns in the set $\mathbb{S}$
$\mathfrak{S}_k$	The set of all permutations over $\{1,\ldots,k\}$

Protocols

PKE	Public Key Encryption
ZK	Zero-Knowledge
ZKAoK	Zero-Knowledge Argument of Knowledge
NIZK	Non-Interactive Zero-Knowledge
QA-NIZK	Quasi-Adaptive Non-Interactive Zero-Knowledge
WI	Witness Indistinguishable
GS	Group Signature
GE	Group Encryption
OT	Oblivious Transfer

Security Notions

$\mathrm{Adv}^{\mathrm{E}}_{\mathcal{A}}$	Advantage of adversary $\mathcal{A}$ for experiment E
EU-CMA	Existentially Unforgeable under chosen-message attacks
EU-RMA	Existentially Unforgeable under random-message attacks
IND-CPA	Indistinguishable under chosen-plaintext attacks (passive adversary)
IND-CCA1	Indistinguishable under non-adaptive active adversary
IND-CCA2	Indistinguishable under adaptive active adversary

Security Models

ROM	Random-Oracle Model
UC	Universal Composability

Security Assumptions

Lattices

SIS	Short Integer Solution (Definition 3.8)
ISIS	Inhomogeneous Short Integer Solution (Definition 3.8)
LWE	Learning-with-Errors (Definition 3.9)
SIVP_{γ}	Shortest Independent Vectors Problem (Definition 3.7)

Cyclic groups

DLP	Discrete Logarithm Problem (Definition 2.9)
DDH	Decisional Diffie-Hellman (Definition 2.11)

Bilinear groups

SXDH	Symmetric eXternal Diffie-Hellman (Definition 3.2)
SDL	Symmetric Discrete Logarithm (Definition 3.3)

Stern-like protocol

$\mathsf{B}^2_{\mathfrak{m}}$	The set of $\{0,1\}$ vector of hamming weight $\mathfrak{m}$
$\mathsf{B}^3_{\mathfrak{m}}$	The set of $\{-1,0,1\}$ vectors with $\mathfrak{m}$ elements in -1, 0 and 1

Résumé substantiel en Français

Les cinquante dernière années, l'utilisation de la cryptographie s'est éloignée de ses origines militaires et de son usage pour le secret commercial afin de se démocratiser à un public plus large. Par exemple, la machine Enigma initialement conçue pour un usage militaire a été déclinée pour un usage commercial (la machine Enigma A26). Aujourd'hui, environ 60% du premier million des sites les plus visités dans le monde propose une connexion chiffrée et authentifiée (à l'aide du protocole `https`), tout comme les canaux de communication des appareils électroniques portatifs (comme la norme `WPA`, en anglais *Wifi Protected Access*).

Dans le même temps, la croissance des données échangée en ligne et la sensibilité de ces informations rendent de plus en plus urgent la protection de ces communications. Pendant que la loi de Moore[2] atteint ses limites, d'autres menaces existent sur nos cryptosystèmes actuels. Par exemple, l'existence d'un ordinateur quantique possédant suffisamment de mémoire [Sho99] rendrait risqué l'utilisation de la majorité des constructions cryptographiques actuellement déployées, puisqu'elles reposent sur des hypothèses issues de l'arithmétique modulaire classique dont la structure algébrique peut-être exploitée par un adversaire quantique. Dans cette situation, il devient alors crucial de construire des schémas cryptographiques qui résisteraient à une menace quantique.

Pour répondre à ce problème, la *cryptographie post-quantique* est née au début des années 2000. Les différents candidats reposent sur différents objets mathématiques, comme les réseaux euclidiens, les codes, les systèmes polynomiaux multivariés, les isognénies, etc. Récemment, le NIST (*National Institude of Standards and Technology*) a organisé une compétition pour évaluer les différentes solutions post-quantiques pour le chiffrement et la signature [NIS17]. Dans cette compétition, 82 protocoles ont été proposés, parmi lesquels 28 reposent sur les réseaux euclidiens, 24 sur les codes correcteurs, 13 sur des systèmes multi-variés, 4 sur des fonctions de hachages et 13 sur d'autres objets.

Si la cryptographie pratique vise principalement à fournir des schémas de signature et de chiffrement, comme l'atteste la compétition du NIST, la recherche théorique propose des solutions à des problèmes plus précis, comme la construction de systèmes de monnaie électronique[3] [CFN88], qui sont l'équivalent numérique de notre monnaie échangée. Les pièces sont délivrées par une autorité centrale (la banque), et les dépenses restent intraçables. En cas de comportement malhonnête (comme une double-dépense), l'identité de l'utilisateur

[2] La loi qui prédit la puissance de calcul des processeurs modernes.

[3] À ne pas confondre avec les cryptomonnaies…

malicieux est révélée.

Les constructions cryptographiques doivent en plus vérifier des propriétés de sécurités. Par exemple, un schéma de chiffrement doit être en mesure cacher un message en présence d'un attaquant passif voire actif (c'est-à-dire un attaquant qui est capable de modifier certains messages). Pour garantir ces exigences, les cryptographes fournissent des preuves de sécurité dans le cadre de modèles de sécurité précis. Une preuve de sécurité nous dit principalement qu'une construction cryptographique est au moins aussi difficile qu'un problème supposé difficile par la littérature.

Finalement, l'importance de la préservation de la vie privé et la protection des données ont été des sujets qui ont fait couler beaucoup d'encres, comme en atteste le développement du règlement général sur la protection des données (RGPD) en 2016, mis en application ce 25 mai. Il est donc intéressant pour les cryptographes de fournir des solutions qui resisteraient, dans le meilleurs des mondes, à un adversaire quantique. Néanmoins, la construction de ces protocoles repose de manière décisive sur les « preuves à divulgation nulles de connaissances ». Ce sont des protocoles interactifs entre un prouveur et un vérifieur où le prouveur cherche à convaincre le vérifieur d'une affirmation sans rien divulguer de plus sur celle-ci que sa valeur de vérité. Dans le contexte de la cryptographie post-quantique, de tels systèmes de preuves sont limités en terme d'expressivité ou en terme de coût de calcul (en temps ou en mémoire).

Cryptographie préservant la vie privée

Dans ce contexte, la « préservation de la vie privée » décrit la capacité d'une primitive cryptographique à fournir certaines fonctionnalités tout en gardant certaines informations privées. Par exemple, l'accréditation anonyme [Cha85, CL01] est un exemple d'une telles primitives. De manière informelle, cette primitive permet à un utilisateur de garantir ses droits d'accès à un vérifieur, sans lui divulguer son identité, ni le motif de ses accès. Pour réaliser cette primitive, le système est composé d'un (ou plusieurs) fournisseur(s) d'accréditations ainsi que d'un ensemble d'utilisateurs qui possèdent leurs propres clefs secrètes ainsi qu'un ensemble d'attributs. Les utilisateurs peuvent obtenir ces accréditations dynamiquement depuis un fournisseur qui ne connaît d'eux qu'un pseudonyme et qui signe de manière inconsciente les clefs secrètes des utilisateurs ainsi que leur attributs. Après avoir obtenu leurs accréditations, les utilisateurs peuvent s'authentifier sous des pseudonymes différents, et prouver la possession d'une certification de la part du/d'un fournisseur, sans divulguer ni la signature, ni la clef secrète. Cette primitive permet ainsi à un utilisateur de s'identifier vis-à-vis du système (par exemple dans le cadre d'un contrôle d'accès anonyme) tout en préservant l'anonymat des utilisateurs. De plus, ce système garantit que les utilisateurs possèdent un droit d'accès suffisant.

L'intérêt pour la cryptographie préservant la vie privée est contemporain de la naissance de la cryptographie à clef publique [Rab81, Cha82, GM82, Cha85]. Une raison pouvant expliquer cela serait la similitude entre les motivations de la cryptographie et les exigences de la protection de la vie privée. De plus, le travail des cryptographes dans ce domaine ont des conséquences directes en terme de services qui pourraient être développés dans le monde réel. En effet, un système d'accréditation anonyme pourrait débloquer le contrôle d'accès anonyme et limiter ainsi le risque de brèches de sécurité. Alors que, actuellement, les systèmes mis en places reposent sur des briques de bases plus élémentaires, comme des

signatures, dont la manipulation peut amener différents problèmes de sécurité [VP17].

De manière similaire, les *primitives avancées* sont construites à partir de briques de bases simples. La principale différence réside en le fait que leur sécurité a été prouvée, ce qui offre une confiance plus forte dans la sécurité de ces constructions. Comme expliqué précédemment, ces preuves permettent de lier la sécurité des systèmes à des hypothèses de sécurité. Ainsi, la sécurité repose sur la validité de ces hypothèses, qui peuvent être étudiées indépendamment des systèmes cryptographiques par les *cryptanalystes*. Dans ce cas contraire, cela peut mener à différents problèmes, comme dans le cas de applications multilinéaires, dont plusieurs candidats ont été rendus caduques par [CHL+15]. Cet exemple montre ainsi l'importance pour la cryptographie de reposer sur des hypothèses simples et robustes comme nous l'expliqueront dans le chapitre 2.

Les schémas développés dans cette thèse reposent sur des hypothèses de réseaux euclidiens et des applications bilinéaires sur les groupes cycliques (ou couplages). La cryptographie à base de réseaux euclidiens est utilisées pour aller d'avant vers la cryptographie post-quantique, tandis que les applications bilinéaires sont utiles pour la réalisation de schémas pratiques. Les détails de ces deux structures sont présentés dans le chapitre 3.

Preuves sans divulgation de connaissance

Comme nous l'avons expliqué précédemment, les preuves sans divulgation de connaissance sont une brique de base pour la cryptographie préservant la vie privée. Elles exigent les propriétés de complétude, de robustesse et de non divulgation de connaissance. La complétude exprime la correction du protocole dans le cas où tout le monde agit honnêtement. Dans le cas d'un prouveur malhonnête, la robustesse demande à ce que le vérifieur ne peut être convaincu qu'avec une probabilité négligeable. Au contraire, si le vérifieur essaye de tricher, la non divulgation de connaissance assure au prouveur que son secret reste protégé.

Dans le cadre de l'identification, la nature du secret reste simple, et des solutions qui reposent sur différentes hypothèses de sécurité existent déjà [Sch96, Ste96, KTX08, Lyu08]. Pour des affirmations plus complexes, comme prouver l'exécution correcte d'un calcul, il existe un écart entre ce qu'on peut prouver dans le cadre de la cryptographie post-quantique et celle qui repose sur le l'arithmétique modulaire classique. Dans le cadre des couplages, il existe des preuves sans divulgation de connaissance non interactives qui permettent de prouver une large classe d'affirmation [GOS06, GS08] sans idéaliser le modèle de sécurité. De telles preuves sont, à ce jour, manquantes dans le cadre de la cryptographie post-quantique.

Pour les réseaux euclidiens, il y a principalement deux familles de preuves : les preuves à la Schnorr [Sch96, Lyu09] et les preuves à la Stern [Ste96], nommés d'après leurs découvreurs respectifs. Les preuves à la Schnorr reposent sur des réseaux structurés en exploitant cette structure pour fournir des preuves compactes au détriment de l'expressivité de ces preuves. Au contraire, les preuves à la Stern reposent sur la combinatoire de la représentation des réseaux généraux en tant que matrices et vecteurs. Par nature, ces preuves sont coûteuses en terme de complexité de communication. En revanche, elles sont suffisamment versatiles pour prouver une large variété d'affirmations, comme nous l'expliqueront plus en détail dans cette thèse, et plus particulièrement dans le chapitre 4.3. Les preuves à divulgation nulle de connaissance sont détaillées dans le chapitre 4.

Signatures avec protocoles efficaces

Pour rendre possible la cryptographie à base de réseaux euclidiens, une approche possible est de coupler les preuves sans divulgation de connaissance avec un schéma de signature. De telles signatures sont dites « avec protocoles efficaces ». Cette primitive étend la fonctionnalité des signatures traditionnelles de deux manière : (i) Elle permet au signataire de signer oublieusement un message caché et (ii) Les utilisateurs peuvent prouver sans divulgation la connaissance d'un couple message-signature cachée.

Ces deux propriétés s'avèrent être extrêmement utiles dans la construction de protocoles préservant l'anonymat tels que l'accréditation anonyme ou la monnaie électronique. La construction effective de ces signatures avec protocoles efficaces est donc un point clef dans la cryptographie protégeant la vie privée.

Dans cette thèse, nous proposons deux constructions de telles signatures. Une première, décrite dans le chapitre 6, repose sur les couplages. Il s'agit d'une traduction du schéma de [LPY15] dans un modèle idéalisé pour obtenir une efficacité et une sécurité raisonnables en pratique. La seconde, décrite dans le chapitre 7, adapte une variante de la signature de Boyen [Boy10, BHJ+15] à la mise en gage de Kawachi-Tanaka-Xagawa [KTX08] pour proposer un schéma de signature à base de réseaux euclidiens compatible avec les preuves à la Stern. Ce schéma a aussi été relaxé dans le cadre du transfert inconscient, où, par endroits, seule la sécurité pour des messages aléatoires est requise. Cela est décrit dans le chapitre 9.

Couplages et réseaux euclidiens

Les constructions proposées dans cette thèse reposent sur la difficulté supposée d'hypothèses sur les groupes compatibles avec des couplages et les réseaux euclidiens. Ces deux objets mathématiques ont été étudiés en long et en large depuis leurs introductions respectives au début des années 2000 [SOK00, Reg05]. Depuis lors, ils ont bénéficié d'une attention considérable de la part des cryptographes, ce qui a mené à la conception de nombreux protocoles avancés (tels que [Jou00, BBS04, BN06, GS08, LYJP14, LPQ17] pour les couplages et [GPV08, ABB10, BV11, GSW13, dPLNS17] pour les réseaux).

Cryptographie à base de couplages

Un couplage est une application bilinéaire qui part de deux groupes cycliques sources vers un groupe cible. La bilinéarité est rendue possible grâce à la structure algébrique forte des groupes abéliens compatibles avec de telles applications. Il n'est donc pas surprenant de voir qu'une large gamme de schémas ont été rendus possibles dans le contexte de la cryptographie à base de couplages. Dans le cadre de la cryptographie pour la protection de la vie privée, la pierre angulaire a été l'introduction des preuves de Groth-Sahai [GOS06, GS08] qui permettent de prouver de manière non interactive et sans divulgation de connaissance une large classe d'affirmations dans le modèle standard (c'est-à-dire sans présupposés). Par exemple, les preuves de Groth-Sahai ont été utilisées pour la construction de signatures de groupe, d'accréditations anonymes [Gro07, BCKL08, BCC+09], ou encore d'e-cash [BCKL09].

Cependant, dans cette thèse, nos constructions à base de couplages visent à la praticité. Ainsi, elles sont instanciées dans le modèle de l'oracle aléatoire, où les preuves à la Schnorr

sont rendues non interactives par la transformée de Fiat-Shamir lorsque l'affirmation à prouver est suffisamment simple.

Des travaux récents en cryptanalyse des couplages [KB16, MSS17, BD18] ont mené à des changements radicaux dans le panorama de la cryptographie à base de couplage. Ces changements nous affectent en ce sens que nos évaluations de paramètres de sécurité sont désormais à revoir pour atteindre le même niveau de sécurité que celui annoncé.

Néanmoins, la cryptographie à base de couplages offre un excellent compromis entre ce qu'on est capable d'y construire et l'efficacité. Comme exemple, nous pouvons citer les travaux de Döttling et Garg [DG17a] qui ont clos le problème de fournir un chiffrement fondé sur l'identité qui repose uniquement sur des hypothèses sur les groupes cycliques. Si leur construction repose sur des objets mathématiques plus simples, elle n'atteint néanmoins pas l'efficacité de celles que proposent la cryptographie à base de couplages [BB04].

Cryptographie reposant sur les réseaux euclidiens

D'un point de vue algébrique, un réseau est un sous-groupe discret d'un certain $\mathbb{R}^n$, ce qui mène à une structure de simple groupe additif. La principale différence avec la cryptographie traditionnelle reposant sur de l'arithmétique modulaire, telle que celle fondée sur le logarithme discret, est l'existence d'une structure géométrique : celle de réseau. C'est de cette géométrie que proviennent les problèmes difficiles sur les réseaux que nous considérons comme difficiles, et ce, même avec la puissance d'un calculateur quantique. Malgré sa structure apparente simple, les réseaux euclidiens ont permis de débloquer des constructions qui ne sont par réalisables autrement, comme par exemple le chiffrement totalement homomorphe [Gen09, BV11, GSW13].

La flexibilité de la cryptographie à base de réseaux euclidiens a été rendue possible par la découverte des *portes dérobées* [GPV08, CHKP10, MP12] telles que nous l'avons expliqué en section 3.2.2. De manière informelle, la connaissance d'une base courte (ou porte dérobée) pour un réseau permet d'échantillonner des vecteurs courts dans ce même réseau. Il s'avère, par ailleurs, que trouver ces vecteurs courts est considéré comme un problème difficile en l'absence de la connaissance d'une base courte. De plus, avoir une base courte pour un réseau $\{\mathbf{v} \in \mathbb{Z}^m \mid \mathbf{A}\mathbf{z} = 0 \bmod q\}$ décrit par une matrice $\mathbf{A} \in \mathbb{Z}^{n \times m}$ permet de générer une base pour un réseau dépendant de celui-ci, décrit par la matrice $[\mathbf{A} \mid \mathbf{B}] \in \mathbb{Z}_q^{n \times m'}$. Cette propriété est utilisée par exemple dans la signature de Boyen [Boy10], largement employée dans la cryptographie à base de réseaux. Dans ce schéma, une signature pour un message m consiste en un vecteur court dans le réseau orthogonal à la matrice $\mathbf{A}_m = [\mathbf{A} \mid \mathbf{B}_m]$ où la matrice $\mathbf{B}_m$ est calculable publiquement. Ainsi, la connaissance d'une porte dérobée pour la matrice $\mathbf{A}$ rend possible le calcul d'un tel vecteur, et le message est lié à la description de la matrice $\mathbf{A}_m$.

Néanmoins, l'utilisation de portes dérobées pour les réseaux est extrêmement coûteuse en terme d'efficacité [Lyu12, LLNW16]. Comme notre but est de fournir les premières constructions pour les primitives que nous présentons, nous nous sommes concentrés sur la réalisations de telles constructions sous des hypothèses bien étudiées, au prix de cette efficacité.

Nos résultats

Dans cette thèse sont présentées différentes constructions cryptographiques pour la préservation de la vie privée. Ces constructions sont le résultat d'améliorations successives des preuves à divulgation nulle de connaissance et des preuves de sécurités liées aux constructions sous des hypothèses calculatoires standards. Nous pensons que ces avancées ont un intérêt indépendant, et que les schémas proposés sont un pas de plus vers la démocratisation d'une cryptographie qui résisterait à un adversaire quantique. Dans la suite, nous détaillons quatre constructions qui ont été développées dans cette thèse. Ces résultats sont issues de ces quatre articles publiés durant ma thèse : [LMPY16, LLM+16a, LLM+16b, LLM+17].

Signatures de groupe dynamique et accréditation anonyme

Dans la partie II nous présentons deux primitives : les signatures de groupes dynamiques et l'accréditation anonyme. Nous avons déjà décrit le comportement de l'accréditation anonyme plus haut. Pour les signatures de groupes, il s'agit d'une primitive qui permet à un utilisateur d'authentifier un message au nom d'un groupe, tout en restant anonyme au sein de ce groupe. Les utilisateurs restent responsables de leurs actions : une autorité tierce (par exemple un juge) disposant d'une information secrète est capable de lever l'anonymat des utilisateurs qui se comporteraient mal.

En tant que telle, cette primitive peut être utilisée pour fournir une authentification anonyme qui garantit la responsabilité de ses utilisateurs (ce qui n'est pas le cas avec l'accréditation anonyme). Par exemple, dans l'internet des objets, comme les voitures intelligentes, il est important de fournir un canal de communication authentifié, alors que l'anonymat de chaque objet est important (puisqu'il possède beaucoup d'information sur son utilisateur).

Dans cette thèse, nous présentons dans le chapitre [?] un schéma de signature de groupe à base de couplages qui vise l'efficacité sous des hypothèses raisonnables. Cette construction est accompagnée d'une implantation en C pour soutenir sa practicité. Le schéma est décrit dans [LMPY16] conjointement avec Benoît Libert, Thomas Peters et Moti Yung, et a été présenté à la conférence AsiaCCS'16.

Le chapitre 7 présente le premier schéma de signature de groupe dynamique qui repose sur la sécurité des réseaux euclidiens. Ces travaux sont décrits dans [LLM+16a] avec Benoît Libert, San Ling, Khoa Nguyen et Huaxiong Wang, et ont été présentés à Asiacrypt'16.

Chiffrement de groupe

Le chiffrement de groupe est l'analogue de la signature de groupe pour le chiffrement. Dans ce contexte, un utilisateur désire envoyer un message à un membre d'un groupe, tout en cachant l'identité du destinataire au sein de ce groupe. De manière similaire, une autorité peut lever l'anonymat des message à l'aide d'une information secrète [KTY07, LYJP14].

Une application possible du chiffrement de groupe est la construction d'un pare-feu d'entreprise, qui permet de garantir qu'un message possède bien un destinataire dans l'entreprise tout en garantissant des propriétés additionnelles comme l'absence de programme malicieux dans le message. En cas de doute, une autorité est capable de lever l'anonymat d'un message suspicieux.

Plus formellement, le chiffrement de groupe est une primitive qui permet à un expéditeur

de générer une preuve publiquement vérifiable que : (1) Le chiffré est bien formé et est destiné à un utilisateur enregistré dans un groupe qui sera capable de déchiffrer le message ; (2) L'autorité d'ouverture sera capable d'identifier le destinataire du message si besoin ; (3) Le message clair vérifie certaines propriétés, comme celle d'être un témoin pour une relation publique. Dans le modèle de Kiayias, Tsiounis et Yung [KTY07], le secret du message et l'anonymat sont définis pour un adversaire actif dans toutes les définitions de sécurité. C'est-à-dire un adversaire capable de demander des requêtes d'ouverture sur n'importe quel message et l'identité du destinataire de messages.

Cette construction nécessite de pouvoir prouver la connaissance d'une clef publique certifiée utilisée pour chiffrer un message. Or, cette clef publique doit rester cachée parmi les différentes clef publiques certifiées. Dans les réseaux euclidiens, cela revient à produire une preuve sans divulgation nulle de connaissance pour une relation dite « quadratique ». Avant ces travaux, les preuves sans divulgation de connaissances que nous connaissons sur les réseaux euclidiens ne permettent que de prouver des relations où le témoin vérifie une relation linéaire en les paramètres publiques. Rappelons qu'une relation d'apprentissage avec erreurs est de la forme $\mathbf{A}\cdot\mathbf{s}+\mathbf{e}+\mathbf{m}\lceil\frac{q}{2}\rceil \bmod q$ où $\mathbf{A}$ est la clef publique du destinataire du message. Comme le chiffrement de groupe demande à ce que le destinataire reste anonyme, cette clef publique $\mathbf{A}$ doit rester privée. Un moyen d'y arriver est d'avoir des preuves sans divulgation nulle de connaissance qui supportent les relations où le témoin est multiplié par une matrice privée.

Cela a été rendu possible en introduisant des techniques nouvelles pour les preuves à la Stern. Ces techniques, qui reposent sur une approche « diviser-pour-régner », sont décrites dans le chapitre 8, ainsi que la construction du schéma de chiffrement de groupe prouvé sûr dans le modèle standard. Ces travaux ont été présentés à Asiacrypt'16 [LLM+16b] et ont été effectués avec Benoît Libert, San Ling, Khoa Nguyen et Huaxiong Wang.

Transfert inconscient adaptatif

Le transfert inconscient est une primitive proposée par Rabin [Rab81] qui a ensuite été étendue par Even, Goldreich et Lempel [EGL85]. Elle met en relation un serveur et un client qui veulent s'échanger des messages indexés de 1 à N. Le protocole permet ainsi à un client d'obtenir le ρ-ième message de la part du serveur sans lui permettre de savoir quoi que ce soit sur le choix du client. De plus, le client n'obtient que le ρ-ième message et n'apprend rien sur les autres messages de la base de donnée.

Dans sa version adaptative [NP99], le client souhaite obtenir *dynamiquement* k messages. Toujours en gardant secret l'indice des messages qu'il a récupéré, ainsi que le motif d'accès des requêtes.

D'un point de vue théorique, le transfert inconscient est une *brique de base complète* pour la cryptographie. Autrement dit, si elle est possible, n'importe quel calcul multipartite sécurisé est rendu possible. Dans sa variante adaptative, le transfert inconscient a des applications pour l'accès préservant la vie privée à des bases de données sensibles (comme les bases de données médicales ou financières) stockées de manière chiffrées sur un serveur distant.

Dans sa forme simple, le transfert inconscient (adaptatif) ne restreint pas l'accès aux données. Dans plusieurs cas de figures (comme des bases de génomes), on souhaite mutualiser la base de données, et il n'est pas souhaitable que n'importe quel utilisateur puisse avoir accès à toutes les données. Il est alors important de protéger l'accès à ces données sensibles

conditionnés par les droits d'accès du client. Dans le même temps, la protection de la vie privée nécessite que seuls les utilisateurs autorisés puissent accéder à la base de donnée, tout en gardant leurs informations anonymes (en particulier, le certificat utilisé pour accéder aux données doit rester secret pour tout le monde, y compris le serveur).

Cette propriété a été formalisée par [CDN09] par la notion de contrôle d'accès. Dans cette variante, chaque données est protégée par une police d'accès différente. À partir de leurs attributs, les utilisateurs peuvent obtenir une accréditation de la part d'une autorité tierce, qui les autorise à récupérer anonymement les données pour lesquelles leurs attributs leurs en permettent l'accès. Durant la phase de transfert, le client prouve sans divulgation de connaissance la possession d'un tel certificat pour un attribut qui vérifie la police d'accès de la donnée qu'il souhaite obtenir. La seule information que la base de donnée peut apprendre est donc qu'un utilisateur a obtenu un élément de la base de données pour lequel il avait accès.

Le système doit être capable de traiter des polices d'accès complexes, tout en gardant la complexité en temps et en mémoire raisonnable. Dans cette thèse, nous proposons dans le chapitre 9 un protocole de preuve sans divulgation de connaissance qui permet de traiter n'importes quelles polices d'accès pouvant être décrites à l'aide d'un circuit booléen de profondeur logarithmique (c'est-à-dire la classe de complexité NC^1) qui repose sur les réseaux euclidiens. Dans le cadre du transfert inconscient adaptatif, la plupart des constructions (à base de couplages) ne permettent que de traiter, sous des hypothèses raisonnables, que des polices d'accès composées de conjonctions [CDN09, CDNZ11, ACDN13]. Sous des hypothèses plus fortes, les polices d'accès dans NC^1 sont néanmoins possibles [ZAW+10].

Cette construction, présentée à Asiacrypt'17 [LLM+17], a été réalisée avec Benoît Libert, San Ling, Khoa Nguyen et Huaxiong Wang.

Chapter 1

Introduction

In the last fifty years, the use of cryptography has shifted from military and commercial secrets to a broader public. For instance, the Enigma machine had a design for military purposes, and another one for companies (Enigma A26). As of today, about 60% of the first million most visited websites propose encrypted and authenticated communications (via `https`), and so are most of the communications channels used by electronic devices (like *Wifi Protected Access*).

At the same time, the growth of exchanged data and the sensitivity of transferred information make the urge of protecting these data efficiently even more critical. While we are reaching the Moore's law barrier, other threats exist against nowadays' cryptosystems. For instance, the existence of a quantum computer with sufficient memory [Sho99] would break most of real-world cryptographic designs, which mostly rely on modular arithmetic assumptions. In this context, it is crucial to design cryptographic schemes that are believed to be quantum-resistant.

To address this problem, *post-quantum cryptography* arose in the early 2000s. The different candidates rely on several mathematical objects, such as lattices, error-correcting codes, systems of multivariate polynomials, etc. Recently, the National Institute of Standards and Technology (or *NIST*) organized a competition to evaluate different post-quantum schemes for encryption and signatures [NIS17]. In this competition, 82 protocols have been proposed out of which: 28 were lattice-based, 24 were code-based, 13 were multi-variate based, 4 were hash-based and the 13 left were categorized as "other".

Though, real-world cryptography mainly aims at designing digital signatures and encryption schemes, as illustrated by the NIST competition. Meanwhile, ongoing research in cryptology proposes different solutions to address more specific problems, such as the design of electronic-cash systems[1] [CFN88], which are the digital analogue of real money. Coins are delivered by a central authority (the bank) and spendings remain untraceable. In case of misbehavior (such as double-spending), the identity of the cheater is revealed.

Cryptographic constructions should additionally verify some security requirements. For instance, an encryption scheme has to hide a message in the presence of an eavesdropper, or even an active adversary who can alter some messages. To guarantee these requirements, cryptographers provide security proofs in the sense of precise security models. A security

[1] Which is not to be confuse with cryptocurrency…

proof mainly states that a given cryptographic scheme is secure if some problems are hard.

At last but not least, the importance of privacy and data protection has been a hot topic in the last years, as reflected by the development of the general data protection regulation law in 2016, which is implemented since may 25th. Hence, it is appealing to have privacy-preserving cryptographic constructions which would ideally resist the advent of a quantum computer. Nevertheless, the design of such protocols crucially relies on "zero-knowledge proofs". These are 2-party protocols between a prover and a verifier where the prover should convince the verifier of a statement without leaking any piece of information about this statement. In the context of post-quantum cryptography, such proofs systems are still limited in power or costly in terms of time, memory and communication consumptions.

1.1 Privacy-Preserving Cryptography

In this context, 'privacy-preserving' refers to the ability of a primitive to provide some functionalities while holding sensitive information private. An example of such primitives are *anonymous credentials* [Cha85, CL01]. Informally, this primitive allows users to prove themselves to some verifiers without telling their identity, nor the pattern of their authentications. To realize this, this system involves one (or more) credential issuer(s) and a set of users who have their own secret keys and pseudonyms that are bound to their secret. Users can dynamically obtain credentials from an issuer that only knows users' pseudonyms and obliviously sign users' secret key as well as a set of attributes. Later on, users can make themselves know to verifiers under a different pseudonym and demonstrate possession of a certification from the issuer, without revealing neither the signature nor the secret key. This primitive thus allows a user to authenticate to a system (e.g., in anonymous access control) while retaining its anonymity. In addition, the system is guaranteed that users indeed possess a valid credential.

Interests in privacy-based cryptography date back to the beginning of public-key cryptography [Rab81, Cha82, GM82, Cha85]. A reason for that could be the similarities between the motivations of cryptography and the requirements of privacy protection. Additionally, cryptographers' work in this field may have direct consequences in term of services that could be developed in the real-world. Indeed, having a practical anonymous credential scheme will enable its use for access control in a way that limits security flaws. Whereas, nowadays' implementations are based on more elementary building blocks, like signatures, whose manipulations may lead to different security holes [VP17].

Similarly, *advanced primitives* often involve simpler building blocks in their design. The difference lies in that provable security conveys security guarantees for the construction. As explained before, these proofs make the security of a set of schemes rely on hardness assumptions. Thus, the security relies on the validity of those assumptions, which are independently studied by cryptanalysts. Hence, security is guaranteed by the study of those assumptions. For example, the security analysis of multilinear maps in [CHL$^+$15] made obsolete a large amount of candidates at this time. This example reflects the importance of relying on well-studied and simple assumptions as we will explain in Chapter 2.

In the context of this thesis, the developed cryptographic schemes rely on lattices and bilinear maps over cyclic groups. Lattice-based cryptography is used to step towards post-quantum cryptography, while the latter proves useful in the design of practical schemes.

The details of these two structures are given in Chapter 3.

1.1.1 Zero-Knowledge Proofs

As explained before, zero-knowledge proofs are a basic building block for privacy-preserving cryptography. They require completeness, soundness and zero-knowledge properties. Completeness captures the correctness of the protocol if everyone is honest. In the case of a dishonest prover, soundness asks the probability that the verifier is convinced to be negligible. On the contrary, if the verifier is cheating, the zero-knowledge property guarantees that the prover's secret remains hidden.

In the case of identification schemes, the nature of the secret remains simple and solutions exist under multiple assumptions [Sch96, Ste96, KTX08, Lyu08]. For more complex statements, such as proving correct computation, a gap appears between post-quantum schemes and modular arithmetic-based schemes. In the case of pairing-based cryptography, there exist non-interactive zero-knowledge proofs which can prove a large variety of statements [GOS06, GS08] without idealized assumptions. Such proofs are still missing in the context of post-quantum cryptography so far.

In the lattice world, there are two main families of proof systems: Schnorr-like proofs [Sch96, Lyu09] and Stern-like proofs [Ste96], named after their respective authors. The first family works on some structured lattices. Exploiting this structure allows for more compact proofs, while the expressiveness of statements is quite restricted. The second kind of proofs is combinatorial and works on the representation of lattice elements (as matrix and vectors). By nature, these proofs are quite expensive in term of communication complexity. However, they can be used to prove a wide variety of statements as we will explain in more details along this thesis and especially in Section 4.3. More generally, zero-knowledge proofs are detailed in Chapter 4.

1.1.2 Signatures with Efficient Protocols

To enable privacy-preserving functionalities, a possible avenue is to couple zero-knowledge proofs with signature schemes. One of such signatures are *signatures with efficient protocols*. This primitive extends the functionalities of ordinary digital signature schemes in two ways: (i) It provides a protocol to allow a signer to obliviously sign a hidden message and (ii) Users are able to prove knowledge of a hidden message-signature pair in a zero-knowledge fashion.

These two properties turn out to be extremely useful when it comes designing efficient anonymity-related protocols such as anonymous credentials or e-cash. The design of effective signatures with efficient protocols is thus important for privacy-preserving cryptography.

In this thesis, we provide two of these signature schemes. One of them, described in Chapter 6, based on pairings, shifts the [LPY15] signature scheme to an idealized but practically acceptable model, aiming at efficiency. The other, described in Chapter 7, adapts a variant of Boyen's signature [Boy10, BHJ+15] along with the Kawachi-Tanaka-Xagawa commitment scheme [KTX08] to provide a lattice-based signature schemes that is compatible with Stern-like proofs. This scheme has also been relaxed in the context of adaptive oblivious

transfer where, in some places, it is only required to have random-message security instead of security against chosen-message security as described in Chapter 9.

1.2 Pairings and Lattices

In this thesis, the proposed constructions rely on the assumed hardness of assumptions over pairing-friendly groups and lattices. These two objects have widely been used in cryptography since the early 2000s [SOK00, Reg05]. Even since, they attracted much attention from cryptographers, leading to multiple constructions in advanced cryptography (as in [Jou00, BBS04, BN06, GS08, LYJP14, LPQ17] for pairings, and [GPV08, ABB10, BV11, GSW13, dPLNS17] for lattices).

1.2.1 Pairing-Based Cryptography

A pairing is a bilinear map from two cyclic source groups to a target group. This bilinear property takes advantage of a rich structure to groups that are compatible with such a map. It is then not surprising to see the variety of schemes that stems from pairing-based cryptography. In the context of privacy-based cryptography, an important breakthrough was the introduction of Groth-Sahai proofs [GOS06, GS08] that allow proving in a non-interactive zero-knowledge fashion a large class of statements in the standard model. For instance, Groth-Sahai proofs have been used in group signatures and anonymous-credential schemes [Gro07, BCKL08, BCC$^+$09], or e-cash systems in the standard model [BCKL09].

In this thesis, however, our pairing-based constructions focus on practicality. Thus, they are instantiated in the random oracle model, where Schnorr's proof are made non-interactive through the Fiat-Shamir transform when the statement to prove is simple enough.

A recent line of work in cryptanalysis of bilinear maps [KB16, MSS17, BD18] led to a change in the panorama of practical pairing-based cryptography. This affects us in the sense that security parameter has to be increased in order to achieve the same security level.

Nevertheless, pairing-based cryptography offers a nice tradeoff between its capabilities and efficiency. As an example, we can cite the work of Döttling and Garg [DG17b], who closed the problem of providing an identity-based encryption scheme which only relies on the Diffie-Hellman assumption (it is construction on cyclic groups that does not need pairings, as defined in Definition 2.11). While their construction relies on a simpler mathematical object, it does not reach the efficiency of pairing-based ones [BB04].

1.2.2 Lattice-Based Cryptography

From an algebraic point of view, a lattice is a discrete subgroup of $\mathbb{R}^n$, which leads to a simple additive structure. The core difference with number-theoretic cryptography, such as discrete-logarithm-based cryptography, is the existence of the geometrical structure of the lattice. From this geometry rises some problems that are believed to withstand quantum computers. Despite this apparently simple structure, some advanced primitives are only known, as of today, to be possible under lattice assumptions, such as fully-homomorphic encryption [Gen09, BV11, GSW13].

The versatility of lattice-based cryptography is enabled by the existence of lattice trapdoors [GPV08, CHKP10, MP12], as we explain in Section 3.2.2. Informally, the knowl-

edge of a short basis for a lattice allows sampling short vectors, which is believed to be hard without such a short basis. Furthermore, knowing a short basis for the lattice $\{\mathbf{v} \in \mathbb{Z}^m \mid \mathbf{Az} = 0 \bmod q\}$ described by matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ makes it possible to generate a short basis for a related lattice described by $[\mathbf{A} \mid \mathbf{B}] \in \mathbb{Z}_q^{n \times m'}$. An application for this property is Boyen's signature scheme [Boy10]. In this scheme, a signature for message m is a short vector in the orthogonal lattice of the matrix $\mathbf{A}_m = [\mathbf{A} \mid \mathbf{B}_m]$, where $\mathbf{B}_m$ is publicly computable. Hence, knowing a trapdoor for $\mathbf{A}$ makes the computation of this short vector possible, and the message is bound to the description of the lattice $\mathbf{A}_m$.

Still, the use of lattice trapdoors comes at a price, as it significantly decreases the efficiency of cryptographic designs that use them [Lyu12, LLNW16]. Given that we provide the first lattice-based construction for the scheme we present, we focused on designing provably-secure scheme under well-studied assumptions.

1.3 Our Results

In this thesis, we present several cryptographic constructions that preserve privacy. These constructions are the result of both improvements we made in the use of zero-knowledge proofs and the ability to prove the security of our constructions under standard assumptions. We believe that these advances on zero-knowledge proofs are of independent interest and that the given schemes are a step towards quantum-secure privacy-preserving cryptography. In the following, we detail four contributions that are developed in this thesis. These results are taken from four published articles: [LMPY16, LLM+16a, LLM+16b, LLM+17].

1.3.1 Dynamic Group Signatures and Anonymous Credentials

In Part II, we present two primitives: dynamic group signatures and anonymous credentials. We already described the behavior of anonymous credential in Section 1.1. As for dynamic group signatures, they are a primitive that allows a group of users to authenticate messages on behalf of the group while remaining anonymous inside this group. The users still remain accountable for their actions, as another authority knows a secret information that gives it the ability to lift anonymity of misbehaving users.

By itself, this primitive can be used to provide anonymous authentications while providing accountability (which is not the case with anonymous credentials). For instance, in the Internet of things, such as smart cars, it is important to provide authenticated communication channels as well as anonymity. For car communications, if exchanged data may not be sensitive by themselves, the identity of the driver could be. We can imagine a scenario where some burglars eavesdrop a specific car to know whenever a house is empty.

In this thesis, we present in Chapter 6 pairing-based group signatures that aims at efficiency while relying on simple assumptions. The resulting scheme shows competitive signature size with other schemes that rely on more ad-hoc assumptions, and its practicality is supported by an implementation. This scheme is presented in [LMPY16], which is joint work with Benoît Libert, Thomas Peters an Moti Yung presented at AsiaCCS'16.

Chapter 7 presents the first *dynamic* group signature scheme relying on lattice assumptions. This has been made possible by adapting Stern-like proofs to properly interact with a signature scheme: a variant of Boyen's signature [Boy10, BHJ+15]. It results in a *signature with efficient protocols* that is of independent interest. Later, it has been adapted in the

design dynamic group encryption [LLM+16b] and adaptive oblivious transfer [LLM+17]. This work is described in [LLM+16a], made with Benoît Libert, San Ling, Khoa Nguyen and Huaxiong Wang and presented at Asiacrypt'16.

1.3.2 Group Encryption

Group encryption schemes [KTY07] are the encryption analogue of group signatures. In this setting, a user is willing to send a message to a group member, while keeping the recipient of the message hidden inside the group. In order to keep user accountable for their actions, an opening authority is further empowered with some secret information allowing it to un-anonymize ciphertexts.

More formally, a group signature scheme is a primitive allowing the sender to generate publicly verifiable proofs that: (1) The ciphertext is well-formed and intended to some registered group member who will be able to decrypt; (2) The opening authority will be able to identify the receiver if necessary; (3) The plaintext satisfies certain properties, such as being a witness for some public relation. In the model of Kiayias, Tsiounis and Yung [KTY07], the message secrecy and anonymity properties are required to withstand active adversaries, which are granted access to decryption oracles in all security definitions.

A natural application is to allow a firewall to filter all incoming encrypted emails except those intended for some certified organization members and the content of which is additionally guaranteed to satisfy certain requirements, like the absence of malware. Furthermore, group encryption schemes are motivated by privacy applications such as anonymous trusted third parties, key recovery mechanisms or oblivious retriever storage system. In cloud storage services, group encryption enables privacy-preserving asynchronous transfers of encrypted datasets. Namely, it allows users to archive encrypted datasets on remote servers while convincing those servers that the data is indeed intended to some anonymous certified client who has a valid account to the storage provider. In case of suspicions on the archive's content, a judge should be able do identify the recipient of the archive.

To tackle the problem of designing lattice-based group encryption, we needed to handle "quadratic relations". Indeed, lattice-based zero-knowledge proof systems were able to handle only relations where witnesses are multiplied by a public value. Let us recall that, in Learning-With-Errors schemes, an encryption have the form $\mathbf{A} \cdot \mathbf{s} + \mathbf{e} + \mathbf{m}\lceil \frac{q}{2} \rceil \bmod q$, where $\mathbf{A}$ is the recipient public-key. As group encryption requires this public-key $\mathbf{A}$ to be private, a way to achieve this is to have a zero-knowledge proof system which handles relations where the witness is multiplied with a private matrix.

We address this issue introducing new technique to handle this kind of relations. These techniques, based on a *divide-and-conquer* strategy, are described in Chapter 8, as well as the construction of the group encryption scheme proven fully-secure in the standard model. This work have been presented at Asiacrypt'16 [LLM+16b] and have been done with Benoît Libert, San Ling, Khoa Nguyen and Huaxiong Wang.

1.3.3 Adaptive Oblivious Transfer

Oblivious transfer is a primitive coined by Rabin [Rab81] and later extended by Even, Goldreich and Lempel [EGL85]. It involves a server with a database of messages indexed

from 1 to N and a receiver with a secret index $\rho \in \{1, \ldots, N\}$. The protocol allows the receiver to retrieve the ρ-th message from the database without letting it infer anything on his choice. Furthermore, the receiver only obtains the ρ-th message and learns nothing about the other messages.

In its adaptive flavor [NP99], oblivious transfer allows the receiver to interact k times with the server to obtain k messages in such a way that, each request may depend on the previously retrieved messages.

From a theoretical point of view, oblivious transfer is known to be a *complete building block* for cryptography in the sense that, if it can be realized, then any secure multiparty computation can be. In its adaptive variant, oblivious transfer has applications in privacy-preserving access to sensitive databases (such as medical records or financial data) stored in an encrypted form on a remote server.

In its basic form, (adaptive) oblivious transfer does not restrict in any way the population of users who can obtain specific records. In many sensitive databases (e.g., DNA samples or patients' medical history), however, not all users should be able to access the whole database. It is thus crucial to protect the access to certain entries conditioned on the receiver holding suitable credentials delivered by authorities. At the same time, privacy protection requires that authorized users should be able to query database records while leaking as little as possible about their interests or activities.

This requirements is handled by endowing oblivious transfer with access control, as stated by Camenish, Dubovitskaya and Neven [CDN09]. In this variant, each database record is protected by a different access control policy. Based on their attributes, users can obtain credentials from pre-determined authorities, which entitle them to anonymously retrieve database records of which the access policy accepts their certified attributes. During the transfer phase, the user demonstrates, in a zero-knowledge manner, possession of an attribute string compatible with the policy of a record in the database, as well as a credential for this attribute. The only information that the database holder eventually learns is that some user retrieved some record which he was authorized to obtain.

To achieve this, an important property is the expressiveness of such access policies. In other words, the system should be able to handle complex attribute policies while keeping time and memory consumption reasonable[2]. In this thesis, we propose in Chapter 9 a zero-knowledge protocol to efficiently handle any access policy that can be described with a logarithmic-depth boolean circuit, also known as $\mathsf{NC1}$, based on lattices. In the context of adaptive oblivious transfer with access control, most of the schemes (based on pairing assumptions) manage to handle the case of conjunctions under reasonable assumptions [CDN09, CDNZ11, ACDN13]. Under strong assumptions, however, the case of $\mathsf{NC1}$ can be taken care of [ZAW+10].

This joint work with Benoît Libert, San Ling, Khoa Nguyen and Huaxiong Wang was presented at Asiacrypt'17 [LLM+17].

[2]Here, "*reasonable*" means (probabilistic) polynomial time.

Part I

Background

Chapter 2

Security Proofs in Cryptography

Provable security is a subfield of cryptography where constructions are proven secure with respect to a security model. To illustrate this notion, let us take the example of public-key encryption schemes. This primitive consists in three algorithms: *key generation*, *encryption* and *decryption*. These algorithms acts according to their names. Naturally, the question of "how to define the security of this set of algorithms" arises. To answer this question, we have to define the power of the adversary, and its goal. In cryptography, many approaches have been used to define this (random oracle model, universal composability (UC) [Can01]…) which give rise to stronger security guarantees. If one aims at the strongest security for its construction, there are known impossibility results in strong models. For instance, in the UC model, it is impossible to realize two-party computation [Yao86] without trusted setup [CKL06], while it is possible in the plain model [LP07].

In this chapter, we will focus on the computational complexity elements we need to define properly the security models we will use in this thesis. Then we will define these security models.

2.1 Security Reductions

Provable security provides constructions for which security is guaranteed by a security proof, or *security reduction*. The name "reduction" comes from computational complexity. In this field of computer science, research focuses on defining equivalence classes for problems or hierarchical relations between them, based on the necessary amount of resources to solve them. In order to define lower bounds for the complexity of some problems, a classical approach is to provide a construction that goes from an instance of a problem A to an instance of problem B such that, if a solution of B is found, then so is a solution of A. This amounts to saying that problem B is at least as hard as problem A up to the complexity of the transformation. For instance, Cook has shown that satisfiability of Boolean formulas is at least as hard as every problem in NP [Coo71] up to a polynomial-time transformation.

Let us now define more formally the notions of reduction and computability using the computational model of Turing machines.

Definition 2.1 (Turing Machine)**.** A k-tape Turing Machine (TM) is described by a triple $M = (\Gamma, Q, \delta)$ containing:

- A finite set Γ, called the *tape alphabet*, which contains symbols that the TM uses in its tapes. In particular, Γ contains a *blank symbol* "$\square$", and "$\triangleright$" that denotes the beginning of a tape.
- A finite set Q called the *states* of the TM. It contains special states q_{start}, q_{halt}, called respectively the *initial state* and the *halt state*.
- A function $\delta : (Q \backslash \{q_{halt}\}) \times \Gamma^{k-1} \to Q \times \Gamma^{k-1} \times \{\leftarrow, \downarrow, \rightarrow\}^k$, called the *transition function*, that describes the behavior of the internal state of the machine and the TM heads.
  Namely, $\delta(q, a_1, \ldots, a_{k-1}) = (r, b_2, \ldots, b_k, m_1, \ldots, m_k)$ means that upon reading symbols $(a_1, \ldots, a_{k-1})$ on tapes 1 to $k-1$ (where the first tape is the input tape, and the k-th tape is the output tape) on state q, the TM will move to state r, write $b_2, \ldots, b_k$ on tapes 2 to k and move its heads as dictated by $m_1, \ldots, m_k$.

A TM M is said to *compute* a function $f : \Sigma^\star \to \Gamma^\star$ if, for any finite input $x \in \Sigma^\star$ on tape T_1, blank tapes $T_2, \ldots, T_k$ with a beginning symbol $\triangleright$ and initial state q_{start}, M halts in a finite number of steps with $f(x)$ written on its output tape T_k.

A TM M is said to *recognize* a language $L \subseteq \Sigma^\star$ if, on a finite input $x \in \Sigma^\star$ written on its input tape T_1, blank tapes $T_2, \ldots, T_k$ with a beginning symbol $\triangleright$ and initial state q_{start}, the machine M eventually ends on the state q_{halt} with 1 written on its output tape if and only if $x \in L$.

A TM M is said to run in $T(n)$-time if, on any input x, it eventually stops within $T(|x|)$ steps.

A TM M is said to run in $S(n)$-space if, on any input x, it eventually stops after having written at most $S(|x|)$ memory cells in its working tapes.

Turing machines are a computational model that proved useful in complexity theory as it is convenient to evaluate the running time of a Turing machine, which amounts to bounding the number of steps the machine can take. Similarly, the working tapes works analogously to the memory of a program, and then counting the number of cells the machine uses is equivalent to evaluating the amount of memory the program requires.

From these considerations, it is possible to describe the time and space complexity of a program from the definition of Turing machines. In our context, we will work with Turing machines that run in polynomial-time and space, as polynomials benefit from good stability properties (sum, product, composition, …).

Definition 2.2 (P [Rab60])**.** The class P describes the set of languages that can be recognized by a Turing machine running in time $T(n) = \mathcal{O}(\mathsf{poly}(n))$.

In theoretical computer science, the class P is often considered as the set of "easy" problems. These problems are considered easy in the sense that the growth of the cost to solve them is asymptotically negligible in front of other functions such as exponential. In this context, it is reasonable to consider the computational power of an adversary as polynomial (or quasi-polynomial) in time and space. As cryptographic algorithms are not deterministic, we also have to consider the probabilistic version of the computation model.

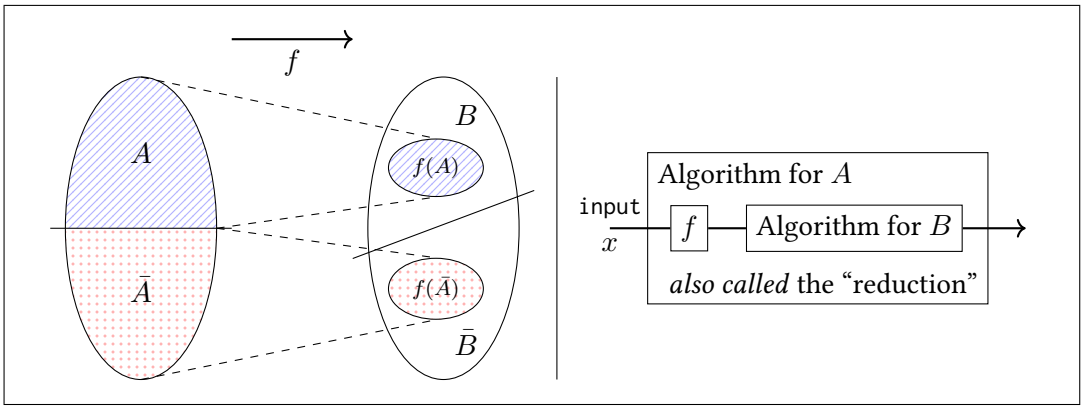


Figure 2.1 – Illustration of a polynomial-time reduction from A to B [AB09, Fig. 2.1].

Definition 2.3 (Probabilistic Turing machine)**.** A *probabilistic Turing machine* is a Turing machine with two different transition functions δ_0 and δ_1 where, at each step, a random coin is tossed to pick δ_0 or δ_1 with probability $1/2$ independently of all the previous choices.

The machine only outputs `accept` and `reject` depending on the content of the output tape at the end of the execution. We denote by $M(x)$ the random variable corresponding to the value M writes on its output tape at the end of its execution.

Definition 2.4 (PP [Gil77])**.** The class PP describes the set of languages $L \subseteq \Sigma^\star$ that a Turing machine M recognizes such that the TM M stops in time $\mathsf{poly}(|x|)$ on every input x and

$$\begin{cases} \Pr[M(x) = 1 \mid x \in L] > \frac{1}{2} \\ \Pr[M(x) = 0 \mid x \notin L] \leq \frac{1}{2} \end{cases}.$$

In the following PPT stands for "probabilistic polynomial time".

We defined complexity classes that corresponds to natural sets of programs that are of interest to us. In order to work with them, we will define the principle of polynomial time reduction.

Definition 2.5 (Polynomial time reduction)**.** A language $A \subseteq \{0,1\}^\star$ is *polynomial-time reducible to* a language $B \subseteq \{0,1\}^\star$, denoted by $A \preceq_P B$, if there is a *polynomial-time computable* function $f : \{0,1\}^\star \to \{0,1\}^\star$ such that for every $x \in \{0,1\}^\star$, $x \in A$ if and only if $f(x) \in B$.

In other words, a polynomial reduction from A to B is the description of a polynomial time algorithm (also called "*the reduction*"), that uses an algorithm for B in a black-box manner to solve A. This is illustrated in Figure 2.1.

To write down that a TM has black-box access to a TM M^O that computes function O, we sometimes use the *oracle* terminology.

Definition 2.6 (Oracle machine)**.** A Turing Machine M is said to have *oracle access* to a function $O(\cdot)$ if it has access to the result of $O(x)$ for any input x of its choice in constant time. We denote the output of M on input x with oracle O by $M^O(x)$.

We can notice that P and PP are both closed under polynomial-time reduction. Namely, if a problem is easier than another problem in P (resp. PP), then the former problem is also in P (resp. PP).

Until know, we mainly focus on the running time of the algorithms. In cryptology, it is also important to consider the success probability of algorithms: an attack is successful if the probability that it succeeds is noticeable.

Definition 2.7 (Landau notations)**.** Let f, g be two functions from $\mathbb{N}$ to $\mathbb{R}$. Let us define the so-called *Landau notations* to asymptotically compare functions.

f **is bounded by** g**:** $f(x) = \mathcal{O}(g(x))$ if there exists a constant $k > 0$ such that $|f(n)| \leq k \cdot |g(n)|$ eventually.

f **is not dominated by** g**:** $f(x) = \Omega(g(x))$ if there exists a constant $k > 0$ such that $|f(n)| \geq k \cdot |g(n)|$ eventually.

f **is bounded by** g **from above and below:** $f(x) = \Theta(g(x))$ if $f(x) = \mathcal{O}(g(x))$ and $f(x) = \Omega(g(x))$.

g **dominates** f**:** $f(x) = o(g(x))$ if for any $k > 0$, $f(n) \geq k \cdot |g(n)|$ eventually.

f **dominates** g**:** $f(x) = \omega(g(x))$ if for any $k > 0$, $|f(n)| > k \cdot |g(n)|$ eventually.

Definition 2.8 (Negligible, noticeable, overwhelming probability)**.** Let $f : \mathbb{N} \to [0, 1]$ be a function. The function f is said to be *negligible* if $f(n) = n^{-\omega(1)}$, and this is written $f(n) = \mathsf{negl}(n)$.
Non-negligible functions are also called *noticeable* functions.
Finally, if $f = 1 - \mathsf{negl}(n)$, f is said to be *overwhelming*.

Now, we have to define two more notions to be able to work on security proofs. Namely, the *security notions* and the *hardness assumptions*. The former are the statements we need to prove, and the latter are the hypotheses on which we rely.

The details of the hardness assumptions we use are given in Chapter 3. Nevertheless, some notions are common to these and are evoked here.

The confidence one can put in a hardness assumption depends on many criteria. First of all, a weaker assumption is preferred to a stronger one. To illustrate this, let us consider the two following assumptions:

Definition 2.9 (Discrete logarithm)**.** The *discrete algorithm problem* is defined as follows. Let $(\mathbb{G}, \cdot)$ be a cyclic group of order p. Given $g, h \in \mathbb{G}$, the goal is to find the integer $a \in \mathbb{Z}_p$ such that: $g^a = h$.

The *discrete logarithm assumption* is the intractability of this problem for any PPT algorithm with noticeable probability.

Definition 2.10 (Indistinguishability)**.** Let D_0 and D_1 be two probabilistic distributions and par be public parameters. Let us define the following experiments $\mathsf{Exp}^{\mathrm{Dist}}_{\mathcal{D},0}$ and $\mathsf{Exp}^{\mathrm{Dist}}_{\mathcal{D},1}$ for any algorithm $\mathcal{D}$:

$\mathsf{Exp}^{\mathrm{Dist}}_{\mathcal{D},b}(\lambda)$

$x \hookleftarrow D_b$

$b' \leftarrow \mathcal{D}(1^\lambda, \mathsf{par}, x)$

return b'

The advantage of an adversary $\mathcal{D}$ for this game is defined as

$$\mathrm{Adv}^{\mathrm{Dist}}_{\mathcal{D}}(\lambda) \triangleq \left|\Pr\left[\mathsf{Exp}^{\mathrm{Dist}}_{\mathcal{D},1}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathrm{Dist}}_{\mathcal{D},0}(\lambda) = 1\right]\right|.$$

A PPT algorithm which has a noticeable advantage for the above experiments is called a *distinguisher* between D_0 and D_1.

Two distributions D_0 and D_1 are *computationally indistinguishable* if there does not exist any PPT distinguisher between those two distributions.

Definition 2.11 (Decisional Diffie-Hellman)**.** Let $\mathbb{G}$ be a cyclic group of order p. The *decisional Diffie-Hellman* (DDH) distribution is

$$\mathsf{D}_{\mathsf{DDH}} \triangleq \{(g, g^a, g^b, g^{ab}) \mid g \hookleftarrow \mathcal{U}(\mathbb{G}), a, b \hookleftarrow \mathcal{U}(\mathbb{Z}_p)\}.$$

The *DDH assumption* states that the distributions $\mathsf{D}_{\mathsf{DDH}}$ and $\mathcal{U}(\mathbb{G}^4)$ are computationally indistinguishable given the public parameter $\mathbb{G}$ (the description of the group).

The discrete logarithm assumption is implied by the decisional Diffie-Hellman assumption for instance. This is why it is preferable to work with the discrete logarithm assumption when it is possible. For instance, there is no security proofs for the El Gamal encryption scheme from DLP.

Another criterion to evaluate the security of an assumption is to look if the assumption is "simple to state" or not. This observation is buttressed by the statement of [KL07, p.25]: "*... there is a general preference for assumptions that are simpler to state, since such assumptions are easier to study and to refute.*".

Indeed, it is complicated to evaluate the security of an assumption as q-Strong Diffie-Hellman assumptions defined as follows.

Definition 2.12 (q-Strong Diffie-Hellman assumption [BB04, BBS04])**.** In a cyclic group $\mathbb{G}$, the *q-Strong Diffie-Hellman* (q-SDH) problem is, given $g, g^a, g^{a^2}, \ldots, g^{a^q}$, compute the element $g^{a^{q+1}}$.

The security of this assumption inherently depends on the parameter q of the assumption. Cheon additionally showed that, for large values of q, this assumption is no more trustworthy [Che06]. These parameterized assumptions are called *q-type assumptions*. There also exist other kinds of non-static assumptions, such as interactive assumptions. An example can be the "1-*more-DL*" assumption. Given oracle access to n discrete logarithm queries (n is not known in advance), the 1-more-DL problem is to solve a $n+1$-th discrete logarithm. These non-interactive assumptions are furthermore *non-falsifiable* according to the definition of Naor [Nao03]. Non-interactive and constant-size assumptions are sometimes called "*standard*".

The next important aspect of a security proof is the model in which it takes place. This is the purpose of the next section.

2.2 Random-Oracle Model and Standard Model

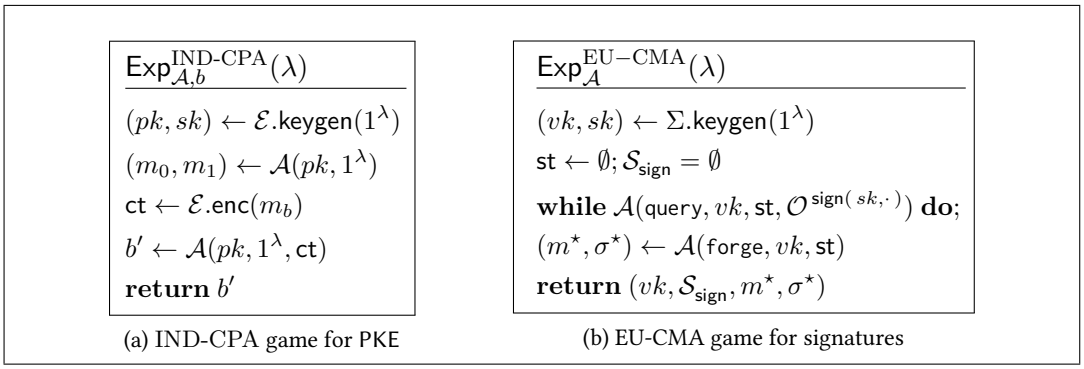
Security proofs should preferably stand in the *standard model* of computation, where no idealization is assumed on behalf of the building blocks. In this model, no implicit assumptions are assumed.

For instance, cryptographic hash functions enjoy several different associated security notions [KL07]. On of the weakest is the collision resistance, that states that it is intractable to find two strings that map to the same digest. A stronger notion is the second pre-image resistance, that states that given $x \in \{0,1\}^\star$, it is not possible for a PPT algorithm to find an $\tilde{x} \in \{0,1\}^\star$ such that $h(x) = h(\tilde{x})$. Similarly to what we saw in the previous section about DDH and DLP, we can see that collision resistance implies second pre-image resistance. Indeed, if there is an attacker against second pre-image, then one can choose a string $x \in \{0,1\}^\star$ and obtains from this attacker another string $\tilde{x} \neq x \in \{0,1\}^\star$ such that $h(x) = h(\tilde{x})$. Hence, a hash function that is collision resistant is also second pre-image resistant.

The *random oracle model* [FS86, BR93], or ROM, is an idealized security model where hash functions are assumed to behave as a truly random function. This implies collision resistance (if the codomain of the hash function is large enough) and other security notions related to hash functions. In this model, hash functions are modeled as oracles in the view of the adversary. These oracles are controlled by the reduction, meaning that the reduction can program the hash function as it likes as long as the responses look random and independent. Moreover, the reduction has access to the conversation between the adversary and the random oracle. It thus eventually knows all inputs for which the adversary chose to evaluate the function.

We can notice that this computation model is unrealistic [CGH98]. Let us construct a *counter-example*. Let Σ be a secure signature scheme, and let Σ_y be the scheme that returns $\Sigma(m)$ as a signature if and only if $h(0) \neq y$ and 0 as a signature otherwise. In the ROM h behaves as a random function. Hence, the probability that $h(0) = y$ is negligible with respect to the security parameter for any fixed y. On the other hand, it appears that when h is instantiated with a real-world hash function, then $\Sigma_{h(0)}$ is the null function, and therefore completely insecure as a signature scheme. □

In this context, one may wonder why is the ROM still used in cryptographic proofs [LMPY16, LLM+16a]. One reason is that some constructions are not known to exist yet from the standard model. For instance, non-interactive zero-knowledge (NIZK) proofs for all NP languages is not known to follow solely from lattice assumptions [Ste96, Lyu08]. NIZK proofs form an elementary building block for privacy-based cryptography. In the lattice setting, we do not have much better options that using random oracles [LLM+16a]. Another reason to use the ROM in cryptography, is because it enables much more efficient constructions and we have no example of a failure in the random oracle methodology for a natural cryptographic construction [BR93]. The example we built earlier is artificial, and in practice there is no known attacks against the ROM for a natural scheme used in real-life applications. Thus, for practical purposes, constructions in the ROM are usually



(a) IND-CPA game for PKE

(b) EU-CMA game for signatures

Figure 2.2 – Some security games examples.

more efficient. For instance, the scheme we present in Chapter 6 adapts the construction of dynamic group signature in the standard model from Libert, Peters and Yung [LPY15] to the ROM. Doing this transform reduces the signature size from 32 elements in $\mathbb{G}$, 14 elements in $\widehat{\mathbb{G}}$ and *one* scalar in the standard model [LPY15, App. J] down to 7 elements in $\mathbb{G}$ and 3 scalars in the ROM.

We now have defined the context we are working on and the base tools that allows security proofs. The following section explains how to define the security of a cryptographic primitive.

2.3 Security Games and Simulation-Based Security

In order to define security properties, a common manner is to define security *games* (or *experiments*) [GM82, Sho06].

Two examples of security game are given in Figure 2.2: to formalize the notions of *indistinguishability under chosen-plaintext attacks* (IND-CPA) for public-key encryption (PKE) schemes and *existential unforgeability under chosen message attacks* (EU-CMA) for signature schemes.

IND-CPA security is modeled by an *indistinguishability* game, meaning that the goal for the adversary $\mathcal{A}$ against this game is to distinguish between two messages from different distributions. To model this, for any adversary $\mathcal{A}$, we define a notion of *advantage* for the IND-CPA game as

$$\mathrm{Adv}^{\text{IND-CPA}}_{\mathcal{A}}(\lambda) \triangleq \left|\Pr\left[\mathsf{Exp}^{\text{IND-CPA}}_{\mathcal{A},1}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{IND-CPA}}_{\mathcal{A},0}(\lambda) = 1\right]\right|.$$

We say that a PKE scheme is IND-CPA if, for any PPT $\mathcal{A}$, the advantage of $\mathcal{A}$ in the IND-CPA game is negligible with respect to λ.

This definition of advantages models that the adversary is unable to distinguish whether the ciphertext ct comes from the experiment $\mathsf{Exp}^{\text{IND-CPA}}_{\mathcal{A},0}$ or the experiment $\mathsf{Exp}^{\text{IND-CPA}}_{\mathcal{A},1}$. As a consequence, the adversary cannot get a single bit of information about the ciphertext.

This kind of definition is also useful to model anonymity. For instance in Section 5.3.4, the definition of anonymity for group signatures is defined in a similar fashion (Definition 5.5).

To handle indistinguishability between distributions, it is useful to quantify the distance between two distributions. In this context, we define the statistical distance as follows.

Definition 2.13 (Statistical Distance)**.** Let P and Q be two distributions. The *statistical distance* $\Delta(P, Q)$ between P and Q is defined as

$$\Delta(P, Q) \triangleq \frac{1}{2} \sum_{x \in \mathrm{Supp}(P) \cup \mathrm{Supp}(Q)} |P(x) - Q(x)|.$$

Two distributions are *statistically close* if their statistical distance is negligible with respect to the security parameter. It is worth noticing that if two distributions are statistically close, then the advantage of an adversary in distinguishing between them is negligible.

Notation. $P \approx_s Q$ means that P is *statistically close* to Q.

Another interesting metric, that will be used in the security proof of is the Rényi Divergence:

Definition 2.14 (Rényi divergence)**.** For any two discrete distributions P and Q such that $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$, and $a \in]1, +\infty[$, we define the *Rényi divergence* of order a by:

$$R_a(P||Q) = \left(\sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{\frac{1}{a-1}}.$$

We define the Rényi divergences of orders 1 and $+\infty$ as:

$$R_1(P||Q) = \exp\left(\sum_{x \in \mathrm{Supp}(P)} P(x) \log \frac{P(x)}{Q(x)} \right) \text{ and } R_\infty(P||Q) = \max_{x \in \mathrm{Supp}(P)} \frac{P(x)}{Q(x)}.$$

The divergence R_1 is the (exponential) of the Kullback-Leibler divergence.

Bai, Langlois, Lepoint, Stehlé and Steinfeld [BLL+15] observed that the Rényi Divergence has a property similar to the *triangular inequality* with respect to multiplication, and can be useful in the context of unforgeability game as we will explain it in the following paragraph. Prest further presented multiple uses of the Rényi Divergence in [Pre17].

We notice that security definitions for signature scheme are not indistinguishability-based experiments, but search experiments (i.e., the adversary has to output a string rather than distinguishing between two experiments by outputting a single bit). The goal of the adversary is not to distinguish between two distributions, but to forge a new signature from what it learns via signature queries.

Those signature queries are handled by an oracle $\mathcal{O}^{\mathsf{sign}(\,sk,\cdot\,)}$, which on input m returns the signature $\sigma = \Sigma.\mathsf{sign}(sk, m)$ and adds σ to $\mathcal{S}_{\mathsf{sign}}$. The initialization of these sets and the oracle's behavior may be omitted in the rest of this thesis for the sake of readability.

For EU-CMA, the advantage of an adversary $\mathcal{A}$ is defined as

$$\mathrm{Adv}^{\text{EU-CMA}}_{\mathcal{A}}(\lambda) \triangleq \Pr\left[\Sigma.\mathsf{verif}(vk, m^\star, \sigma^\star) = \top \ \wedge \ \sigma^\star \notin \mathcal{S}_{\mathsf{sign}} \right].$$

A signature scheme is considered unforgeable under chosen message attacks if, for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible with respect to λ.

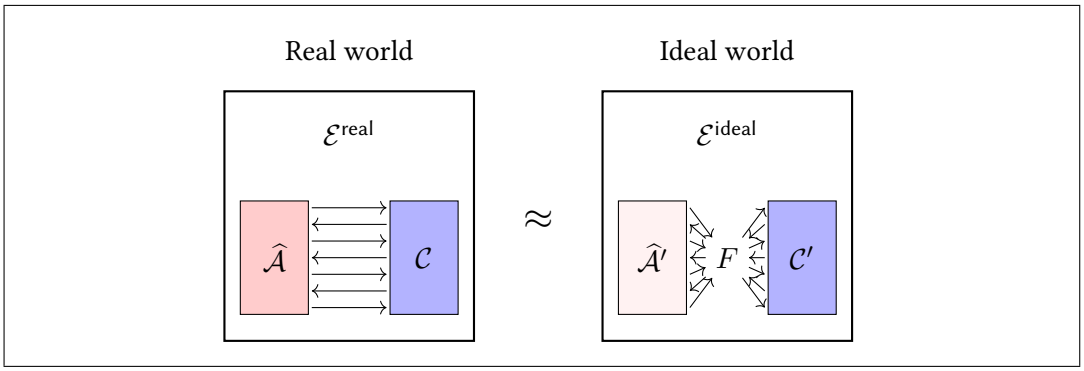


Figure 2.3 – Simulation-based cryptography.

This means that, within reasonable expected time[1], no adversary can create a new valid signature without the signing key (sk). This kind of definitions are often used in the case of authentication primitives. In our example of group signatures in Part II, the *security against misidentification attacks* (or *traceability*) experiment follows the same structure. This security notion illustrates that no collusion between malicious users and the group authority can create valid signatures that open on an honest user, or do not open to a valid registered user.

The security definition of IND-CPA is defined via an indistinguishability experiment. The first security definition for PKE was nevertheless a simulation-based definition [GM82]. In this context, instead of distinguishing between two messages, the goal is to distinguish between two different environments. In the following, we will use the *Real world*/*Ideal world* paradigm [Can01] to describe those different environments. Namely, for PKE, it means that, for any PPT adversary $\widehat{\mathcal{A}}$ – in the *Real world* – that, interacts with a challenger $\mathcal{C}$, there exists a PPT *simulator* $\widehat{\mathcal{A}}'$ – in the *Ideal world* – that interacts with the same challenger $\mathcal{C}'$ with the difference that the functionality F is replaced by a trusted third party in the *Ideal word*.

In other words, it means that the information that $\widehat{\mathcal{A}}$ obtains from its interaction with the challenger $\mathcal{C}$ does not allow $\mathcal{A}$ to lean any more information than it does via black-box access to the functionality.

In the context of PKE, the functionality is the access to the public key pk as described in Line 2 of $\mathsf{Exp}^{\text{IND-CPA}}_{\mathcal{A},b}(\lambda)$. Therefore, the existence of a simulator $\widehat{\mathcal{A}}$ that does not use pk shows that $\mathcal{A}$ does not learn anything from pk.

For PKE, the simulation-based definition for chosen-plaintext security is equivalent to the indistinguishability security [Gol04, Se. 5.2.3], even if the two security definitions are conceptually different. As indistinguishability-based model are often easier to work with, they are more commonly used to prove security of PKE schemes. For other primitives, such as Oblivious Transfer (OT) described in Chapter 9, the simulation-based definitions are strictly stronger than indistinguishability definitions [NP99]. Therefore, it is preferable to have security proofs of the strongest *possible* definitions in theoretical cryptography.

Even though, the question of which security model is the strongest remains a complex one, as it depends on many parameters: the answer mainly depends on the manner the scheme

[1]Reasonable time may have multiple definitions, in the context of theoretical cryptography, we assume that quasi-polynomial time is the upper bound of reasonable.

will be used as well as the adversarial model. For example, we know from the work of Canetti and Fischlin [CF01] that it is impossible to construct a UC-secure bit commitment scheme[2] in the plain model, while the design of such a primitive is possible assuming a *trusted setup*. In the *trusted setup* model or *common reference string* (CRS) model, all the participants are assumed to have access to a common string $\mathsf{crs} \in \{0,1\}^\star$ that is drawn from some specific distribution D_{crs}.

[2] The definition of a commitment scheme is given in Definition 4.5. To put it short, it is the digital equivalent of a safe.

Chapter 3

Underlying Structures

In the previous chapter, we saw that cryptography has to rely on *computational hardness assumptions*. Besides *information-theoretic cryptography*, most hardness assumptions are built on top of suitable algebraic structures. For instance the discrete logarithm assumption (Definition 2.9) is based on a cyclic group structure.

The existence of such structures proves useful when it comes to designing protocols. For this purpose, constructions take advantage of the mathematical properties of the structure to enable the functionality. An example is the multiplicative homomorphism of the ElGamal cryptosystem which is made possible by underlying cyclic group structure.

In this chapter, we describe the different structures on which the cryptographic primitives we design in this thesis are based on, namely bilinear groups and lattices, as well as related hardness assumptions.

3.1 Pairing-Based Cryptography

Pairing-based cryptography was introduced by Sakai, Ohgishi and Kasahara [SOK00] to generalize Diffie-Hellman key exchange to three users in one round. Since then, many constructions have been proposed for cryptographic constructions, such as identity-based encryption [BF01, Wat05] or group signatures [BBS04]. Multiple constructions and parameter sets coexist for pairings. Real-world implementation are based on elliptic curves [BN06, KSS08], but recent advances in cryptanalysis requires to reassess the security level of pairing-based cryptography [KB16, MSS17, BD18].

In the following, we adopt black-box definitions of cryptographic pairings as bilinear maps, and on the assumed hardness of classical constant-size assumptions over pairing-friendly groups, namely SXDH and SDL. The notations $1_{\mathbb{G}}$, $1_{\widehat{\mathbb{G}}}$ and $1_{\mathbb{G}_T}$ denote the identity elements in $\mathbb{G}$, $\widehat{\mathbb{G}}$ and $\mathbb{G}_T$ respectively.

Definition 3.1 (Pairings [BSS05])**.** A pairing is a map $e : \mathbb{G} \times \widehat{\mathbb{G}} \to \mathbb{G}_T$ over cyclic groups of order p that verifies the following properties for any $g \in \mathbb{G}, \hat{g} \in \widehat{\mathbb{G}}$:

(i) bilinearity: for any $a, b \in \mathbb{Z}_p$, we have $e(g^a, \hat{g}^b) = e(g^b, \hat{g}^a) = e(g, \hat{g})^{ab}$.

(ii) non-degeneracy: $e(g, \hat{g}) = 1_{\mathbb{G}_T} \iff g = 1_{\mathbb{G}}$ or $\hat{g} = 1_{\widehat{\mathbb{G}}}$.

(iii) the map is computable in polynomial time in the size of the input.

For cryptographic purposes, pairings are usually defined over elliptic curves, hence $\mathbb{G}_T$ is a multiplicative subgroup of the multiplicative group of a finite field.

The most standard assumptions over pairings are derived from the equivalent of the Diffie-Hellman assumptions from cyclic groups, described in Definition 2.11. This hypothesis is used to defined the SXDH assumption [Sco02] as follows.

Definition 3.2 (SXDH [BGdMM05, As. 1])**.** The *Symmetric eXternal Diffie-Hellman* (SXDH) assumption holds if the DDH assumption holds both in $\mathbb{G}$ and $\widehat{\mathbb{G}}$.

The advantages of the best PPT adversary against DDH in group $\mathbb{G}$ and $\widehat{\mathbb{G}}$ are written $\mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}}$ and $\mathrm{Adv}^{\mathsf{DDH}}_{\widehat{\mathbb{G}}}$ respectively. Both of those quantities are assumed negligible under the SXDH assumption.

In Chapter 6, the security of our group signature scheme relies on the SXDH assumption, which is a well-studied assumption. Moreover, this assumption is static, meaning that the size of the assumption is independent of the number of queries made py the adversary or any feature (e.g., the maximal number of users) of the system, and is non-interactive, in the sense that it does not involve any oracle.

This gives us stronger confidence in the security of schemes proven under this kind of assumptions. For instance, Cheon gave an attack against the q-Strong Diffie-Hellmann problem for large values of q [Che06] (which usually represents the number of adversarial queries).

In Chapter 6, we also rely on the following assumption, which generalizes the Discrete Logarithm problem to asymmetric groups.

Definition 3.3 (SDL)**.** In bilinear groups $(\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T)$ of prime order p, the *Symmetric Discrete Logarithm* (SDL) problem consists in, given $(g, \hat{g}, g^a, \hat{g}^a) \in (\mathbb{G} \times \widehat{\mathbb{G}})^2$ where $a \hookleftarrow \mathbb{Z}_p$, computing $a \in \mathbb{Z}_p$.

Like SXDH, this assumption is also static (i.e., constant-size) and non-interactive.

3.2 Lattice-Based Cryptography

During the last decade, lattice-based cryptography has emerged as a promising candidate for post-quantum cryptography. For example, on the first round of the NIST post-quantum competition, there are 28 out of 82 submissions stem from lattice-based cryptography [NIS17]. Lattice-based cryptography takes advantage of a simple mathematical structure in order to realize advanced functionalities, beyond encryption and signature schemes. For instance, fully homomorphic encryption [Gen09, GSW13] is only known to be possible in the lattice-based world for now.

In the context of provable security, lattice assumptions benefit from a worst-case-to-average-case reduction [Reg05, GPV08, MP12, AFG14]. Concurrently, worst-case lattice problems have been extensively analyzed in the last decade [ADSD15, ADRSD15, HK17], both classically and quantumly.

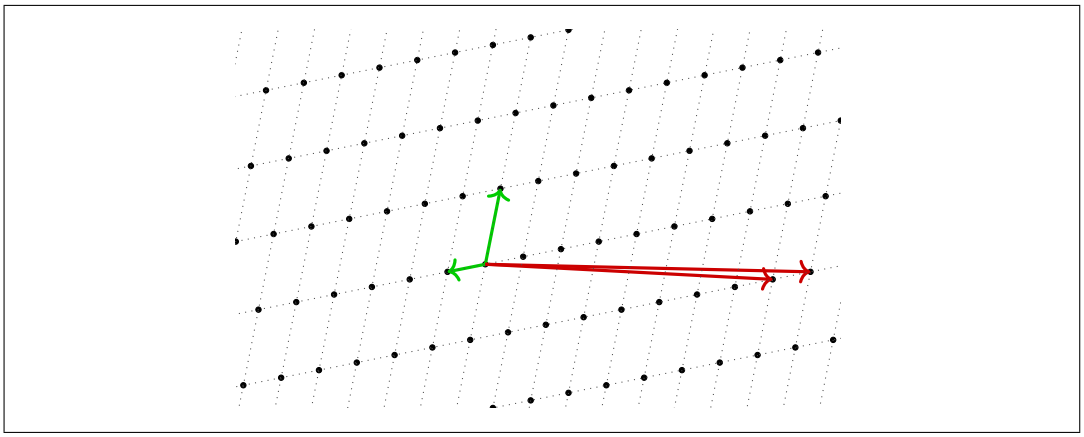


Figure 3.1 – A lattice Λ with two different basis.

This gives us a good confidence in lattice assumptions (given the *caveats* of Chapter 2) such as Learning-with-Errors (LWE) and Short Integer Solutions (SIS) which are defined in Section 3.2.1. The rest of this section will describe some useful tools that rely on *lattice trapdoors*.

3.2.1 Lattices and Hard Lattice Problems

A (full-rank) lattice Λ is defined as the set of all integer linear combinations of some linearly independent basis vectors $(\mathbf{b}_i)_{1\leq i\leq n}$ of $\mathbb{R}^n$. The integer n denotes the *dimension* of the lattice. A lattice basis is not unique, as illustrated in Figure 3.1 with a dimension 2 lattice. In the following, we work with q-ary lattices, for some prime number q, defined as follows.

Definition 3.4 (q-ary lattices)**.** Let two integers $m \geq n \geq 1$, a prime $q \geq 2$, a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$, define

$$\Lambda_q(\mathbf{A}) \triangleq \{\mathbf{e} \in \mathbb{Z}^m \mid \exists\, \mathbf{s} \in \mathbb{Z}_q^n \text{ s.t. } \mathbf{A}^T \cdot \mathbf{s} = \mathbf{e} \bmod q\} \text{ as well as}$$
$$\Lambda_q^{\perp}(\mathbf{A}) \triangleq \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{0}^n \bmod q\}, \text{ and}$$
$$\Lambda_q^{\mathbf{u}}(\mathbf{A}) \triangleq \{\mathbf{e} \in \mathbb{Z}^m \mid \mathbf{A} \cdot \mathbf{e} = \mathbf{u} \bmod q\}.$$

For any lattice point $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$, it holds that $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^{\perp}(\mathbf{A}) + \mathbf{t}$, meaning that $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is a shift of $\Lambda_q^{\perp}(\mathbf{A})$.

Definition 3.5 (Gaussian distribution over a lattice)**.** For a lattice Λ, a vector $\mathbf{c} \in \mathbb{R}^n$ and a real $\sigma > 0$, define the distribution function $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) \triangleq \exp(-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2)$. The discrete Gaussian distribution of support Λ, parameter σ and center $\mathbf{c}$ is defined as $D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{y}) = \rho_{\sigma,\mathbf{c}}(\mathbf{y})/\rho_{\sigma,\mathbf{c}}(\Lambda)$ for any $\mathbf{y} \in \Lambda$, where $\rho_{\sigma,\mathbf{c}}(\Lambda) \triangleq \sum_{\mathbf{x}\in\Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{x})$. We denote by $D_{\Lambda,\sigma}(\mathbf{y})$ the distribution centered in $\mathbf{c} = \mathbf{0}$.

Lemma 3.1 ([Ban93, Le. 1.5])**.** *For any lattice $\Lambda \subseteq \mathbb{R}^n$ and positive real number $\sigma > 0$, we have* $\Pr_{\mathbf{b}\hookleftarrow D_{\Lambda,\sigma}}[\|\mathbf{b}\| \leq \sigma\sqrt{n}] \geq 1 - 2^{-\Omega(n)}$.

In order to work with lattices in cryptography, hard lattice problems have to be defined [Ajt96]. In the following we state the *Shortest Independent Vectors Problem* (SIVP$_\gamma$).

This problem reduces to the *Learning-with-Errors* (LWE) problems and the Short Integer Solution (SIS) problem as explained later in Lemma 3.2 and 3.3. These links are important as those are "worst-case-to-average-case" reductions.

By itself, the SIVP_γ assumption is not very handy in order to construct new cryptographic designs. On the other hand, the LWE and SIS assumptions – which are "average-case" assumptions – are more suitable to design cryptographic schemes.

In order to define the SIVP_γ problem and assumption, let us first define the successive minima of a lattice, a generalization of the minimum of a lattice (i.e., the length of a shortest non-zero vector in a lattice).

Definition 3.6 (Successive minima)**.** For a lattice Λ of dimension n, let us define for each $i \in \{1, \ldots, n\}$ the i-th successive minimum as

$$\lambda_i(\Lambda) = \inf\{r \mid \dim(\mathrm{span}(\Lambda \cap \mathcal{B}(\mathbf{0}, r))) \geq i\},$$

where $\mathcal{B}(\mathbf{c}, r)$ denotes the ball of radius r centered in $\mathbf{c}$.

This leads us to the SIVP_γ problem, which is to find a set of sufficiently short linearly independent vectors given a lattice basis.

Definition 3.7 (SIVP_γ)**.** For a dimension-n lattice described by a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$, and a parameter $\gamma > 0$, the shortest independent vectors problem is to find n linearly independent vectors $v_1, \ldots, v_n$ such that $\|v_1\| \leq \|v_2\| \leq \ldots \leq \|v_n\|$ and $\|v_n\| \leq \gamma \cdot \lambda_n(\mathbf{B})$.

As explained before, the hardness of this assumption for worst-case lattices implies the hardness of the following two assumptions in their average-case setting, which are illustrated in Figure 3.2. In particular, it means that no polynomial-time algorithm can solve those problems with non-negligible probability and non-negligible advantage given that SIVP_γ is hard.

Definition 3.8 (The SIS and ISIS problem)**.** Let m, q, β be functions of $n \in \mathbb{N}$ and $\|\cdot\|$ be a norm (e.g., Euclidean norm $\|\cdot\|_2$ or infinite norm $\|\cdot\|_\infty$). The *Short Integer Solution* problem $\mathsf{SIS}_{n,m,q,\beta}$ is, given $\mathbf{A} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n \times m})$, find $\mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$ with $0 < \|\mathbf{x}\| \leq \beta$.

The *Inhomogeneous Short Integer Solution* $\mathsf{ISIS}_{n,m,q,\beta}$ problem is, given $\mathbf{A} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n \times m})$ and $\mathbf{u} \in \mathbb{Z}_q^n$, find $\mathbf{x} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$ with $0 < \|\mathbf{x}\| \leq \beta$.

Evidence of the hardness of the SIS and ISIS assumptions is given by the following Lemma, which reduced these problems from SIVP_γ.

Lemma 3.2 ([GPV08, Se. 9])**.** *If $q \geq \sqrt{n}\beta$ and $m, \beta \leq \mathsf{poly}(n)$, then $\mathsf{SIS}_{n,m,q,\beta}$ and $\mathsf{ISIS}_{n,m,q,\beta}$ problems are both at least as hard as standard worst-case lattice problem SIVP_γ with $\gamma = \tilde{\mathcal{O}}(\beta\sqrt{n})$.*

Definition 3.9 (The LWE problem)**.** Let $n, m \geq 1$, $q \geq 2$, and let χ be a probability distribution on $\mathbb{Z}$. For a fixed $\mathbf{s} \in \mathbb{Z}_q^n$, let $A_{\mathbf{s},\chi}$ be the distribution obtained by sampling $\mathbf{a} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^n)$ and $e \hookleftarrow \chi$, and outputting $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. The *Learning-with-Errors* problem $\mathsf{LWE}_{n,q,\chi}$ asks to distinguish m samples chosen according to $\mathcal{A}_{\mathbf{s},\chi}$ (for $\mathbf{s} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^n)$) and m samples chosen according to $\mathcal{U}(\mathbb{Z}_q^n \times \mathbb{Z}_q)$.

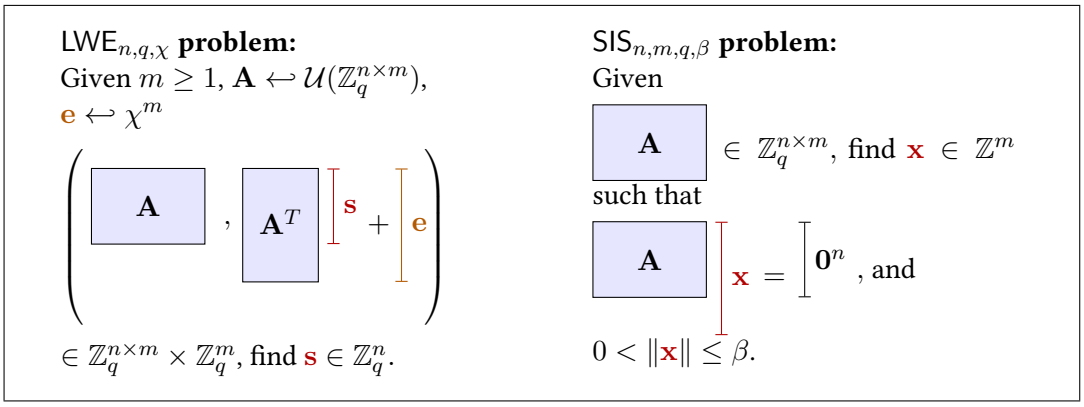


Figure 3.2 – Illustration of the LWE and SIS problems.

The worst-case-to-average-case reduction for LWE is stated by the following Lemma.

Lemma 3.3 ([Reg05, Pei09, BLP+13]). *If q is a prime power, $B \geq \sqrt{n}\omega(\log n)$, $\gamma = \widetilde{\mathcal{O}}(nq/B)$, then there exists an efficient sampleable B-bounded distribution χ (i.e., χ outputs samples with norm at most B with overwhelming probability) such that* LWE$_{n,q,\chi}$ *is as least as hard as* SIVP$_\gamma$.

3.2.2 Lattice Trapdoors

In this section, we recall the specifications of different algorithms that use "*lattice trapdoors*". A trapdoor for a lattice Λ is a *short* basis of this lattice. The knowledge of such a basis allows sampling elements in $D_{\Lambda,\sigma}$ within some restrictions given in Lemma 3.5. The existence of this sampler allows sampling short vectors which is believed to be infeasible without knowing such a short basis. Indeed, Lemma 3.5 shows that it is possible to sample a (statistically close to) uniform matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ along with a short basis for $\Lambda_q^\perp(\mathbf{A})$. Thus, a vector sampled from $D_{\Lambda_q^\perp(\mathbf{A}),\sigma}$, which is short with overwhelming probabilities according to Lemma 3.1, is a solution to SIS$_{n,m,q,\sigma\sqrt{n}}$.

Gentry *et al.* [GPV08] showed that Gaussian distributions with lattice support can be sampled efficiently given a sufficiently short basis of the lattice.

Recall. Given a matrix $\mathbf{A}$, $\widetilde{\mathbf{A}}$ denotes the Gram-Schmidt orthogonalization of $\mathbf{A}$.

Lemma 3.4 ([BLP+13, Le. 2.3]). *There exists a* PPT *(probabilistic polynomial-time) algorithm GPVSample that inputs a basis $\mathbf{B}$ of a lattice $\Lambda \subseteq \mathbb{Z}^n$ and a rational $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \Omega(\sqrt{\log n})$, and outputs vectors $\mathbf{b} \in \Lambda$ with distribution $D_{\Lambda,\sigma}$.*

The following Lemma states that it is possible to efficiently compute a statistically uniform $\mathbf{A}$ along with a short basis of its orthogonal lattice $\Lambda_q^\perp(\mathbf{A})$.

Lemma 3.5 ([AP09, 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 $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$ such that $\mathbf{A}$ is within statistical distance $2^{-\Omega(n)}$ to $\mathcal{U}(\mathbb{Z}_q^{n\times m})$, and $\|\widetilde{\mathbf{T_A}}\| \leq \mathcal{O}(\sqrt{n \log q})$.*

Lemma 3.5 is often combined with the sampler from Lemma 3.5. Micciancio and Peikert [MP12] proposed a more efficient approach for this combined task, which is to be be

preferred in practice but, for the sake of simplicity, schemes are presented using TrapGen and GPVSample in this thesis.

We also make use of an algorithm that extends a trapdoor for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ to a trapdoor of any $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ for which a m-subset of its columns is $\mathbf{A}$. For the sake of simplicity we will consider the case where $\mathbf{A}$ is the left $n \times m$ submatrix of $\mathbf{B}$.

Lemma 3.6 ([CHKP10, Le. 3.2]). *There exists a* PPT *algorithm ExtBasis that takes as inputs a matrix* $\mathbf{B} \in \mathbb{Z}_q^{n \times m'}$ *whose first* m *columns span* $\mathbb{Z}_q^n$*, and a basis* $\mathbf{T_A}$ *of* $\Lambda_q^\perp(\mathbf{A})$ *where* $\mathbf{A}$ *is the left* $n \times m$ *submatrix of* $\mathbf{B}$*, and outputs a basis* $\mathbf{T_B}$ *of* $\Lambda_q^\perp(\mathbf{B})$ *with* $\|\widetilde{\mathbf{T_B}}\| \leq \|\widetilde{\mathbf{T_A}}\|$.

In some of our security proofs, analogously to [Boy10, BHJ+15], we also use a technique due to Agrawal, Boneh and Boyen [ABB10] that implements an all-but-one trapdoor mechanism (akin to the one of Boneh and Boyen [BB04]) in the lattice setting.

Lemma 3.7 ([ABB10, Th. 19]). *There exists a* PPT *algorithm SampleRight that takes as inputs matrices* $\mathbf{A}, \mathbf{C} \in \mathbb{Z}_q^{n \times m}$*, a low-norm matrix* $\mathbf{R} \in \mathbb{Z}^{m \times m}$*, a short basis* $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ *of* $\Lambda_q^\perp(\mathbf{C})$*, a vector* $\mathbf{u} \in \mathbb{Z}_q^n$ *and a rational* σ *such that* $\sigma \geq \|\widetilde{\mathbf{T_C}}\| \cdot \Omega(\sqrt{\log n})$*, and outputs a short vector* $\mathbf{b} \in \mathbb{Z}^{2m}$ *such that* $\begin{bmatrix} \mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{C} \end{bmatrix} \cdot \mathbf{b} = \mathbf{u} \bmod q$ *and with distribution statistically close to* $D_{L,\sigma}$ *where* L *denotes the shifted lattice* $\Lambda_q^{\mathbf{u}}\left(\begin{bmatrix} \mathbf{A} \mid \mathbf{A} \cdot \mathbf{R} + \mathbf{C} \end{bmatrix}\right)$.

Chapter 4

Zero-Knowledge Arguments

A *Zero-Knowledge proof* [GMR85] (or **ZK proofs**) is an *interactive proof* between a prover and a verifier at the end of which the verifier should be convinced of the truth of a statement (within some probability, called *soundness error*), while the prover is guaranteed that the verifier learns nothing more that the authenticity of the statement.

One of the early applications of ZK proofs in cryptography was the design of identification systems [FS86]. The goal is for a user A to prove the knowledge of a secret (such as a password) to user B without revealing any piece of information about the secret, otherwise user B would be able to impersonate A. Since then, the use of zero-knowledge proofs is now widespread in privacy-preserving protocols like anonymous credentials [Cha85, CL01], revocable group signatures [NFHF09], e-cash [CHL05b], oblivious transfer [CDN09] ...

If these primitives flourish in the context of number-theory-based cryptography (such as RSA groups or pairing groups), they are still elusive in the lattice world.

In this section, we first present the general principles and basic tools to handle ZK proofs. Then we will describe two families of ZK proofs that may prove useful in the context of pairing-based and lattice-based cryptography. Namely, Schnorr-like proofs and Stern-like proofs.

4.1 Definitions

4.1.1 Zero-Knowledge proofs and arguments

Definition 4.1 (Zero-knowledge proofs and arguments)**.** Let $R = \{(x, w) \in \mathcal{L} \times \mathcal{R}\}$ be a binary relation. A *zero-knowledge proof* for a relation R is an interactive protocol between a prover $P(x, w)$ and a verifier $V(x)$ where V outputs a bit b at the end of the interaction. This is written as $\langle P(x, w), V(x)\rangle = b$. The aforementioned protocol should also verify the following properties.

Completeness. For any $(x, w) \in R$, $\Pr[\langle P(x, w), V(x)\rangle = 1] \geq 1 - \mathsf{negl}(|\lambda|)$.

Soundness. For all $x \in \mathcal{L}$, for any $\bar{w} \in \mathcal{R}$ such that $(x, \bar{w}) \notin R$, and for any cheating prover $P^\star(x, \bar{w})$, $\Pr[\langle P^\star(x, \bar{w}), V(x)\rangle = 1] \leq s < 1 - \mathsf{negl}(|x|)$, where s is called the *soundness error*. We want s to be as small as possible, ideally negligible.

Zero-Knowledge. Let $\mathsf{trans}(\cdot,\cdot)$ be the transcript of the interaction during the proof. There exists a PPT simulator S such that for all (possibly cheating) PPT verifier $V^\star$, the distributions $\{\mathsf{trans}(P(x,w),V^\star(x))\}_{(x,w)\in R}$ and $\{S^{V^\star}(x)\}_{(x,w)\in R}$ are computationally indistinguishable.

If, in the *soundness* definition, the adversary $P^\star$ is restricted to be a PPT algorithm, then the proof system is called an *argument* system.

We can notice that the soundness error can be reduced to be negligible by repeating the proof.

If the two ensembles in the definition of *zero-knowledge* are the same, then the proof is *perfect zero-knowledge*.

Definition 4.2 (Proof of knowledge [GMR85, BG92])**.** Let κ be a function from $\{0,1\}^\star$ to $[0,1]$. A complete interactive proof system (P,V) is said to be a *proof of knowledge* for the relation R with knowledge error κ if it verifies the knowledge soundness property.

Knowledge soundness. There exists a PPT algorithm $\mathcal{E}$, called the knowledge extractor. This algorithm takes as input x and rewindable black-box access to the prover, and targets to compute a w such that $(x,w)\in R$. For any prover $\hat{P}$, let $\varepsilon(x)$ be the probability that V accepts on input x. There exists a constant c such that, whenever $\varepsilon(x)>\kappa(x)$, M will output a correct w with expected time at most $\frac{|x|^c}{\varepsilon(x)-\kappa(x)}$, where access to $\hat{P}$ counts as one step.

This extractor represents the fact that an effective prover actually knows the secret (while a zero-knowledge proof only attests the existence of a witness w). In the following, ZKAoK denotes *Zero-Knowledge Argument of Knowledge*.

Another useful property that a proof system can have in the context of privacy-preserving cryptography is witness indistinguishability (WI). This property states that if a proof system has multiple witnesses, it is impossible to tell apart which one has been used during the proof.

Definition 4.3 (Witness indistinguishable proofs [FS90])**.** Let (P,V) be a complete interactive proof system for relation R. It is said to be *witness indistinguishable* if, for every PPT algorithm $\hat{V}$ and every two sequences $\{w_x\}_{(x,w_x)\in R}$, $\{w'_x\}_{(x,w'_x)\in R}$, the following ensembles are computationally indistinguishable:

$$\{\mathsf{trans}(P(x,w_x),\hat{V}(x)\}_x \qquad \text{and} \qquad \{\mathsf{trans}(P(x,w'_x),\hat{V}(x)\}_x.$$

The WI property is implied by the zero-knowledge property. Whereas the latter, *witness indistinguishability* is preserved through parallel repetitions of the protocol [FS90].

4.1.2 Σ-protocols

A way to construct zero-knowledge proofs – that will be described with more details in Section 4.2 – is a blackbox transformation from a Σ-*protocol* and a *commitment scheme* [Dam00, GMY03]. The resulting proof remains secure against malicious verifiers.

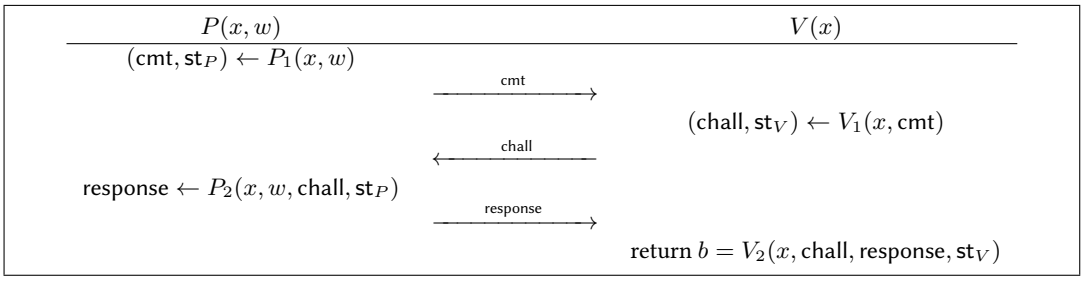


Figure 4.1 – Abstract description of a Σ-protocol.

Definition 4.4 (Σ-protocol [Cra96])**.** Let $R = \{(x, w)\}$ be a binary relation. A Σ*-protocol* is a 3-move interactive protocol between P and V that follows Figure 4.1 and verifies the following properties.

Completeness. For any $(x, w) \in R$, $P(x, w)$ and $V(x)$ that follows the protocol, the verifier always accepts.

2-Special soundness. For any x and any pair of accepting transcripts on input x of the form $(\mathsf{cmt}, \mathsf{chall}, \mathsf{response})$ and $(\mathsf{cmt}, \mathsf{chall}', \mathsf{response}')$, there exists a PPT algorithm extract that inputs the two aforementioned transcripts and outputs an element w such that $(x, w) \in R$.

Honest-Verifier Zero-Knowledge. There exists a PPT simulator S, such that the two probability distributions $\{\mathsf{trans}(P(x, w), V(x))\}$ and $\{S(x)\}$ with honest P and V are statistically indistinguishable.

An example of Σ-protocol will be given in Section 4.2, and its transformation into a Zero-Knowledge proof using a commitment scheme as well.

4.1.3 Commitment schemes

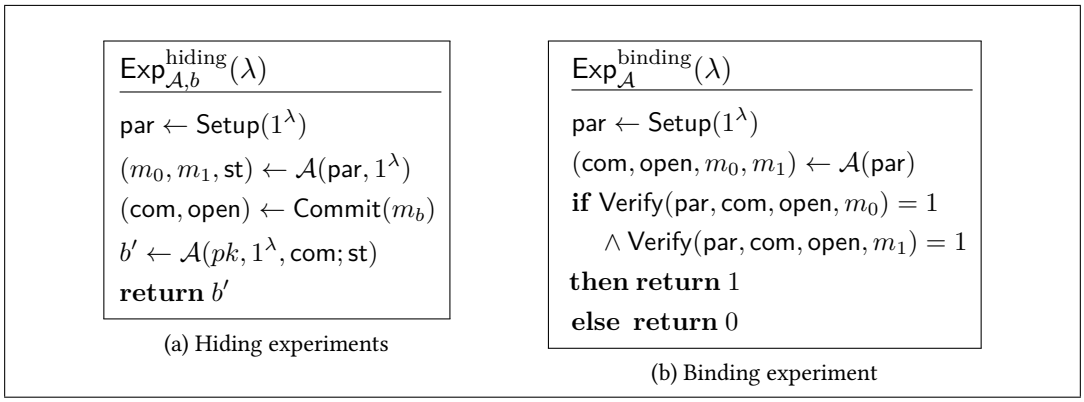
Commitment schemes [Blu81] are the digital analogue of a safe. The goal is to commit a message M into a commitment string com that verifies the *hiding* and *binding* properties. The former is, that once a message is committed, it is impossible to know what is inside, while the latter states that, it is impossible to alter a commitment string to modify the underlying message.

Definition 4.5 (Commitment schemes)**.** A *commitment scheme* is given by a triple of algorithms $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ that act as follows:

Setup(1^λ)**:** This algorithm outputs the commitment scheme's common public parameters par.

Commit(par, M)**:** From a message M and parameters par, this algorithms outputs a commitment com and an opening open. The randomness ρ used in the commitment is sometimes made explicit.

Verify$(\mathsf{par}, \mathsf{com}, \mathsf{open}, M)$**:** Using parameters par a message M, its commitment com and its opening open, this algorithms returns bit b.



(a) Hiding experiments

(b) Binding experiment

Figure 4.2 – Security experiments for commitment schemes.

These algorithms should verify correctness, hiding and binding properties, where $\mathsf{Exp}^{\text{hiding}}_{\mathcal{A},b}$ and $\mathsf{Exp}^{\text{binding}}_{\mathcal{A}}$ are described in Figure 4.2.

Correctness. For any public parameters $\mathsf{par} \leftarrow \mathsf{Setup}(1^\lambda)$, message M, commitment and opening $(\mathsf{com}, \mathsf{open}) \leftarrow \mathsf{Commit}(\mathsf{par}, M)$, it holds that $\mathsf{Open}(\mathsf{par}, \mathsf{open}, M) = 1$.

Hiding. For any PPT adversary $\mathcal{A}$ against the hiding experiment, we have that

$$\mathrm{Adv}^{\text{hiding}}_{\mathcal{A}}(\lambda) = \left|\Pr\left[\mathsf{Exp}^{\text{hiding}}_{\mathcal{A},1}(\lambda) = 1\right] - \Pr\left[\mathsf{Exp}^{\text{hiding}}_{\mathcal{A},0}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda)\,,$$

over the randomness of Commit.

Binding. For any PPT adversary $\mathcal{A}$ against the binding experiment,

$$\Pr\left[\mathsf{Exp}^{\text{binding}}_{\mathcal{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda)\,.$$

Commitment schemes are thus used to *force* the verifier of the Σ-protocol to behave honestly: it commits its challenge at the outset of the interaction, and opens it at the challenge phase, so that it cannot change its challenge with respect to the commitment of the prover.

An example of commitment scheme that will prove useful in Section 4.3 is the Kawachi, Tanaka, Xagawa SIS-based commitment scheme [KTX08].

This construction relies on the following hash function:

Definition 4.6 (SIS-based hash function)**.** Let $n, \ell, q \in \mathbb{Z}$ be parameters such that the $\mathsf{SIS}_{n,\ell,q,\sqrt{\ell}}$ assumption holds. Let $\mathbf{A} \in \mathbb{Z}_q^{n\times\ell}$, and let $f_{\mathbf{A}} : \{0,1\}^\ell \to \mathbb{Z}_q^n$ be the function that maps its input string x into a binary vector $\mathbf{x} \in \mathbb{Z}_q^n$ and outputs $\mathbf{A}\mathbf{x} \bmod q \in \mathbb{Z}_q^n$.

One can notice that $f_{\mathbf{A}}$ is indeed a collision-resistant one-way function under the SIS assumption, as finding two inputs $x \neq \tilde{x}$ such that $\mathbf{A}\cdot\mathbf{x} = \mathbf{A}\cdot\tilde{\mathbf{x}} \bmod q$ leads to a non-zero vector $\mathbf{x}' = \mathbf{x} - \tilde{\mathbf{x}} \in \mathbb{Z}$ such that $\|\mathbf{x}'\|_2 \leq \sqrt{\ell}$.

It is thus possible to apply the *Merkle-Damgård construction* [Mer79, Mer89, Dam89] on $f_{\mathbf{A}}$ to obtain a *collision resistant hash function* $h_{\mathbf{A}} : \{0,1\}^\star \to \mathbb{Z}_q^n$ that is secure under the $\mathsf{SIS}_{n,\ell,q,\sqrt{\ell}}$ assumption.

It is then possible to use this hash function $h_{\mathbf{A}}$ to construct the following string commitment scheme.

Definition 4.7 (SIS-based commitment scheme)**.** Given parameters $n, m, q \in \mathbb{Z}$, let us define the following commitment scheme due to [KTX08].

Setup(1^λ)**:** Pick two random matrices $\mathbf{A}_M, \mathbf{A}_\rho \in \mathcal{U}(\mathbb{Z}_q^{n\times m})$ and define the public parameters as the matrix $\mathbf{A} = [\mathbf{A}_M \mid \mathbf{A}_\rho]$.

Commit$(\mathbf{A}, M; \rho)$**:** To commit to a string $M \in \{0,1\}^\star$ under randomness $\rho \in \{0,1\}^m$, first parse $\mathbf{A} \in \mathbb{Z}_q^{n\times 2m}$ as $[\mathbf{A}_M \mid \mathbf{A}_\rho]$ as in the **Setup** algorithm, then compute $\mathsf{com} = h_{\mathbf{A}_M}(M) + f_{\mathbf{A}_\rho}(\rho) \in \mathbb{Z}_q^n$, where $h_{\mathbf{A}_M}$ and $f_{\mathbf{A}_\rho}$ are the hash function and the one-way collision resistant function described in Definition 4.6. The opening corresponds to the randomness ρ used in the computation.

Verify$(\mathbf{A}, \mathsf{com}, \mathsf{open}, M)$**:** First parse $\mathbf{A}$ as in the **Setup** algorithm. Then accept if and only if $\mathsf{open} \in \{0,1\}^m$ and $\mathsf{com} = h_{\mathbf{A}_M}(M) + f_{\mathbf{A}_\rho}(\rho)$.

Lemma 4.1 ([KTX08, Le. 3.4])**.** *If $m > 2n \log q$, the above commitment scheme is* statistically hiding *and* binding *under the $\mathsf{SIS}_{n,m,q,\sqrt{m}}$ assumption in the trusted setup model.*

4.1.4 Non-Interactive Proofs

Another useful primitives are the non-interactive version of zero-knowledge proofs.

Definition 4.8 (Non Interactive Zero Knowledge)**.** A *non-interactive zero-knowledge* proof (or **NIZK proof**) for a relation $R = \{(x, w) \in \mathcal{L} \times \mathcal{W}\}$ is a pair of PPT algorithms (P, V) such that P takes as inputs $x \in \mathcal{L}$ and $w \in \mathcal{W}$ and outputs a proof π, and V takes as inputs x and π and outputs a bit b. These algorithms should verify the following properties.

Completeness. For any $(x, w) \in R$, $\Pr[V(x, P(x, w)) = 1] \geq 1 - \mathsf{negl}(|x|)$.

Soundness. For all $x \in \mathcal{L}$, for any $\bar{w} \in \mathcal{W}$ such that $(x, \bar{w}) \notin R$, and for any cheating prover $P^\star(x, \bar{w})$, $\Pr[V(x, P^\star(x)) = 1] < \mathsf{negl}(|x|)$.

Zero-Knowledge. There exists a PPT simulator S such that the probability ensembles $\{(x, P(x, w))\}_{(x,w)\in R}$ and $\{S(x)\}_{(x,w)\in R}$ are computationally indistinguishable.

In the random oracle model [BR93, PS96], it is possible to transform a ZK proof into an NIZK proof [FS86]. This techniques is called the Fiat-Shamir transform.

Definition 4.9 (Fiat-Shamir Transform [FS86])**.** Let (P, V) be a three-move ZK proof system for relation $R = \{(x, w)\}$ as in Figure 4.1 and $\mathcal{H}$ be a cryptographic hash function. Let $\hat{P}$ be the following non-interactive prover that takes as inputs x and w:

1. First run $P_1(x, w)$ to get a random commitment cmt and a state information st_P;
2. Generate the challenge as $\mathsf{chall} \leftarrow \mathcal{H}(x, \mathsf{cmt})$;
3. Run $\mathsf{response} \leftarrow P_2(x, w, \mathsf{chall}, \mathsf{st}_P)$;

4. Return the proof $\pi = (\mathsf{cmt}, \mathsf{response})$.

And let $\hat{V}$ be the following non-interactive verifier that takes as inputs x and π:

1. Parse π as $(\mathsf{cmt}, \mathsf{response})$;
2. Generate the challenge $\mathsf{chall} = \mathcal{H}(x, \mathsf{cmt})$;
3. Return $V_2(x, \mathsf{chall}, \mathsf{response}, \emptyset)$.

Then $(\hat{P}, \hat{V})$ forms a non-interactive zero-knowledge proof in the ROM.

For the sake of completeness, we also mention NIZK in the standard model, such as Groth-Sahai proofs [GOS06, GS08] for bilinear groups, but these will not be used in the context of this thesis.

In the trusted setup model (also known as common reference string model) described in Section 2.3, there is also another type of NIZK proofs that is useful for us, for instance in Chapter 6. Quasi-adaptive NIZK (QA-NIZK) [JR13] are NIZK where the common reference string crs may depend on the language for which proofs have to be generated (that is, the distribution $\mathsf{D}_{\mathsf{crs}}$ is a function of the language we want to prove). A formal definition can be found in [JR13, KW15, LPJY15], where completeness, soundness and zero-knowledge properties are adapted to take into account the crs.

Definition 4.10 (Quasi-Adaptive Non-Interactive Zero-Knowledge Argument)**.** A *Quasi-Adaptive Non-Interactive Zero-Knowledge Argument* argument (or **QA-NIZK**) over a collection of relations $\mathcal{R} = \{R_\rho\}$ parametrized by a string ρ consists in four PPT algorithms $(\mathsf{Gen}_0, \mathsf{Gen}_1, P, V)$.

The algorithms Gen_0 and Gen_1 both generate the crs. Gen_0 inputs 1^λ and outputs Γ the fixed part of the crs from which ρ is sampled according to a distribution D_Γ, while Gen_1 inputs Γ and ρ to output a language-dependent part ψ (or directly the $\mathsf{crs} = (\Gamma, \psi, \rho)$). The prover P and the verifier V act as in Definition 4.8 with the difference that, they also take as input the common reference string crs.

We consider proof systems where the prover and the verifier both take a label τ as additional input. Formally, a tuple $(\mathsf{Gen}_0, \mathsf{Gen}_1, P, V)$ of PPT algorithms is a QA-NIZK proof system for $\mathcal{R}$ if, there exists a PPT simulator (S_1, S_2) such that for any PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$ and $\mathcal{A}_3$, the following properties hold:

Quasi-Adaptive Completeness.

$$\Pr\left[\begin{array}{c} V(\mathsf{crs}, x, \pi, \tau) = 1 \\ \text{if } R_\rho(x, w) = 1 \end{array} \middle| \begin{array}{c} \Gamma \leftarrow \mathsf{Gen}_0(1^\lambda); \rho \leftarrow \mathsf{D}_\Gamma; \\ \mathsf{crs} \leftarrow \mathsf{Gen}_1(\Gamma, \rho); (x, w, \tau) \leftarrow \mathcal{A}_1(\mathsf{crs}, \rho) \\ \pi \leftarrow P(\mathsf{crs}, x, w); \end{array}\right] = 1.$$

Quasi-Adaptive Soundness.

$$\Pr\left[\begin{array}{c} (\forall w : (x, w) \notin R_\rho) \\ \wedge V(\mathsf{crs}, x, \pi, \tau) = 1 \end{array} \middle| \begin{array}{c} \Gamma \leftarrow \mathsf{Gen}_0(1^\lambda); \rho \leftarrow \mathsf{D}_\Gamma; \\ \mathsf{crs} \leftarrow \mathsf{Gen}_1(\Gamma, \rho); (x, \pi, \tau) \leftarrow \mathcal{A}_2(\mathsf{crs}) \end{array}\right] \leq \mathsf{negl}(\lambda)\,.$$

Common input: A prime-order group $\mathbb{G}$ of order p with a generator g.

Schnorr's Protocol for DLOG

$P(h, a)$		$V(h)$
$r \hookleftarrow \mathbb{Z}_p^\star$		
$\rho = g^r \in \mathbb{G}$		
	$\xrightarrow{\rho}$	
		$c \hookleftarrow \mathbb{Z}_p$
	$\xleftarrow{c}$	
$d \leftarrow c \cdot a + r \bmod p$		
	$\xrightarrow{d}$	
		if $g^d = h^c \cdot \rho$ **then**
		return 1
		else
		return 0

Figure 4.3 – The Schnorr Σ-protocol for discrete logarithm.

Quasi-Adaptive Zero-Knowledge.

$$\Pr[\mathcal{A}_3^{P(\psi,\cdot)}(\Gamma,\psi,\rho) = 1 \mid \Gamma \leftarrow \mathsf{Gen}_0(1^\lambda); \rho \leftarrow \mathsf{D}_\Gamma; \mathsf{crs} \leftarrow \mathsf{Gen}_1(\Gamma,\rho)] \approx_s \Pr\left[\mathcal{A}_3^{S(\psi,\tau_{sim},\cdot)}(\Gamma,\psi,\rho) = 1 \;\middle|\; \begin{array}{c} \Gamma \leftarrow \mathsf{Gen}_0(1^\lambda); \rho \leftarrow \mathsf{D}_\Gamma; \\ (\psi,\tau_{sim}) \leftarrow S_1(\Gamma,\rho) \end{array}\right]$$

Where

- $P(\psi,\cdot)$ emulates the actual prover. It inputs (x, w, τ) and outputs a proof π if $(x, w) \in R_\rho$. Otherwise, it outputs $\perp$.
- $S(\psi, \tau_{sim}, \cdot)$ is an oracle that takes as input (x, w, τ) and outputs a simulated proof $S_2(\psi, \tau_{sim}, x, \tau)$ if $(x, w) \in R_\rho$ and $\perp$ otherwise.

4.2 Schnorr Proofs

Schnorr's methodology [Sch96] to construct proofs is based on the Σ-protocol technique to design zero-knowledge proofs. It has been introduced in order to prove the knowledge of a discrete logarithm (which can bee seen at the relation $R_{\mathsf{dlog}} = \{(h, a) \in \mathbb{G} \times \mathbb{Z}_p \mid h = g^a\}$ with $\mathbb{G} = \langle g \rangle$ be a cyclic group of prime order $p > 2$) and is described in Figure 4.3.

An interpretation of this methodology is the following: given a commitment scheme (Setup, Commit, Verify), where the randomness r used in Commit is made explicit, the first move of the prover P consists in binding the randomness used in the commitment scheme r using the transmitted value $\rho = g^r$, then the verifier asks the prover to commit to a challenge

Common input: A public element $\mathbf{a} \in R$ where $R = \mathbb{Z}_p[\mathbf{x}]/\langle \mathbf{x}^n + 1 \rangle$.

Schnorr's Protocol for Ring-SIS

$P(\mathbf{t} = \mathbf{a} \cdot \mathbf{s}_1 + \mathbf{s}_2, (\mathbf{s}_1, \mathbf{s}_2))$		$V(\mathbf{t})$
$\mathbf{y}_1, \mathbf{y}_2 \hookleftarrow D_y \in R$		
$\mathbf{w} = \mathbf{a} \cdot \mathbf{y}_1 + \mathbf{y}_2 \in R$		
	$\xrightarrow{\mathbf{w}}$	
		$\mathbf{c} \hookleftarrow D_c \in R$ (small)
	$\xleftarrow{\mathbf{c}}$	
$\mathbf{z}_1 \leftarrow \mathbf{s}_1\mathbf{c} + \mathbf{y}_1 \in R$		
$\mathbf{z}_2 \leftarrow \mathbf{s}_2\mathbf{c} + \mathbf{y}_2 \in R$		
[**if** $\mathbf{z}_1, \mathbf{z}_2 \notin G^2$ **then**		
$\mathbf{z}_1, \mathbf{z}_2 \leftarrow \perp, \perp$]		
	$\xrightarrow{\mathbf{z}_1, \mathbf{z}_2}$	
		if $\mathbf{z}_1 \in G \wedge \mathbf{z}_2 \in G \wedge$
		$\mathbf{a} \cdot \mathbf{z}_1 + \mathbf{z}_2 = \mathbf{tc} + \mathbf{w}$ **then**
		return 1
		else
		return 0

Figure 4.4 – The Schnorr Σ-protocol for Ring-SIS.

message c using the randomness carried by ρ, and the prover sends the opening for this commitment open. Finally, the verifier accepts if and only if $\mathsf{Verify}(\mathsf{par}, \mathsf{com}, \mathsf{open}, c) = 1$.

In the protocol described in Figure 4.3, the underlying commitment is the Pedersen commitment scheme [Ped91]: a commitment of a message $m \in \mathbb{Z}_p$ is $g^m \cdot h^r \in \mathbb{G}$ and the opening is the randomness r used to commit.

For efficiency reasons, Schnorr's protocol is used along with Fiat-Shamir heuristic in the pairing-based group signature described in Chapter 6.

This methodology has also been adapted to the ideal lattice-setting by Lyubashevsky [Lyu08, Lyu09] along with a technique called *rejection sampling* in order to construct ZK proofs from ideal lattice assumptions and is described in Figure 4.4. In this description D_y and D_c are the distributions from which $\mathbf{y}_1, \mathbf{y}_2$ and $\mathbf{c}$ have to be sampled respectively, and G describes the set of *good* responses $\mathbf{z}_1, \mathbf{z}_2$ in order not to leak informations about $\mathbf{s}_1, \mathbf{s}_2$. The part between brackets is called the *rejection* phase, and ensure that the transmitted $\mathbf{z}_1, \mathbf{z}_2$ will not leak any information about $\mathbf{s}_1, \mathbf{s}_2$ to V. This part induced a noticeable error-rate where the prover aborts the proof. As the protocol is proven *witness indistinguishable* [Lyu09], one can run the protocol multiple times in parallel and hope that one of them will not abort [FS90].

One can notice that this is *not* a Σ-protocol in the strict sense as the knowledge extractor

outputs *witnesses* that can be up to $\tilde{\mathcal{O}}(n)$ larger than the actual witness in infinity norm. This behavior is sometimes called "*imperfect soundness*" or "*soundness slack*".

However, this method suffers from *limited expressiveness*: the relations that can be proved with this proof system are essentially restricted to be knowledge of a Ring-SIS secret, which is not sufficient to prove, for instance, the knowledge of a signature on a committed message. Moreover, the gap in the extraction makes it hard, although, to prove that an underlying message under an encryption is binary [dPLNS17].

4.3 Stern-like Proofs

Stern's protocol has originally been introduced in the context of code-base cryptography [Ste96]. Initially, it was designed for Syndrome Decoding Problem (SDP): given a matrix $\mathbf{M} \in \mathbb{F}_2^{n \times m}$ and a syndrome $\mathbf{v} \in \mathbb{F}_2^n$, the goal is to find a binary vector $\mathbf{w} \in \mathbb{F}_2^m$ with fixed hamming weight w such that $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2$.

This problem bears similarities with the ISIS problem defined in Definition 3.8 where the constraint on the norm of $\mathbf{x}$ is replaced by a constraint on Hamming weight, and operations are in $\mathbb{F}_2$ instead of $\mathbb{Z}_q$.

After the first work of Kawachi, Tanaka and Xagawa [KTX08] that extended Stern's proofs to statements $\bmod q$, the results of Ling, Nguyen, Stehlé and Wang [LNSW13] enable the use of Stern's protocol to prove general SIS or LWE statements (meaning proving knowledge of a solution to these problems). These advances in the expressiveness of Stern-like protocols has been used to further improve them and therefore enable privacy-based primitives for which no constructions previously existed in the post-quantum world, such as dynamic group signatures [LLM+16a], group encryption [LLM+16b], electronic cash [LLNW17], etc.

Unlike Schnorr-like proofs that we described in the previous section, Stern-like proofs are mainly combinatorial and rely on the fact that every permutation on a binary vector $\mathbf{w} \in \{0,1\}^m$ leaves its Hamming weight w invariant. As a consequence, for $\pi \in \mathfrak{S}_m$, $\mathbf{w}$ satisfies these conditions if and only if $\pi(\mathbf{x})$ also does. Therefore, the randomness of π is used to verify these two constraints (being binary and having fixed Hamming weight) in a zero-knowledge fashion. We can notice that this can be extended to vectors $\mathbf{w} \in \{-1,0,1\}^m$ having fixed numbers of -1 and 1. This property allowed [LNSW13] to propose the generalization of this protocol to any $\mathsf{ISIS}_{n,m,q,\beta}$ statements. In Section 4.3.2, we describes these permutations while abstracting the set of ZK-provable statements as the set VALID that satisfies conditions (4.3).

It is worth noticing that this argument on knowledge does not strictly follow the definition of a Σ-protocol in Definition 4.4. The challenge space is ternary as described in Section 4.3.2, hence the protocol verifies 3-special soundness. Thus, standard theorems on Σ-protocols have to be adapted in this setting.

In this Section, we describe in a high-level manner the behavior of Stern-like protocols before detailing it.

- $\mathsf{B}^2_{\mathfrak{m}}$: the set of vectors in $\{0,1\}^{2\mathfrak{m}}$ with Hamming weight $\mathfrak{m}$.
- $\mathsf{B}^3_{\mathfrak{m}}$: the set of vectors in $\{-1,0,1\}^{3\mathfrak{m}}$ which has exactly $\mathfrak{m}$ coordinates equal to j for each $j \in \{-1,0,1\}$.

Figure 4.5 – Notations for Stern-like protocols.

4.3.1 The Decomposition-Extension Framework

The original Stern protocol was designed to prove knowledge of a SDP preimage. That is, to prove the knowledge of a vector $\mathbf{w} \in \{0,1\}^m$ that verifies

$$\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod 2. \tag{4.1}$$

A first improvement by [KTX08] was to extend this protocol using a statistically hiding SIS-based commitment scheme as described in 4.6 to prove in (statistical) zero-knowledge that

$$\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q. \tag{4.2}$$

The details of this proof is given in Section 4.3.2, but it can be summarized in the following Lemma.

Lemma 4.2 ([KTX08, Se. 4]). *There exists a statistical ZKAoK with perfect completeness and soundness error 2/3 to prove the knowledge of a secret vector* $\mathbf{w} \in \{0,1\}^m$ *that verifies relation* (4.2) *for public input* $(\mathbf{M}, \mathbf{v}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$.

Ling, Nguyen, Stehlé and Wang [LNSW13] noticed that the ZKAoK of Lemma 4.2 works in a straightforward manner to prove knowledge of a vector in $\{-1,0,1\}^m$.

To prove the knowledge of an ISIS preimage, i.e. the knowledge of a bounded vector $\mathbf{w} \in [-B, B]^m$ that satisfies relation (4.2), the goal is to rewrite $\mathbf{w}$ as $\bar{\mathbf{w}} = \mathbf{K} \cdot \mathbf{w} \bmod q$ with a public transformation matrix $\mathbf{K}$ such that $\bar{\mathbf{w}} \in \{-1,0,1\}^{m'}$ and of known numbers of elements equal to j for each $j \in \{-1,0,1\}$. This reduces to use Lemma 4.2 to prove the knowledge of $\bar{\mathbf{w}} \in \{-1,0,1\}^{m'}$ for public input $(\mathbf{M} \cdot \mathbf{K}, \mathbf{v})$.

To construct such a transfer matrix $\mathbf{K}$, [LNSW13] showed that *decomposing* a vector $\mathbf{x} \in [-B, B]^m$ as a vector $\tilde{\mathbf{x}} \in \{-1,0,1\}^{m \cdot \delta_B}$ and *extending* the resulting vector into $\bar{\mathbf{x}} \in \mathsf{B}^3_{m\delta_B}$ leads to a new statement that can be proven using the variant of Stern's protocol described in [KTX08]. The resulting matrix $\mathbf{K} = \left[\mathbf{K}_{m,B} \mid \mathbf{0}^{m \times 2m\delta_B}\right] \in \mathbb{Z}^{m \times 3m\delta_B}$, where $\mathbf{K}_{m,B}$ is the $\{-1,0,1\}$-decomposition matrix $\mathbf{K}_{m,B} = \mathbf{I}_m \otimes [B_1 \mid \cdots \mid B_{\delta_B}]$ with $B_j = \left\lfloor \frac{B + 2^{j-1}}{2^j} \right\rfloor$, for all $j \in \{1, \ldots, j\}$, can be computed from public parameters.

In Chapter 8, we extend Stern-like protocols to handle statements where the matrix $\mathbf{M}$ of (4.2) is kept hidden. For this purpose, we define the decomposition-extension method in more detail in Section 8.3.

4.3.2 Abstraction of Stern's Protocol

Let K, D, q be positive integers with $D \geq K$ and $q \geq 2$, and let VALID be a subset of $\mathbb{Z}^D$. Suppose that $\mathcal{S}$ is a finite set such that every element $\phi \in \mathcal{S}$ can be associated with a

1. **Commitment:** Prover samples $\mathbf{r}_w \leftarrow \mathcal{U}(\mathbb{Z}_q^D)$, $\phi \leftarrow \mathcal{U}(\mathcal{S})$ and randomnesses ρ_1, ρ_2, ρ_3 for COM. Then, he sends $\mathrm{CMT} = \big(C_1, C_2, C_3\big)$ to the verifier, where

$$C_1 = \mathsf{COM}(\phi, \mathbf{M} \cdot \mathbf{r}_w \bmod q; \rho_1), \;\; C_2 = \mathsf{COM}(\Gamma_\phi(\mathbf{r}_w); \rho_2),$$
$$C_3 = \mathsf{COM}(\Gamma_\phi(\mathbf{w} + \mathbf{r}_w \bmod q); \rho_3).$$

2. **Challenge:** The verifier sends a challenge $Ch \hookleftarrow \mathcal{U}(\{1, 2, 3\})$ to the prover.
3. **Response:** Depending on Ch, the prover sends RSP computed as follows:
   - $Ch = 1$: Let $\mathbf{t}_w = \Gamma_\phi(\mathbf{w})$, $\mathbf{t}_r = \Gamma_\phi(\mathbf{r}_w)$, and $\mathrm{RSP} = (\mathbf{t}_w, \mathbf{t}_r, \rho_2, \rho_3)$.
   - $Ch = 2$: Let $\phi_2 = \phi$, $\mathbf{w}_2 = \mathbf{w} + \mathbf{r}_w \bmod q$, and $\mathrm{RSP} = (\phi_2, \mathbf{w}_2, \rho_1, \rho_3)$.
   - $Ch = 3$: Let $\phi_3 = \phi$, $\mathbf{w}_3 = \mathbf{r}_w$, and $\mathrm{RSP} = (\phi_3, \mathbf{w}_3, \rho_1, \rho_2)$.

Verification: Receiving RSP, the verifier proceeds as follows:

- $Ch = 1$: Check that

$$\mathbf{t}_w \in \mathsf{VALID},$$
$$C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2), \qquad C_3 = \mathsf{COM}(\mathbf{t}_w + \mathbf{t}_r \bmod q; \rho_3).$$

- $Ch = 2$: Check that

$$C_1 = \mathsf{COM}(\phi_2, \mathbf{M} \cdot \mathbf{w}_2 - \mathbf{v} \bmod q; \rho_1), \qquad C_3 = \mathsf{COM}(\Gamma_{\phi_2}(\mathbf{w}_2); \rho_3).$$

- $Ch = 3$: Check that

$$C_1 = \mathsf{COM}(\phi_3, \mathbf{M} \cdot \mathbf{w}_3; \rho_1), \qquad C_2 = \mathsf{COM}(\Gamma_{\phi_3}(\mathbf{w}_3); \rho_2).$$

In each case, the verifier outputs 1 if and only if all the conditions hold.

Figure 4.6 – Stern-like ZKAoK for the relation $\mathrm{R}_{\mathrm{abstract}}$.

permutation $\Gamma_\phi \in \mathfrak{S}_D$ satisfying the following conditions:

$$\begin{cases} \mathbf{w} \in \mathsf{VALID} \iff \Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}, \\ \text{If } \mathbf{w} \in \mathsf{VALID} \text{ and } \phi \text{ is uniform in } \mathcal{S}, \text{ then } \Gamma_\phi(\mathbf{w}) \text{ is uniform in } \mathsf{VALID}. \end{cases} \tag{4.3}$$

We aim to construct a statistical Zero-Knowledge Argument of Knowledge (ZKAoK) for the following abstract relation:

$$\mathrm{R}_{\mathrm{abstract}} = \{((\mathbf{M}, \mathbf{v}), \mathbf{w}) \in \mathbb{Z}_q^{K \times D} \times \mathbb{Z}_q^K \times \mathsf{VALID} : \mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q.\}$$

Note that, Stern's original protocol corresponds to the special case when the set $\mathsf{VALID} = \{\mathbf{w} \in \{0,1\}^D : \mathsf{wt}(\mathbf{w}) = k\}$, where $\mathsf{wt}(\cdot)$ denotes the Hamming weight and $k < D$ is a given integer, $\mathcal{S} = \mathfrak{S}_D$ is the set of all permutations of D elements and $\Gamma_\phi(\mathbf{w}) = \phi(\mathbf{w})$.

The conditions in (4.3) play a crucial role to prove in zero-knowledge that $\mathbf{w} \in \mathsf{VALID}$. To this end, the prover samples a random $\phi \hookleftarrow \mathcal{U}(\mathcal{S})$ and lets the verifier check that $\Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}$ without learning any additional information about $\mathbf{w}$ due to the randomness of ϕ. Furthermore, to prove in a zero-knowledge manner that the linear equation is satisfied, the

prover samples a masking vector $\mathbf{r}_w \hookleftarrow \mathcal{U}(\mathbb{Z}_q^D)$, and convinces the verifier instead that $\mathbf{M} \cdot (\mathbf{w} + \mathbf{r}_w) = \mathbf{M} \cdot \mathbf{r}_w + \mathbf{v} \bmod q$.

The interaction between prover $\mathcal{P}$ and verifier $\mathcal{V}$ is described in Figure 4.6. The protocol uses a statistically hiding and computationally binding string commitment scheme COM (e.g., the SIS-based scheme from [KTX08] described in Definition 4.7).

Theorem 4.3. *The protocol in Figure 4.6 is a statistical* ZKAoK *with perfect completeness, soundness error* $2/3$*, and communication cost* $\mathcal{O}(D \cdot \log q)$*. Namely:*

- *There exists a polynomial-time simulator that, on input* $(\mathbf{M}, \mathbf{v})$*, outputs an accepted transcript statistically close to that produced by the real prover.*
- *There exists a polynomial-time knowledge extractor that, on input a commitment* CMT *and* 3 *valid responses* $(\mathrm{RSP}_1, \mathrm{RSP}_2, \mathrm{RSP}_3)$ *to all* 3 *possible values of the challenge* Ch*, outputs* $\mathbf{w}' \in \mathsf{VALID}$ *such that* $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$.

The proof of the theorem relies on standard simulation and extraction techniques for Stern-like protocols [KTX08, LNSW13, LLM+16a].

Proof. Note that, by construction, the protocol is perfectly complete: if an honest prover follows the protocol, then he always gets accepted by the verifier. It is also easy to see that the communication cost is bounded by $\widetilde{\mathcal{O}}(D \cdot \log q)$.

We will now prove that the protocol is a statistical zero-knowledge argument of knowledge for the relation $\mathrm{R}_{\mathrm{abstract}}$ and is given below.

Zero-Knowledge Property. We construct a PPT simulator SIM interacting with a (possibly dishonest) verifier $\widehat{\mathcal{V}}$ such that, given only the public input, SIM outputs with probability negligibly close to $2/3$ a simulated transcript that is statistically close to the one produced by the honest prover in the real interaction.

The simulator first chooses a random $\overline{Ch} \in \{1, 2, 3\}$. This is a prediction of the challenge value that $\widehat{\mathcal{V}}$ will *not* choose.

Case $\overline{Ch} = 1$: Using basic linear algebra over $\mathbb{Z}_q$, SIM computes a vector $\mathbf{w}' \in \mathbb{Z}_q^D$ such that $\mathbf{M} \cdot \mathbf{w}' = \mathbf{v} \bmod q$. Next, it samples $\mathbf{r} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^D)$, $\pi \hookleftarrow \mathcal{U}(\mathcal{S})$, and randomnesses ρ_1, ρ_2, ρ_3 for COM.

Then, it sends the commitment $\mathrm{CMT} = (C'_1, C'_2, C'_3)$ to $\widehat{\mathcal{V}}$, where

$$C'_1 = \mathsf{COM}(\pi, \mathbf{M} \cdot \mathbf{r}; \rho_1), \qquad C'_2 = \mathsf{COM}(\Gamma_\pi(\mathbf{r}); \rho_2),$$
$$C'_3 = \mathsf{COM}(\Gamma_\pi(\mathbf{w}' + \mathbf{r}); \rho_3).$$

Receiving a challenge Ch from $\widehat{\mathcal{V}}$, the simulator responds as follows:

- If $Ch = 1$: Output $\perp$ and abort.
- If $Ch = 2$: Send $\mathrm{RSP} = (\pi, \mathbf{w}' + \mathbf{r}, \rho_1, \rho_3)$.
- If $Ch = 3$: Send $\mathrm{RSP} = (\pi, \mathbf{r}, \rho_1, \rho_2)$.

Case $\overline{Ch} = 2$: SIM samples $\mathbf{w}' \hookleftarrow \mathcal{U}(\mathsf{VALID})$, $\mathbf{r} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^D)$, $\pi \hookleftarrow \mathcal{U}(\mathcal{S})$, and randomnesses ρ_1, ρ_2, ρ_3 for COM.

Then, it sends the commitment $\mathrm{CMT} = (C'_1, C'_2, C'_3)$ to $\widehat{\mathcal{V}}$, where

$$C'_1 = \mathsf{COM}(\pi, \mathbf{M} \cdot \mathbf{r}; \rho_1), \qquad C'_2 = \mathsf{COM}(\Gamma_\pi(\mathbf{r}); \rho_2),$$
$$C'_3 = \mathsf{COM}(\Gamma_\pi(\mathbf{w}' + \mathbf{r}); \rho_3)$$

as previously.

Receiving a challenge Ch from $\widehat{\mathcal{V}}$, the simulator responds as follows:

- If $Ch = 1$: Send $\mathrm{RSP} = (\Gamma_\pi(\mathbf{w}'), \Gamma_\pi(\mathbf{r}), \rho_2, \rho_3)$.
- If $Ch = 2$: Output $\perp$ and abort.
- If $Ch = 3$: Send $\mathrm{RSP} = (\pi, \mathbf{r}, \rho_1, \rho_2)$.

Case $\overline{Ch} = 3$: SIM samples $\mathbf{w}' \hookleftarrow \mathcal{U}(\mathsf{VALID})$, $\mathbf{r} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^D)$, $\pi \hookleftarrow \mathcal{U}(\mathcal{S})$, and randomnesses ρ_1, ρ_2, ρ_3 for COM.

Then, it sends the commitment $\mathrm{CMT} = (C'_1, C'_2, C'_3)$ to $\widehat{\mathcal{V}}$, where

$$C'_2 = \mathsf{COM}(\Gamma_\pi(\mathbf{r}); \rho_2), \qquad C'_3 = \mathsf{COM}(\Gamma_\pi(\mathbf{w}' + \mathbf{r}); \rho_3)$$

as in the previous two cases, while

$$C'_1 = \mathsf{COM}(\pi, \mathbf{M} \cdot (\mathbf{w}' + \mathbf{r}) - \mathbf{v}; \rho_1),$$

Receiving a challenge Ch from $\widehat{\mathcal{V}}$, it responds as follows:

- If $Ch = 1$: Send RSP computed as in the case $(\overline{Ch} = 2, Ch = 1)$.
- If $Ch = 2$: Send RSP computed as in the case $(\overline{Ch} = 1, Ch = 2)$.
- If $Ch = 3$: Output $\perp$ and abort.

We observe that, in all the above cases, since COM is statistically hiding, the distribution of the commitment CMT and the distribution of the challenge Ch from $\widehat{\mathcal{V}}$ are statistically close to those in the real interaction. Hence, the probability that the simulator outputs $\perp$ is negligibly close to $1/3$. Moreover, one can check that whenever the simulator does not halt, it provides an accepted transcript, the distribution of which is statistically close to that of the prover in the real interaction. In other words, we have designed a simulator that can successfully emulate the honest prover with probability negligibly far from $2/3$.

Argument of Knowledge. Let us assume that

$$\mathrm{RSP}_1 = (\mathbf{t}_x, \mathbf{t}_r, \rho_2^{(1)}, \rho_3^{(1)}), \qquad \mathrm{RSP}_2 = (\phi_2, \mathbf{y}, \rho_1^{(2)}, \rho_3^{(2)}),$$
$$\text{and } \mathrm{RSP}_3 = (\phi_3, \mathbf{w}_3, \rho_1^{(3)}, \rho_2^{(3)})$$

are 3 valid responses to the same commitment $\mathrm{CMT} = (C_1, C_2, C_3)$, with respect to all 3 possible values of the challenge. The validity of these responses implies that:

$$\begin{cases} \mathbf{t}_x \in \mathsf{VALID}; \\ C_1 = \mathsf{COM}(\phi_2, \mathbf{M} \cdot \mathbf{w}_2 - \mathbf{v}; \rho_1^{(2)}) = \mathsf{COM}(\phi_3, \mathbf{M} \cdot \mathbf{w}_3; \rho_1^{(3)}); \\ C_2 = \mathsf{COM}(\mathbf{t}_r; \rho_2^{(1)}) = \mathsf{COM}(\Gamma_{\phi_3}(\mathbf{w}_3); \rho_2^{(3)}); \\ C_3 = \mathsf{COM}(\mathbf{t}_x + \mathbf{t}_r; \rho_3^{(1)}) = \mathsf{COM}(\Gamma_{\phi_2}(\mathbf{w}_2); \rho_3^{(2)}). \end{cases}$$

Since COM is computationally binding, we can deduce that:

$$\begin{cases} \mathbf{t}_x \in \mathsf{VALID}; \\ \phi_2 = \phi_3; \\ \mathbf{t}_r = \Gamma_{\phi_3}(\mathbf{w}_3); \\ \mathbf{t}_x + \mathbf{t}_r = \Gamma_{\phi_2}(\mathbf{w}_2); \\ \mathbf{M} \cdot \mathbf{w}_2 - \mathbf{v} = \mathbf{M} \cdot \mathbf{w}_3 \bmod q. \end{cases}$$

Let $\mathbf{w}' = \mathbf{w}_2 - \mathbf{w}_3$, then we have $\Gamma_{\phi_2}(\mathbf{w}') = \mathbf{t}_x \in \mathsf{VALID}$ which implies that $\mathbf{w}' \in \mathsf{VALID}$. Furthermore, we have $\mathbf{M} \cdot \mathbf{w}' = \mathbf{M} \cdot (\mathbf{w}_2 - \mathbf{w}_3) = \mathbf{v} \bmod q$.

This concludes the proof. □

Part II

Group Signatures and Anonymous Credentials

Chapter 5

Dynamic Group Signatures

In this part, we will present two constructions of dynamic group signatures. The construction that will be explained in Chapter 6 is an adaptation of the Libert, Peters and Yung short group signature in the standard model from classical pairing assumptions [LPY15] to the random oracle model, which allows us to gain in efficiency while keeping the assumptions simple. This gives us a constant-size group signature scheme that is shown to be competitive with other constructions based on less standard assumptions such as the q-SDH assumption. An implementation is available and detailed in Chapter 6.

The second construction, described in Chapter 7, is a lattice-based dynamic group signature based on the scheme of Ling, Nguyen and Wang [LNW15] for static groups. This construction was improved to match the requirements for dynamic groups, which closes an open-problem [GKV10]. This construction has been the first fully secure group signature scheme from lattices.

Before describing those schemes, this chapter recalls the definition of dynamic group signatures and their related security definitions.

5.1 Background

Dynamic group signatures are a primitive that allows a user to authenticate a message on behalf of a set of users it belongs to (the *group*). This can be publicly verified while the user remains anonymous inside his group. On the other hand, the user remains accountable for the signatures he generates as there exists an authority, the *opening authority*, that can lift the anonymity of a given signature using his own secret key. In the dynamic setting, a group signature scheme has a second authority: the *group manager*, that allows a user to join the group after an interaction with him. These interactions are summarized in Figure 5.1.

The concept of group signatures was introduced by Chaum and van Heyst in 1991 [CvH91]. Nevertheless, the work of Ateniese, Camenisch, Joye and Tsudik in 2001 [ACJT00] were the first to provide scalable and secure group signatures. In 2003, Bellare, Micciancio and Warinschi [BMW03] proposed formal security definitions for *static* group signatures, where the group is defined *once-and-for-all* at the setup phase. This model was extended to dynamic groups by Bellare, Shi and Zhang (BSZ) and Kiayias and Yung (KY) in 2005 [BSZ05, KY06]. These two security models are slightly different, and we choose in this thesis to build our

proofs in the [KY06] model as described in Section 5.2.

The [BMW03] model summarizes the security of a group signatures in two notions: *anonymity* and *traceability*. The former notions models the fact that, without the opening authority's secret, even if everyone colludes, no one can identify the author of a signature; the latter sums up the fact that, even if everyone is corrupted (even the opening authority), it is infeasible to forge a valid signature that does not open to a valid user.

In the dynamic setting, the *group signing-keys issuing* phase is replaced by an interactive *join* protocol where a user who wants to join the group interacts with the group manager. In this context, the two notions of the BMW model are retained, and a third one is added: the "*non-frameability*" property. This notion expresses the infeasibility to frame a group of honest users (which can be reduced to a singleton) in order to provide a signature that opens to one of them, *even if the group manager and the opening authority are colluding*.

One possible application of this primitive is anonymous access control for public transportation systems. In order to commute, a person should prove possession of a valid subscription to the transportation service. Thus, at registration to the service, the commuter joins the group of "*users with a valid subscription*". When he uses the transportation service, he is asked to sign the timestamp of his entry in the name of the group. In case of misbehavior, another entity – let say the police – is able to lift the anonymity of the signatures logged by the reading machine. Then, the public transportation company is unable to learn anything from the signatures, except the validity of the subscription of a user. On the other hand, the police does not have access to the logs except if the public transportation company hands them to them.

Other applications of group signatures can be found as authentication of low-range communications for intelligent cars or anonymous access control of a building. As we can see, most applications necessitate the use of *dynamically growing* groups in order to be meaningful.

Bootle, Cerulli, Chaidos, Ghadafi and Groth [BCC+16] raised the problem of revocation and proposed a model that handles the issues that arose from the introduction of revocation called "*fully-dynamic*" group signatures. As the main difficulty is to allow users to dynamically enroll in the group – revocation has been known to be implemented in a modular manner [LLNW14] – this approach is not considered here, even if it is of interest [LNWX17].

5.2 Formal Definition and Correctness

This section recalls the syntax and the security definitions of dynamic group signatures based on the model of Kiayias and Yung [KY06].

In the setting of *dynamic groups*, the syntax of group signatures includes an interactive protocol which allows users to register as new members of the group at any time. The syntax and the security model are those defined by Kiayias and Yung [KY06]. Like the very similar BSZ model [BSZ05], the Kiayias-Yung (KY) model assumes an interactive *join* protocol whereby a prospective user becomes a group member by interacting with the group manager. This protocol provides the user with a membership certificate, cert_i, and a membership secret, sec_i.

We denote by $N_{\mathsf{gs}} \in \mathsf{poly}(\lambda)$ the maximal number of group members that the system will

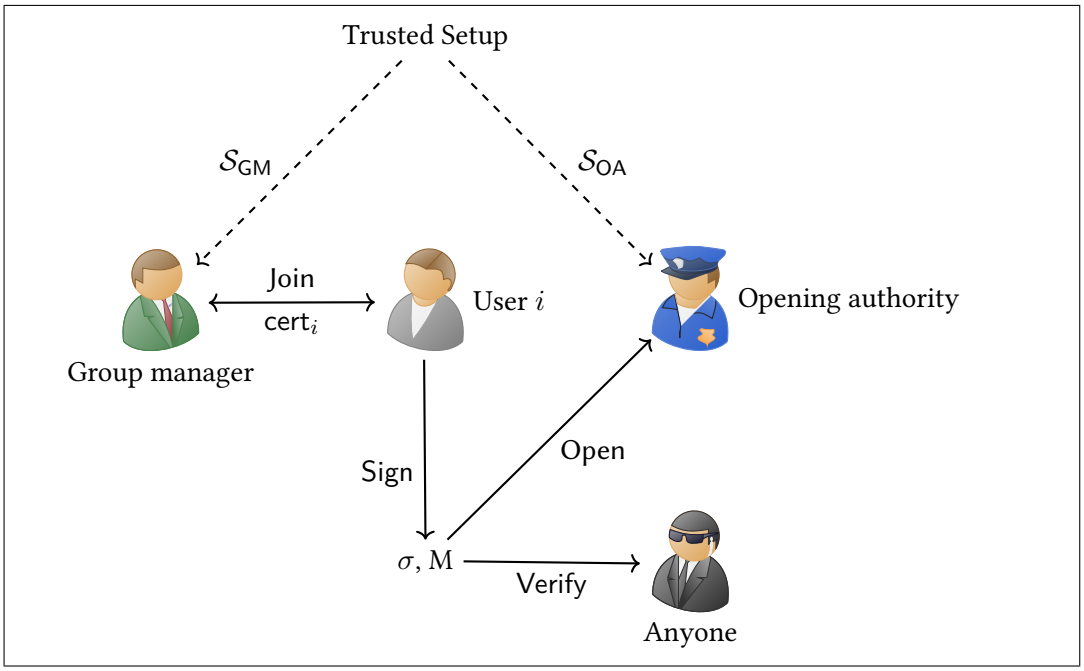


Figure 5.1 – Relations between the protagonists in a dynamic group signature scheme.

be able to handle.

Definition 5.1 (Dynamic Group Signature)**.** A *dynamic group signature* scheme consists of the algorithms or protocols (Setup, Join, Sign, Verify, Open) described as follows.

Setup$(1^\lambda, 1^{N_{\mathsf{gs}}})$**:** given a security parameter λ and a maximal number of group members $N_{\mathsf{gs}} \in \mathbb{N}$, this algorithm is run by a *trusted party* to generate a group public key $\mathcal{Y}$, the group manager's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$. Each key is given to the appropriate authority while $\mathcal{Y}$ is made public. The algorithm also initializes a public state st comprising a set data structure $\mathsf{st}_{\mathsf{users}} = \emptyset$ and a string data structure $\mathsf{st}_{\mathsf{trans}} = \epsilon$.
In the following, all algorithms have access to the public parameters $\mathcal{Y}$.

Join$^{\mathsf{J}_{\mathsf{user}},\mathsf{J}_{\mathsf{GM}}}$**:** is an *interactive* protocol between the group manager GM and a user $\mathcal{U}_i$ where the latter becomes a group member. The protocol involves two interactive Turing machines $\mathsf{J}_{\mathsf{user}}$ and J_{GM} that both take the group public key $\mathcal{Y}$ as input. The execution $\langle \mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, \mathsf{st}, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}) \rangle$, ends with user $\mathcal{U}_i$ obtaining a membership secret sec_i, that no one else knows, and a membership certificate cert_i. If the protocol is successful, the group manager updates the public state st by updating the following state informations $\mathsf{st}_{\mathsf{users}} := \mathsf{st}_{\mathsf{users}} \cup \{i\}$ as well as $\mathsf{st}_{\mathsf{trans}} := \mathsf{st}_{\mathsf{trans}} || (i, \mathsf{transcript}_i)$.

Sign$(\mathsf{cert}_i, \mathsf{sec}_i, M)$**:** given a membership certificate cert_i, a membership secret sec_i and a message M, this *probabilistic* algorithm outputs a signature σ.

Verify(σ, M)**:** given a signature σ, a message M and a group public key $\mathcal{Y}$, this *deterministic* algorithm returns either 0 or 1.

Open($\mathcal{S}_{\mathsf{OA}}, M, \sigma$): takes as input a message M, a valid signature σ w.r.t. $\mathcal{Y}$, the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ and the public state st. It outputs $i \in \mathsf{st}_{\mathsf{users}} \cup \{\perp\}$, which is the identity of a group member or a symbol indicating an opening failure.

Each membership certificate contains a unique *tag* that identifies the user.

The correctness requirement basically captures that, if all parties *honestly* run the protocols, all algorithms are correct with respect to their specification described as above.

The Kiayias-Yung model [KY06] considers three security notions: the security against *misidentification attacks* requires that, even if the adversary can introduce users under its control in the group, it cannot produce a signature that traces outside the set of dishonest users. The notion of security against *framing attacks* implies that honest users can never be accused of having signed messages that they did not sign, even if the whole system conspired against them. And finally the *anonymity* property is also formalized by granting the adversary access to a signature opening oracle as in the models of [BSZ05].

Correctness for Dynamic Group Signatures. Following the Kiayias-Yung terminology [KY06], we say that a public state st is *valid* if it can be reached from $\mathsf{st} = (\emptyset, \epsilon)$ by a Turing machine having oracle access to J_{GM}. Also, a state st' is said to *extend* another state st if it is within reach from st.

Moreover, as in [KY06], when we write $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, it means that there exists coin tosses ϖ for J_{GM} and J_{user} such that, for some valid public state st', the execution of the interactive protocol $\langle \mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, \mathsf{st}', \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})\rangle_{\varpi}$ provides $\mathsf{J}_{\mathsf{user}}$ with $(i, \mathsf{sec}_i, \mathsf{cert}_i)$.

Definition 5.2 (Correctness)**.** A dynamic group signature scheme is correct if the following conditions are all satisfied:

(1) In a valid state st, $|\mathsf{st}_{users}| = |\mathsf{st}_{trans}|$ always holds and two distinct entries of st_{trans} always contain certificates with distinct tag.

(2) If $\langle \mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, \mathsf{st}, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}})\rangle$ is run by two honest parties following the protocol and $\langle i, \mathsf{cert}_i, \mathsf{sec}_i\rangle$ is obtained by $\mathsf{J}_{\mathsf{user}}$, then we have $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$.

(3) For each $(i, \mathsf{cert}_i, \mathsf{sec}_i)$ such that $\mathsf{cert}_i \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}_i$, satisfying condition 2, we have $\mathsf{Verify}(\mathsf{Sign}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M), M, \mathcal{Y}) = 1$.

(4) For any outcome $(i, \mathsf{cert}_i, \mathsf{sec}_i)$ of $\langle \mathsf{J}_{\mathsf{user}}(.,.), \mathsf{J}_{\mathsf{GM}}(., \mathsf{st}, ., .)\rangle$ for some valid state information st, if $\sigma = \mathsf{Sign}(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M)$, then $\mathsf{Open}(M, \sigma, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, \mathsf{st}') = i$.

5.3 Associated Security Notions

5.3.1 Oracles for Security Experiments

We formalize security properties via experiments where the adversary interacts with a stateful interface $\mathcal{I}$ that maintains the following variables:

- $\mathsf{state}_{\mathcal{I}}$: is a data structure representing the state of the interface as the adversary invokes the various oracles available in the attack games. It is initialized as $\mathsf{state}_{\mathcal{I}} =$

$(\mathsf{st}, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^{\lambda}, 1^{N_{\mathsf{gs}}})$. It includes the (initially empty) set st_{users} of group members and a dynamically growing database st_{trans} storing the transcripts of previously executed join protocols.

- $n = |\mathsf{st}_{users}| < N_{\mathsf{gs}}$ denotes the current cardinality of the group.
- Sigs: is a database of signatures created by the signing oracle. Each entry consists of a triple (i, M, σ) indicating that message M was signed by user i.
- U^a: is the set of users that were introduced by the adversary in the system in an execution of the join protocol.
- U^b: is the set of honest users that the adversary, acting as a dishonest group manager, introduced in the system. For these users, the adversary obtains the transcript of the join protocol but not the user's membership secret.

In attack games, adversaries are granted access to the following oracles:

- Q_{pub}, Q_{keyGM} and Q_{keyOA}: when these oracles are invoked, the interface looks up $\mathsf{state}_{\mathcal{I}}$ and returns the group public key $\mathcal{Y}$, the GM's private key $\mathcal{S}_{\mathsf{GM}}$ and the opening authority's private key $\mathcal{S}_{\mathsf{OA}}$ respectively.
- $Q_{\mathsf{a\text{-}join}}$: allows the adversary to introduce users under its control in the group. On behalf of the GM, the interface runs J_{GM} in interaction with the $\mathsf{J}_{\mathsf{user}}$-executing adversary who plays the role of the prospective user in the join protocol. If this protocol successfully ends, the interface increments n, updates st by inserting the new user n in both sets st_{users} and U^a. It also sets $\mathsf{st}_{\mathsf{trans}} := \mathsf{st}_{\mathsf{trans}} || (n, \mathsf{transcript}_n)$.
- $Q_{\mathsf{b\text{-}join}}$: allows the adversary, acting as a corrupted group manager, to introduce new honest group members of its choice. The interface triggers an execution of $\langle \mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}} \rangle$ and runs $\mathsf{J}_{\mathsf{user}}$ in interaction with the adversary who runs J_{GM}. If the protocol successfully completes, the interface increments n, adds user n to st_{users} and U^b and sets $\mathsf{st}_{\mathsf{trans}} := \mathsf{st}_{\mathsf{trans}} || (n, \mathsf{transcript}_n)$. It stores the membership certificate cert_n and the membership secret sec_n in a *private* part of $\mathsf{state}_{\mathcal{I}}$.
- Q_{sig}: given a message M, an index i, the interface checks whether the private area of $\mathsf{state}_{\mathcal{I}}$ contains a certificate cert_i and a membership secret sec_i. If no such elements $(\mathsf{cert}_i, \mathsf{sec}_i)$ exist or if $i \notin U^b$, the interface returns $\perp$. Otherwise, it outputs a signature σ on behalf of user i and also sets $\mathsf{Sigs} \leftarrow \mathsf{Sigs} || (i, M, \sigma)$.
- Q_{open}: when this oracle is invoked on input of a valid pair (M, σ), the interface runs algorithm Open using the current state st. When S is a set of pairs of the form (M, σ), $Q_{\mathsf{open}}^{\neg S}$ denotes a restricted oracle that only applies the opening algorithm to pairs (M, σ) which are not in S.
- Q_{read} and Q_{write}: are used by the adversary to read and write the content of $\mathsf{state}_{\mathcal{I}}$. At each invocation, Q_{read} outputs the whole $\mathsf{state}_{\mathcal{I}}$ but the public/private keys and the private part of $\mathsf{state}_{\mathcal{I}}$ where membership secrets are stored after $Q_{\mathsf{b\text{-}join}}$-queries. By using Q_{write}, the adversary can modify $\mathsf{state}_{\mathcal{I}}$ at will as long as it does not remove or alter elements of st_{users}, st_{trans} or invalidate the public state st: for example, the

adversary is allowed to create dummy users as long as it does not re-use already existing certificate tags.

Based on the above syntax, the security properties are formalized as follows.

5.3.2 Security Against Misidentification Attacks

Experiment $\mathsf{Exp}^{\text{mis-id}}_{\mathcal{A}}(\lambda)$

$\mathsf{state}_{\mathcal{I}} = (\mathsf{st}, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^{\lambda}, 1^{N_{\mathsf{gs}}})$
$(M^{\star}, \sigma^{\star}) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{a\text{-}join}}, Q_{\mathsf{read}}, Q_{\mathsf{keyOA}})$
if $\mathsf{Verify}(\sigma^{\star}, M^{\star}, \mathcal{Y}) = 0$ **then**
  return 0
$i = \mathsf{Open}(M^{\star}, \sigma^{\star}, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, \mathsf{st}')$
if $i \notin U^{a}$ **then**
  return 1
else
  return 0

Figure 5.2 – Experiment for security against misidentification attacks.

In a misidentification attack, the adversary can corrupt the opening authority using the Q_{keyOA} oracle and introduce malicious users in the group via $Q_{\mathsf{a\text{-}join}}$-queries. It aims at producing a valid signature $\sigma^{\star}$ that does not open to any adversarially-controlled user.

Definition 5.3. A dynamic group signature scheme is secure against *misidentification attacks* if, for any PPT adversary $\mathcal{A}$ involved in Experiment $\mathsf{Exp}^{\text{mis-id}}_{\mathcal{A}}(\lambda)$ described in Figure 5.2, we have:

$$\mathrm{Adv}^{\mathcal{A}}_{\text{mis-id}}(\lambda) \triangleq \Pr\left[\mathsf{Exp}^{\text{mis-id}}_{\mathcal{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda)\,.$$

5.3.3 Non-Frameability

Experiment $\mathsf{Exp}^{\mathrm{fra}}_{\mathcal{A}}(\lambda)$

$\mathsf{state}_{\mathcal{I}} = (\mathsf{st}, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^{\lambda}, 1^{N_{\mathsf{gs}}})$

$(M^{\star}, \sigma^{\star}) \leftarrow \mathcal{A}(Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{keyOA}}, Q_{\mathsf{b\text{-}join}}, Q_{\mathsf{sig}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$

if $\mathsf{Verify}(\sigma^{\star}, M^{\star}, \mathcal{Y}) = 0$ **then**

  return 0

if $i = \mathsf{Open}(M^{\star}, \sigma^{\star}, \mathcal{S}_{\mathsf{OA}}, \mathcal{Y}, \mathsf{st}') \notin U^b$ **then**

  return 0

if $\bigwedge_{j \in U^b \text{ s.t. } j=i} (j, M^{\star}, *) \notin \mathsf{Sigs}$ **then**

  return 1

else

  return 0

Figure 5.3 – Experiment for security against framing attacks.

Framing attacks consider the situation where the entire system is colluding against some honest user. The adversary can corrupt the group manager as well as the opening authority (via oracles Q_{keyGM} and Q_{keyOA}, respectively). It can also introduce honest group members (via $Q_{\mathsf{b\text{-}join}}$-queries), observe the system while these users sign messages and create dummy users using Q_{write}. The adversary eventually aims at framing an honest group member.

Definition 5.4. A dynamic group signature scheme is secure against *framing attacks* if, for any PPT adversary $\mathcal{A}$ involved in the experiment $\mathsf{Exp}^{\mathrm{fra}}_{\mathcal{A}}(\lambda)$ described Figure 5.3), it holds that

$$\mathrm{Adv}^{\mathcal{A}}_{\mathrm{fra}}(\lambda) = \Pr\left[\mathsf{Exp}^{\mathrm{fra}}_{\mathcal{A}}(\lambda) = 1\right] \leq \mathsf{negl}(\lambda)\,.$$

5.3.4 Full Anonymity

Experiment $\mathsf{Exp}^{\mathrm{anon}}_{\mathcal{A},d}(\lambda)$

$\mathsf{state}_{\mathcal{I}} = (\mathsf{st}, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}}) \leftarrow \mathsf{Setup}(1^{\lambda}, 1^{N_{\mathsf{gs}}})$

$\big(aux, M^{\star}, (\mathsf{sec}^{\star}_0, \mathsf{cert}^{\star}_0), (\mathsf{sec}^{\star}_1, \mathsf{cert}^{\star}_1)\big) \leftarrow \mathcal{A}(\mathsf{play};\, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q_{\mathsf{open}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$

if $\neg((\mathsf{cert}^{\star}_0 \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}^{\star}_0) \wedge (\mathsf{cert}^{\star}_1 \leftrightharpoons_{\mathcal{Y}} \mathsf{sec}^{\star}_1) \wedge (\mathsf{cert}^{\star}_0 \neq \mathsf{cert}^{\star}_1))$ **then**

  return $\perp$

$\sigma^{\star} \leftarrow \mathsf{Sign}(\mathcal{Y}, \mathsf{cert}^{\star}_d, \mathsf{sec}^{\star}_d, M^{\star})$

$d' \leftarrow \mathcal{A}(\mathsf{guess};\, \sigma^{\star}, aux, Q_{\mathsf{pub}}, Q_{\mathsf{keyGM}}, Q^{\neg\{(M^{\star}, \sigma^{\star})\}}_{\mathsf{open}}, Q_{\mathsf{read}}, Q_{\mathsf{write}})$

return d';

Figure 5.4 – Security experiments for (full-)anonymity game.

The notion of anonymity is formalized by means of a game involving a two-stage adversary. The first stage is called play stage and allows the adversary $\mathcal{A}$ to modify $\mathsf{state}_{\mathcal{I}}$ via Q_{write}-queries and open arbitrary signatures by probing Q_{open}. When the play stage ends, $\mathcal{A}$

chooses a message $M^\star$ as well as two pairs $(\mathsf{sec}_0^\star, \mathsf{cert}_0^\star)$ and $(\mathsf{sec}_1^\star, \mathsf{cert}_1^\star)$, consisting of a valid membership certificate and a corresponding membership secret. Then, the challenger flips a coin $d \leftarrow \{0,1\}$ and computes a challenge signature $\sigma^\star$ using $(\mathsf{sec}_d^\star, \mathsf{cert}_d^\star)$. The adversary is given $\sigma^\star$ with the task of eventually guessing the bit $d \in \{0,1\}$. Before doing so, it is allowed further oracle queries throughout the second stage, called guess stage, but is restricted not to query Q_{open} for $(M^\star, \sigma^\star)$.

Definition 5.5. A dynamic group signature scheme is fully anonymous if, for any PPT adversary $\mathcal{A}$ in the experiment $\mathsf{Exp}_{\mathcal{A},d}^{\mathrm{anon}}(\lambda)$ described in Figure 5.4, the following distance is negligible:

$$\mathrm{Adv}_{\mathrm{anon}}^{\mathcal{A}}(\lambda) \triangleq \left| \Pr\left[\mathbf{Exp}_{\mathcal{A},1}^{\mathrm{anon}}(\lambda) = 1 \right] - \Pr\left[\mathbf{Exp}_{\mathcal{A},0}^{\mathrm{anon}}(\lambda) = 1 \right] \right|$$

Chapter 6

Pairing-Based Dynamic Group Signatures

In this chapter, we aim at lifting the *signature with efficient protocols* from [LPY15] to the random oracle model [BR93] in order to get an *efficient* construction. In the Camenish and Lysyanskaya terminology, signatures with efficient protocols [CL04a] are digital signatures which come with two companion protocols: a protocol whereby a signer can obliviously sign a committed message known only to the user and a zero-knowledge proof to efficiently attest possession of a hidden message-signature pair.

This building block proved useful in the design of many efficient anonymity-related protocols such as anonymous credentials [Cha85, CL01], which are similar to group signatures except that anonymity is irrevocable (meaning that there is no opening authority). In other words, an anonymous credential scheme involves one (or more) credential issuer(s) and a set of users who have a long term secret key which can be seen as their digital identity, and pseudonyms that can be seen as commitments to their secret key. Users can dynamically obtain credentials from an issuer that only knows users' pseudonyms and obliviously sign users' secret keys as well as a set of attributes. Later on, users can make themselves known to verifiers under a different pseudonym and demonstrate possession of the issuer's certificate on their secret key without revealing neither the signature nor the key. In this context, signature with efficient protocols can typically be used as follows: the user obtains the issuer's signature on a committed message via an interactive protocol, and uses a protocol for proving that two commitments open to the same value (which allows proving that the same secret underlies two distinct pseudonyms) and finally a protocol for proving possession of a secret message-signature pair.

As explained in Chapter 2, the *quality* of a scheme depends on both its efficiency and the reliability of the assumptions it relies on. Before the works described in this chapter, most signature schemes rely on groups of hidden order [CL04a] or non-standard assumptions in groups with bilinear maps [CL04b, BBS04, Oka06]. To illustrate this multi-criteria quality evaluation, we can see that Camenisch and Lysyanskaya proposed a signature scheme that is secure in pairing-friendly groups but relies on the interactive LRSW assumption [LRSW99]; but this signature scheme requires $\mathcal{O}(n)$ group elements to encode an ℓ-block message. Pointcheval and Sanders [PS18] improved this signature to go down to $\mathcal{O}(1)$ group elements for an ℓ-block message, but which is only proven secure in the generic group model (a

model where group accesses are handled by an oracle that performs the group operations).

We note that besides the scheme presented in this section, we are only aware of two schemes based on fixed-size assumptions: (1) A variant of the Camenisch and Lysyanskaya scheme [CL04b] due to Gerbush, Lewko, O'Neill and Waters [GLOW12] in composite order groups. Due to this assumption, the groups that are used are inherently bigger and lead to less efficient representations than in prime-order groups: for equivalent security levels, Freeman [Fre10] estimates that computing a pairing over a group $N = pq$ is at least 50 times slower than the same pairing in the prime order group setting. (2) A construction by Yuen, Chow, Zhang and Yu [YCZY14] under the decision linear assumption [BBS04] that unfortunately does not support "randomizable signature", which is an important property in privacy-enhancing cryptography. An application of this property is, in the context of group signatures, the re-randomization of credentials accross distinct privacy-preserving authentication.

In this chapter, we describe a new signature scheme with efficient protocols and re-randomizable signatures under a simple and well-studied assumption. Namely, the security of our scheme relies on the SXDH assumption in groups of prime order with a bilinear map. From an efficiency point of view, the signature for an ℓ-block message consists of only 4 groups elements.

This signature length is made possible by using efficient QA-NIZK arguments – as presented in Section 4.1.4 and formally defined in [JR13] – to prove the belonging to some linear subspace spanned by the rows of a matrix. For this purpose, it was shown that for this specific task, the size of the argument may be independent of the dimension of the considered subspace [JR14, LPJY14, KW15]. The signature scheme described in this chapter (Section 6.2) crucially takes advantage of this observation as ℓ-block messages are certified using a QA-NIZK argument for a subspace of dimension $\mathcal{O}(\ell)$. This construction natively supports efficient protocols to enhance privacy as described in Section 6.3. Hence, our signature scheme enables the design of an efficient anonymous credentials system based on the sole SXDH assumption.

As another showcase for this signature, we also design another primitive. Namely, a dynamic group signature scheme, as described in Chapter 5, which is practical and relies on simple assumptions (namely, SXDH and SDL). This construction is competitive both in term of signature size and computation time with the best solutions based on non-interactive assumptions [BBS04, DP06] (in these cases, the Strong Diffie-Hellman assumption [BB04]). Concretely, at the 128-bits security, each signature fits within 320 bytes while providing the strongest sense of anonymity (meaning the definition in Section 5.3.4).

Our Contribution. In this chapter, we propose a new signature scheme with efficient protocols and re-randomizable signatures under simple, well-studied assumptions. The security of our scheme is proved in the standard model under the Symmetric eXternal Diffie-Hellman (SXDH) assumption, which is a well-established, constant-size assumption (i.e., described using a constant number of elements, regardless of the number of adversarial queries) in groups with a bilinear map. Remarkably, we can sign ℓ-block messages using only 4 group elements under the SXDH assumption.

Our signature length is made possible by the use of efficient Quasi-Adaptive Non-Interactive Zero-Knowledge (QA-NIZK) arguments for linear subspaces (described in Definition 4.10).

It was shown [LPJY14, JR14, KW15] that, for the task of arguing that a vector of group elements belongs to some linear subspace, the size of arguments may be independent of the dimensions of the considered subspace. Our signature scheme crucially exploits this observation as ℓ-block messages are signed by generating a QA-NIZK argument for a subspace of dimension $O(\ell)$.

Our signature natively supports efficient privacy-enhancing protocols. We describe a two-party protocol allowing a user to obtain a signature on a committed multi-block message as well as a honest-verifier zero-knowledge protocol for efficiently demonstrating knowledge of a signature on a committed message revealing neither the message nor the signature. Hence, our scheme readily enables the design of an efficient anonymous credentials system based on the sole SXDH assumption.

As another application of our signature scheme, we describe a truly practical group signature (for dynamic groups) based on simple assumptions in the random oracle model. Our scheme is competitive with the best solutions [BBS04, DP06] based on non-interactive assumptions (which are those relying on the Strong Diffie-Hellman assumption [BB04]) in terms of computational cost and signature length. Concretely, at the 128-bit security level, each signature fits within 320 bytes while providing anonymity in the strongest sense (i.e., against adversaries equipped with a signature opening oracle). To the best of our knowledge, the new scheme thus features the shortest group signatures based on standard assumptions.

It seems that our signature scheme has many other potential applications. For example, combining it with the ideas of [CHL05a] and a pseudo-random function based on standard assumptions (e.g., [NR97]) readily gives a compact e-cash system based on simple hardness assumptions.

The rest of the chapter is organized as follows. We will first recall the useful building blocks that are used to design and prove our signature scheme that supports efficient protocols in the [CL02a] fashion. Then we describe this scheme and we next give the construction and the proof for the group signature scheme for dynamically growing groups. Finally, we show the experimental results we obtain for this group signature scheme.

6.1 Building blocks

We use bilinear maps $e : \mathbb{G} \times \widehat{\mathbb{G}} \to \mathbb{G}_T$ over groups of prime order p and we rely on the assumed security of the SDL and SXDH problems defined in Section 3.1. All these definitions are recalled below.

Definition 3.1 (Pairings [BSS05])**.** A pairing is a map $e : \mathbb{G} \times \widehat{\mathbb{G}} \to \mathbb{G}_T$ over cyclic groups of order p that verifies the following properties for any $g \in \mathbb{G}, \hat{g} \in \widehat{\mathbb{G}}$:

(i) bilinearity: for any $a, b \in \mathbb{Z}_p$, we have $e(g^a, \hat{g}^b) = e(g^b, \hat{g}^a) = e(g, \hat{g})^{ab}$.

(ii) non-degeneracy: $e(g, \hat{g}) = 1_{\mathbb{G}_T} \iff g = 1_{\mathbb{G}}$ or $\hat{g} = 1_{\widehat{\mathbb{G}}}$.

(iii) the map is computable in polynomial time in the size of the input.

Definition 3.2 (SXDH [BGdMM05, As. 1])**.** The *Symmetric eXternal Diffie-Hellman* (SXDH) assumption holds if the DDH assumption holds both in $\mathbb{G}$ and $\widehat{\mathbb{G}}$.

Definition 3.3 (SDL). In bilinear groups $(\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T)$ of prime order p, the *Symmetric Discrete Logarithm* (SDL) problem consists in, given $(g, \hat{g}, g^a, \hat{g}^a) \in (\mathbb{G} \times \widehat{\mathbb{G}})^2$ where $a \hookleftarrow \mathbb{Z}_p$, computing $a \in \mathbb{Z}_p$.

6.1.1 Quasi-Adaptive NIZK Arguments for Linear Subspaces

Quasi-Adaptive NIZK (QA-NIZK) proofs [JR13] are NIZK proofs where the common reference string (CRS) may depend on the language for which proofs have to be generated. Formal definitions are given in [JR13, LPJY14, KW15].

This section recalls the QA-NIZK argument of [KW15] for proving membership in the row space of a matrix. In the description below, we assume that all algorithms take as input the description of common public parameters cp consisting of asymmetric bilinear groups $(\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T, p)$ of prime order $p > 2^\lambda$, where λ is the security parameter. In this setting the problem is to convince that $\boldsymbol{v}$ is a linear combination of the rows of a given $\mathbf{M} \in \mathbb{G}^{t \times n}$.

Kiltz and Wee [KW15] suggested the following construction which simplifies [LPJY14] and remains secure under SXDH. We stress that cp is independent of the matrix $\mathbf{M} = (\vec{M}_1 \cdots \vec{M}_t)^T$.

Keygen$(\mathsf{cp}, \mathbf{M})$**:** Given public parameters $\mathsf{cp} = (\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T, p)$ and the matrix $\mathbf{M} = (M_{i,j}) \in \mathbb{G}^{t \times n}$.

First, choose $\hat{g}_z \hookleftarrow \mathcal{U}(\widehat{\mathbb{G}})$. Pick $\mathsf{tk} = (\chi_1, \ldots, \chi_n) \hookleftarrow \mathcal{U}(\mathbb{Z}_p^n)$ and compute $\hat{g}_j = \hat{g}_z{}^{\chi_j}$, for all $j = 1$ to n.

Then, for $i = 1$ to t, compute $z_i = \prod_{j=1}^n M_{i,j}^{-\chi_j}$ and output

$$\mathsf{crs} = \left(\{z_i\}_{i=1}^t,\ \hat{g}_z,\ \{\hat{g}_j\}_{j=1}^n\right) \in \mathbb{G}^t \times \widehat{\mathbb{G}}^{n+1}.$$

Prove$(\mathsf{crs}, \boldsymbol{v}, \{\omega_i\}_{i=1}^t)$**:** To prove that $\boldsymbol{v} = \vec{M}_1^{\omega_1} \cdots \vec{M}_t^{\omega_t}$, for some witness $\omega_1, \ldots, \omega_t \in \mathbb{Z}_p$, where $\vec{M}_i$ denotes the i-th row of $\mathbf{M}$, parse crs as above and return $\pi = \prod_{i=1}^t z_i^{\omega_i}$.

Sim$(\mathsf{tk}, \boldsymbol{v})$**:** In order to simulate a proof for a vector $\boldsymbol{v} \in \mathbb{G}^n$ using $\mathsf{tk} = \{\chi_i\}_{i=1}^n$, output $\pi = \prod_{j=1}^n v_j^{-\chi_j}$.

Verify$(\mathsf{crs}, \boldsymbol{v}, \pi)$**:** Given $\pi \in \mathbb{G}$, $\boldsymbol{v} = (v_1, \ldots, v_n)$ and crs parsed as above, return 1 if and only if $(v_1, \ldots, v_n) \neq (1_\mathbb{G}, \ldots, 1_\mathbb{G})$ and π satisfies $1_{\mathbb{G}_T} = e(\pi, \hat{g}_z) \cdot \prod_{j=1}^n e(v_j, \hat{g}_j)$.

The proof of the soundness of this QA-NIZK argument system requires the matrix $\mathbf{M}$ to be witness-samplable. This means that the reduction has to know the discrete logarithms of the group elements of $\mathbf{M}$. This requirement is compatible with our security proofs.

6.2 A Randomizable Signature on Multi-Block Messages

In [LPY15], Libert *et al.* described an F-unforgeable signature[1] based on the SXDH assumption. We show that their scheme implies an efficient ordinary digital signature which

[1] In F-unforgeability, the adversary only has to output a forgery for a message M without outputting the message, but the image $F(M)$ for an injective function F that is not necessarily efficiently invertible instead [BCKL08]. In [LPY15], the function F is $M \mapsto \hat{g}^M$.

makes it possible to efficiently sign multi-block messages in $\mathbb{Z}_p^\ell$ while keeping the scheme compatible with efficient protocols. In order to keep the signature length independent of the number of blocks, we exploit the property that the underlying QA-NIZK argument [KW15] has constant size, regardless of the dimensions of the considered linear subspace. Moreover, we show that their scheme remains unforgeable under the SXDH assumption.

Keygen(λ, ℓ) : Choose bilinear groups $\mathsf{cp} = (\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T, p)$ of prime order $p > 2^\lambda$ with $g \hookleftarrow \mathcal{U}(\mathbb{G})$, $\hat{g} \hookleftarrow \mathcal{U}(\widehat{\mathbb{G}})$.

1. Choose $\omega, a \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$, and set $h = g^a$, $\Omega = h^\omega$.
2. Choose $\vec{v} = (v_1, \ldots, v_\ell, w) \hookleftarrow \mathcal{U}(\mathbb{G}^{\ell+1})$.
3. Define a matrix $\mathbf{M} = (M_{j,i})_{j,i} \in \mathbb{G}^{(\ell+2)\times(2\ell+4)}$

$$\mathbf{M} = \left(\begin{array}{c|c|c|c} g & \mathbf{1}_{\ell+1} & \mathbf{1}_{\ell+1} & h \\ \hline \vec{v}^T & g^{\mathbf{I}_{\ell+1}} & h^{\mathbf{I}_{\ell+1}} & \mathbf{1}_{\ell+1}^T \end{array}\right), \tag{6.1}$$

   where $\mathbf{1}_{\ell+1} = (1_\mathbb{G}, \ldots, 1_\mathbb{G}) \in \mathbb{G}^{\ell+1}$.
4. Run $\mathsf{Keygen}(\mathsf{cp}, M)$ of the QA-NIZK argument of Section 6.1.1 to get the common reference string $\mathsf{crs} = \Big(\{z_i\}_{i=1}^{\ell+2},\ \hat{g}_z,\ \{\hat{g}_j\}_{j=1}^{2\ell+4}\Big)$.

The private key is $\mathsf{sk} := \omega$ and the public key is

$$\mathsf{pk} = \Big(\mathsf{cp},\ g,\ h,\ \hat{g},\ \vec{v},\ \Omega = h^\omega,\ \mathsf{crs}\Big).$$

Sign$(\mathsf{sk}, \vec{m} = (m_1, \ldots, m_\ell))$: given the private key $\mathsf{sk} = \omega$ and a message $\vec{m} \in \mathbb{Z}_p^\ell$, choose $s \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ to compute

$$\sigma_1 = g^\omega \cdot (v_1^{m_1} \cdots v_\ell^{m_\ell} \cdot w)^s, \qquad \sigma_2 = g^s, \qquad \sigma_3 = h^s.$$

Then, run Prove of the QA-NIZK argument to prove that the following vector of $\mathbb{G}^{2\ell+4}$

$$(\sigma_1, \sigma_2^{m_1}, \ldots, \sigma_2^{m_\ell}, \sigma_2, \sigma_3^{m_1}, \ldots, \sigma_3^{m_\ell}, \sigma_3, \Omega) \tag{6.2}$$

is in the row space of $\mathbf{M}$. This QA-NIZK proof $\pi \in \mathbb{G}$ consists of $\pi = z_1^\omega \cdot (z_2^{m_1} \cdots z_{\ell+1}^{m_\ell} \cdot z_{\ell+2})^s$.

Return the signature $\sigma = (\sigma_1, \sigma_2, \sigma_3, \pi) \in \mathbb{G}^4$.

Verify$(\mathsf{pk}, \sigma, \vec{m})$: parse σ as above and $\vec{m}$ as a tuple $(m_1, \ldots, m_\ell)$ in $\mathbb{Z}_p^\ell$ and return 1 if and only if

$$\begin{aligned} e(\Omega, \hat{g}_{2\ell+4})^{-1} = {} & e(\pi, \hat{g}_z) \cdot e(\sigma_1, \hat{g}_1) \\ & \cdot e(\sigma_2, \hat{g}_2^{m_1} \cdots \hat{g}_{\ell+1}^{m_\ell} \cdot \hat{g}_{\ell+2}) \cdot e(\sigma_3, \hat{g}_{\ell+3}^{m_1} \cdots \hat{g}_{2\ell+2}^{m_\ell} \cdot \hat{g}_{2\ell+3}). \end{aligned} \tag{6.3}$$

The signature on ℓ scalars thus only consists of 4 elements in $\mathbb{G}$ while the verification equation only involves a computation of 5 pairings[2].

Theorem 6.1. *The above signature scheme is existentially unforgeable under chosen-message attacks (eu-cma) if the SXDH assumption holds in* $(\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T)$.

Proof. We will proceed as in [LPY15] to prove that the scheme of section 6.2 is secure under chosen-message attacks. Namely we will consider a sequence of hybrid games involving two kinds of signatures.

Type A signatures: These are real signatures:

$$\begin{aligned} \sigma_1 &= g^{\omega} \cdot (v_1^{m_1} \cdots v_{\ell}^{m_{\ell}} \cdot w)^s, \qquad \sigma_2 = g^s, \\ \pi &= z_1^{\omega} \cdot (z_2^{m_1} \cdots z_{\ell+1}^{m_{\ell}} \cdot z_{\ell+2})^s, \quad \sigma_3 = h^s. \end{aligned} \tag{6.4}$$

Since $(\sigma_1, \sigma_2^{m_1}, \ldots, \sigma_2^{m_{\ell}}, \sigma_2, \sigma_3^{m_1}, \ldots, \sigma_3^{m_{\ell}}, \sigma_3, \Omega)$ is in the row space of $\mathbf{M}$, the QA-NIZK proof π has the same distribution as if it were computed as

$$\begin{aligned} \pi = \sigma_1^{-\chi_1} \cdot \left(\prod_{i=2}^{\ell+1} \sigma_2^{-\chi_i m_{i-1}} \right) \cdot \sigma_2^{-\chi_{\ell+2}} \cdot \\ \left(\prod_{i=\ell+3}^{2\ell+2} \sigma_3^{-\chi_i m_{i-\ell-2}} \right) \cdot \sigma_3^{-\chi_{2\ell+3}} \cdot \Omega^{-\chi_{2\ell+4}}. \end{aligned} \tag{6.5}$$

We also define **Type A′** signatures as a generalization of Type A signatures where only condition (6.4) are imposed and no restriction is given on π beyond the fact that it should be a valid homomorphic signature on vector (6.2).

Type B signatures: These use a random value $\omega' \in_R \mathbb{Z}_p$ instead of the secret key ω. We pick random $\omega', s, s_1 \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and compute:

$$(\sigma_1, \sigma_2, \sigma_3) = (g^{\omega'} \cdot (v_1^{m_1} \cdots v_{\ell}^{m_{\ell}} \cdot w)^s,\ g^s,\ h^{s+s_1}),$$

The QA-NIZK proof π is computed as in (6.5) by using $\mathsf{tk} = \{\chi_i\}_{i=1}^{2\ell+4}$. Note that Type B signatures can be generated without using $\omega \in \mathbb{Z}_p$.

We consider a sequence of games. In Game i, S_i denotes the event that $\mathcal{A}$ produces a valid signature $\sigma^\star$ on $M^\star$ such that $(M^\star, \sigma^\star)$ was not queried before, and by E_i the event that $\mathcal{A}$ produces a Type A′ signature.

Game 0: This is the real game. The challenger $\mathcal{B}$ produces a key pair $(\mathsf{sk}, \mathsf{pk})$ and sends pk to $\mathcal{A}$. Then $\mathcal{A}$ makes Q signature queries: $\mathcal{A}$ sends messages M_i to $\mathcal{B}$, and $\mathcal{B}$ answers by sending $\sigma_i = \mathsf{Sign}(\mathsf{sk}, M_i)$ to $\mathcal{A}$. Finally $\mathcal{A}$ sends a pair $(M^\star, \sigma^\star) \notin \{(M_i, \sigma_i)\}_{i=1}^{Q}$ and wins if $\mathsf{Verify}(\mathsf{pk}, \sigma^\star, M^\star) = 1$.

[2] Actually only 4 pairing computations are necessary, as $e(\Omega, \hat{g}_{2\ell+4})$ is independent of the inputs π and $\vec{m}$, and can hence be precomputed.

Game 1: We change the way $\mathcal{B}$ answers signing queries. The QA-NIZK proofs π are then computed as simulated QA-NIZK proofs using tk as in (6.5). These QA-NIZK proofs are thus simulated proofs for true statements, and then their distribution remains unchanged. We have $\Pr[S_1] = \Pr[S_1 \wedge E_1] + \Pr[S_1 \wedge \neg E_1]$. Lemma 6.2 states that the event $S_1 \wedge \neg E_1$ happens with all but negligible probability: $\Pr[S_1 \wedge \neg E_1] \leq \mathrm{Adv}^{\mathsf{DDH}}_{\widehat{\mathbb{G}}}(\lambda) - 1/p$. Thus our task is now to upper-bound the probability $\Pr[S_1 \wedge E_1]$.

Game 2.k ($0 \leq k \leq Q$): In Game $2.k$, the challenger returns a Type B signature for the first k queries. At the last $Q - k$ signature queries, the challenger answers a type A signature. Lemma 6.3 ensures that

$$\left|\Pr\left[S_{2.k} \wedge E_{2.k}\right] - \Pr\left[S_{2.(k-1)} \wedge E_{2.(k-1)}\right]\right|$$

is smaller than $\mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}}(\lambda) + 1/p$.

In Game $2.Q$, we know that if SXDH holds, $\mathcal{A}$ can only output a type A' forgery even if it only obtains type B signatures during the game. Nevertheless, lemma 6.4 shows that a type A' forgery in Game $2.Q$ contradicts the DDH assumptions in $\mathbb{G}$. Therefore we have $\Pr[S_{2.Q} \wedge E_{2.Q}] \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}}(\lambda)$. Putting the above altogether, the probability $\Pr[S_0]$ is upper-bounded by

$$\mathrm{Adv}^{\mathsf{DDH}}_{\widehat{\mathbb{G}}}(\lambda) + \frac{1}{p} + Q\left(\mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}}(\lambda) + \frac{1}{p}\right) + \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}}(\lambda) < (Q+2) \cdot \left(\mathrm{Adv}^{\mathrm{SXDH}}_{\mathbb{G},\widehat{\mathbb{G}}}(\lambda) + \frac{1}{p}\right).$$

□

Lemma 6.2. *In **Game 1**, if the DDH assumption holds in $\widehat{\mathbb{G}}$, $\mathcal{A}$ can only output a type A' forgery.*

Proof. Let $\mathcal{A}$ be an attacker that does not output a type A' forgery. We will build an attacker $\mathcal{B}$ against the soundness of the Quasi-Adaptive NIZK (QA-NIZK) scheme, which security is implied from the double-pairing problem that reduces from DDH as explained in [LPJY13]. Let us define the vector $\boldsymbol{\sigma} \in \mathbb{G}^{2\ell+4}$ as

$$\boldsymbol{\sigma} \triangleq (\sigma_1^\star, \sigma_2^{\star m_1}, \ldots, \sigma_2^{\star m_\ell}, \sigma_2^\star, \sigma_3^{\star m_1}, \ldots, \sigma_3^{\star m_\ell}, \sigma_3^\star, \Omega) \in \mathbb{G}^{2\ell+4}.$$

If $(M^\star, \sigma^\star)$ is not a type A' forgery, $\boldsymbol{\sigma}$ is then not in the row space of $\mathbf{M}$.

Our reduction $\mathcal{B}$ receives as input $\mathsf{cp} = (\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T, p)$, a matrix $\mathbf{M}$ as in (6.1) and a common reference string crs (depending on the matrix) for an instance of the QA-NIZK scheme allowing to prove that vectors of dimension $2\ell+4$ are in the row space of $\mathbf{M}$. The generation of the matrix $\mathbf{M}$ fixes g, h and $\vec{v} = (v_1, \ldots, v_\ell, w) \in \mathbb{G}^{\ell+1}$. After that, $\mathcal{B}$ picks $\omega \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and $\hat{g} \hookleftarrow \mathcal{U}(\widehat{\mathbb{G}})$, and set $\Omega = h^\omega$. Then, the reduction $\mathcal{B}$ sends to $\mathcal{A}$ cp and the verification key:

$$\mathsf{pk} = (g, h, \hat{g}, \vec{v}, \omega, \mathsf{crs}).$$

Since $\mathcal{B}$ knows the secret key $\omega \in \mathbb{Z}_p$, it can answer all signing queries by honestly running the Sign algorithm, in particular, it does not need to know tk to do this.

When $\mathcal{A}$ halts, it outputs $(M^\star, \sigma^\star)$ where $\sigma^\star$ is not a Type A′ forgery, so that $\boldsymbol{\sigma}$ is not in the row space of $\mathbf{M}$. Therefore, outputting $\pi^\star$ constitutes a valid proof against the soundness property of the scheme, and thus implies an algorithm against DDH as in [KW15] since the matrix can be witness-samplable. □

Lemma 6.3. *If DDH holds in $\mathbb{G}$, for each $k \in \{1, \ldots, Q\}$, $\mathcal{A}$ produces a type A' forgery with negligibly different probabilities in* ***Game 2.k*** *and* ***Game 2.$(k-1)$****.*

Proof. Let us assume there exists an index $k \in \{1, \ldots, Q\}$ and an adversary $\mathcal{A}$ that outputs a Type A′ forgery with smaller probability in Game $2.k$ than in Game $2.(k-1)$. We build a DDH distinguisher $\mathcal{B}$.

Algorithm $\mathcal{B}$ takes in $(g^a, g^b, \eta) \in \mathbb{G}^3$, where $\eta = g^{a(b+c)}$, and decides if $c = 0$ or $c \in_R \mathbb{Z}_p$. To do this, $\mathcal{B}$ sets $h = g^a$. It picks $\omega, a_{v_1}, b_{v_1}, \ldots, a_{v_\ell}, b_{v_\ell}, a_w, b_w \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and sets $\Omega = h^\omega$ as well as:

$$\forall i \in \{1, \ldots, \ell\} : \quad v_i = g^{a_{v_i}} \cdot h^{b_{v_i}}, \quad w = g^{a_w} \cdot h^{b_w}.$$

The reduction $\mathcal{B}$ also chooses $\mathsf{tk} = \{\chi_i\}_{i=1}^{2\ell+4}$ and computes $\mathsf{crs} = (\{z_j\}_{j=1}^{2\ell+4}, \hat{g}_z, \{\hat{g}_i\}_{i=1}^{2\ell+4})$ as in steps 3-4 of Keygen. It then outputs $\mathsf{pk} = (g, h, \hat{g}, \vec{v}, \omega, \mathsf{crs})$.

Then, queries are answered depending on their index j:
Case $j < k$: $\mathcal{B}$ computes a Type B signature, $\sigma = (\sigma_1, \sigma_2, \sigma_3, \pi)$, using $\mathsf{tk} = \{\chi_i\}_{i=1}^{2\ell+4}$ with the QA-NIZK simulator to computes π.

Case $j > k$: The last $Q - k - 1$ signing queries are computed as Type A signatures, which $\mathcal{B}$ is able to generate using the secret key $\omega \in \mathbb{Z}_p$ he knows and crs or $\mathsf{tk} = \{\chi_i\}_{i=1}^{2\ell+4}$ to produces valid proofs.

Case $j = k$: In the k-th signing query $(m_1, \ldots, m_\ell)$, $\mathcal{B}$ embeds the DDH instance in the signature and simulates either Game $2.k$ or Game $2.(k-1)$ depending on whether $\eta = g^{ab}$ or $\eta = g^{a(b+c)}$ for some $c \in_R \mathbb{Z}_p$. Namely, $\mathcal{B}$ computes $\sigma_2 = g^b$, $\sigma_3 = \eta$, and $\sigma_1 = g^\omega \sigma_2^{a_w + \sum_{i=1}^\ell a_{v_i} m_i} \sigma_3^{b_w + \sum_{i=1}^\ell b_{v_i} m_i}$. Then $\mathcal{B}$ simulates QA-NIZK proofs π as recalled in (6.5), and sends $\sigma = (\sigma_1, \sigma_2, \sigma_3, \pi)$ to $\mathcal{A}$.

If $\eta = g^{ab}$, the k-th signature σ is a Type A signature with $s = b$. If $\eta = g^{a(b+c)}$ for some $c \in_R \mathbb{Z}_p$, we have:

$$\begin{aligned} \sigma_1 &= g^\omega g^{ac \cdot (b_w + \sum_{i=1}^\ell b_{v_i} m_i)} (v_1^{m_1} \cdots v_\ell^{m_\ell} w)^b \\ &= g^{\omega'} (v_1^{m_1} \cdots v_\ell^{m_\ell} w)^b \\ \sigma_2 &= g^b, \qquad\qquad \sigma_3 = h^{b+c} \end{aligned}$$

Where $\omega' = \omega + ac \cdot (b_w + \sum_{i=1}^\ell b_{v_i} m_i)$. Since the term $b_w + \sum_{i=1}^\ell b_{v_i} m_i$ is uniform and independent of $\mathcal{A}$'s view, σ is distributed as a Type B signature if $\eta = g^{a(b+c)}$.

When $\mathcal{A}$ terminates, it outputs a couple $(m_1^\star \cdots m_\ell^\star, \sigma^\star)$ that has not been queried during the signing queries. Now the reduction $\mathcal{B}$ has to determine whether $\sigma^\star$ is a Type A′ forgery

or not. To this end, it tests if the equality:

$$\sigma_1^\star = g^\omega \sigma_2^{\star a_w + \sum_{i=1}^{\ell} a_{v_i} m_i^\star} \sigma_3^{\star b_w + \sum_{i=1}^{\ell} b_{v_i} m_i^\star} \tag{6.6}$$

is satisfied. If it is, $\mathcal{B}$ outputs 1 to indicate that $\eta = g^{ab}$. Otherwise it outputs 0 and rather bets that $\eta \in_R \mathbb{G}$.

To see why this test allows recognizing Type A′ forgeries, we remark that $\sigma^\star$ is of the form:

$$\sigma_2^\star = g^s, \qquad \sigma_3^\star = h^{s+s_1}, \qquad \sigma_1^\star = g^{\omega+s_0}(v_1^{m_1^\star} \cdots v_\ell^{m_\ell^\star} w)^s,$$

and the goal of $\mathcal{B}$ is to decide whether $(s_0, s_1) = (0, 0)$ or not. We notice that $s_0 = a \cdot s_1 \cdot (b_w + \sum_{i=1}^{\ell} b_{v_i} \cdot m_i^\star)$ if the forgery fulfills relation (6.6) and we show this to only happen with probability $1/p$ for any $s_1 \neq 0$ meaning that Type B forgery passes the test with the same probability.

From the entire game, and assuming a forgery which passes the test, we have the following linear system:

$$\left(\begin{array}{c|c} \mathbf{I}_{\ell+1} & a \cdot \mathbf{I}_{\ell+1} \\ \hline \mathbf{0}_{\ell+1}^T & ac \cdot (m_1|\cdots|m_\ell|1) \\ \hline \mathbf{0}_{\ell+1}^T & as_1 \cdot (m_1^\star|\cdots|m_\ell^\star|1) \end{array}\right) \cdot \begin{pmatrix} a_{v_1} \\ \vdots \\ a_{v_\ell} \\ a_w \\ b_{v_1} \\ \vdots \\ b_{v_\ell} \\ b_w \end{pmatrix} = \begin{pmatrix} \log_g(v_1) \\ \vdots \\ \log_g(v_\ell) \\ \log_g(w) \\ \omega' - \omega \\ s_0 \end{pmatrix}$$

where, $\mathbf{0}_{\ell+1}$ denotes the zero vector of length $\ell+1$ and $m_1, \ldots, m_\ell$ is the message involved in the k-th signing query. Note that the $(l+2)$-th equation is meaningless when $c = 0$ since then $\omega' = \omega$. However, even if $c \neq 0$ the information that $\mathcal{A}$ can infer about $(a_{v_1}, \ldots, a_{v_\ell}, a_w, b_{v_1}, \ldots, b_{v_\ell}, b_w) \in \mathbb{Z}_p^{2\ell+2}$ during the game amounts to the first $\ell+2$ equations of the system which is of full rank. It means that this vector is unpredictable since all the solutions of this linear system live in a sub-space of dimension at least one (actually $\ell = (2\ell+2) - (\ell+2)$). Finally, as long as $s_1 \neq 0$, the right value s_0 can only be guessed with probability $1/p$ since the last row of the matrix is independent of the others as soon as $(m_1, \ldots, m_\ell) \neq (m_1^\star, \ldots, m_\ell^\star) \neq 0$.

To conclude the proof, since $\mathcal{B}$ is able the tell apart the type of the forgery, if $\mathcal{A}$'s probability to output a forgery of some Type in Game $k-1$ (*i.e.*, $c = 0$) was significantly different than in Game k (*i.e.*, $c \neq 0$) then B would be able to solve the DDH problem with non-negligible advantage.

□

Lemma 6.4. *In* ***Game 2.Q****, a PPT adversary outputting a type A′ forgery would contradict the DDH assumption in* $\mathbb{G}$*:* $\Pr[S_{2.Q} \wedge E_{2.Q}] \leq \mathrm{Adv}_{\mathbb{G}}^{DDH}(\lambda)$.

Proof. We will build an algorithm $\mathcal{B}$ for solving the Computational Diffie Hellman problem (CDH) which is at least as hard as the DDH problem. The reduction $\mathcal{B}$ takes as

input a tuple $(g, h, \Omega = h^{\omega})$ and computes g^{ω}. To generate pk, $\mathcal{B}$ picks $\hat{g} \hookleftarrow \mathcal{U}(\widehat{\mathbb{G}})$, $a_{v_1}, \ldots, a_{v_\ell}, a_w \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and computes $v_1 = g^{a_{v_1}}, \ldots, v_\ell = g^{a_{v_\ell}}$, and $w = g^{a_w}$. Then $\mathcal{B}$ generates $\mathsf{tk} = \{\chi_i\}_{i=1}^{2\ell+4}$, $\mathsf{crs} = (\{z_j\}_{j=1}^{\ell+2}, \hat{g}_z, \{\hat{g}_i\}_{i=1}^{2\ell+4})$ as in step 3-4 of the key generation algorithm, then sends the public key $pk = (g, h, \hat{g}, \boldsymbol{v}, \Omega = h^{\omega}, \mathsf{crs})$ to $\mathcal{A}$.

$\mathcal{B}$ also retains $\mathsf{tk} = \{\chi_i\}_{i=1}^{2\ell+4}$ to handle signing queries. We recall that during the game, signing queries are answered by returning a Type B signature so that, using tk, $\mathcal{B}$ can answer all queries without knowing the $\omega = \log_h(\Omega)$ which is part of the CDH challenge.

The results of Lemma 6.3 implies that even if $\mathcal{A}$ only obtains Type B signatures, it will necessarily output a Type A′ forgery $\sigma^{\star} = (\sigma_1^{\star}, \sigma_2^{\star}, \sigma_3^{\star}, \pi^{\star})$ unless the DDH assumption does not hold in $\mathbb{G}$. This event thus allows $\mathcal{B}$ to compute

$$g^{\omega} = \sigma_1^{\star} \cdot \sigma_2^{\star -a_w - \sum_{i=1}^{\ell} a_{v_i} m_i^{\star}},$$

which contradicts the DDH assumption in $\mathbb{G}$. □

6.3 Companion Protocols

In this section, we give Σ-protocols (Section 4.1.2) for issuing a signature on a committed multi-block message and for proving knowledge of a valid message-signature pair.

6.3.1 Proof of Knowledge of a Signature on a Committed Message

We give Σ-protocols for proving the knowledge of a signature-message pair $(\sigma, \vec{m})$ satisfying the verification equation (6.3) of the scheme of Section 6.2

$$\begin{aligned} e(\Omega, \hat{g}_{2\ell+4})^{-1} = \; & e(\sigma_1, \hat{g_1}) \cdot e(\sigma_2, \hat{g}_2^{m_1} \cdots \hat{g}_{\ell+1}^{m_\ell} \cdot \hat{g}_{\ell+2}) \\ & \cdot e(\sigma_3, \hat{g}_{\ell+3}^{m_1} \cdots \hat{g}_{2\ell+2}^{m_\ell} \cdot \hat{g}_{2\ell+3}) \cdot e(\pi, \hat{g}_z), \end{aligned} \tag{6.7}$$

where $\sigma = (\sigma_1, \sigma_2, \sigma_3, \pi)$ and $\vec{m} = (m_1, \ldots, m_\ell)$. We note that, as shown in the proof of Theorem 6.1, a candidate signature $(\sigma_1, \sigma_2, \sigma_3, \pi)$ may satisfy the verification equation although $\log_g(\sigma_2) \neq \log_h(\sigma_3)$. In applications to anonymous credentials, a malicious credential issuer could take advantage of this fact in attempts to break the anonymity of the scheme (e.g., by linking two authentications involving the same credential). For this reason, we consider a protocol for proving possession of a possibly maliciously generated signature.

We thus consider the case of arbitrary valid signatures that may have been maliciously computed by a signer who, e.g., aims at tracing provers across different authentications. In this setting, we can still obtain a perfect SHVZK Σ-protocol to hedge against such attacks.

A first attempt to efficiently build such a protocol is to "linearize" the verification equation (6.7) by making sure that two witnesses are never paired together. However, we will still have to deal with (parallelizable) intermediate Σ-protocols for quadratic scalar relations. Even though a quadratic pairing-product equation $e(x_1, \hat{a}) \cdot e(x_2, \hat{y})$ – for variables $x_1, x_2, \hat{y}$ and constant $\hat{a}$ – can be linearized by partially randomizing the variables so as to get the equation $e(x_1 \cdot x_2^r, \hat{a}) \cdot e(x_2, \hat{y} \cdot \hat{a}^{-r})$ (which allows $\hat{y}' = \hat{y} \cdot \hat{a}^{-r}$ to appear in the clear), proving knowledge of a valid signature still requires proving a statement about some

representation of $\hat{y}$ which now appears in committed form. Somehow, going through the randomizing factor $\hat{a}^{-r}$ involves a quadratic relation between some known exponents to get special-soundness. To ease the entire proof we rather directly commit to the variables in $\mathbb{G}$ and $\hat{\mathbb{G}}$ using their available generator g and $\hat{g}$ which are not among the constants of the verification equation of the signature. We additionally need an extra generator f of $\mathbb{G}$ whose discrete logarithm is unknown.

Commit Given $(\sigma, \vec{m})$, conduct the following steps.

1. Commit to $d_1 \coloneqq \hat{g}_2^{m_1} \cdots \hat{g}_{\ell+1}^{m_\ell} \cdot \hat{g}_{\ell+2} \in \hat{\mathbb{G}}$ and $d_2 \coloneqq \hat{g}_{\ell+3}^{m_1} \cdots \hat{g}_{2\ell+2}^{m_\ell} \cdot \hat{g}_{2\ell+3} \in \hat{\mathbb{G}}$. To this end, choose $r_1, r_2 \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and compute $\hat{D}_1 = d_1 \cdot \hat{g}^{r_1}$ and $\hat{D}_2 = d_2 \cdot \hat{g}^{r_2}$.

2. In order to prove knowledge of an opening of commitments $\hat{D}_1, \hat{D}_2 \in \widehat{\mathbb{G}}$ to the same message $\vec{m} = (m_1, \ldots, m_\ell) \in \mathbb{Z}_p^\ell$, choose $s_1, s_2, u_1, \ldots, u_\ell \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and compute $\hat{E}_1 = \hat{g}_2^{u_1} \cdots \hat{g}_{\ell+1}^{u_\ell} \cdot \hat{g}^{s_1}$ and $\hat{E}_2 = \hat{g}_{\ell+3}^{u_1} \cdots \hat{g}_{2\ell+2}^{u_\ell} \cdot \hat{g}^{s_2}$.

3. Using the randomness $r_1, r_2 \in \mathbb{Z}_p$ from step 1, define $\sigma_0 = \sigma_2^{r_1} \cdot \sigma_3^{r_2}$ and commit to $(\pi, \sigma_0, \sigma_1, \sigma_2, \sigma_3) \in \mathbb{G}^5$. For this purpose, choose $t_z, t_0, t_1, t_2, t_3 \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ at random and set $C_z = \pi \cdot g^{t_z}$, $C_i = \sigma_i \cdot g^{t_i}$, for $i \in \{0, \ldots, 3\}$, and $\hat{D}_0 = \hat{g}_z^{t_z} \cdot \hat{g}_1^{t_1} \cdot \hat{D}_1^{t_2} \cdot \hat{D}_2^{t_3} \cdot \hat{g}^{-t_0}$.

4. In order to prove (partial) knowledge of an opening to $(C_z, C_0, C_1, C_2, C_3, \hat{D}_0)$, compute $\hat{E}_0 = \hat{g}_z^{v_z} \cdot \hat{g}_1^{v_1} \cdot \hat{D}_1^{v_2} \cdot \hat{D}_2^{v_3} \cdot \hat{g}^{-v_0}$ for random $v_z, v_0, v_1, v_2, v_3 \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$.

5. Prove that C_0 is well-formed relatively to the committed values in C_1, C_2 and the coins $r_1, r_2 \in \mathbb{Z}_p$ used in $\hat{D}_1, \hat{D}_2$. To this end, prove knowledge of the representation $C_0 = C_2^{r_1} \cdot C_3^{r_2} \cdot g^{t_4}$, where $t_4 = t_0 - r_1 \cdot t_2 - r_2 \cdot t_3$. To do this, compute $F_0 = C_2^{s_1} \cdot C_3^{s_2} \cdot g^{v_4}$, for $v_4 \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and where $s_1, s_2 \in \mathbb{Z}_p$ are the random coins used in $\hat{E}_1, \hat{E}_2$.

6. To prove that $t_4 = t_0 - r_1 \cdot t_2 - r_2 \cdot t_3$, (re-)commit to $t_0, t_2, t_3, t_4 \in \mathbb{Z}_p$ by picking $x_2, x_3, x_4 \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and computing

   $$T_i = g^{t_i} \cdot f^{x_i} \qquad \forall i \in \{0, 2, 3, 4\},$$

   where $x_0 = x_2 \cdot r_1 + x_3 \cdot r_2 + x_4$. Ensure that committed variables coincide with those of previous steps by computing

   $$\{V_i = g^{v_i} \cdot f^{y_i}\}_{i \in \{0,2,3,4\}},$$

   where $y_0, y_2, y_3, y_4 \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$. To prove the equality $T_0 = T_2^{r_1} \cdot T_3^{r_2} \cdot T_4$, re-use $s_1, s_2 \in \mathbb{Z}_p$ from steps 2 and 5 to compute $S_0 = T_2^{s_1} \cdot T_3^{s_2}$.

Finally, keep $C_z \in \mathbb{G}$ and all the random coins in aux,

and output

$$\begin{aligned} \mathsf{com} = \Big(& \{C_i\}_{i=0}^3, F_0, \{(T_i, V_i)\}_{i=0,2,3,4}, \\ & S_0, \{(\hat{D}_i, \hat{E}_i)\}_{i=0}^2 \Big) \in \mathbb{G}^{14} \times \hat{\mathbb{G}}^6 \end{aligned} \tag{6.8}$$

Challenge Given com as per (6.8), pick $\rho \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ uniformly at random and return $\mathsf{chall} = \rho$.

Response On inputs com, aux and $\mathsf{chall} = \rho$, compute:

1. $\bar{m}_i = \rho \cdot m_i + u_i$, for $i = 1$ to ℓ, $\bar{r}_1 = \rho \cdot r_1 + s_1$, and $\bar{r}_2 = \rho \cdot r_2 + s_2$;
2. $w_z = \rho \cdot t_z + v_z$ and $w_i = \rho \cdot t_i + v_i$, for $i = 0$ to 3;
3. $w_4 = \rho \cdot t_4 + v_4$, where $t_4 \coloneqq t_0 - t_1 \cdot r_1 - t_2 \cdot r_2$;
4. $z_i = \rho \cdot x_i + y_i$ for each $i \in \{0, 2, 3, 4\}$.

Output $\mathsf{resp} \in \mathbb{G} \times \mathbb{Z}_p^{\ell+12}$ as

$$\big(C_z, \{\bar{m}_i\}_{i=1}^{\ell}, \bar{r}_1, \bar{r}_2, w_z, \{w_i\}_{i=0}^{4}, \{z_i\}_{i=0,2,3,4}\big).$$

Verify Given $(\mathsf{com}; \mathsf{chall}; \mathsf{resp})$ return 0 if it does not parse correctly or if the following relations do not hold:

1. $(\hat{D}_1/\hat{g}_{\ell+2})^{\rho} \cdot \hat{E}_1 = \hat{g}_2^{\bar{m}_1} \cdots \hat{g}_{\ell+1}^{\bar{m}_\ell} \cdot g^{\bar{r}_1}$ and $(\hat{D}_2/\hat{g}_{2\ell+3})^{\rho} \cdot \hat{E}_2 = \hat{g}_{\ell+3}^{\bar{m}_1} \cdots \hat{g}_{2\ell+2}^{\bar{m}_\ell} \cdot g^{\bar{r}_2}$;
2. $\hat{D}_0^{\rho} \cdot \hat{E}_0 = \hat{g}_z^{w_z} \cdot \hat{g}_1^{w_1} \cdot \hat{D}_1^{w_2} \cdot \hat{D}_2^{w_3} \cdot \hat{g}^{-w_0}$ and $C_0^{\rho} \cdot F_0 = C_2^{\bar{r}_1} \cdot C_3^{\bar{r}_2} \cdot g^{w_4}$.
3. $T_i^{\rho} \cdot V_i = g^{w_i} f^{z_i}$ for each $i \in \{0, 2, 3, 4\}$ and

$$(T_0/T_4)^{\rho} \cdot S_0 = T_2^{\bar{r}_1} \cdot T_3^{\bar{r}_2}. \tag{6.9}$$

Then, return 1 if and only if

$$\begin{aligned} &e(C_0, \hat{g}) \cdot e(g, \hat{D}_0) \cdot e(\Omega, \hat{g}_{2\ell+4})^{-1} \\ &\quad = e(C_1, \hat{g}_1) \cdot e(C_2, \hat{D}_1) \cdot e(C_3, \hat{D}_2) \cdot e(C_z, \hat{g}_z). \end{aligned} \tag{6.10}$$

It is worth noticing that no pairing evaluation is required until the final step of Verify, which is almost as efficient as the verification of underlying signatures. Moreover, the prover's first message com is of constant-size and the communication complexity of the protocol exceeds the length of the witness by a constant additive overhead.

Theorem 6.5. *The above interactive scheme is a secure Σ-protocol for the language L_{sig} induced by the relation $R_{sig}(\mathsf{pk}, (\vec{\sigma}, \vec{m})) = 1$ if and only if $\mathsf{Verify}'(\mathsf{pk}, \vec{\sigma}, \vec{m}) = 1$, where $(\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}')$ is the signature of Section 6.2.*

Proof. Correctness. Expanding an honestly generated $\hat{D}_0 = \hat{g}_z^{t_z} \cdot \hat{g}_1^{t_1} \cdot \hat{D}_1^{t_2} \cdot \hat{D}_2^{t_3} \cdot \hat{g}^{-t_0}$ in equation (6.10) and regrouping the pairing factors gives

$$\begin{aligned} &e(C_0 \cdot g^{-t_0}, \hat{g}) \cdot e(\Omega, \hat{g}_{2\ell+4})^{-1} \\ &\qquad = e(C_1 \cdot g^{-t_1}, \hat{g}_1) \cdot e(C_2 \cdot g^{-t_2}, \hat{D}_1) \\ &\qquad\qquad \cdot e(C_3 \cdot g^{-t_3}, \hat{D}_2) \cdot e(C_z \cdot g^{-t_z}, \hat{g}_z). \end{aligned}$$

Now, expanding the commitments to group elements in $\mathbb{G}$ reduces this equation to

$$\begin{aligned} & e(\sigma_2^{r_1} \cdot \sigma_3^{r_2}, \hat{g}) \cdot e(\Omega, \hat{g}_{2\ell+4})^{-1} \\ & \quad = e(\sigma_1, \hat{g_1}) \cdot e(\sigma_2, \hat{D}_1) \cdot e(\sigma_3, \hat{D}_2) \cdot e(\pi, \hat{g}_z) \end{aligned}$$

which holds true for valid witnesses when $\hat{D}_1 = d_1 \cdot \hat{g}^{r_1}$ and $\hat{D}_2 = d_2 \cdot \hat{g}^{r_2}$. Remaining verifications of items 1,2,3 follow from the correctness of the built-in Σ-protocols.

Special-Soundness. We assume two accepting transcripts $(\mathsf{com}, \rho, \mathsf{resp})$, $(\mathsf{com}, \rho', \mathsf{resp}')$ with $\rho \neq \rho'$. The special soundness of the sub-protocols involving $\hat{D}_1, \hat{D}_2$ (with $\hat{E}_1, \hat{E}_2$) – consisting of steps 1 and 2 of Commit and step 1 of Verify – ensures the extraction of $m_1, \ldots, m_\ell, r_1, r_2$ satisfying $\hat{D}_1 = d_1 \cdot \hat{g}^{r_1}$, where $d_1 = \hat{g}_2^{m_1} \cdots \hat{g}_{\ell+1}^{m_\ell} \cdot \hat{g}_{\ell+2}$, and $\hat{D}_2 = d_2 \cdot \hat{g}^{r_2}$, where $d_2 = \hat{g}_{\ell+3}^{m_1} \cdots \hat{g}_{2\ell+2}^{m_\ell} \cdot \hat{g}_{2\ell+3}$. From step 2 of Verify, a similar argument on $\hat{D}_0$ (with $\hat{E}_0$) implies the extractability of $(t_z, t_0, t_1, t_2, t_3, t_4)$ such that $\hat{D}_0 = \hat{g}_z^{t_z} \cdot \hat{g}_1^{t_1} \cdot \hat{D}_1^{t_2} \cdot \hat{D}_2^{t_3} \cdot \hat{g}^{-t_0}$. Moreover, together with previously extracted (r_1, r_2), step 2 of Verify also guarantees that t_4 satisfies $C_0 = C_2^{r_1} \cdot C_3^{r_2} \cdot g^{t_4}$.

We now state that quantities $\{\sigma_i = C_i \cdot g^{-t_i}\}_{i \in \{1,2,3\}}$ and $\pi = C_z \cdot g^{-t_z}$ satisfy (6.3), so that, together with $\vec{m} = (m_1, \ldots, m_\ell)$, they form a valid witness for R_{sig}. Namely, $(\sigma, \vec{m}) = ((\sigma_1, \sigma_2, \sigma_3, \pi), (m_1, \ldots, m_\ell))$ is a valid message-signature pair.

To see this, define $\sigma_0 = C_0 \cdot g^{-t_0}$. Since equation (6.10) holds by hypothesis, if we expand all commitments using extracted values, we find

$$\begin{aligned} & e(\sigma_0, \hat{g}) \cdot e(\Omega, \hat{g}_{2\ell+4})^{-1} \\ & \quad = e(\sigma_1, \hat{g_1}) \cdot e(\sigma_2, d_1 \cdot \hat{g}^{r_1}) \cdot e(\sigma_3, d_2 \cdot \hat{g}^{r_2}) \cdot e(\pi, \hat{g}_z). \end{aligned}$$

We are thus left with showing that $\sigma_0 = \sigma_2^{r_1} \cdot \sigma_3^{r_2}$ or, equivalently, $e(\sigma_0, \hat{g}) = e(\sigma_2, \hat{g}^{r_1}) \cdot e(\sigma_3, \hat{g}^{r_2})$. Remember that, from step 2 of Verify, we know that extracted $(r_1, r_2, t_4) \in \mathbb{Z}_p^3$ form a representation of C_0 w.r.t. the base (C_0, C_2, g): i.e., $C_0 = C_2^{r_1} \cdot C_3^{r_2} \cdot g^{t_4}$, which, from the definition of $\sigma_0, \sigma_2, \sigma_3$, yields $\sigma_0 \cdot g^{t_0} = \sigma_2^{r_1} \cdot \sigma_3^{r_2} \cdot g^{t_2 \cdot r_1 + t_3 \cdot r_2 + t_4}$. Hence, we are done if we can show that $t_0 = t_2 r_1 + t_3 r_2 + t_4$. But this exactly what step 3 of Verify and the special soundness of the sub-protocol involving (T_0, T_2, T_3, T_4) tells us. First, we have a representation of these T_i's w.r.t. the basis $(g, f) \in \mathbb{G}^2$ which guarantees that we are working on the already extracted (t_0, t_2, t_3, t_4) involved in the expressions of $\hat{D}_0$ and C_0. Second, the verification equation (6.9) ensures that $T_0 = T_2^{r_1} \cdot T_3^{r_2} \cdot T_4$ and the final result follows by replacing them by their representation.

Perfect SHVZK. To show this property we must build a simulator that, on input of a challenge $\mathsf{chall} = \rho \in_R \mathbb{Z}_p$, emulates a valid transcript without any witness. First, we need to compute a random tuple $C_z, \{C_i\}_{i=0}^3, \{\hat{D}\}_{i=0}^2$ constrained to satisfy the verification equation (6.10).

From the identity $e(\Omega, \hat{g}_{2\ell+4})^{-1} = e(\Omega^{-1}, \hat{g}_{2\ell+4})$ we first pick $a_0, a_1, a_2, a_z \leftarrow \mathbb{Z}_p$, $\hat{D}_1 \leftarrow \widehat{\mathbb{G}}$ and we have $e(\Omega, \hat{g}_{2\ell+4})^{-1} = e(\Omega^{-1}, \hat{g}_{2\ell+4} \cdot \hat{g}^{a_0} \hat{g}_1^{a_1} \hat{D}_1^{a_2} \hat{g}_z^{a_z}) \cdot e(\Omega^{a_0}, \hat{g}) \cdot e(\Omega^{a_1}, \hat{g}_1) \cdot e(\Omega^{a_2}, \hat{D}_1) \cdot e(\Omega^{a_z}, \hat{g}_z)$, so that we can set $C_0 = \Omega^{-a_0}$, $C_1 = \Omega^{a_1}$, $C_2 = \Omega^{a_2}$ and $C_z = \Omega^{a_z}$. Let $\hat{B} := \hat{g}_{2\ell+4} \cdot \hat{g}^{a_0} \hat{g}_1^{a_1} \hat{D}_1^{a_2} \hat{g}_z^{a_z}$. Now, we can introduce the constant $g \in \mathbb{G}$ in the equation by picking $a_g \leftarrow \mathbb{Z}_p$ since $e(\Omega^{-1}, \hat{B}) = e(\Omega^{-1} \cdot g^{a_g}, \hat{B}) \cdot e(g, \hat{B}^{-a_g})$. Then, we finally set $\hat{D}_0 = \hat{B}^{a_g}$, $\hat{D}_2 = \hat{B}^{a_3}$ and $C_3 = (\Omega^{-1} \cdot g^{a_g})^{1/a_3}$ for a random $a_3 \leftarrow \mathbb{Z}_p$.

To complete the simulated transcript, we run a parallel execution of the simulators of all Σ-protocols used as subroutines.

More explicitly, first pick $\rho \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and

$$\bar{m}_1, \ldots, \bar{m}_\ell, \bar{r}_1, \bar{r}_2, w_z, w_0, \ldots, w_4, z_0, z_2, z_3, z_4 \hookleftarrow \mathcal{U}(\mathbb{Z}_p).$$

Also, choose $T_0, T_2, T_3, T_4 \hookleftarrow \mathcal{U}(\mathbb{G})$ and do the following:

1. Compute
$$\hat{E}_1 = (\hat{D}_1/\hat{g}_{\ell+2})^{-\rho} \cdot \hat{g}_2^{\bar{m}_1} \cdots \hat{g}_{\ell+1}^{\bar{m}_\ell} \cdot g^{\bar{r}_1}$$
and, similarly,
$$\hat{E}_2 = (\hat{D}_2/\hat{g}_{2\ell+3})^{-\rho} \cdot \hat{g}_{\ell+3}^{\bar{m}_1} \cdots \hat{g}_{2\ell+2}^{\bar{m}_\ell} \cdot g^{\bar{r}_2};$$
2. Compute
$$F_0 = C_0^{-\rho} \cdot C_2^{\bar{r}_1} \cdot C_3^{\bar{r}_2} \cdot g^{w_4}$$
as well as
$$\hat{E}_0 = \hat{D}_0^{\rho} \cdot \hat{g}_z^{w_z} \cdot \hat{g}_1^{w_1} \cdot \hat{D}_1^{w_2} \cdot \hat{D}_2^{w_3} \cdot \hat{g}^{-w_0};$$
3. Compute
$$V_i = T_i^{-\rho} \cdot g^{v_i} f^{z_i},$$
for each $i \in \{0, 2, 3, 4\}$, and
$$S_0 = (T_0/T_4)^{-\rho} \cdot T_2^{\bar{r}_1} \cdot T_3^{\bar{r}_2}.$$

This concludes the proof. □

6.3.2 Signing a Committed Message

At a high level, the protocol involves a committer who wants to get a signature on $\mathbf{m} = (m_1, \ldots, m_\ell)$ and first computes a commitment of the form $c_v = v_1^{m_1} \cdots v_\ell^{m_\ell} \cdot u^r$, where u is the extra public parameter (with unknown discrete log). The signer gives back elements of the form $\tau_1 = g^\omega c_v^s$, $\tau_2 = g^s$, $\tau_3 = h^s$ which is almost the desired signature. To get the component σ_1 of the right form relatively to τ_2, τ_3 the committer has to remove the factor u^{rs} from τ_1. Then, the signer also sends $\tau_0 = u^s$ to enable removing τ_0^r. In the protocol some randomizing steps are included as well as other additional components allowing the committer to extract π, the QA-NIZK part of the signature. In the security proof of the protocol we thus have to show that the additional value $\tau_0 = u^s$ does not affect the unforgeability of the signature.

The protocol. At the beginning of a new run of the protocol, the committer has a vector $\mathbf{m} = (m_1, \ldots, m_\ell)$, the public-key of the signature scheme and the extra generator $u \in \mathbb{G}$ (which can be a hashed point), the signer also has the secret key of the signature scheme but not $\mathbf{m}$. To get a signature on $\mathbf{m}$, the committer picks $r \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and computes a perfectly hiding commitment $c_v = v_1^{m_1} \cdots v_\ell^{m_\ell} \cdot u^r \in \mathbb{G}$. Besides, it also computes the elements $c_z = z_2^{m_1} \cdots z_{\ell+1}^{m_\ell} \cdot u^{t_z}$. The signer receives these commitments and they both engage in an interactive proof of knowledge of an equal representation of c_v relatively to the basis $(v_1, \ldots, v_\ell; u)$ and c_z relatively to the basis $(z_2, \ldots, z_{\ell+1}; u)$, where the signer

plays the role of the verifier. Depending on the success of the proof the signer computes what we can call a "pre-signature" consisting of the following group elements

$$\tau_1 = g^{\omega} \cdot (c_v \cdot w)^s, \qquad \tau_3 = h^s, \qquad \pi_0 = z_1^{\omega} \cdot c_z^s \cdot z_{\ell+2}^s,$$
$$\tau_2 = g^s, \qquad \tau_0 = u^s,$$

for a random $s \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$. In the final step, the user received the pre-signature, then picks $s' \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and computes $(\sigma_1, \sigma_2, \sigma_3, \pi) \in \mathbb{G}^4$ as follows

$$\sigma_1 = \tau_1 \cdot \tau_0^{-r} \cdot (v_1^{m_1} \cdots v_\ell^{m_\ell} \cdot w)^{s'}, \qquad \sigma_2 = \tau_2 \cdot g^{s'},$$
$$\pi = \pi_0 \cdot \tau_0^{-t_z} \cdot (z_2^{m_1} \cdots z_{\ell+1}^{m_\ell} \cdot z_{\ell+2})^{s'}, \qquad \sigma_3 = \tau_3 \cdot h^{s'}.$$

Finally the user checks the validity of the signature. Depending on the validity, the user outputs the signature or a failure symbol $\perp$.

We notice that the number of transmitted group elements is constant and no pairing is needed before the signature verification phase. In comparison, the construction of [CL02a] requires groups of larger hidden order and their protocol for signing committed message blocks requires a linear number of range proofs.

Security. We briefly sketch the proof of the above protocol in front of malicious entities since classical arguments can be applied. Assuming that the committer uses secure ZKPK and does not output $\perp$, a malicious signer which receives perfectly hiding commitments c_v, c_z cannot tell apart an honest proof from a simulated proof. Consequently the signer learns nothing from $\mathbf{m}$ during the execution of the protocol. In the other case, we have to show that a corrupted committer remains unable to produce valid signature on a new vector $\mathbf{m}^\star$. First, since the generation of u is not under the controlled of the committer but of the random oracle, u can be made independent of rest of pk. Then, we only need to show that the signature remains unforgeable when τ_0 is given in the signature. Since $\mathbf{m}$ and s can be extracted from the proof of knowledge the reduction can output a signature on $\mathbf{m}$. Moreover it is easy to see from the security proof (in Section 6.1.1) of the signature how this additional element can be simulated. Actually the only place in the reduction where τ_0 could not be computed directly as u^s for a known s is when the challenger $\mathcal{B}$ has to embed an SXDH challenge in a simulated signature. Given (g, h, g^b, h^{b+c}), $\mathcal{B}$ can compute $u = g^{a_u} h^{b_u}$ from random $a_u, b_u \leftarrow \mathbb{Z}_p$ and program the random oracle to output this element u as the specification of the public-key would do. Then to simulate τ_0 $\mathcal{B}$ simply has to compute $\tau_0 = (g^b)^{a_u}(h^{b+c})^{b_v} = u^b h^{c \cdot b_v}$ which is u^b or random. The rest of the reduction remains unchanged since the value a_u, b_u are completely independent of those already described in the sketch of proof in Section 6.1.1.

Remark. Since a malicious signer may know the simulation trapdoor $\mathsf{tk} = \{\chi_i\}_{i=1}^{2\ell+4}$ of the underlying QA-NIZK argument, he could produce valid signature so that $\log_g \sigma_2 \neq \log_h \sigma_3$. Then, if the committer later needs to proof knowledge of the received signature it then has to use the sigma protocol of Section 6.2 where both σ_2 and σ_3 only appear in committed form.

6.4 The Dynamic Group Signatures Scheme

We adapt the protocol of section 6.2 to build a dynamic group signature [BSZ05, KY06]. At a high level, each group member obtains a membership certificate consisting of a signature

$(\sigma_1, \sigma_2, \sigma_3, \pi)$ on a message $\mathsf{ID} \in \mathbb{Z}_p$ which is only known to the group member. During the joining protocol, each group member thus obtains a signature on a committed message $\mathsf{ID} \in \mathbb{Z}_p$. Here, we use a deterministic commitment to ID, which suffices to ensure security against framing attacks and allows for a better efficiency. When signing a message, each group member verifiably encrypts the components (σ_1, π) of his membership certificate that depend on ID (and not σ_2, σ_3 which can be assumed to be honestly computed here, unlike in the previous section). For the sake of efficiency, we use a randomness re-using [BBKS07] variant of the Cramer-Shoup encryption scheme [CS98] whereby σ_1 and π are both encrypted using the same encryption exponent $\theta \in \mathbb{Z}_p$. For public verifiability purposes, the validity of Cramer-Shoup ciphertexts is demonstrated using Σ-protocols and the Fiat-Shamir heuristic [FS86] (somewhat in the fashion of [SG98]) rather than designated verifier NIZK proofs [CS98].

In the join protocol, the user proves knowledge of his membership secret $\mathsf{ID} \in \mathbb{Z}_p$ in a zero-knowledge manner, which restricts the group manager to sequentially interact with prospective users. However, this limitation can be removed using an extractable commitment as in [DP06].

Keygen(λ, N)**:** given $\lambda \in \mathbb{N}$, and the maximum number of users $N \in \mathsf{poly}(n)(\lambda)$, choose asymmetric bilinear groups $\mathsf{cp} = (\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T, p)$ of order $p > 2^\lambda$.

1. Generate a key pair $(\mathsf{pk}_s, \mathsf{sk}_s)$ for the scheme of section 6.2 for a one-block message (i.e., $\ell = 1$). The secret key is $\mathsf{sk}_s = \omega$, while the public key is
$$\mathsf{pk}_s = (\mathsf{cp},\ g,\ h,\ \hat{g},\ \vec{v} = (v, w),\ \Omega = h^\omega,\ \mathsf{crs}),$$
where $\mathsf{crs} = (\{z_j\}_{j=1}^3, \hat{g}_z, \{\hat{g}_i\}_{i=1}^6)$.
2. Pick $x_z, y_z, x_\sigma, y_\sigma, x_{\mathsf{ID}}, y_{\mathsf{ID}} \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and set
$$X_z = g^{x_z} h^{y_z}, \qquad X_\sigma = g^{x_\sigma} h^{y_\sigma}, \qquad X_{\mathsf{ID}} = g^{x_{\mathsf{ID}}} h^{y_{\mathsf{ID}}}.$$
3. Choose a hash function $H : \{0,1\}^* \times \mathbb{G}^{10} \times \mathbb{G}_T \to \mathbb{Z}_p$ that will be modeled as a random oracle.
4. Define $\mathcal{Y} = \{\mathsf{pk}_s, X_z, X_\sigma, X_{\mathsf{ID}}\}$ to be the group public key. The group manager's private key is $\mathcal{S}_{\mathsf{GM}} = \omega = \mathsf{sk}_s$ whereas the opening authority's private key consists of $\mathcal{S}_{\mathrm{OA}} = (x_z, y_z, x_\sigma, y_\sigma, x_{\mathsf{ID}}, y_{\mathsf{ID}})$.

Join$^{(\mathsf{GM}, \mathcal{U}_i)}$**:** The group manager GM, and the prospective user $\mathcal{U}_i$ run the following interactive protocol:

1. $\mathcal{U}_i$ chooses $\mathsf{ID} \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and sends the following to GM: $(V_{\mathsf{ID}}, Z_{\mathsf{ID}}, \hat{G}_{2,\mathsf{ID}}, \hat{G}_{4,\mathsf{ID}}) = (v^{\mathsf{ID}}, z_2^{\mathsf{ID}}, \hat{g}_2^{\mathsf{ID}}, \hat{g}_4^{\mathsf{ID}})$
2. GM checks that V_{ID} does not appear in any transcript of St and abort if it does. Otherwise (i.e., if V_{ID} is fresh), GM verifies that: for $k = 2, 4$,
$$e(V_{\mathsf{ID}}, \hat{g}_k) \stackrel{?}{=} e(v, \hat{G}_{k,\mathsf{ID}}), \qquad e(Z_{\mathsf{ID}}, \hat{g}_k) \stackrel{?}{=} e(z_2, \hat{G}_{k,\mathsf{ID}}).$$
If all tests pass, samples a fresh index $i \in \mathbb{Z}_p$ and sends it to $\mathcal{U}_i$, otherwise abort.

3. $\mathcal{U}_i$ runs an interactive zero-knowledge proof of knowledge of $\mathsf{ID} = \log_v(V_{\mathsf{ID}})$ in interaction with GM. For instance, the 4-round protocol of Cramer *et al.* [CDM00] can be used for this purpose. Let $\pi_K(\mathsf{ID})$ denote the interaction transcript.

4. GM uses $V_{\mathsf{ID}} = v^{\mathsf{ID}}$ to sign ID using the scheme of section 6.2: i.e., GM picks $s \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$, and uses $\mathcal{S}_{\mathsf{GM}} = \omega$ to compute $\sigma_1 = g^{\omega} \cdot (V_{\mathsf{ID}} \cdot w)^s = g^{\omega} \cdot (v^{\mathsf{ID}} \cdot w)^s$ and

$$\sigma_2 = g^s, \qquad \sigma_3 = h^s.$$

   Then GM uses Z_{ID} to generate the QA-NIZK proof $\pi \in \mathbb{G}$ as

$$\pi = z_1^{\omega} \cdot (Z_{\mathsf{ID}} \cdot z_3)^s = z_1^{\omega} \cdot (z_2^{\mathsf{ID}} \cdot z_3)^s$$

   and finally sends $\mathsf{cert}_i = (i, V_{\mathsf{ID}}, \sigma_1, \sigma_2, \sigma_3, \pi)$

5. Finally GM and $\mathcal{U}_i$ respectively store

$$\mathsf{transcript}_i = \Big((Z_{\mathsf{ID}}, \hat{G}_{2,\mathsf{ID}}, \hat{G}_{4,\mathsf{ID}}), \pi_K(\mathsf{ID}), \mathsf{cert}_i \Big) \tag{6.11}$$

   and $(\mathsf{cert}_i, \mathsf{sec}_i) = ((i, V_{\mathsf{ID}}, \sigma_1, \sigma_2, \sigma_3, \pi), \mathsf{ID})$.

Sign$(\mathcal{Y}, \mathsf{sec}_i, \mathsf{cert}_i, M)$**:** Given a message $M \in \{0,1\}^*$ and a secret $\mathsf{sec}_i = \mathsf{ID}$, the user $\mathcal{U}_i$ does the following:

1. Re-randomize the certificate cert_i. Namely, choose $r \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and compute $\tilde{\sigma}_2 = \sigma_2 \cdot g^r$, $\tilde{\sigma}_3 = \sigma_3 \cdot h^r$, $\tilde{\sigma}_1 = \sigma_1 \cdot (v^{\mathsf{ID}} \cdot w)^r$, $\tilde{\pi} = \pi \cdot (z_2^{\mathsf{ID}} \cdot z_3)^r$.

2. Encrypt elements $\tilde{\pi}$, $\tilde{\sigma}_1$ and v^{ID} from the membership certificate. Specifically, choose $\theta \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and compute the Cramer-Shoup ciphertext $C_{\mathsf{CS}} = (C_1, C_2, C_z, C_\sigma, C_{\mathsf{ID}})$, where $C_1 = g^\theta$, $C_2 = h^\theta$,

$$C_z = \tilde{\pi} \cdot X_z^\theta, \qquad C_\sigma = \tilde{\sigma}_1 \cdot X_\sigma^\theta, \qquad C_{\mathsf{ID}} = v^{\mathsf{ID}} \cdot X_{\mathsf{ID}}^\theta.$$

3. Then, prove knowledge of $(\mathsf{ID}, \theta) \in \mathbb{Z}_p^2$ such that

$$C_1 = g^\theta, \qquad C_2 = h^\theta, \qquad C_{\mathsf{ID}} = v^{\mathsf{ID}} \cdot X_{\mathsf{ID}}^\theta,$$

$$\begin{aligned} &\big(e(C_z, \hat{g}_z) \cdot e(C_\sigma, \hat{g}_1) \cdot e(\tilde{\sigma}_2, \hat{g}_3) \cdot e(\tilde{\sigma}_3, \hat{g}_5) \cdot e(\Omega, \hat{g}_6)\big) \\ &= \big(e(X_z, \hat{g}_z) \cdot e(X_\sigma, \hat{g}_1)\big)^{\theta} \cdot \big(e(\tilde{\sigma}_2, \hat{g}_2) \cdot e(\tilde{\sigma}_3, \hat{g}_4)\big)^{-\mathsf{ID}}. \end{aligned}$$

   Namely, sample random $r_{\mathsf{ID}}, r_\theta \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$, compute

$$\begin{aligned} R_1 = g^{r_\theta}, \quad R_2 = h^{r_\theta}, \quad R_3 = v^{r_{\mathsf{ID}}} \cdot X_{\mathsf{ID}}^{r_\theta}, \\ R_4 = \big(e(X_z, \hat{g}_z) \cdot e(X_\sigma, \hat{g}_1)\big)^{r_\theta} \cdot \big(e(\tilde{\sigma}_2, \hat{g}_2) \cdot e(\tilde{\sigma}_3, \hat{g}_4)\big)^{-r_{\mathsf{ID}}} \end{aligned}$$

   and then define c as $c = H(M, C_{\mathsf{CS}}, \tilde{\sigma}_2, \tilde{\sigma}_3, R_1, R_2, R_3, R_4)$. Finally compute the two responses $s_\theta = r_\theta + c \cdot \theta$, $s_{\mathsf{ID}} = r_{\mathsf{ID}} + c \cdot \mathsf{ID}$ both in $\mathbb{Z}_p$.

4. Return the signature Σ which consists of

$$\Sigma = (C_{\mathsf{CS}}, \tilde{\sigma}_2, \tilde{\sigma}_3, c, s_{\mathsf{ID}}, s_\theta) \in \mathbb{G}^7 \times \mathbb{Z}_p^3 \tag{6.12}$$

Verify$(\mathcal{Y}, M, \Sigma)$**:** Parse the signature Σ as in (6.12) and C_{CS} as $(C_1, C_2, C_z, C_\sigma, C_{\mathsf{ID}})$. Then, output 1 if the the zero-knowledge proof verifies. Namely,

1. Compute the group elements $R_1, R_2, R_3 \in \mathbb{G}$ as:

$$\begin{aligned} R_1 = g^{s_\theta} \cdot C_1^{-c}, \quad R_2 = h^{s_\theta} \cdot C_2^{-c}, \\ R_3 = v^{s_{\mathsf{ID}}} \cdot X_{\mathsf{ID}}^{s_\theta} \cdot C_{\mathsf{ID}}^{-c}; \end{aligned} \tag{6.13}$$

and the element $R_4 \in \mathbb{G}_T$ as

$$\begin{aligned} &\big(e(X_z, \hat{g}_z) \cdot e(X_\sigma, \hat{g}_1)\big)^{s_\theta} \cdot \big(e(\tilde{\sigma}_2, \hat{g}_2) \cdot e(\tilde{\sigma}_3, \hat{g}_4)\big)^{-s_{\mathsf{ID}}} \\ &\quad \cdot \big(e(C_z, \hat{g}_z) \cdot e(C_\sigma, \hat{g}_1) \cdot e(\tilde{\sigma}_2, \hat{g}_3) \cdot e(\tilde{\sigma}_3, \hat{g}_5) \cdot e(\Omega, \hat{g}_6)\big)^{-c}. \end{aligned} \tag{6.14}$$

2. Return 1 if $c = H(M, C_{\mathsf{CS}}, \tilde{\sigma}_2, \tilde{\sigma}_3, R_1, R_2, R_3, R_4)$ and 0 otherwise.

Open$(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \Sigma)$**:** Given a message-signature pair (M, Σ) and the OA's private key $S_{\mathrm{OA}} = (x_z, y_z, x_\sigma, y_\sigma, x_{\mathsf{ID}}, y_{\mathsf{ID}})$:

1. Decrypt $C_{\mathsf{CS}} = (C_1, C_2, C_z, C_\sigma, C_{\mathsf{ID}})$ by computing

$$\begin{aligned} \sigma_1 = C_\sigma \cdot C_1^{-x_\sigma} \cdot C_2^{-y_\sigma}, \qquad \pi = C_z \cdot C_1^{-x_z} \cdot C_2^{-y_z}, \\ V_{\mathsf{ID}} = C_{\mathsf{ID}} \cdot C_1^{-x_{\mathsf{ID}}} C_2^{-y_{\mathsf{ID}}}. \end{aligned}$$

2. Search V_{ID} in the database of joining transcripts (6.11) and check that it corresponds to a valid signature $(\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3, \tilde{\pi})$ for the committed value V_{ID}. If so, return the corresponding i, otherwise return $\perp$.

It is possible to spare one group element in the signature by eliminating the encryption C_{ID} of v^{ID} which is only used to open signatures in constant time. Then, the opening algorithm has to check for each transcript if $(\tilde{\sigma}_1, \tilde{\sigma}_2, \tilde{\sigma}_3, \tilde{\pi})$ corresponds to the identifier ID embedded in $(\sigma_1, \hat{G}_{2,\mathsf{ID}}, \hat{G}_{4,\mathsf{ID}})$ by testing the relation

$$1 \stackrel{?}{=} e(\tilde{\pi}, \hat{g}_z) \cdot e(\tilde{\sigma}_1, \hat{g}_1) \cdot e(\tilde{\sigma}_2, \hat{G}_{2,\mathsf{ID}} \cdot \hat{g}_3) \cdot e(\tilde{\sigma}_3, \hat{G}_{4,\mathsf{ID}} \cdot \hat{g}_5) \cdot e(\Omega, \hat{g}_6).$$

This results in a modified opening algorithm which takes $O(N)$ in the worst-case. In applications where signature openings are infrequent, this is acceptable.

6.4.1 Security

The security of the above dynamic group signature scheme, namely full anonymity, security against misidentifications and security against framing attacks that are defined in Section 5.3 are expressed in Theorem 6.6, Theorem 6.9 and Theorem 6.10 respectively. The security relies on the SXDH assumption for anonymity and misidentification, and on the SDL assumption for non-frameability.

Theorem 6.6. *If SXDH holds in* $(\mathbb{G}, \widehat{\mathbb{G}}, \mathbb{G}_T)$*, the scheme is CCA-anonymous in the random oracle model.*

Proof. We use a sequence of games where, for each i, W_i is the event that the adversary $\mathcal{A}$ wins in Game i.
At the first transition, we need to rely on the security of the computational soundness of the QA-NIZK argument of Section 6.1.1 which relies on the SXDH assumption, since $\tilde{\sigma}_2$ and $\tilde{\sigma}_3$ appear un-encrypted in each group signature.

Game 0: This is the real CCA-anonymity game.
In the challenge phase, the adversary outputs two valid membership certificates and membership secrets $(\mathsf{cert}_0^\star, \mathsf{sec}_0^\star)$, $(\mathsf{cert}_1^\star, \mathsf{sec}_1^\star)$ and obtains a challenge signature which the challenger computes using $(\mathsf{cert}_d^\star, \mathsf{sec}_d^\star)$, where $d \hookleftarrow \mathcal{U}(\{0,1\})$. We define W_0 to be the event that the adversary outputs $d' = d$.

Game 1: This game is as Game 0, except that the challenger $\mathcal{B}$ aborts in the event, which we call F_1, that $\mathcal{A}$ chooses membership certificates $\mathsf{cert}_0^\star, \mathsf{cert}_1^\star$ for which one of the underlying signatures $(\sigma_1^\star, \sigma_2^\star, \sigma_3^\star, \pi^\star)$ correctly verifies but $\log_g(\sigma_2^\star) \neq \log_h(\sigma_3^\star)$. This implies that the vector $(\sigma_1^\star, \sigma_2^{\star\mathsf{ID}}, \sigma_2^\star, \sigma_3^{\star\mathsf{ID}}, \sigma_3^\star, \Omega)$ is outside the row space of the matrix $\mathbf{M}$ (6.1), so that F_1 would contradict the soundness of the QA-NIZK proof of [KW15] (via the same arguments as in Theorem 9 of [LPY15] since the matrix can be witness-samplable here) and thus the DDH assumption in $\widehat{\mathbb{G}}$. We have $|\Pr[W_1] - P[W_0]| \leq \mathrm{Adv}^{\mathsf{DDH}}_{\widehat{\mathbb{G}}}(\lambda)$.

Game 2: We change the way to generate the challenge signature $\Sigma^\star$. Instead of faithfully running the Schnorr-like protocol, we use the HVZK-simulator to produce the proofs $s_\theta, s_{\mathsf{ID}}$ without knowing the witnesses θ, ID. Namely, we pick $c,\ s_\theta,\ s_{\mathsf{ID}} \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ at random and set $R_1 = g^{s_\theta} \cdot C_1^{-c}$, $R_2 = h^{s_\theta} \cdot C_2^{-c}$, $R_3 = v^{s_{\mathsf{ID}}} \cdot X_{\mathsf{ID}}^{s_\theta} \cdot C_{\mathsf{ID}}^{-c}$ as well as $R_4 \in \mathbb{G}_T$ as in (6.14). Then, we program the random oracle and assign the output c to the hash value $H(M, C_{\mathsf{CS}}, \tilde{\sigma}_2, \tilde{\sigma}_3, R_1, R_2, R_3, R_4)$. In the unlikely event that this value was previously defined (which only happens with probability at most $1/p^3$), the challenger aborts. Thus $|\Pr[W_2] - \Pr[W_1]| \leq 1/p^3$

Game 3: We modify again the generation of the challenge signature $\Sigma^\star$. Namely, the challenger computes $C_z, C_\sigma, C_{\mathsf{ID}}$ using $\mathcal{S}_{\mathsf{OA}}$ as follows

$$C_z = \tilde{\pi} \cdot C_1^{x_z} \cdot C_2^{y_z},$$
$$C_\sigma = \tilde{\sigma} \cdot C_1^{x_\sigma} \cdot C_2^{y_\sigma}, \qquad C_{\mathsf{ID}} = v^{\mathsf{ID}} \cdot C_1^{x_{\mathsf{ID}}} \cdot C_2^{y_{\mathsf{ID}}}.$$

The distribution of $(C_z, C_\sigma, C_{\mathsf{ID}})$ remains the same and we have $\Pr[W_3] = \Pr[W_2]$.

Game 4: Here, we modify the distribution of the challenge signature and replace $C_2 = h^{\theta}$ by $C_2 = h^{\theta+\theta'}$, for a randomly chosen $\theta' \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$. We prove in Lemma 6.7 that $|\Pr[W_4] - \Pr[W_3]| \leq \mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}}(\lambda)$.

Game 5: We introduce one more change. Instead of sampling $h \in_R \mathbb{Z}_p$, the challenger chooses a random $\alpha \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ at the beginning of the game, sets $h = g^{\alpha}$ and retains the information $\alpha = \log_g(h)$ (note that we are done with the DDH assumption and we can henceforth use $\alpha = \log_g(h)$). At each signature opening query, the challenger returns $\perp$ on any signature $\Sigma = (C_1, C_2, C_z, C_\sigma, C_{\mathsf{ID}}, \tilde{\sigma}_2, \tilde{\sigma}_3, c, s_{\mathsf{ID}}, s_\theta)$ such that $C_2 \neq C_1^{\alpha}$. Game 5 remains the same as Game 4. until the event E_5 that $\mathcal{A}$ queries the opening of a signature that properly verifies although $C_2 \neq C_1^{\alpha}$. Lemma 6.8 states that $\Pr[E_5] \leq q_O \cdot q_H/p$, where q_O is the number of opening queries and q_H is the number of random oracle queries.

In Game 5, $\Sigma^{\star}$ perfectly hides $(\tilde{\pi}, \tilde{\sigma}_1, v^{\mathsf{ID}})$. Indeed,

$$C_1 = g^{\theta}, \quad C_2 = h^{\theta+\theta'}, \quad C_z = (\tilde{z} \cdot h^{\theta' \cdot y_z}) \cdot X_z^{\theta},$$
$$C_\sigma = (\tilde{\sigma}_1 \cdot h^{\theta' \cdot y_\sigma}) \cdot X_\sigma^{\theta}, \quad C_{\mathsf{ID}} = (v^{\mathsf{ID}} \cdot h^{\theta' \cdot y_{\mathsf{ID}}}) \cdot X_{\mathsf{ID}}^{\theta}$$

and $(y_\sigma, y_z, y_{\mathsf{ID}}) \in \mathbb{Z}_p^3$ are completely independent of $\mathcal{A}$'s view. The only way for $\mathcal{A}$ to infer information about $(y_\sigma, y_z, y_{\mathsf{ID}})$ is to make opening queries on signatures such that $C_2 \neq C_1^{\alpha}$. However, all such signatures are declared invalid in Game 5. It comes that $\Pr[W_5] = 1/2$.

Finally, $\mathcal{A}$'s advantage $\left|\Pr[W_0] - 1/2\right|$ is bounded by

$$\mathrm{Adv}^{\mathsf{DDH}}_{\mathbb{G}}(\lambda) + \mathrm{Adv}^{\mathsf{DDH}}_{\widehat{\mathbb{G}}}(\lambda) + \frac{q_O \cdot q_H}{p} + \frac{1}{p^3},$$

which concludes the proof. □

Lemma 6.7. *In Game 4, the adversary $\mathcal{A}$ wins the anonymity game with negligibly different probabilities than in Game 3 if the DDH assumption holds in $\mathbb{G}$.*

Proof. Let us assume that an adversary $\mathcal{A}$ wins with noticeably different probabilities in Game 4 and Game 3. We then construct a DDH distinguisher $\mathcal{B}$ from $\mathcal{A}$.

Our reduction $\mathcal{B}$ takes as input a DDH instance (g^a, g^b, η), where $\eta = g^{a(b+c)}$ and has to decide with non-negligible probability ε whether $c = 0$ or $c \in_R \mathbb{Z}_p$. To achieve this, $\mathcal{B}$ sets $h = g^a$ and computes the challenge signature as $C_1 = g^b$ and $C_2 = \eta$. The rest of the game continues like in Game 3 (which is also the same as in Game 2). If $\mathcal{A}$ wins and correctly guesses $d' = d \in \{0, 1\}$, $\mathcal{B}$ outputs 1, meaning that $C_2 = h^b = g^{ab}$. Otherwise, $\mathcal{B}$ returns 0 meaning that $(g^a, g^b, \eta) \in_R \mathbb{G}^3$.
It is easy to see that $\mathcal{B}$'s advantage as a DDH distinguisher is ε if $|\Pr[W_4] - \Pr[W_3]| = \varepsilon$. □

Lemma 6.8. *In Game 5, we have $\Pr[E_5] \leq q_O \cdot q_H/p$.*

Proof. This proof uses idea similar to the security proof of the Katz-Wang [KW03] signature scheme. In Game 5, event E_5 happens if $\log_g(C_1) \neq \log_h(C_2)$ and the verification

equations (6.13) and (6.14) holds. In particular, we have $R_1 = g^{s_\theta} \cdot C_1^{-c}$ and $R_2 = h^{s_\theta} \cdot C_2^{-c}$, which can be interpreted as a linear system with unknowns $(c, s_\theta) \in \mathbb{Z}_p^2$

$$\begin{cases} \log_g(R_1) = s_\theta - \log_g(C_1) \cdot c \mod p, \\ \log_h(R_2) = s_\theta - \log_h(C_2) \cdot c \mod p. \end{cases} \tag{6.15}$$

We can assume w.l.o.g. that each opening query is preceded by the corresponding random oracle query (otherwise, the reduction can simply make the hash query for itself). The input of each hash query contains a pair (R_1, R_2) determining the non-homogeneous terms of the linear system (6.15). Since $\log_g(C_1) \neq \log_h(C_2)$, the system is full-rank, so that for each (R_1, R_2), there is exactly one pair $(c, s_\theta) \in \mathbb{Z}_p^2$ that satisfies (6.15). The probability that, in response to a random oracle query, the reduction returns the value of c which is uniquely determined by (6.15) is at most $1/p$. For all hash queries, the probability that one of them be answered with the uniquely determined $c \in \mathbb{Z}_q$ is *at most* q_H/p. A union bound over all opening queries implies that the probability that the event E_4 happens is smaller than $\Pr[E_4] \leq q_O \cdot q_H/p$. □

The proof of security against misidentification attacks requires the reduction to rewind a the proof of knowledge of ID at each execution of the join protocol with the adversary attempting to escape traceability. For this reason, we need to assume that users join the system sequentially, rather than concurrently. However, this problem can be solved as in [DP06] by having the user send an extractable commitment to ID and non-interactively prove (via the Fiat-Shamir heuristic) that he did so correctly. This allows the reduction to extract ID without rewinding the user at each execution of Join. Then, the proof of security against framing attacks must be modified by having the reduction simulate the proof of knowledge of ID (by programming a random oracle) and rely on the hiding property of the extractable commitment.

Theorem 6.9. *In the ROM, the scheme is secure against misidentification attacks under the SXDH assumption in* $(\mathbb{G}, \widehat{\mathbb{G}})$.

Proof. The proof uses the forking technique [PS00] which consists in implicitly rewinding the zero-knowledge proof by running the adversary twice and changing the outputs of the random oracle after the hash query that involves the forgery message. The Forking Lemma [PS00] – more precisely, its generalization given by Bellare and Neven [BN06] – ensures that, after two runs of the adversary, the reduction can extract witnesses of which knowledge is demonstrated by the signature of knowledge.

Let us assume an attacker $\mathcal{A}$ against the misidentification game that wins with non-negligible probability ε. We build an adversary $\mathcal{B}$ against the chosen-message security of the signature scheme of section 6.2.

Keygen. At the key generation, $\mathcal{B}$ invokes its own challenger for the chosen-message security game to obtain the public key pk_s for the signature scheme. pk_s is embedded in the group public key $\mathcal{Y}$. Except for $\mathcal{S}_{\mathsf{GM}}$, all keys are generated as in the normal Keygen algorithm.

Join. To answer joining queries without knowing sk_s, $\mathcal{B}$ uses the knowledge extractor of the proof of knowledge of $\mathsf{ID} = \log_v(V_{\mathsf{ID}})$ to extract the identity to be signed. Namely,

on a Join query, the reduction $\mathcal{B}$ rewinds the adversary $\mathcal{A}$ in order to extract the witness $\mathsf{ID} = \log_v(V_{\mathsf{ID}})$ of which $\mathcal{A}$ demonstrates knowledge at step 3 of the join protocol. Having extracted $\mathsf{ID} \in \mathbb{Z}_p$, $\mathcal{B}$ invokes its own signing oracle on the message ID to obtain $(\sigma_1, \sigma_2, \sigma_3, z, r)$. Then, $\mathcal{B}$ returns $\mathsf{cert}_i = (i, V_{\mathsf{ID}}, \sigma_1, \sigma_2, \sigma_3, z, r)$ as in a normal execution of the join protocol.

At some point, the attacker $\mathcal{A}$ produces a valid forgery

$$(M^\star, \Sigma^\star = (C_1^\star, C_2^\star, C_z^\star, C_\sigma^\star, C_{\mathsf{ID}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, c^\star, s_{\mathsf{ID}}^\star, s_\theta^\star))$$

for which the opening algorithm does not reveal a properly registered identity. With all but negligible probability, $\mathcal{A}$ must have queried the random oracle value

$$H(M^\star, C_{\mathsf{CS}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, R_1^\star, R_2^\star, R_3^\star, R_4^\star)$$

which would have been unpredictable otherwise.

Thus, $\mathcal{B}$ replays the adversary $\mathcal{A}$ with the *same* input and random tape as in the first run. In the second run, the random oracle is also the same until the hash query

$$H(M^\star, C_{\mathsf{CS}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, R_1^\star, R_2^\star, R_3^\star, R_4^\star).$$

At this point, the forking occurs and $\mathcal{B}$ outputs fresh random oracle values. By the Forking Lemma of [BN06], $\mathcal{B}$ obtains two suitably related forgeries with non-negligible probability $\varepsilon \cdot (\varepsilon/q_H - 1/p)$. Namely, $\mathcal{B}$ will obtain two matching transcripts $(C_{\mathsf{CS}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, c^\star, s_{\mathsf{ID}}^\star, s_\theta^\star)$, $(C_{\mathsf{CS}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, c^\dagger, s_{\mathsf{ID}}^\dagger, s_\theta^\dagger)$ of the Σ-protocol for the commitment message

$$\mathsf{com} = (C_{\mathsf{CS}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, R_1^\star, R_2^\star, R_3^\star, R_4^\star).$$

From the responses $s_{\mathsf{ID}}^\star$ and $s_{\mathsf{ID}}^\dagger$ (that necessarily involve the same identifier $\mathsf{ID}^\star$ which is uniquely determined by $C_{\mathsf{CS}}^\star = (C_1^\star, C_2^\star, C_z^\star, C_\sigma^\star, C_{\mathsf{ID}}^\star)$), $\mathcal{B}$ runs the knowledge extractor of to obtain $\mathsf{ID}^\star \in \mathbb{Z}_p$. Namely, given $(c^\star, c'^\star, s_\theta^\star, s_\theta'^\star, s_{\mathsf{ID}}^\star, s_{\mathsf{ID}}'^\star) \in \mathbb{Z}_p^6$ with

$$c^\star \neq c^\dagger, \qquad s_\theta^\star \neq s_\theta^\dagger \qquad s_{\mathsf{ID}}^\star \neq s_{\mathsf{ID}}^\dagger$$

which verifies the relation (6.13) , (6.14) for the same commitment $(R_1^\star, R_2^\star, R_3^\star, R_4^\star) \in \mathbb{G}^4$, one can compute the secrets $\mathsf{ID}^\star = \frac{s_{\mathsf{ID}}^\dagger - s_{\mathsf{ID}}^\star}{c^\star - c^\dagger} \bmod p$ and $\theta^\star = \frac{s_\theta^\dagger - s_\theta^\star}{c^\star - c^\dagger} \bmod p$.

Finally $\mathcal{B}$ uses $\mathcal{S}_{\mathsf{OA}}$ to extract $\tilde{\sigma}_1^\star, \tilde{r}^\star, \tilde{z}^\star$ and outputs $\big(\mathsf{ID}^\star, \sigma^\star = (\tilde{\sigma}_1^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, \tilde{r}^\star, \tilde{z}^\star)\big)$ as a forgery for the signature scheme of Section 6.2. □

Theorem 6.10. *In the ROM, the scheme is secure against framing attacks under the SDL assumption.*

Proof. Let us assume that a PPT adversary $\mathcal{A}$ can create, with advantage ε, a forgery $(M^\star, \sigma^\star)$ that opens to some honest user $i \in U^b$ who did not sign $M^\star$. We give a reduction $\mathcal{B}$ that uses $\mathcal{A}$ to break SDL.

Algorithm $\mathcal{B}$ takes as input an SDL instance $(g, \hat{g}, g^a, \hat{g}^a)$ and uses its interaction with the adversary $\mathcal{A}$ to compute $a \in \mathbb{Z}_p$. To generate the group public key $\mathcal{Y}$, $\mathcal{B}$ runs all the steps of the real setup algorithm Keygen except step 1. At step 1, $\mathcal{B}$ defines the generators

$g, \hat{g}$ in pk_s to be those of its input and computes $h = g^{\alpha_h}$, $v = g^{\alpha_v}$, $w = g^{\alpha_w}$, $\hat{g}_z = \hat{g}^{\alpha_z}$ for randomly chosen scalars $\alpha_h, \alpha_v, \alpha_w, \alpha_z \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$. In order to compute $\{z_j\}_{j=1}^3$ of crs contained in pk_s, $\mathcal{B}$ chooses $\mathsf{tk} = \{\chi_j\}_{j=1}^6$ of step 4 of the key generation algorithm of the signature scheme of Section 6.2 with $\ell = 1$. (Note that when $\ell = 1$, $n = 6$ and that $\{z_j\}_{j=1}^3$ are QA-NIZK argument for the vectors $(g, 1, 1, 1, 1, h)$, $(v, g, 1, h, 1, 1)$ and $(w, 1, g, 1, h, 1)$. Moreover $\{\hat{g}_i = \hat{g}_z^{\chi_i}\}_{i=1}^6$ are the verifying key.) As a result of this setup phase, $\mathcal{B}$ knows $\mathcal{S}_{\mathsf{GM}} = \mathsf{sk}_s = \omega$, $\mathcal{S}_{\mathsf{OA}} = (x_z, y_z, x_\sigma, y_\sigma, x_{\mathsf{ID}}, y_{\mathsf{ID}})$ and even tk. The adversary $\mathcal{A}$ is run on input of the group public key $\mathcal{Y} \coloneqq (\mathsf{pk}_s, (X_z, X_\sigma, X_{\mathsf{ID}}), H)$, which has the same distribution as in the real attack game.

Should $\mathcal{A}$ decide to corrupt the group manager or the opening authority during the game, $\mathcal{B}$ is able to reveal $\mathcal{S}_{\mathsf{GM}} = \mathsf{sk}_s$ and $\mathcal{S}_{\mathsf{OA}}$ when requested. In addition, $\mathcal{B}$ must be able to answer the following queries.

- $Q_{\mathsf{b\text{-}join}}$-queries: At any time $\mathcal{A}$ can act as a corrupted group manager and introduce a new honest user i in the group by invoking the $Q_{\mathsf{b\text{-}join}}$ oracle. Then, $\mathcal{B}$ runs $\mathsf{J}_{\mathsf{user}}$ on behalf of the honest user in an execution of Join. At step 1 of Join, $\mathcal{B}$ picks a random $\delta_i \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and uses tk to compute the tuple $(V_i, Z_i, \hat{G}_{2,i}, \hat{G}_{4,i})$, for an unknown $\mathsf{sec}_i = \mathsf{ID}_i = a \cdot \delta_i \in \mathbb{Z}_p$, that J_{GM} expects at step 1 of the join protocol. Namely, $\mathcal{B}$ computes the vector $\vec{v}_i = (V_i, G_i, 1, H_i, 1, 1) = (v, g, 1, h, 1, 1)^{\mathsf{ID}_i}$ as

$$V_i = (g^a)^{\alpha_v \cdot \delta_i}, \quad G_i = (g^a)^{\delta_i}, \quad H_i = (g^a)^{\alpha_h \cdot \delta_i},$$

  and then computes Z_i as a simulated QA-NIZK proof for $\vec{v}_i \in \mathbb{G}^6$ using tk. A straightforward calculation shows that $Z_i = z_2^{\mathsf{ID}_i}$ since the QA-NIZK argument of Section 6.1.1 has a deterministic proving algorithm, so that $(V_i, Z_i, \hat{G}_{2,i}, \hat{G}_{4,i})$ successfully passes the test of step 2. As for the last two components, for each $j \in \{2, 4\}$, $\mathcal{B}$ computes

$$\hat{G}_{j,i} \coloneqq (\hat{g}^a)^{\delta_i(\alpha_z \chi_j + \alpha_r \gamma_j)} = (\hat{g}_z^{\chi_j} \hat{g}_r^{\gamma_j})^{\mathsf{ID}_i} = \hat{g}_j^{\mathsf{ID}_i},$$

  At step 3 of Join, $\mathcal{B}$ simulates the interactive proof of knowledge of $\mathsf{ID}_i = \log_v(V_i)$ using the simulator. In the rest of the protocol, $\mathcal{B}$ proceeds like the actual run and obtains $\mathsf{cert}_i = (i, V_i, \sigma_1, \sigma_2, \sigma_3, \pi)$. Finally, $\mathcal{B}$ stores $(\mathsf{cert}_i, Z_i, \delta_i, \hat{G}_{2,i}, \hat{G}_{4,i})$.

- Q_{sig}-queries: When $\mathcal{A}$ requests user $i \in U^b$ to sign a message M, $\mathcal{B}$ is able to use the membership certificate $\mathsf{cert}_i = (i, V_i, \sigma_1, \sigma_2, \sigma_3, \pi)$ to compute the ciphertext C_{CS} at steps 1-2 of the signing algorithm. While $\mathcal{B}$ does not know the witness $\mathsf{ID}_i = a \cdot \delta_i \in \mathbb{Z}_p$ to generate a proof at step 3, $\mathcal{B}$ is able to simulate the non-interactive proof $(c, s_{\mathsf{ID}}, s_\theta)$, for a randomly chosen challenge $c \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ by programming the random oracle. More precisely, $\mathcal{B}$ re-randomizes the certificate cert_i by picking $r \hookleftarrow \mathcal{U}(\mathbb{Z}_p)$ and computing

$$\tilde{\sigma}_1 = \sigma_1 \cdot (V_i \cdot w)^r \qquad \tilde{\sigma}_2 = \sigma_2 \cdot g^r,$$
$$\tilde{\pi} = \pi \cdot (Z_i \cdot z_3)^r, \qquad \tilde{\sigma}_3 = \sigma_3 \cdot h^r.$$

  Then $\mathcal{B}$ encrypts $\tilde{\pi}$, $\tilde{\sigma}_1$ and V_i as in the real signing algorithm to get the encryption $C_{\mathsf{CS}} = (C_1, C_2, C_z, C_\sigma, C_{\mathsf{ID}})$. Then, $\mathcal{B}$ chooses $c, s_{\mathsf{ID}}, s_\theta \in \mathbb{Z}_p$ and computes R_1, R_2, R_3, R_4 as in (6.13) and (6.14) of Verify. Finally, $\mathcal{B}$ programs H to return c on inputs $(M, C_{\mathsf{CS}}, \tilde{\sigma}_2, \tilde{\sigma}_3, R_1, R_2, R_3, R_4)$. In the event that H is already defined at

that point, $\mathcal{B}$ aborts. The probability to fail at one signing query is $\leq q_s/p^3$, where q_s is the number of signing queries.

When $\mathcal{A}$ halts, it presumably frames some honest user $i^\star \in U^b$ by outputting a signature

$$\Sigma^\star = (C_1^\star, C_2^\star, C_z^\star, C_\sigma^\star, C_{\mathsf{ID}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, c^\star, s_{\mathsf{ID}}^\star, s_\theta^\star),$$

for some message $M^\star$, that opens to $i^\star \in U^b$ although user $i^\star$ did not sign $M^\star$. With high probability, $\mathcal{A}$ must have queried the hash value $H(M^\star, C_{\mathsf{CS}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, R_1^\star, R_2^\star, R_3^\star, R_4^\star)$, which would be unpredictable otherwise. Hence, $\mathcal{B}$ can run $\mathcal{A}$ a second time with the same input and random tape.

At the moment when $\mathcal{A}$ queries $H(M^\star, C_{\mathsf{CS}}^\star, \tilde{\sigma}_2^\star, \tilde{\sigma}_3^\star, R_1^\star, R_2^\star, R_3^\star, R_4^\star)$ in the second run, $\mathcal{B}$ starts responding with different random oracle values which depart from those of the initial run. The Forking Lemma of [BN06] ensures that, with non-negligible probability the second run will result in a forgery

$$\Sigma^\dagger = (C_1^\star, C_2^\star, C_z^\star, C_\sigma^\star, C_{\mathsf{ID}}^\star, \tilde{\sigma}_2^\dagger, \tilde{\sigma}_3^\dagger, c^\dagger, s_{\mathsf{ID}}^\dagger, s_\theta^\dagger)$$

on the same message $M^\star$, with distinct challenges $c^\dagger \neq c^\star$. From the two responses $(s_{\mathsf{ID}}^\star, s_\theta^\star)$, $(s_{\mathsf{ID}}^\dagger, s_\theta^\dagger)$, $\mathcal{B}$ can extract witnesses $(\theta^\star, \mathsf{ID}^\star)$ satisfying $C_{\mathsf{ID}}^\star = v^{\mathsf{ID}^\star} X_{\mathsf{ID}}^{\theta^\star}$ and which identifies $V_i^\star = v^{\mathsf{ID}^\star}$. At this stage, $\mathcal{B}$ can compute and output the sought-after SDL solution $a := \mathsf{ID}^\star/\delta_i$ in $\mathbb{Z}_p$.
This observation tells us that, if $\mathcal{A}$ has advantage ε as a framing adversary making q_H random oracle queries, then $\mathcal{B}$ implies an algorithm solving the SDL problem with probability $\varepsilon(\varepsilon/q_H - 1/p)$.

□

We stress that the proofs can be easily adapted to the case where the opening algorithm has linear complexity in the number of users.

6.4.2 Comparison with Existing Schemes

Name	Signature length			Assumptions	Group Type	Anonymity
	$\mathbb{G}$	$\mathbb{Z}_p$	bits			
Ours	7	3	2560 bits	SXDH + SDL	Dynamic	CCA
Boneh-Boyen-Shacham	3	6	2304 bits	q-SDH + DLIN	Static	CPA
Delerablée-Pointcheval	4	5	2304 bits	q-SDH + XDH	Dynamic	CCA
Bichsel *et al.*	3	2	1280 bits	LRSW + SDL	Dynamic	CCA-
Pointcheval-Sanders	2	2	1024 bits	LRSW-like	Dynamic	CCA-

Table 6.1 – Comparison between different group signature schemes

Table 6.1 compares our scheme with previous practical group signatures based on pairing-related assumptions. Since we focus on practical schemes, we only consider those in the random oracle model. To make the comparison possible, we use 256-bit group orders, so that elements of $\mathbb{G}$ and $\mathbb{Z}_p$ are encoded using 256 bits each.

The scheme of Boneh, Boyen and Shacham [BBS04] is the first scheme providing short signatures: each signature is comprised of 3 group elements and 6 elements of $\mathbb{Z}_p$. However,

Algorithm/Protocol	Our scheme	Standard deviation
Keygen	9.70 ms	2.18 ms
Join	23.16 ms	0.11 ms
Sign	15.70 ms	0.04 ms
Verify	26.91 ms	0.04 ms

Table 6.2 – Experimental results for the Pairing-Base group signature scheme

this scheme is designed for static groups only and relies on the Strong Diffie-Hellmann assumption, which is a non-standard q-type assumption, and its anonymity is only proved in the CPA sense.

Delerablée and Pointcheval [DP06] presented a scheme designed for a dynamically growing group and which is also fully (i.e., CCA) anonymous. The security of their scheme is based on the eXternal Diffie-Hellman assumption (XDH), which we also use here, and the q-SDH assumption. In [DP06], each signature consists of 4 group elements and 5 scalars in $\mathbb{Z}_p$, which leads to the same signature size as previously. They also proposed a variant to get rid of the XDH assumption at the cost of 2 more group elements and one more scalar, but they still rely on the q-SDH assumption.

Bichsel *et al.* [BCN+10] proposed a very short group signature for dynamic groups, where each signature consists of 3 group elements and 2 elements in $\mathbb{Z}_p$. The downsides are their use the LRSW assumption [LRSW99], which is a very *ad-hoc* interactive assumption, and their security notion is not fully-anonymous, but is an hybrid security with selfless-anonymity, which is marked "CCA-" in Table 6.1. Another caveat is that, unlike the two previous systems, the opening complexity of their scheme is linear in the number of group members.

In 2015, Pointcheval and Sanders [PS16] gave another instantiation of [BCN+10] based on a variant of the LRSW assumption in the asymmetric setting (meaning using only Type III pairings), which provides even shorter signatures than [BCN+10] with the same downsides. Their scheme provides signatures composed of only 2 group elements in $\mathbb{G}$ and 2 scalars in $\mathbb{Z}_p$.

Our main contribution compared to these schemes is to provide size-comparable signatures – we recall that our scheme is composed of 7 group elements and 3 scalars in $\mathbb{Z}_p$ – while relying on standard, constant-size assumptions. Moreover, we can notice that we can save one element in $\mathbb{G}$ at the expense of a linear-time opening algorithm in the number N_{gs} of group users (like [BCN+10]).

6.4.3 Experimental Results

An implementation of the aforementioned group signature scheme has been made in `C` using the *Relic toolkit* for pairing-based cryptography [AG] and is available at the following URL: `https://gforge.inria.fr/projects/sigmasig-c/`.

The relic toolkit provides an implementation for pairing computations, hash functions (SHA-256 in this case) and benchmarking macros. The benchmarking was made on a single-core of an *Intel® Core™ i5-7500 CPU @ 3.40GHz* (Kaby Lake architecture) with 6MB of cache. To implement pairings, the relic library implements the Barreto-Naehrig [BN06] curve over

a 256 bits curve. As explained previously, since recent advances in pairing-friendly elliptic curve cryptanalysis, there is no more curve that shows the best timing results in every aspect. Figures are available in Table 6.2.

Chapter 7

Lattice-Based Dynamic Group Signatures

In this chapter, we present the first dynamic group signature scheme based on lattice assumptions. This construction relies on a signature scheme with efficient protocols as in Chapter 6, which is used in a similar manner. As a consequence, it is possible to design lattice-based anonymous credentials from this building block. The group signature scheme relies on the Gentry, Peikert and Vaikuntanathan identity-based encryption [GPV08] with the Canetti, Halevi and Katz [CHK04] transform to obtain a CCA2-secure public key encryption scheme which will be used to provide full-anonymity.

The group signature is proven secure in the ROM under the SIS and LWE assumptions, which are fixed-size and well-studied assumptions. As of the security parameter λ and groups of up to N_{gs} members, the scheme features public key size $\tilde{\mathcal{O}}(\lambda^2) \cdot \log N_{\mathsf{gs}}$, user's secret key size $\tilde{\mathcal{O}}(\lambda)$ and signature size $\tilde{\mathcal{O}}(\lambda) \cdot \log N_{\mathsf{gs}}$. Our scheme thus achieves a level of efficiency comparable to recent proposals based on standard (i.e. non-ideal) lattices [LLLS13, NZZ15, LNW15, LLNW16] in the static setting as depicted in Table 7.1. In particular, the cost of moving to dynamic group is reasonable: while using the scheme from [LNW15] as a building block, our construction lengthens the signature size only by a (small) constant factor.

The signature scheme with efficient protocols is built upon the SIS-based signature of Böhl *et al.* [BHJ+15], which is itself a variant of Boyen's signature [Boy10]. The latter scheme involves a public key containing matrices $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n \times m}$ and signs an ℓ-bit message $\mathfrak{m} \in \{0,1\}^\ell$ by computing a short vector $\mathbf{v} \in \mathbb{Z}^{2m}$ such that $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \mathfrak{m}[j]\mathbf{A}_j] \cdot \mathbf{v} = \mathbf{0}^n \bmod q$. The variant proposed by Böhl *et al.* only uses a constant number of matrices $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1 \in \mathbb{Z}_q^{n \times m}$ where each signature is assigned with a single-use tag τ and the public key involves an extra matrix $\mathbf{D} \in \mathbb{Z}_q^{n \times m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$. A message $\mathfrak{m}$ is then signed by first applying a chameleon hash function $\mathbf{h} = \mathcal{H}(\mathfrak{m}, \mathbf{s}) \in \{0,1\}^m$ and signing $\mathbf{h}$ by computing a short $\mathbf{v} \in \mathbb{Z}^{2m}$ such that $[\mathbf{A} \mid \mathbf{A}_0 + \tau\mathbf{A}_1] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathbf{h} \bmod q$.

Our scheme extends [BHJ+15] so that an N-block message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N) \in (\{0,1\}^L)^N$, for some $L \in \mathbb{N}$, is signed by outputting a tag $\tau \in \{0,1\}^\ell$ and a short $\mathbf{v} \in \mathbb{Z}^{2m}$ such that $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j]\mathbf{A}_j] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathcal{H}(\mathfrak{m}_1, \ldots, \mathfrak{m}_N, \mathbf{s}) \bmod q$, where the chameleon hash function computes $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \bmod q$, for some short vector $\mathbf{s}$, before re-encoding $\mathbf{c}_M$ so as to enable multiplication by $\mathbf{D}$.

Scheme	[LLLS13]	[NZZ15]	[LNW15]	[LLNW16]	Ours
Group PK	$\widetilde{\mathcal{O}}(\lambda^2) \cdot \log N_{\mathsf{gs}}$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2) \cdot \log N_{\mathsf{gs}}$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2) \cdot \log N_{\mathsf{gs}}$
User's SK	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda^2)$	$\widetilde{\mathcal{O}}(\lambda)$	$\widetilde{\mathcal{O}}(\lambda) \cdot \log N_{\mathsf{gs}}$	$\widetilde{\mathcal{O}}(\lambda)$
Signature	$\widetilde{\mathcal{O}}(\lambda) \cdot \log N_{\mathsf{gs}}$	$\widetilde{\mathcal{O}}(\lambda + \log^2 N_{\mathsf{gs}})$	$\widetilde{\mathcal{O}}(\lambda) \cdot \log N_{\mathsf{gs}}$	$\widetilde{\mathcal{O}}(\lambda) \cdot \log N_{\mathsf{gs}}$	$\widetilde{\mathcal{O}}(\lambda) \cdot \log N_{\mathsf{gs}}$

Table 7.1 – Efficiency comparison among recent lattice-based group signatures for static groups and our dynamic scheme. The evaluation is done with respect to 2 governing parameters: security parameter λ and the maximum expected group size N_{gs}. We do not include the earlier schemes [GKV10, CNR12] that have signature size $\widetilde{\mathcal{O}}(\lambda^2) \cdot N_{\mathsf{gs}}$.

In order to obtain a signature scheme that possesses efficient protocols akin to Camenish and Lysyanskaya [CL02b], our idea is to have the tag $\tau \in \{0,1\}^\ell$ play the same role as the prime exponent in Strong-RSA-based schemes [CL02a]. To adapt this idea in the context of signatures with efficient protocols, we have to overcome several difficulties. The first one is to map $\mathbf{c}_M$ back in the domain of the chameleon hash function while preserving the compatibility with ZK proofs. To solve this issue, we extend a technique used in [LLNW16] in order to build a "zero-knowledge-friendly" chameleon hash function. This function hashes the message by outputting the coordinate-wise binary decomposition $\mathbf{w}$ of $\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k$. Using the "power-of-two" matrix $\mathbf{H} = \mathbf{I} \otimes [1 \mid 2 \mid \cdots \mid 2^{\lceil \log q \rceil}]$, we can prove that $\mathbf{w} = \mathcal{H}(\mathfrak{m}_1, \ldots, \mathfrak{m}_N, \mathbf{s})$ by demonstrating the knowledge of short vectors $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N, \mathbf{s}, \mathbf{w})$ that verifies $\mathbf{H} \cdot \mathbf{w} = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \bmod q$ which can be proven using the ZKAoK of Section 4.3.

The second problem is to prove knowledge of $(\tau, \mathbf{v}, \mathbf{s})$ and $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ satisfying $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j] \cdot \mathbf{A}_j] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathsf{CMHash}(\mathfrak{m}_1, \ldots, \mathfrak{m}_N, \mathbf{s})$, without revealing any of the witnesses. To this end, we provide a framework for proving all the involved statement (and many other relations that naturally arise in lattice-based cryptography) as special cases. We reduce the statements to asserting that a short integer vector $\mathbf{x}$ satisfies an equation of the form $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$, for some public matrix $\mathbf{P}$ and vector $\mathbf{v}$, and belongs to a set VALID of short vectors with a particular structure. While the small-norm property of $\mathbf{x}$ is provable using standard techniques (e.g., [Lyu08]), we argue its membership of VALID by leveraging the properties of Stern-like protocols [Ste96, KTX08, LNSW13]. In particular, we rely on the fact that their underlying permutations interact well with combinatorial statements pertaining to $\mathbf{x}$, especially $\mathbf{x}$ being a bitstring with a specific pattern. We believe our framework to be of independent interest as it provides a blueprint for proving many other intricate relations in a modular manner.

When we extend the scheme with a protocol for signing committed messages, we need the signer to re-randomize the user's commitment before signing the hidden messages. This is indeed necessary to provide the reduction with a backdoor allowing to correctly answer the $i^\dagger$-th query by "programming" the randomness of the commitment. Since we work with integers vectors, a straightforward simulation incurs a non-negligible statistical distance between the simulated distributions of re-randomization coins and the real one (which both have a discrete Gaussian distribution). Camenisch and Lysyanskaya [CL02b] address a similar problem by choosing the signer's randomness to be exponentially larger than that of the user's commitment so as to statistically "drown" the aforementioned discrepancy. Here, the same idea would require to work with an exponentially large modulus q. Instead, we adopt a more efficient solution, inspired by Bai *et al.* [BLL+15], which is to apply

an analysis based on the Rényi divergence rather than the statistical distance. In short, the Rényi divergence's properties tell us that, if some event E occurs with noticeable probability in some probability space P, so does it in a different probability space Q for which the second order divergence $R_2(P||Q)$ is sufficiently small. In our setting, $R_2(P||Q)$ is precisely polynomially bounded since the two probability spaces only diverge in one signing query.

Our dynamic group signature scheme avoids these difficulties because the group manager only signs known messages: instead of signing the user's secret key as in anonymous credentials, it creates a membership certificate by signing the user's public key. Our zero-knowledge arguments accommodate the requirements of the scheme in the following way. In the joining protocol that dynamically introduces new group members, the user i chooses a membership secret consisting of a short discrete Gaussian vector $\mathbf{z}_i$. This user generates a public syndrome $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i \mod q$, for some public matrix $\mathbf{F}$, which constitutes his public key. In order to certify $\mathbf{v}_i$, the group manager computes the coordinate-wise binary expansion $\mathsf{bin}(\mathbf{v}_i)$ of $\mathbf{v}_i$. The vector $\mathsf{bin}(\mathbf{v}_i)$ is then signed using our signature scheme. Using the resulting signature $(\tau, \mathbf{v}, \mathbf{s})$ as a membership certificate, the group member is able to sign a message by proving that: (i) He holds a valid signature $(\tau, \mathbf{v}, \mathbf{s})$ on some secret binary message $\mathsf{bin}(\mathbf{v}_i)$; (ii) The latter vector $\mathsf{bin}(\mathbf{v}_i)$ is the binary expansion of some syndrome $\mathbf{v}_i$ of which he knows a GPV pre-image $\mathbf{z}_i$. We remark that condition (ii) can be proved by providing evidence that we have $\mathbf{v}_i = \mathbf{H} \cdot \mathsf{bin}(\mathbf{v}_i) = \mathbf{F} \cdot \mathbf{z}_i \bmod q$, for some short integer vector $\mathbf{z}_i$ and some binary $\mathsf{bin}(\mathbf{v}_i)$, where $\mathbf{H}$ is the "powers-of-2" matrix. Our abstraction of Stern-like protocols [Ste96, KTX08, LNSW13] allows us to efficiently argue such statements. The fact that the underlying chameleon hash function smoothly interacts with Stern-like zero-knowledge arguments is the property that maintains the user's capability of efficiently proving knowledge of the underlying secret key.

Given the state of NIZK proofs in the lattice setting, it seems hard to provide group signature schemes in the standard model.

In the forthcoming sections, we first provide the description of our signature with efficient protocols; then a description of our dynamic group signature will be given and finally, we will explain how to use the Stern abstraction of Section 4.3 to provide the required zero-knowledge arguments.

7.1 A Lattice-Based Signature with Efficient Protocols

Our scheme can be seen as a variant of the Böhl *et al.* signature [BHJ+15], where each signature is a triple $(\tau, \mathbf{v}, \mathbf{s})$, made of a tag $\tau \in \{0,1\}^\ell$ and integer vectors $(\mathbf{v}, \mathbf{s})$ satisfying $[\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j] \cdot \mathbf{A}_j] \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathbf{h} \bmod q$, where matrices $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell, \mathbf{D} \in \mathbb{Z}_q^{n \times m}$ are public random matrices and $\mathbf{h} \in \{0,1\}^m$ is a chameleon hash of the message which is computed using randomness $\mathbf{s}$. A difference is that, while [BHJ+15] uses a short single-use tag $\tau \in \mathbb{Z}_q$, we need the tag to be an ℓ-bit string $\tau \in \{0,1\}^\ell$ which will assume the same role as the prime exponent of Camenisch-Lysyanskaya signatures [CL02a] in the security proof.

We show that a suitable chameleon hash function makes the scheme compatible with Stern-like zero-knowledge arguments [LNSW13, LNW15] for arguing possession of a valid message-signature pair. Section 4.3 shows how to translate such a statement into asserting

that a short witness vector $\mathbf{x}$ with a particular structure satisfies a relation of the form $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$, for some public matrix $\mathbf{P}$ and vector $\mathbf{v}$. The underlying chameleon hash can be seen as a composition of the chameleon hash of [CHKP10, Se. 4.1] with a technique used in [PSTY13, LLNW16]: on input of a message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, it outputs the binary decomposition of $\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k$, for some discrete Gaussian vector $\mathbf{s}$.

7.1.1 Description

We assume that messages are vectors of N blocks $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, where each block is a $2m$-bit string $\mathfrak{m}_k = \mathfrak{m}_k[1] \ldots \mathfrak{m}_k[2m] \in \{0,1\}^{2m}$ for $k \in \{1, \ldots, N\}$.

For each vector $\mathbf{v} \in \mathbb{Z}_q^L$, we denote by $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{L\lceil \log q \rceil}$ the vector obtained by replacing each coordinate of $\mathbf{v}$ by its binary representation.

Keygen$(1^\lambda, 1^N)$**:** Given a security parameter $\lambda > 0$ and the number of blocks $N = \mathsf{poly}(\lambda)$, choose the following parameters: $n = \mathcal{O}(\lambda)$; a prime modulus $q = \widetilde{\mathcal{O}}(N \cdot n^4)$; dimension $m = 2n\lceil \log q \rceil$; an integer $\ell = \Theta(\lambda)$; and Gaussian parameters $\sigma = \Omega(\sqrt{n \log q} \log n)$, $\sigma_0 = 2\sqrt{2}(N+1)\sigma m^{3/2}$, and $\sigma_1 = \sqrt{\sigma_0^2 + \sigma^2}$. Define the message space as $(\{0,1\}^{2m})^N$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T}_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$. This basis allows computing short vectors in $\Lambda_q^\perp(\mathbf{A})$ with a Gaussian parameter σ. Next, choose $\ell + 1$ random $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n \times m})$.
2. Choose random matrices $\mathbf{D} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n \times m})$, $\mathbf{D}_0, \mathbf{D}_1, \ldots, \mathbf{D}_N \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{2n \times 2m})$ as well as a random vector $\mathbf{u} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^n)$.

The private key consists of $SK \coloneqq \mathbf{T}_\mathbf{A} \in \mathbb{Z}^{m \times m}$ and the public key is

$$PK \coloneqq (\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \{\mathbf{D}_k\}_{k=0}^{N},\ \mathbf{D},\ \mathbf{u}).$$

Sign(SK, Msg)**:** To sign an N-block message $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_N) \in \left(\{0,1\}^{2m}\right)^N$,

1. Choose a random string $\tau \hookleftarrow \mathcal{U}(\{0,1\}^\ell)$. Then, using $SK \coloneqq \mathbf{T}_\mathbf{A}$, compute with $\mathsf{ExtBasis}$ a short delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$ for the matrix

$$\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau[j]\mathbf{A}_j] \in \mathbb{Z}_q^{n \times 2m}. \tag{7.1}$$

2. Sample a vector $\mathbf{s} \hookleftarrow D_{\mathbb{Z}^{2m}, \sigma_1}$. Compute $\mathbf{c}_M \in \mathbb{Z}_q^{2n}$ as a chameleon hash of $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$: i.e., compute $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \in \mathbb{Z}_q^{2n}$, which is used to define $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{c}_M) \in \mathbb{Z}_q^n$. Then, using the delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$, sample a short vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D_{\Lambda_q^{\mathbf{u}_M}(\mathbf{A}_\tau), \sigma}$.

Output the signature $sig = (\tau, \mathbf{v}, \mathbf{s}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$.

Verify(PK, Msg, sig)**:** Given PK, a message $\mathsf{Msg} = (\mathfrak{m}_1, \ldots, \mathfrak{m}_N) \in (\{0,1\}^{2m})^N$ and a purported signature $sig = (\tau, \mathbf{v}, \mathbf{s}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$, return 1 if

$$\mathbf{A}_\tau \cdot \mathbf{v} \quad = \quad \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k) \bmod q. \tag{7.2}$$

and $\|\mathbf{v}\| < \sigma\sqrt{2m}$, $\|\mathbf{s}\| < \sigma_1\sqrt{2m}$.

When the scheme is used for obliviously signing committed messages, the security proof follows Bai *et al.* [BLL+15] in that it applies an argument based on the Rényi divergence in one signing query. This argument requires to sample $\mathbf{s}$ from a Gaussian distribution whose standard deviation σ_1 is polynomially larger than σ.

We note that, instead of being included in the public key, the matrices $\{\mathbf{D}_k\}_{k=0}^N$ can be part of common public parameters shared by many signers. Indeed, only the matrices $(\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^\ell)$ should be specific to the user who holds the secret key $SK = \mathbf{T}_\mathbf{A}$. In Section 7.1.3, we use a variant where $\{\mathbf{D}_k\}_{k=0}^N$ belong to public parameters.

7.1.2 Security Analysis

The security analysis in Theorem 7.1 requires that $q > \ell$.

Theorem 7.1. *The signature scheme is secure under chosen-message attacks under the* SIS *assumption.*

Proof. To prove the result, we will distinguish three kinds of attacks:

Type I attacks are attacks where, in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$, $\tau^\star$ did not appear in any output of the signing oracle.

Type II attacks are such that, in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$, $\tau^\star$ is recycled from an output $sig^{(i^\star)} = (\tau^{(i^\star)}, \mathbf{v}^{(i^\star)}, \mathbf{s}^{(i^\star)})$ of the signing oracle, for some index $i^\star \in \{1, \ldots, Q\}$. However, if $\mathsf{Msg}^\star = (\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star)$ and $\mathsf{Msg}^{(i^\star)} = (\mathfrak{m}_1^{(i^\star)}, \ldots, \mathfrak{m}_N^{(i^\star)})$ denote the forgery message and the $i^\star$-th signing query, respectively, we have $\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^\star \neq \mathbf{D}_0 \cdot \mathbf{s}^{(i^\star)} + \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\star)}$.

Type III attacks are those where the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$ recycles $\tau^\star$ from an output $sig^{(i^\star)} = (\tau^{(i^\star)}, \mathbf{v}^{(i^\star)}, \mathbf{s}^{(i^\star)})$ of the signing oracle (i.e., $\tau^{(i^\star)} = \tau^\star$ for some index $i^\star \in \{1, \ldots, Q\}$) and we have the collision

$$\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star = \mathbf{D}_0 \cdot \mathbf{s}^{(i^\star)} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\star)}. \tag{7.3}$$

Type III attacks imply a collision for the chameleon hash function of Kawachi *et al.* [KTX08]: if (7.3) holds, a short vector of $\Lambda_q^\perp([\mathbf{D}_0 \mid \mathbf{D}_1 \mid \ldots \mid \mathbf{D}_N])$ is obtained as

$$\left(\mathbf{s}^{\star T} - \mathbf{s}^{(i^\star)^T} \mid \mathfrak{m}_1^{\star T} - \mathfrak{m}_1^{(i^\star)^T} \mid \ldots \mid \mathfrak{m}_N^{\star T} - \mathfrak{m}_N^{(i^\star)^T}\right)^T,$$

so that a collision breaks the SIS assumption.

The security against Type I attacks is proved by Lemma 7.2 which applies the same technique as in [Boy10, MP12]. In particular, the prefix guessing technique of [HW09] allows keeping the modulus smaller than the number Q of adversarial queries as in [MP12]. In order to deal with Type II attacks, we can leverage the technique of [BHJ+15]. In Lemma 7.3, we prove that Type II attack would also contradict SIS. □

Lemma 7.2. *The scheme is secure against Type I attacks if the* $\mathsf{SIS}_{n,m,q,\beta'}$ *assumption holds for* $\beta' = m^{3/2}\sigma^2(\ell+3) + m^{1/2}\sigma_1$

Proof. Let $\mathcal{A}$ be a PPT adversary that can mount a Type I attack with non-negligible success probability ε. We construct a PPT algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to break the $\mathsf{SIS}_{n,m,q,\beta'}$ assumption. It takes as input $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times m}$ and computes $\mathbf{v} \in \Lambda_q^{\perp}(\bar{\mathbf{A}})$ with $0 < \|\mathbf{v}\| \leq \beta'$.

Algorithm $\mathcal{B}$ first chooses the ℓ-bit strings $\tau^{(1)}, \ldots, \tau^{(Q)} \hookleftarrow \mathcal{U}(\{0,1\}^\ell)$ to be used in signing queries. As in [HW09], it guesses the shortest prefix such that the string $\tau^\star$ contained in $\mathcal{A}$'s forgery differs from all prefixes of $\tau^{(1)}, \ldots, \tau^{(Q)}$. To this end, $\mathcal{B}$ chooses $i^\dagger \hookleftarrow \mathcal{U}(\{1,\ldots,Q\})$ and $t^\dagger \hookleftarrow \mathcal{U}(\{1,\ldots,\ell\})$ so that, with probability $1/(Q\cdot\ell)$, the longest common prefix between $\tau^\star$ and one of the $\left\{\tau^{(i)}\right\}_{i=1}^{Q}$ is the string $\tau^\star[1]\ldots\tau^\star[t^\dagger-1] = \tau^{(i^\dagger)}[1]\ldots\tau^{(i^\dagger)}[t^\dagger-1] \in \{0,1\}^{t^\dagger-1}$ comprised of the first $(t^\dagger-1)$-th bits of $\tau^\star \in \{0,1\}^\ell$. We define $\tau^\dagger \in \{0,1\}^{t^\dagger}$ as the $t^\dagger$-bit string $\tau^\dagger = \tau^\star[1]\ldots\tau^\star[t^\dagger]$. By construction, with probability $1/(Q\cdot\ell)$, we have $\tau^\dagger \notin \left\{\tau^{(1)}_{|t^\dagger}, \ldots, \tau^{(Q)}_{|t^\dagger}\right\}$, where $\tau^{(i)}_{|t^\dagger}$ denotes the $t^\dagger$-th prefix of $\tau^{(i)}$ for each $i \in \{1,\ldots,Q\}$.

Then, $\mathcal{B}$ runs $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ 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 picks $\ell+1$ matrices $\mathbf{Q}_0, \ldots, \mathbf{Q}_\ell \in \mathbb{Z}^{m\times m}$, where each matrix $\mathbf{Q}_i$ has its columns sampled independently from $D_{\mathbb{Z}^m,\sigma}$. The reduction $\mathcal{B}$ defines the matrices $\{\mathbf{A}_j\}_{j=0}^{\ell}$ as

$$\begin{cases} \mathbf{A}_0 = \bar{\mathbf{A}}\cdot\mathbf{Q}_0 + (\sum_{j=1}^{t^\dagger}\tau^\star[j])\cdot\mathbf{C} & \\ \mathbf{A}_j = \bar{\mathbf{A}}\cdot\mathbf{Q}_j + (-1)^{\tau^\star[j]}\cdot\mathbf{C}, & \text{for } j\in[1,t^\dagger] \\ \mathbf{A}_j = \bar{\mathbf{A}}\cdot\mathbf{Q}_j, & \text{for } j\in[t^\dagger+1,\ell] \end{cases}$$

It also sets $\mathbf{A} = \bar{\mathbf{A}}$. We note that we have

$$\begin{aligned} \mathbf{A}_{\tau^{(i)}} &= \left[\ \bar{\mathbf{A}} \ \middle|\ \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell}\tau^{(i)}[j]\mathbf{A}_j\ \right] \\ &= \left[\ \bar{\mathbf{A}} \ \middle|\ \bar{\mathbf{A}}\cdot(\mathbf{Q}_0 + \textstyle\sum_{j=1}^{\ell}\tau^{(i)}[j]\mathbf{Q}_j) + (\sum_{j=1}^{t^\dagger}\tau^\star[j] + (-1)^{\tau^\star[j]}\tau^{(i)}[j])\cdot\mathbf{C}\ \right] \\ &= \left[\ \bar{\mathbf{A}} \ \middle|\ \bar{\mathbf{A}}\cdot(\mathbf{Q}_0 + \textstyle\sum_{j=1}^{\ell}\tau^{(i)}[j]\mathbf{Q}_j) + h_{\tau^{(i)}}\cdot\mathbf{C}\ \right] \end{aligned}$$

where $h_{\tau^{(i)}} \in [1,t^\dagger] \subset [1,\ell]$ stands for the Hamming distance between $\tau^{(i)}_{|t^\dagger}$ and $\tau^\star_{|t^\dagger}$. Note that, with probability $1/(Q\cdot\ell)$ and since $q > \ell$, we have $h_{\tau^{(i)}} \neq 0 \bmod q$ whenever $\tau^{(i)}_{|t^\dagger} \neq \tau^\star_{|t^\dagger}$.

Next, $\mathcal{B}$ chooses the matrices $\mathbf{D}_k \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{2n\times 2m})$ uniformly at random for each $k \in [0,N]$. Then, it picks a random short matrix $\mathbf{R} \in \mathbb{Z}^{m\times m}$ which has its columns independently sampled from $D_{\mathbb{Z}^m,\sigma}$ and computes

$$\mathbf{D} \ = \ \bar{\mathbf{A}}\cdot\mathbf{R}.$$

Finally, $\mathcal{B}$ samples a short vector $\mathbf{e}_u \hookleftarrow D_{\mathbb{Z}^m,\sigma_1}$ and computes the vector $\mathbf{u} \in \mathbb{Z}_q^n$ as $\mathbf{u} = \bar{\mathbf{A}} \cdot \mathbf{e}_u \in \mathbb{Z}_q^n$. The public key

$$PK \coloneqq (\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \{\mathbf{D}_k\}_{k=0}^{N},\ \mathbf{D},\ \mathbf{u})$$

is given to $\mathcal{A}$.

At the i-th signing query $\mathsf{Msg}^{(i)} = (\mathfrak{m}_1^{(i)}, \ldots, \mathfrak{m}_N^{(i)}) \in (\{0,1\}^{2m})^N$, $\mathcal{B}$ can use the trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m\times m}$ to generate a signature. To do this, $\mathcal{B}$ first samples $\mathbf{s}^{(i)} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_1}$ and computes a vector $\mathbf{u}_M \in \mathbb{Z}_q^m$ as

$$\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \{0,1\}\big(\sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i)} + \mathbf{D}_0 \cdot \mathbf{s}^{(i)}\big) \mod q.$$

Using $\mathbf{T_C} \in \mathbb{Z}^{m\times m}$, $\mathcal{B}$ can then sample a short vector $\mathbf{v}^{(i)} \in \mathbb{Z}^{2m}$ in $D^{\mathbf{u}_M}_{\Lambda^{\perp}(\mathbf{A}_{\tau^{(i)}}),\sigma}$ such that $\big(\tau^{(i)}, \mathbf{v}^{(i)}, \mathbf{s}^{(i)}\big)$ satisfies the verification equation (7.2).

When $\mathcal{A}$ halts, it outputs a valid signature $sig^{\star} = \big(\tau^{(i^{\dagger})}, \mathbf{v}^{\star}, \mathbf{s}^{\star}\big)$ on a message $\mathsf{Msg}^{\star} = (\mathfrak{m}_1^{\star}, \ldots, \mathfrak{m}_N^{\star})$ with $\|\mathbf{v}^{\star}\| \leq \sigma\sqrt{2m}$ and $\|\mathbf{s}^{\star}\| \leq \sigma_1\sqrt{2m}$. At this point, $\mathcal{B}$ aborts and declares failure if it was unfortunate in its choice of $i^{\dagger} \in \{1,\ldots,Q\}$ and $t^{\dagger} \in \{1,\ldots,\ell\}$. Otherwise, with probability $1/(Q\cdot\ell)$, $\mathcal{B}$ correctly guessed $i^{\dagger} \in \{1,\ldots,Q\}$ and $t^{\dagger} \in \{1,\ldots,\ell\}$, in which case it can solve the given SIS instance as follows.

If we parse $\mathbf{v}^{\star} \in \mathbb{Z}^{2m}$ as $(\mathbf{v}_1^{\star T} \mid \mathbf{v}_2^{\star T})^T$ with $\mathbf{v}_1^{\star}, \mathbf{v}_2^{\star} \in \mathbb{Z}^m$, we have the equality

$$\begin{aligned}
\Big[\ \bar{\mathbf{A}}\ \Big|\ \bar{\mathbf{A}} \cdot (\mathbf{Q}_0 + \textstyle\sum_{j=1}^{\ell} \tau^{\star}[j]\mathbf{Q}_j)\ \Big] \cdot \left[\frac{\mathbf{v}_1^{\star}}{\mathbf{v}_2^{\star}}\right] & \\
&= \mathbf{u} + \mathbf{D} \cdot \{0,1\}\big(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star}\big) \bmod q \\
&= \bar{\mathbf{A}} \cdot \Big(\mathbf{e}_u + \mathbf{R} \cdot \{0,1\}\big(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star}\big)\Big) \bmod q,
\end{aligned}$$

which implies that the vector

$$\mathbf{w} \quad = \quad \mathbf{v}_1^{\star} + (\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^{\star}[j]\mathbf{Q}_j) \cdot \mathbf{v}_2^{\star} - \mathbf{e}_u - \mathbf{R} \cdot \{0,1\}\big(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star}\big) \in \mathbb{Z}^m$$

is in $\Lambda_q^{\perp}(\bar{\mathbf{A}})$. Moreover, with overwhelming probability, this vector is non-zero since, in $\mathcal{A}$'s view, the distribution of $\mathbf{e}_u \in \mathbb{Z}^m$ is $D_{\Lambda_q^{\mathbf{u}}(\bar{\mathbf{A}}),\sigma_1}$, which ensures that $\mathbf{e}_u$ is statistically hidden by the syndrome $\mathbf{u} = \bar{\mathbf{A}} \cdot \mathbf{e}_u$. Finally, the norm of $\mathbf{w}$ is smaller than $\beta' = m^{3/2}\sigma^2(\ell+3) + m^{1/2}\sigma_1$ which yields a valid solution of the given $\mathsf{SIS}_{n,m,q,\beta'}$ instance with overwhelming probability. □

Lemma 7.3. *The scheme is secure against Type II attacks if the* $\mathsf{SIS}_{n,m,q,\beta''}$ *assumption holds for* $\beta'' = \sqrt{2}(\ell+2)\sigma^2 m^{3/2} + m^{1/2}$.

Proof. We prove the result using a sequence of games. For each i, we denote by W_i the event that the adversary wins by outputting a Type II forgery in Game i.

Game 0: This is the real game where, at the i-th signing query $\mathsf{Msg}^{(i)} = (\mathfrak{m}_1^{(i)}, \ldots, \mathfrak{m}_N^{(i)})$, the adversary obtains a signature $sig^{(i)} = (\tau^{(i)}, \mathbf{v}^{(i)}, \mathbf{s}^{(i)})$ for each $i \in \{1, \ldots, Q\}$ from the signing oracle. At the end of the game, the adversary outputs a forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$ on a message $\mathsf{Msg}^\star = (\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star)$. By hypothesis, the adversary's advantage is $\varepsilon = \Pr[W_0]$. We assume without loss of generality that the random ℓ-bit strings $\tau^{(1)}, \ldots, \tau^{(Q)}$ are chosen at the very beginning of the game. Since $(\mathsf{Msg}^\star, sig^\star)$ is a Type II forgery, there exists an index $i^\star \in \{1, \ldots, Q\}$ such that $\tau^\star = \tau^{(i^\star)}$.

Game 1: This game is identical to Game 0 with the difference that the reduction aborts the experiment in the unlikely event that, in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$, $\tau^\star$ coincides with more than one of the random ℓ-bit strings $\tau^{(1)}, \ldots, \tau^{(Q)}$ used by the challenger. If we call F_1 the latter event, we have $\Pr[F_1] < Q^2/2^\ell$ since we are guaranteed to have $\neg F_1$ as long as no two $\tau^{(i)}$, $\tau^{(i')}$ collide. Given that Game 1 is identical to Game 0 until F_1 occurs, we have $|\Pr[W_1] - \Pr[W_0]| \leq \Pr[F_1] < Q^2/2^\ell$.

Game 2: This game is like Game 1 with the following difference. At the outset of the game, the challenger $\mathcal{B}$ chooses a random index $i^\dagger \hookleftarrow (\{1, \ldots, Q\})$ as a guess that $\mathcal{A}$'s forgery will recycle the ℓ-bit string $\tau^{(i^\dagger)} \in \{0,1\}^\ell$ of the $i^\dagger$-th signing query. When $\mathcal{A}$ outputs its Type II forgery $sig^\star = (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star)$, the challenger aborts in the event that $\tau^{(i^\dagger)} \neq \tau^\star$ (i.e., $i^\dagger \neq i^\star$). Since the choice of $i^\dagger$ in $\{1, \ldots, Q\}$ is independent of $\mathcal{A}$'s view, we have $\Pr[W_2] = \Pr[W_1]/Q$.

Game 3: In this game, we modify the key generation phase and the way to answer signing queries. First, the challenger $\mathcal{B}$ randomly picks $h_0, h_1, \ldots, h_\ell \in \mathbb{Z}_q$ subject to the constraints

$$\begin{aligned} h_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot h_j &= 0 \bmod q \\ h_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot h_j &\neq 0 \bmod q \qquad\qquad i \in \{1, \ldots, Q\} \setminus \{i^\dagger\} \end{aligned}$$

It runs $(\mathbf{C}, \mathbf{T}_\mathbf{C}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, $(\mathbf{D}_0, \mathbf{T}_{\mathbf{D}_0}) \leftarrow \mathsf{TrapGen}(1^{2n}, 1^{2m}, q)$ so as to obtain statistically random matrices $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{D}_0 \in \mathbb{Z}_q^{2n \times 2m}$ with trapdoors $\mathbf{T}_\mathbf{C} \in \mathbb{Z}^{m \times m}$, $\mathbf{T}_{\mathbf{D}_0} \in \mathbb{Z}^{2m \times 2m}$ consisting of short bases of $\Lambda_q^\perp(\mathbf{C})$ and $\Lambda_q^\perp(\mathbf{D}_0)$, respectively. Then, $\mathcal{B}$ chooses a uniformly random $\mathbf{D} \hookleftarrow (\mathbb{Z}_q^{n \times m})$ and re-randomizes it using short matrices $\mathbf{S}, \mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_\ell \hookleftarrow \mathbb{Z}^{m \times m}$, which are obtained by sampling their columns from the distribution $D_{\mathbb{Z}^m, \sigma}$. Namely, from $\mathbf{D} \in \mathbb{Z}_q^{n \times m}$, $\mathcal{B}$ defines

$$\begin{aligned} \mathbf{A} &= \mathbf{D} \cdot \mathbf{S} \\ \mathbf{A}_0 &= \mathbf{D} \cdot \mathbf{S}_0 + h_0 \cdot \mathbf{C} \\ \mathbf{A}_j &= \mathbf{D} \cdot \mathbf{S}_j + h_j \cdot \mathbf{C} \qquad\qquad \forall j \in \{1, \ldots, \ell\} \end{aligned} \tag{7.4}$$

In addition, $\mathcal{B}$ picks random matrices $\mathbf{D}_1, \ldots, \mathbf{D}_N \hookleftarrow (\mathbb{Z}_q^{2n \times 2m})$ and a random vector $\mathbf{c}_M \hookleftarrow (\mathbb{Z}_q^{2n})$. It samples short vectors $\mathbf{v}_1, \mathbf{v}_2 \hookleftarrow D_{\mathbb{Z}^m, \sigma}$ and computes $\mathbf{u} \in \mathbb{Z}_q^n$ as

$$\mathbf{u} = \mathbf{A}_{\tau^{(i^\dagger)}} \cdot \left[\frac{\mathbf{v}_1}{\mathbf{v}_2}\right] - \mathbf{D} \cdot \mathsf{bin}(\mathbf{c}_M) \bmod q, \text{ where}$$

$$\begin{aligned} \mathbf{A}_{\tau^{(i^\dagger)}} &= \left[\begin{array}{c|c} \mathbf{A} & \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{A}_j \end{array}\right] \\ &= \left[\begin{array}{c|c} \mathbf{D} \cdot \mathbf{S} & \mathbf{D} \cdot (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{S}_j) \end{array}\right]. \end{aligned}$$

The adversary's signing queries are then answered as follows.

- At the i-th signing query $(\mathfrak{m}_1^{(i)}, \ldots, \mathfrak{m}_N^{(i)})$, whenever $i \neq i^\dagger$, we have

$$\begin{aligned} \mathbf{A}_{\tau^{(i)}} &= \left[\begin{array}{c|c} \mathbf{A} & \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot \mathbf{A}_j \end{array}\right] \\ &= \left[\begin{array}{c|c} \mathbf{A} & \mathbf{D} \cdot (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot \mathbf{S}_j) + h_{\tau^{(i)}} \cdot \mathbf{C} \end{array}\right] \in \mathbb{Z}_q^{n \times 2m}, \end{aligned}$$

  with $h_{\tau^{(i)}} = h_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot h_j \neq 0$. This implies that $\mathcal{B}$ can use the trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ to generate a signature. To this end, $\mathcal{B}$ first samples a discrete Gaussian vector $\vec{s}^{(i)} \hookleftarrow D_{\mathbb{Z}^{2m}, \sigma_1}$ and computes $\mathbf{u}_M \in \mathbb{Z}_q^n$ as

$$\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i)} + \mathbf{D}_0 \cdot \mathbf{s}^{(i)}) \mod q.$$

  Then, using $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$, it samples a short vector $\mathbf{v}^{(i)} \in \mathbb{Z}^{2m}$ in $D^{\mathbf{u}_M}_{\Lambda^\perp(\mathbf{A}_{\tau^{(i)}}), \sigma}$ such that $(\tau^{(i)}, \mathbf{v}^{(i)}, \mathbf{s}^{(i)})$ satisfies (7.2).

- At the $i^\dagger$-th signing query $(\mathfrak{m}_1^{(i^\dagger)}, \ldots, \mathfrak{m}_N^{(i^\dagger)})$, we have

$$\begin{aligned} \mathbf{A}_{\tau^{(i^\dagger)}} &= \left[\begin{array}{c|c} \mathbf{A} & \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{A}_j \end{array}\right] \\ &= \left[\begin{array}{c|c} \mathbf{D} \cdot \mathbf{S} & \mathbf{D} \cdot (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{S}_j) \end{array}\right] \in \mathbb{Z}_q^{n \times 2m} \end{aligned} \tag{7.5}$$

  due to the constraint $h_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot h_j = 0 \bmod q$. To answer the query, $\mathcal{B}$ uses the trapdoor $\mathbf{T}_{\mathbf{D}_0} \in \mathbb{Z}^{2m \times 2m}$ of $\Lambda_q^\perp(\mathbf{D}_0)$ to sample a short vector $\mathbf{s}^{(i^\dagger)} \in D_{\Lambda_q^{\mathbf{c}'_M}(\mathbf{D}_0), \sigma_1}$, where $\mathbf{c}'_M = \mathbf{c}_M - \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)} \in \mathbb{Z}_q^{2n}$. The obtained vector $\mathbf{s}^{(i^\dagger)} \in \mathbb{Z}^{2m}$ thus verifies

$$\mathbf{D}_0 \cdot \mathbf{s}^{(i^\dagger)} = \mathbf{c}_M - \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)} \bmod q, \tag{7.6}$$

  and $\mathcal{A}$ receives $sig^{(i^\dagger)} = (\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}^{(i^\dagger)})$, where $\mathbf{v}^{(i^\dagger)} = (\mathbf{v}_1^T \mid \mathbf{v}_2^T)^T$. By construction, the returned signature $sig^{(i^\dagger)}$ satisfies

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \left[\frac{\mathbf{v}_1}{\mathbf{v}_2}\right] = \mathbf{u} + \mathbf{D} \cdot \{0,1\}(\mathbf{D}_0 \cdot \mathbf{s}^{(i^\dagger)} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)}) \mod q,$$

  and the distribution of $(\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}^{(i^\dagger)})$ is statistically the same as in Game 2.

We conclude that $\Pr[W_2]$ is negligibly far apart from $\Pr[W_3]$ since, by the Leftover Hash Lemma (see [ABB10, Le. 13]), the public key PK in Game 3 is statistically close to its distribution in Game 2.

In Game 3, we claim that the challenger $\mathcal{B}$ can use $\mathcal{A}$ to solve the SIS problem by finding a short vector of $\Lambda_q^{\perp}(\mathbf{D})$ with probability $\Pr[W_3]$. Indeed, with probability $\Pr[W_3]$, the adversary outputs a valid signature $sig^{\star} = (\tau^{(i^{\dagger})}, \mathbf{v}^{\star}, \mathbf{s}^{\star})$ on a message $\mathsf{Msg}^{\star} = (\mathfrak{m}_1^{\star}, \ldots, \mathfrak{m}_N^{\star})$ with $\|\mathbf{v}^{\star}\| \leq \sigma\sqrt{2m}$ and $\|\mathbf{s}^{\star}\| \leq \sigma_1\sqrt{2m}$. If we parse $\mathbf{v}^{\star} \in \mathbb{Z}^{2m}$ as $(\mathbf{v}_1^{\star T} \mid \mathbf{v}_2^{\star T})^T$ with $\mathbf{v}_1^{\star}, \mathbf{v}_2^{\star} \in \mathbb{Z}^m$, we have the equality

$$\mathbf{A}_{\tau^{(i^{\dagger})}} \cdot \left[\frac{\mathbf{v}_1^{\star}}{\mathbf{v}_2^{\star}}\right] = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star}) \mod q. \tag{7.7}$$

Due to the way $\mathbf{u} \in \mathbb{Z}_q^n$ was defined at the outset of the game, $\mathcal{B}$ also knows short vectors $\mathbf{v}^{(i^{\dagger})} = (\mathbf{v}_1^T \mid \mathbf{v}_2^T)^T \in \mathbb{Z}^{2m}$ such that

$$\mathbf{A}_{\tau^{(i^{\dagger})}} \cdot \left[\frac{\mathbf{v}_1}{\mathbf{v}_2}\right] = \mathbf{u} + \mathbf{D} \cdot \mathsf{bin}(\mathbf{c}_M) \bmod q. \tag{7.8}$$

Relation (7.6) implies that $\mathbf{c}_M \neq \mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star} \bmod q$ by hypothesis. It follows that $\mathsf{bin}(\mathbf{c}_M) - \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star})$ is a non-zero vector in $\{-1, 0, 1\}^m$. Subtracting (7.8) from (7.7), we get

$$\mathbf{A}_{\tau^{(i^{\dagger})}} \cdot \left[\frac{\mathbf{v}_1^{\star} - \mathbf{v}_1}{\mathbf{v}_2^{\star} - \mathbf{v}_1}\right] = \mathbf{D} \cdot \left(\mathsf{bin}(\mathbf{c}_M) - \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star})\right) \mod q,$$

which implies

$$\left[\begin{array}{c|c} \mathbf{D} \cdot \mathbf{S} & \mathbf{D} \cdot (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i^{\dagger})}[j] \cdot \mathbf{S}_j) \end{array}\right] \cdot \left[\frac{\mathbf{v}_1^{\star} - \mathbf{v}_1}{\mathbf{v}_2^{\star} - \mathbf{v}_2}\right] = \mathbf{D} \cdot \left(\mathsf{bin}(\mathbf{c}_M) - \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star})\right) \mod q. \tag{7.9}$$

The above implies that the vector

$$\mathbf{w} = \mathbf{S} \cdot (\mathbf{v}_1^{\star} - \mathbf{v}_1) + (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i^{\dagger})}[j] \cdot \mathbf{S}_j) \cdot (\mathbf{v}_2^{\star} - \mathbf{v}_2) + \{0, 1\}(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star}) - \mathsf{bin}(\mathbf{c}_M)$$

is a short integer vector of $\Lambda_q^{\perp}(\mathbf{D})$. Indeed, its norm can be bounded as $\|\mathbf{w}\| \leq \beta'' = \sqrt{2}(\ell + 2)\sigma^2 m^{3/2} + m^{1/2}$. We argue that it is non-zero with overwhelming probability. We already observed that $\mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s}^{\star} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^{\star}) - \mathsf{bin}(\mathbf{c}_M)$ is a non-zero vector of $\{-1, 0, 1\}^m$, which rules out the event that $(\mathbf{v}_1^{\star}, \mathbf{v}_2^{\star}) = (\mathbf{v}_1, \mathbf{v}_2)$. Hence, we can only have

$\mathbf{w} = \mathbf{0}^m$ when the equality

$$\mathbf{S} \cdot (\mathbf{v}_1^\star - \mathbf{v}_1) + (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{S}_j) \cdot (\mathbf{v}_2^\star - \mathbf{v}_2)$$
$$= \mathsf{bin}(\mathbf{c}_M) - \{0,1\}(\mathbf{D}_0 \cdot \mathbf{s}^\star + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k^\star) \tag{7.10}$$

holds over $\mathbb{Z}$. However, as long as either $\mathbf{v}_1^\star \neq \mathbf{v}_1$ or $\mathbf{v}_2^\star \neq \mathbf{v}_2$, the left-hand-side member of (7.10) is information theoretically unpredictable since the columns of matrices $\mathbf{S}$ and $\{\mathbf{S}_j\}_{j=0}^{\ell}$ are statistically hidden in the view of $\mathcal{A}$. Indeed, conditionally on the public key, each column of $\mathbf{S}$ and $\{\mathbf{S}_j\}_{j=0}^{\ell}$ has at least n bits of min-entropy, as shown by, e.g., [MP12, Le. 2.7]. □

7.1.3 Protocols for Signing a Committed Value and Proving Possession of a Signature

We first show a two-party protocol whereby a user can interact with the signer in order to obtain a signature on a committed message.

In order to prove that the scheme still guarantees unforgeability for obliviously signed messages, we will assume that each message block $\mathfrak{m}_k \in \{0,1\}^{2m}$ is obtained by encoding the actual message $M_k = M_k[1] \dots M_k[m] \in \{0,1\}^m$ as $\mathfrak{m}_k = \mathsf{Encode}(M_k) = (\bar{M}_k[1], M_k[1], \dots, \bar{M}_k[m], M_k[m])$. Namely, each 0 (respectively each 1) is encoded as a pair $(1,0)$ (resp. $(0,1)$). The reason for this encoding is that the proof of Theorem 7.4 requires that at least one block $\mathfrak{m}_k^\star$ of the forgery message is 1 while the same bit is 0 at some specific signing query. We will show (see Section 7.3) that the correctness of this encoding can be efficiently proved using Stern-like [Ste96] protocols.

To sign committed messages, a first idea is exploit the fact that our signature of Section 7.1.1 blends well with the SIS-based commitment scheme suggested by Kawachi *et al.* [KTX08]. In the latter scheme, the commitment key consists of matrices $(\mathbf{D}_0, \mathbf{D}_1) \in \mathbb{Z}_q^{2n \times 2m} \times \mathbb{Z}_q^{2n \times 2m}$, so that message $\mathfrak{m} \in \{0,1\}^{2m}$ can be committed to by sampling a Gaussian vector $\mathbf{s} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and computing $\mathbf{C} = \mathbf{D}_0 \cdot \mathbf{s} + \mathbf{D}_1 \cdot \mathfrak{m} \in \mathbb{Z}_q^{2n}$. This scheme extends to commit to multiple messages $(\mathfrak{m}_1, \dots, \mathfrak{m}_N)$ at once by computing $\mathbf{C} = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \in \mathbb{Z}_q^{2n}$ using a longer commitment key $(\mathbf{D}_0, \mathbf{D}_1, \dots, \mathbf{D}_N) \in (\mathbb{Z}_q^{2n \times 2m})^{N+1}$. It is easy to see that the resulting commitment remains statistically hiding and computationally binding under the SIS assumption.

In order to make our construction usable in the definitional framework of Camenisch *et al.* [CKL+15], we assume common public parameters (i.e., a common reference string) and encrypt all witnesses of which knowledge is being proved under a public key included in the common reference string. The resulting ciphertexts thus serve as statistically binding commitments to the witnesses. To enable this, the common public parameters comprise public keys $\mathbf{G}_0 \in \mathbb{Z}_q^{n \times \ell}$, $\mathbf{G}_1 \in \mathbb{Z}_q^{n \times 2m}$ for multi-bit variants of the dual Regev cryptosystem [GPV08] and all parties are denied access to the underlying private keys. The flexibility of Stern-like protocols allows us to prove that the content of a perfectly hiding commitment $\mathbf{c}_\mathfrak{m}$ is consistent with encrypted values.

Global-Setup: Let $B = \sqrt{n}\omega(\log n)$ and let χ be a B-bounded distribution. Let $p = \sigma \cdot \omega(\sqrt{m})$ upper-bound entries of vectors sampled from the distribution $D_{\mathbb{Z}^{2m},\sigma}$. Generate two public keys for the dual Regev encryption scheme in its multi-bit variant. These keys consists of a public random matrix $\mathbf{B} \hookleftarrow (\mathbb{Z}_q^{n\times m})$ and random matrices $\mathbf{G}_0 = \mathbf{B}\cdot\mathbf{E}_0 \in \mathbb{Z}_q^{n\times\ell}$, $\mathbf{G}_1 = \mathbf{B}\cdot\mathbf{E}_1 \in \mathbb{Z}_q^{n\times 2m}$, where $\mathbf{E}_0 \in \mathbb{Z}^{m\times\ell}$ and $\mathbf{E}_1 \in \mathbb{Z}^{m\times 2m}$ are short Gaussian matrices with columns sampled from $D_{\mathbb{Z}^m,\sigma}$. These matrices will be used to encrypt integer vectors of dimension ℓ and $2m$, respectively. Finally, generate public parameters $CK \coloneqq \{\mathbf{D}_k\}_{k=0}^N$ consisting of uniformly random matrices $\mathbf{D}_k \hookleftarrow (\mathbb{Z}_q^{2n\times 2m})$ for a statistically hiding commitment to vectors in $(\{0,1\}^{2m})^N$. Return public parameters consisting of

$$\mathsf{par} \coloneqq \{\mathbf{B} \in \mathbb{Z}_q^{n\times m}, \mathbf{G}_0 \in \mathbb{Z}_q^{n\times\ell}, \mathbf{G}_1 \in \mathbb{Z}_q^{n\times 2m}, CK\}.$$

Issue $\leftrightarrow$ Obtain : The signer S, who holds a key pair $PK \coloneqq \{\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{D},\ \mathbf{u}\}$, $SK \coloneqq \mathbf{T}_{\mathbf{A}}$, interacts with the user U who has a message $(\mathfrak{m}_1,\ldots,\mathfrak{m}_N)$, in the following interactive protocol.

1. U samples $\mathbf{s}' \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and computes $\mathbf{c}_{\mathfrak{m}} = \mathbf{D}_0\cdot\mathbf{s}' + \sum_{k=1}^N \mathbf{D}_k\cdot\mathfrak{m}_k \in \mathbb{Z}_q^{2n}$ which is sent to S as a commitment to $(\mathfrak{m}_1,\ldots,\mathfrak{m}_N)$. In addition, U encrypts $\{\mathfrak{m}_k\}_{k=1}^N$ and $\mathbf{s}'$ under the dual-Regev public key $(\mathbf{B},\mathbf{G}_1)$ by computing for all $k \in \{1,\ldots,N\}$:

$$\begin{aligned}\mathbf{c}_k &= (\mathbf{c}_{k,1},\mathbf{c}_{k,2})\\ &= (\mathbf{B}^T\cdot\mathbf{s}_k + \mathbf{e}_{k,1},\ \mathbf{G}_1^T\cdot\mathbf{s}_k + \mathbf{e}_{k,2} + \mathfrak{m}_k\cdot\lfloor q/2\rfloor) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m}\end{aligned} \tag{7.11}$$

for randomly chosen $\mathbf{s}_k \hookleftarrow \chi^n$, $\mathbf{e}_{k,1} \hookleftarrow \chi^m$, $\mathbf{e}_{k,2} \hookleftarrow \chi^{2m}$, and

$$\begin{aligned}\mathbf{c}_{s'} &= (\mathbf{c}_{s',1},\mathbf{c}_{s',2})\\ &= (\mathbf{B}^T\cdot\mathbf{s}_0 + \mathbf{e}_{0,1},\ \mathbf{G}_1^T\cdot\mathbf{s}_0 + \mathbf{e}_{0,2} + \mathbf{s}'\cdot\lfloor q/p\rfloor) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m}\end{aligned} \tag{7.12}$$

where $\mathbf{s}_0 \hookleftarrow \chi^n$, $\mathbf{e}_{0,1} \hookleftarrow \chi^m$, $\mathbf{e}_{0,2} \hookleftarrow \chi^{2m}$. The ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$ and $\mathbf{c}_{s'}$ are sent to S along with $\mathbf{c}_{\mathfrak{m}}$.

Then, U generates an interactive zero-knowledge argument to convince S that $\mathbf{c}_{\mathfrak{m}}$ is a commitment to $(\mathfrak{m}_1,\ldots,\mathfrak{m}_N)$ with the randomness $\mathbf{s}'$ such that $\{\mathfrak{m}_k\}_{k=1}^N$ and $\mathbf{s}'$ were honestly encrypted to $\{\mathbf{c}_k\}_{i=1}^N$ and $\mathbf{c}_{s'}$, as in (7.11) and (7.12). For convenience, this argument system will be described in Section 7.3.1, where we demonstrate that, together with other zero-knowledge protocols used in this work, it can be derived from a Stern-like [Ste96] protocol constructed in Section 7.3.

2. If the argument of step 1 properly verifies, S samples $\mathbf{s}'' \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_0}$ and computes a vector $\mathbf{u}_{\mathfrak{m}} = \mathbf{u} + \mathbf{D}\cdot\{0,1\}(\mathbf{c}_{\mathfrak{m}} + \mathbf{D}_0\cdot\mathbf{s}'') \in \mathbb{Z}_q^n$. Next, S randomly picks $\tau \hookleftarrow \{0,1\}^{\ell}$ and uses $\mathbf{T}_{\mathbf{A}}$ to compute a delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m\times 2m}$ for the matrix $\mathbf{A}_\tau \in \mathbb{Z}_q^{n\times 2m}$ of (7.1). Using $\mathbf{T}_\tau \in \mathbb{Z}^{2m\times 2m}$, S samples a short vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D^{\mathbf{u}_M}_{\Lambda^\perp(\mathbf{A}_\tau),\sigma}$. It returns the vector $(\tau,\mathbf{v},\mathbf{s}'') \in \{0,1\}^{\ell}\times\mathbb{Z}^{2m}\times\mathbb{Z}^{2m}$ to U.

3. U computes $\mathbf{s} = \mathbf{s}' + \mathbf{s}''$ over $\mathbb{Z}$ and verifies that

$$\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \{0,1\}(\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k) \bmod q.$$

   If so, it outputs $(\tau, \mathbf{v}, \mathbf{s})$. Otherwise, it outputs $\perp$.

Note that, if both parties faithfully run the protocol, the user obtains a valid signature $(\tau, \mathbf{v}, \mathbf{s})$ for which the distribution of $\mathbf{s}$ is $D_{\mathbb{Z}^{2m},\sigma_1}$, where $\sigma_1 = \sqrt{\sigma^2 + \sigma_0^2}$.
The following protocol allows proving possession of a message-signature pair.

Prove: On input of a signature $(\tau, \mathbf{v} = (\mathbf{v}_1^T \mid \mathbf{v}_2^T)^T, \mathbf{s}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$ on the message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$, the user does the following.

1. Using $(\mathbf{B}, \mathbf{G}_0)$ and $(\mathbf{B}, \mathbf{G}_1)$ generate perfectly binding commitments to $\tau \in \{0,1\}^\ell$, $\{\mathfrak{m}_k\}_{k=1}^N$, $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{Z}^m$ and $\mathbf{s} \in \mathbb{Z}^{2m}$. Namely, compute

$$\begin{aligned}
\mathbf{c}_\tau &= (\mathbf{c}_{\tau,1}, \mathbf{c}_{\tau,2}) \\
&= (\mathbf{B}^T \cdot \mathbf{s}_\tau + \mathbf{e}_{\tau,1},\ \mathbf{G}_0^T \cdot \mathbf{s}_\tau + \mathbf{e}_{\tau,2} + \tau \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^\ell, \\
\mathbf{c}_k &= (\mathbf{c}_{k,1}, \mathbf{c}_{k,2}) \\
&= (\mathbf{B}^T \cdot \mathbf{s}_k + \mathbf{e}_{k,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_k + \mathbf{e}_{k,2} + \mathfrak{m}_k \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m} \\
&\qquad\qquad \forall k \in \{1, \ldots, N\}
\end{aligned}$$

   where $\mathbf{s}_\tau, \mathbf{s}_k \hookleftarrow \chi^n$, $\mathbf{e}_{\tau,1}, \mathbf{e}_{k,1} \hookleftarrow \chi^m$, $\mathbf{e}_{\tau,2} \hookleftarrow \chi^\ell$, $\mathbf{e}_{k,2} \hookleftarrow \chi^{2m}$, as well as

$$\begin{aligned}
\mathbf{c}_\mathbf{v} &= (\mathbf{c}_{\mathbf{v},1}, \mathbf{c}_{\mathbf{v},2}) \\
&= (\mathbf{B}^T \cdot \mathbf{s}_\mathbf{v} + \mathbf{e}_{\mathbf{v},1},\ \mathbf{G}_1^T \cdot \mathbf{s}_\mathbf{v} + \mathbf{e}_{\mathbf{v},2} + \mathbf{v} \cdot \lfloor q/p \rfloor) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m} \\
\mathbf{c}_s &= (\mathbf{c}_{s,1}, \mathbf{c}_{s,2}) \\
&= (\mathbf{B}^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,1},\ \mathbf{G}_1^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,2} + \mathbf{s} \cdot \lfloor q/p \rfloor) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m},
\end{aligned}$$

   where $\mathbf{s}_\mathbf{v}, \mathbf{s}_0 \hookleftarrow \chi^n$, $\mathbf{e}_{\mathbf{v},1}, \mathbf{e}_{0,1} \hookleftarrow \chi^m$, $\mathbf{e}_{\mathbf{v},2}, \mathbf{e}_{0,2} \hookleftarrow \chi^{2m}$.

2. Prove in zero-knowledge that $\mathbf{c}_\tau$, $\mathbf{c}_s$, $\mathbf{c}_\mathbf{v}$, $\{\mathbf{c}_k\}_{k=1}^N$ encrypt a valid message-signature pair. In Section 7.3.2, we show that this involved zero-knowledge protocol can be derived from the statistical zero-knowledge argument of knowledge for a simpler, but more general relation that we explicitly present in Section 7.3. The proof system can be made statistically ZK for a malicious verifier using standard techniques (assuming a common reference string, we can use [Dam00]). In the random oracle model, it can be made non-interactive using the Fiat-Shamir heuristic [FS86].

We require that the adversary be unable to prove possession of a signature of a message $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$ for which it did not legally obtain a credential by interacting with the issuer. Note that the messages that are blindly signed by the issuer are uniquely defined since, at each signing query, the adversary is required to supply perfectly binding commitments $\{\mathbf{c}_k\}_{k=1}^N$ to $(\mathfrak{m}_1, \ldots, \mathfrak{m}_N)$.

In instantiations using non-interactive proofs, we assume that these can be bound to a verifier-chosen nonce to prevent replay attacks, as suggested in [CKL+15].

The security proof (in Theorem 7.4) makes crucial use of the Rényi divergence using arguments in the spirit of Bai *et al.* [BLL+15]. The reduction has to guess upfront the index $i^\star \in \{1, \ldots, Q\}$ of the specific signing query for which the adversary will re-use $\tau^{(i^\star)}$. For this query, the reduction will have to make sure that the simulation trapdoor of Agrawal *et al.* [ABB10] (used by the SampleRight algorithm of Lemma 3.7) vanishes: otherwise, the adversary's forgery would not be usable for solving SIS. This means that, as in the proof of [BHJ+15], the reduction must answer exactly one signing query in a different way, without using the trapdoor. While Böhl *et al.* solve this problem by exploiting the fact that they only need to prove security against non-adaptive forgers, we directly use a built-in chameleon hash function mechanism which is implicitly realized by the matrix $\mathbf{D}_0$ and the vector $\mathbf{s}$. Namely, in the signing query for which the Agrawal *et al.* trapdoor [ABB10] cancels, we assign a special value to the vector $\mathbf{s} \in \mathbb{Z}^{2m}$, which depends on the adaptively-chosen signed message $(\mathsf{Msg}_1^{(i^\star)}, \ldots, \mathsf{Msg}_N^{(i^\star)})$ and some Gaussian matrices $\{\mathbf{R}_k\}_{k=1}^N$ hidden behind $\{\mathbf{D}_k\}_{k=1}^N$.

One issue is that this results in a different distribution for the vector $\mathbf{s} \in \mathbb{Z}^m$. However, we can still view $\mathbf{s}$ as a vector sampled from a Gaussian distribution centered away from $\mathbf{0}^{2m}$. Since this specific situation occurs only once during the simulation, we can apply a result proved in [LSS14] which upper-bounds the Rényi divergence between two Gaussian distributions with identical standard deviations but different centers. By choosing the standard deviation σ_1 of $\mathbf{s} \in \mathbb{Z}^{2m}$ to be polynomially larger than that of the columns of matrices $\{\mathbf{R}_k\}_{k=1}^N$, we can keep the Rényi divergence between the two distributions of $\mathbf{s}$ (i.e., the one of the simulation and the one of the real game) sufficiently small to apply the probability preservation property (which still gives a polynomial reduction since the argument must only be applied on one signing query). Namely, the latter implies that, if the Rényi divergence $R_2(\mathbf{s}^{\mathsf{real}} || \mathbf{s}^{\mathsf{sim}})$ is polynomial, the probability that the simulated vector $\mathbf{s}^{\mathsf{sim}} \in \mathbb{Z}^{2m}$ passes the verification test will only be polynomially smaller than in the real game and so will be the adversary's probability of success.

Another option would have been to keep the statistical distance between $\mathbf{s}^{\mathsf{real}}$ and $\mathbf{s}^{\mathsf{sim}}$ negligible using the smudging technique of [AJLA+12]. However, this would have implied to use an exponentially large modulus q since σ_1 should have been exponentially larger than the standard deviations of the columns of $\{\mathbf{R}_k\}_{k=1}^N$.

Theorem 7.4. *Under the* $\mathsf{SIS}_{n,2m,q,\hat{\beta}}$ *assumption, where* $\hat{\beta} = N\sigma(2m)^{3/2} + 4\sigma_1 m^{3/2}$*, the above protocols are secure protocols for obtaining a signature on a committed message and proving possession of a valid message-signature pair.*

In the following proof, we make use of the Rényi divergence in a similar way to [BLL+15]: instead of the classical statistical distance we sometimes use the Rényi divergence, which is a measurement of the distance between two distributions. Its use in security proofs for lattice-based systems was first considered by Bai *et al.* [BLL+15] and further improved by Prest [Pre17]. We first recall its definition.

Definition 2.14 (Rényi divergence)**.** For any two discrete distributions P and Q such that

$\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$, and $a \in]1, +\infty[$, we define the *Rényi divergence* of order a by:

$$R_a(P||Q) = \left(\sum_{x \in \mathrm{Supp}(P)} \frac{P(x)^a}{Q(x)^{a-1}} \right)^{\frac{1}{a-1}}.$$

We define the Rényi divergences of orders 1 and $+\infty$ as:

$$R_1(P||Q) = \exp\left(\sum_{x \in \mathrm{Supp}(P)} P(x) \log \frac{P(x)}{Q(x)} \right) \text{ and } R_\infty(P||Q) = \max_{x \in \mathrm{Supp}(P)} \frac{P(x)}{Q(x)}.$$

The divergence R_1 is the (exponential) of the Kullback-Leibler divergence.

We will focus on the following properties of the Rényi divergence, the proofs can be found in [LSS14].

Lemma 7.5 ([BLL+15, Le. 2.7]). *Let $a \in [1, +\infty]$. Let P and Q denote distributions with $\mathrm{Supp}(P) \subseteq \mathrm{Supp}(Q)$. Then the following properties hold:*

Log. Positivity: $R_a(P||Q) \geq R_a(P||P) = 1$

Data Processing Inequality: $R_a(P^f||Q^f) \leq R_a(P||Q)$ *for any function f, where P^f denotes the distribution of $f(y)$ induced by sampling $y \hookleftarrow P$ (resp. $y \hookleftarrow Q$)*

Multiplicativity: *Assume P and Q are two distributions of a pair of random variables (Y_1, Y_2). For $i \in \{1, 2\}$, let P_i (resp. Q_i) denote 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:*

- $R_a(P||Q) = P_a(P_1||Q_1) \cdot R_a(P_2||Q_2)$ *if Y_B and Y_2 are independent;*
- $R_a(P||Q) \leq R_\infty(P_1||Q_1) \cdot max_{y_1 \in X} R_a\left(P_{2|1}(\cdot|y_1)||Q_{2|1}(\cdot|y_1)\right)$.

Probability Preservation: *Let $A \subseteq \mathrm{Supp}(Q)$ be an arbitrary event. If $a \in]1, +\infty[$, then $Q(A) \geq P(A)^{\frac{a}{a-1}}/R_a(P||Q)$. Further we have:*

$$Q(A) \geq P(A)/R_\infty(P||Q)$$

Weak Triangle Inequality: *Let P_1, P_2, P_3 be three distributions with*

$$\mathrm{Supp}(P_1) \subseteq \mathrm{Supp}(P_2) \subseteq \mathrm{Supp}(P_3).$$

Then we have:

$$R_a(P_1||P_3) \leq \begin{cases} R_a(P_1||P_2) \cdot R_\infty(P_2||P_3), \\ R_\infty(P_1||P_2)^{\frac{a}{a-1}} \cdot R_a(P_2||P_3) \quad \text{if } a \in]1, +\infty[. \end{cases}$$

In our proofs, we mainly use the probability preservation to bound the probabilities during hybrid games where the two distributions are not close in terms of statistical distance.

Proof. The proof is very similar to the proof of Theorem 7.1 and we will only explain the changes.

Let us assume that an adversary $\mathcal{A}$ can prove possession of a signature on a message $(\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star)$ which has not been blindly signed by the issuer, we outline an algorithm $\mathcal{B}$ that solves a $\mathsf{SIS}_{n,2m,q,\beta}$ instance $\bar{\mathbf{A}}$, where $\bar{\mathbf{A}} = [\bar{\mathbf{A}}_1 \mid \bar{\mathbf{A}}_2] \in \mathbb{Z}_q^{n\times 2m}$ with matrices $\bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2 \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$.

At the outset of the game, $\mathcal{B}$ generates the common parameters par by choosing $\mathbf{B} \in_R \mathbb{Z}_q^{n\times m}$ and defining $\mathbf{G}_0 = \mathbf{B}\cdot\mathbf{E}_0 \in \mathbb{Z}_q^{n\times \ell}$, $\mathbf{G}_1 = \mathbf{B}\cdot\mathbf{E}_1 \in \mathbb{Z}_q^{n\times 2m}$. The short Gaussian matrices $\mathbf{E}_0 \in \mathbb{Z}^{m\times\ell}$ and $\mathbf{E}_1 \in \mathbb{Z}^{m\times 2m}$ are retained for later use. Also, $\mathcal{B}$ flips a coin $coin \in \{0,1,2\}$ as a guess for the kind of attack that $\mathcal{A}$ will mount. If $coin = 0$, $\mathcal{B}$ expects a Type I forgery, where $\mathcal{A}$'s forgery involves a new $\tau^\star \in \{0,1\}^\ell$ that was never used by the signing oracle. If $coin = 1$, $\mathcal{B}$ expects $\mathcal{A}$ to recycle a tag $\tau^\star$ involved in some signing query in its forgery. Namely, if $coin = 1$, $\mathcal{B}$ expects an attack which is either a Type II forgery or a Type III forgery. If $coin = 2$, $\mathcal{B}$ rather bets that $\mathcal{A}$ will break the soundness of the interactive argument systems used in the signature issuing protocol or the Prove protocol. Depending on the value of $coin \in \{0,1,2\}$, $\mathcal{B}$ generates the issuer's public key PK and simulates $\mathcal{A}$'s view in different ways.

- If $coin = 0$, $\mathcal{B}$ undertakes to find a short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}}_1)$, which in turn yields a short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$. To this end, it defines $\mathbf{A} = \bar{\mathbf{A}}_1$ and generates PK by computing $\{\mathbf{A}_j\}_{j=0}^\ell$ as re-randomizations of $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ as in the proof of Lemma 7.2. This implies that $\mathcal{B}$ can always answer signing queries using the trapdoor $\mathbf{T}_\mathbf{C} \in \mathbb{Z}^{m\times m}$ of the matrix $\mathbf{C}$ without even knowing the messages hidden in the commitments $\mathbf{c}_\mathfrak{m}$ and $\{\mathbf{c}_k\}_{k=1}^N$, $\mathbf{c}_{s'}$. When the adversary generates a proof of possession of its own at the end of the game, $\mathcal{B}$ uses the matrices $\mathbf{E}_0 \in \mathbb{Z}^{m\times\ell}$ and $\mathbf{E}_1 \in \mathbb{Z}^{m\times 2m}$ as an extraction trapdoor to extract a plain message-signature pair $((\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star), (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star))$ from the ciphertexts $\{\mathbf{c}_k^\star\}_{k=1}^N$ $(\mathbf{c}_{\mathbf{v}_1}^\star, \mathbf{c}_{\mathbf{v}_2}^\star)$, $\mathbf{c}_\tau^\star$, $\mathbf{c}_\mathbf{s}^\star$ produced by $\mathcal{A}$ as part of its forgery. If the extracted $\tau^\star$ is not a new tag, then $\mathcal{B}$ aborts. Otherwise, it can solve the given SIS instance exactly as in the proof of Lemma 7.2.

- If $coin = 1$, the proof proceeds as in the proof of Lemma 7.3 with one difference in Game 3. This difference is that Game 3 is no longer statistically indistinguishable from Game 2: instead, we rely on an argument based on the Rényi divergence. In Game 3, $\mathcal{B}$ generates PK exactly as in the proof of Lemma 7.3. This implies that $\mathcal{B}$ takes a guess $i^\dagger \leftarrow U(\{1,\ldots,Q\})$ with the hope that $\mathcal{A}$ will choose to recycle the tag $\tau^{(i^\dagger)}$ of the $i^\dagger$-th signing query (i.e., $\tau^\star = \tau^{(i^\dagger)}$). As in the proof of Lemma 7.3, $\mathcal{B}$ defines $\mathbf{D} = \bar{\mathbf{A}}_1 \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{A} = \bar{\mathbf{A}}_1 \cdot \mathbf{S}$ for a small-norm matrix $\mathbf{S} \in \mathbb{Z}^{m\times m}$ with Gaussian entries. It also "programs" the matrices $\{\mathbf{A}_j\}_{j=0}^\ell$ in such a way that the trapdoor precisely vanishes at the $i^\dagger$-th signing query: in other words, the sum

$$\mathbf{A}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j]\mathbf{A}_j = \bar{\mathbf{A}}_1 \cdot (\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j]\cdot\mathbf{S}_j) + (h_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j]\cdot h_j)\cdot\mathbf{C}$$

  does not depend on the matrix $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$ (of which a trapdoor $\mathbf{T}_\mathbf{C} \in \mathbb{Z}^{m\times m}$ is known to $\mathcal{B}$) when $\tau^{(i)} = \tau^{(i^\dagger)}$, but it does for all other tags $\tau^{(i)} \neq \tau^{(i^\dagger)}$. In the

setup phase, $\mathcal{B}$ also sets up a random matrix $\mathbf{D}_0 \in U(\mathbb{Z}_q^{2n\times 2m})$ which it obtains by choosing $\mathbf{A}' \hookleftarrow (\mathbb{Z}_q^{n\times 2m})$ to define

$$\mathbf{D}_0 = \left[\frac{\bar{\mathbf{A}}}{\mathbf{A}'}\right] \in \mathbb{Z}_q^{2n\times 2m}. \tag{7.13}$$

Then, it computes $\mathbf{c}_M = \mathbf{D}_0 \cdot \mathbf{s}_0 \in \mathbb{Z}_q^{2n}$ for a short Gaussian vector $\mathbf{s}_0 \hookleftarrow D_{\mathbb{Z}^{2m},\sigma_0}$, which will be used in the $i^\dagger$-th query. Next, it samples short vectors $\mathbf{v}_1, \mathbf{v}_2 \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ to define

$$\mathbf{u} = \mathbf{A}_{\tau^{(i^\dagger)}} \cdot \begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\end{bmatrix} - \mathbf{D}\cdot \mathsf{bin}(\mathbf{c}_M) \ \in \mathbb{Z}_q^n.$$

In addition, $\mathcal{B}$ picks extra small-norm matrices $\mathbf{R}_1, \ldots, \mathbf{R}_N \in \mathbb{Z}^{2m\times 2m}$ whose columns are sampled from $D_{\mathbb{Z}^m,\sigma}$, which are used to define randomizations of $\mathbf{D}_0$ by computing $\mathbf{D}_k = \mathbf{D}_0 \cdot \mathbf{R}_k$ for each $k \in \{1, \ldots, N\}$. The adversary is given public parameters $\mathsf{par} := \{\mathbf{B}, \mathbf{G}_0, \mathbf{G}_1, CK\}$, where $CK = \{\mathbf{D}_k\}_{k=0}^N$, and the public key $PK := (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^\ell, \mathbf{D}, \mathbf{u})$.

Using $\mathbf{T_C}$, $\mathcal{B}$ can perfectly emulate the signing oracle at all queries, except the $i^\dagger$-th query where the vector $\mathbf{s}''^{(i^\dagger)}$ chosen by $\mathcal{B}$ is sampled from a distribution that departs from $D_{\mathbb{Z}^{2m},\sigma_0}$. At the $i^\dagger$-th query, $\mathcal{B}$ uses the extraction trapdoor $\mathbf{E}_1 \in \mathbb{Z}^{m\times 2m}$ to obtain $\mathbf{s}'^{(i^\dagger)} \in \mathbb{Z}^{2m}$ and $\{\mathfrak{m}_k\}_{k=1}^N$ – which form a valid opening of $\mathbf{c}_\mathfrak{m}$ unless the soundness of the proof system is broken (note that the latter case is addressed by the situation $coin = 3$) – from the ciphertexts $\mathbf{c}_{s'}^{(i^\dagger)}$ and $\{\mathbf{c}_k\}_{k=1}^N$ sent by $\mathcal{A}$ at step 1 of the signing protocol. Then, $\mathcal{B}$ computes the vector $\mathbf{s}''^{(i^\dagger)}$ as

$$\mathbf{s}''^{(i^\dagger)} = \mathbf{s}_0 - \sum_{k=1}^N \mathbf{R}_k \cdot \mathfrak{m}_k^{(i^\dagger)} - \mathbf{s}'^{(i^\dagger)} \in \mathbb{Z}^{2m}, \tag{7.14}$$

which satisfies $\mathbf{c}_M = \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)} + \mathbf{D}_0 \cdot (\mathbf{s}'^{(i^\dagger)} + \mathbf{s}''^{(i^\dagger)})$ and allows returning $(\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}''^{(i^\dagger)})$ such that $(\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}'^{(i^\dagger)} + \mathbf{s}''^{(i^\dagger)})$ satisfies the verification equation of the signature scheme. Moreover, we argue that, with noticeable probability, the integer vector $\mathbf{s}^{(i^\dagger)} = \mathbf{s}'^{(i^\dagger)} + \mathbf{s}''^{(i^\dagger)}$ will be accepted by the verification algorithm since the Rényi divergence between the simulated distribution of $\mathbf{s}''^{(i^\dagger)}$ and its distribution in the real game will be sufficiently small. Indeed, its distribution is now that of a Gaussian vector $D_{\mathbb{Z}^{2m},\sigma_0,\mathbf{z}^\dagger}$ centered in

$$\mathbf{z}^\dagger = -\sum_{k=1}^N \mathbf{R}_k \cdot \mathfrak{m}_k^{(i^\dagger)} - \mathbf{s}'^{(i^\dagger)} \in \mathbb{Z}^{2m},$$

whose norm is at most $\|\mathbf{z}^\dagger\|_2 \leq N\sigma(2m)^{3/2} + \sigma(2m)^{1/2}$. By choosing the standard deviation σ_0 to be at least $\sigma_0 > N\sigma(2m)^{3/2} + \sigma(2m)^{1/2}$, the Rényi divergence between the simulated distribution of $\mathbf{s}''^{(i^\dagger)}$ (in Game 3) and its real distribution (which is the one of Game 2) can be kept constant: we have

$$R_2(\mathbf{s}''^{(i^\dagger),2} || \mathbf{s}''^{(i^\dagger),3}) \leq \exp\left(2\pi \cdot \frac{\|\mathbf{z}^\dagger\|_2^2}{\sigma_0^2}\right) \leq \exp(2\pi). \tag{7.15}$$

This ensures that, with noticeable probability, $(\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)}, \mathbf{s}^{(i^\dagger)})$ will pass the verification test and lead $\mathcal{A}$ to eventually output a valid forgery. So, the success probability of $\mathcal{A}$ in Game 3 remains noticeable as (7.15) implies $\Pr[W_3] \geq \Pr[W_2]^2/\exp(2\pi)$.

When W_3 occurs in Game 3, $\mathcal{B}$ uses the matrices $(\mathbf{E}_0, \mathbf{E}_1)$ to extract a plain message-signature pair $((\mathfrak{m}_1^\star, \ldots, \mathfrak{m}_N^\star), (\tau^\star, \mathbf{v}^\star, \mathbf{s}^\star))$ from the extractable commitments $\{\mathbf{c}_k^\star\}_{k=1}^N$ $(\mathbf{c}_{\mathbf{v}_1}^\star, \mathbf{c}_{\mathbf{v}_2}^\star)$, $\mathbf{c}_\tau^\star$, $\mathbf{c}_{\mathbf{s}}^\star$ generated by $\mathcal{A}$. At this point, two cases can be distinguished. First, if $\mathbf{c}_M \neq \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^\star + \mathbf{D}_0 \cdot \mathbf{s}^\star \bmod q$, then algorithm $\mathcal{B}$ can find a short vector of $\Lambda_q^\perp(\bar{\mathbf{A}}_1) = \Lambda_q^\perp(\mathbf{D})$ exactly as in the proof of Lemma 7.3. In the event that $\mathbf{c}_M = \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^\star + \mathbf{D}_0 \cdot \mathbf{s}^\star$, $\mathcal{B}$ can use the fact that the collision $\mathbf{c}_M = \sum_{k=1}^N \mathbf{D}_k \cdot \mathfrak{m}_k^{(i^\dagger)} + \mathbf{D}_0 \cdot \mathbf{s}^{(i^\dagger)}$ allows computing

$$\mathbf{w} = \mathbf{s}^\star - \mathbf{s}^{(i^\dagger)} + \sum_{k=1}^N \mathbf{R}_k \cdot \left(\mathfrak{m}_k^\star - \mathfrak{m}_k^{(i^\dagger)}\right) \in \mathbb{Z}^{2m},$$

which belongs to $\Lambda_q^\perp(\mathbf{D}_0)$ and has norm $\|\mathbf{w}\|_2 \leq N\sigma(2m)^{3/2} + 4\sigma_1 m^{3/2}$. Moreover, it is non-zero with overwhelming probability. Indeed, there exists at least one $k \in [1, N]$ such that $\mathfrak{m}_k^{(i^\dagger)} \neq \mathfrak{m}_k^\star$. Let us assume w.l.o.g. that they differ in their first two bits where $\mathfrak{m}_k^{(i^\dagger)}$ contains a 0 and $\mathfrak{m}_k^\star$ contains a 1 (recall that each bit b is encoded as $(\bar{b}, b)$ in both messages). This implies that $\mathbf{s}''^{(i^\dagger)}$ (as computed in (7.14)) does not depend on the first column of $\mathbf{R}_k$ but $\mathbf{w}$ does. Hence, given that the columns of $\mathbf{R}_k$ have at least n bits of min-entropy conditionally on $\mathbf{D}_k = \mathbf{D}_0 \cdot \mathbf{R}_k$, the vector $\mathbf{w} \in \mathbb{Z}^{2m}$ is unpredictable to the adversary.

Due to the definition of $\mathbf{D}_0 \in \mathbb{Z}_q^{2n \times 2m}$ in (7.13), we finally note that $\mathbf{w} \in \mathbb{Z}^{2m}$ is also a short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$.

- If $coin = 2$, $\mathcal{B}$ faithfully generates par and PK, but it retains the extraction trapdoor $(\mathbf{E}_0, \mathbf{E}_1)$ associated with the dual Regev public keys $(\mathbf{G}_0, \mathbf{G}_1)$. Note that $\mathcal{A}$ can break the soundness of the proof system by either: (i) Generating ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$ and $\mathbf{c}_{s'}$ that do not encrypt an opening of $\mathbf{c}_{\mathfrak{m}}$ in the signature issuing protocol; (ii) Generating ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$, $\mathbf{c}_\tau$, $\mathbf{c}_{\mathbf{v}_1}$, $\mathbf{c}_{\mathbf{v}_2}$ and $\mathbf{c}_s$ that do not encrypt a valid signature in the Prove protocol. In either case, the reduction $\mathcal{B}$ is able to detect the event by decrypting dual Regev ciphertext using $(\mathbf{E}_0, \mathbf{E}_1)$ and create a breach in the soundness of the argument system.

It it easy to see that, since $coin \in \{0, 1, 2\}$ is chosen independently of $\mathcal{A}$'s view, it turns out to be correct with probability $1/3$. As a consequence, if $\mathcal{A}$'s advantage is non-negligible, so is $\mathcal{B}$'s. □

Theorem 7.6. *The scheme provides anonymity under the* $\mathsf{LWE}_{n,q,\chi}$ *assumption.*

Proof. The proof is rather straightforward and consists of a sequence of three games.

Game 0: This is the real game. Namely, the adversary is given common public parameters par and comes up with a public key PK of its own. The adversary can run oblivious signing protocols with honest users. At each query, the adversary chooses a user index i and triggers an execution of the signing protocol with the challenger emulating the

honest users. At some point, the adversary chooses some user index $i^\star$ for which the execution of the signing protocol ended successfully. At this point, the challenger $\mathcal{B}$ runs the real Prove protocol on behalf of user i. At the end of the game, the adversary outputs a bit $b' \in \{0,1\}$. We define W_0 to be the event that $b' = 1$.

Game 1: This game is like Game 0 with the difference that, at each execution of the Prove protocol, the challenger runs the zero-knowledge simulator of the interactive proof system. The latter simulator uses either a trapdoor hidden in the common reference string (if Damgård's technique [Dam00] is used) or proceeds by programming the random oracle which allows implementing the Fiat-Shamir heuristic. In either case, the statistical zero-knowledge property ensures that the adversary cannot distinguish Game 1 from Game 0 and $|\Pr[W_1] - \Pr[W_0]| \in \mathsf{negl}(\lambda)$.

Game 3: This game is like Game 1 except that, at each execution of the Prove protocol, the ciphertexts $\{\mathbf{c}_k\}_{k=1}^N$, $\mathbf{c}_s$, $\mathbf{c}_\tau$, and $\mathbf{c}_{\mathbf{v}_1}$, $\mathbf{c}_{\mathbf{v}_2}$ encrypt random messages instead of the actual witnesses. The semantic security of the dual Regev cryptosystem ensures that, under the $\mathsf{LWE}_{n,q,\chi}$ assumption, the adversary is unable to see the difference. Hence, we have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(\lambda)$.

In Game 2, we can notice that the adversary is interacting with a simulator that emulates the user in the Prove protocol *without* using any message-signature pair. We thus conclude that, under the $\mathsf{LWE}_{n,q,\chi}$ assumption, $\mathcal{A}$'s view cannot distinguish a real proof of signature possession from a simulated proof produced without any witness. □

7.2 A Dynamic Lattice-Based Group Signature

In this section, the signature scheme of Section 7.1 is used to design a group signature for dynamic groups using the syntax and the security model of Kiayias and Yung [KY06], which is recalled in Section 5.2.

In the notations hereunder, for any positive integers $\mathfrak{n}$, and $q \geq 2$, we define the "powers-of-2" matrix $\mathbf{H}_{\mathfrak{n}\times\mathfrak{n}\lceil\log q\rceil} \in \mathbb{Z}_q^{\mathfrak{n}\times\mathfrak{n}\lceil\log q\rceil}$ to be:

$$\mathbf{H}_{\mathfrak{n}\times\mathfrak{n}\lceil\log q\rceil} \quad = \quad \mathbf{I}_{\mathfrak{n}} \otimes [1 \mid 2 \mid 4 \mid \ldots \mid 2^{\lceil\log q\rceil-1}].$$

Also, for each vector $\mathbf{v} \in \mathbb{Z}_q^{\mathfrak{n}}$, we define $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{\mathfrak{n}\lceil\log q\rceil}$ to be the vector obtained by replacing each entry of $\mathbf{v}$ by its binary expansion. Hence, we have $\mathbf{v} = \mathbf{H}_{\mathfrak{n}\times\mathfrak{n}\lceil\log q\rceil} \cdot \mathsf{bin}(\mathbf{v})$ for any $\mathbf{v} \in \mathbb{Z}_q^{\mathfrak{n}}$.

In our scheme, each group membership certificate is a signature generated by the group manager on the user's public key. Since the group manager only needs to sign known (rather than committed) messages, we can use a simplified version of the signature, where the chameleon hash function does not need to choose the discrete Gaussian vector $\mathbf{s}$ with a larger standard deviation than other vectors.

A key component of the scheme is the two-message joining protocol whereby the group manager admits new group members by signing their public key. The first message is sent by the new user $\mathcal{U}_i$ who samples a membership secret consisting of a short vector $\mathbf{z}_i \hookleftarrow D_{\mathbb{Z}^{4m},\sigma}$ (where $m = 2n\lceil\log q\rceil$), which is used to compute a syndrome $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i \in \mathbb{Z}_q^{4n}$ for some

public matrix $\mathbf{F} \in \mathbb{Z}_q^{4n \times 4m}$. This syndrome $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$ must be signed by $\mathcal{U}_i$ using his long term secret key $\mathsf{usk}[i]$ (as in [KY06, BSZ05], we assume that each user has a long-term key $\mathsf{upk}[i]$ for a digital signature, which is registered in some PKI) and will uniquely identify $\mathcal{U}_i$. In order to generate a membership certificate for $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$, the group manager GM signs its binary expansion $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{4n\lceil \log q \rceil}$ using the scheme of Section 7.1.

Equipped with his membership certificate $(\tau, \mathbf{d}, \mathbf{s}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}$, the new group member $\mathcal{U}_i$ can sign a message using a Stern-like protocol for demonstrating his knowledge of a valid certificate for which he also knows the secret key associated with the certified public key $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$. This boils down to providing evidence that the membership certificate is a valid signature on some binary message $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{4n\lceil \log q \rceil}$ for which he also knows a short $\mathbf{z}_i \in \mathbb{Z}^{4m}$ such that $\mathbf{v}_i = \mathbf{H}_{4n \times 2m} \cdot \mathsf{bin}(\mathbf{v}_i) = \mathbf{F} \cdot \mathbf{z}_i \in \mathbb{Z}_q^{4n}$.

Interestingly, the process does not require any proof of knowledge of the membership secret $\mathbf{z}_i$ during the joining phase, which is round-optimal. Analogously to the Kiayias-Yung technique [KY05] and constructions based on structure-preserving signatures [AFG+10], the joining protocol thus remains secure in environments where many users want to register at the same time in concurrent sessions.

We remark that a similar Stern-like protocol could also be directly used to prove knowledge of a Boyen signature [Boy10] on a binary expansion of the user's syndrome $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$ while preserving the user's ability to prove knowledge of a short $\mathbf{z}_i \in \mathbb{Z}^{4m}$ such that $\mathbf{F} \cdot \mathbf{z}_i = \mathbf{v}_i \bmod q$. However, this would require considerably longer private keys containing $4n \cdot \log q$ matrices $\{\mathbf{A}_j\}_{j=0}^{\ell}$ of dimension $n \times m$ each (i.e., we would need $\ell = \Theta(n \cdot \log q)$). In contrast, by using the signature scheme of Section 7.1, we only need the group public key $\mathcal{Y}$ to contain $\ell = \log N_{\mathsf{gs}}$ matrices in $\mathbb{Z}_q^{n \times m}$. Since the number of users N_{gs} is polynomial, we have $\log N_{\mathsf{gs}} \ll n$, which results in a much more efficient scheme.

7.2.1 Description of the Scheme

Setup$(1^\lambda, 1^{N_{\mathsf{gs}}})$**:** Given a security parameter $\lambda > 0$ and the maximal expected number of group members $N_{\mathsf{gs}} = 2^\ell \in \mathsf{poly}(\lambda)$, choose lattice parameter $n = \mathcal{O}(\lambda)$; prime modulus $q = \tilde{\mathcal{O}}(\ell n^3)$; dimension $m = 2n\lceil \log q \rceil$; Gaussian parameter $\sigma = \Omega(\sqrt{n \log q} \log n)$; infinity norm bounds $\beta = \sigma\omega(\log m)$ and $B = \sqrt{n}\omega(\log n)$. Let χ be a B-bounded distribution. Choose a hash function $H : \{0,1\}^* \to \{1,2,3\}^t$ for some $t = \omega(\log n)$, which will be modeled as a random oracle in the security analysis. Then, do the following.

1. Generate a key pair for the signature of Section 7.1.1 for signing single-block messages. Namely, run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T}_{\mathbf{A}}$ of $\Lambda_q^\perp(\mathbf{A})$, which allows computing short vectors in $\Lambda_q^\perp(\mathbf{A})$ with Gaussian parameter σ. Next, choose matrices $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell, \mathbf{D} \hookleftarrow (\mathbb{Z}_q^{n \times m})$, $\mathbf{D}_0, \mathbf{D}_1 \hookleftarrow (\mathbb{Z}_q^{2n \times 2m})$ and a vector $\mathbf{u} \hookleftarrow (\mathbb{Z}_q^n)$.
2. Choose an additional random matrix $\mathbf{F} \hookleftarrow (\mathbb{Z}_q^{4n \times 4m})$ uniformly. Looking ahead, this matrix will be used to ensure security against framing attacks.
3. Generate a master key pair for the Gentry-Peikert-Vaikuntanathan IBE scheme in its multi-bit variant. This key pair consists of a statistically uniform matrix

$\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T_B} \in \mathbb{Z}^{m \times m}$ of $\Lambda_q^{\perp}(\mathbf{B})$. This basis will allow us to compute GPV private keys with a Gaussian parameter $\sigma_{\mathrm{GPV}} \geq \|\widetilde{\mathbf{T}_\mathbf{B}}\| \cdot \sqrt{\log m}$.

4. Choose a one-time signature scheme $\Pi^{\mathrm{OTS}} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ and a hash function $H_0 : \{0,1\}^* \to \mathbb{Z}_q^{n \times 2m}$, that will be modeled as random oracles.

The group public key is defined as

$$\mathcal{Y} \coloneqq (\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{B},\ \mathbf{D},\ \mathbf{D}_0,\ \mathbf{D}_1,\ \mathbf{F},\ \mathbf{u},\ \Pi^{\mathrm{OTS}},\ H,\ H_0).$$

The opening authority's private key is $\mathcal{S}_{\mathsf{OA}} \coloneqq \mathbf{T_B}$ and the private key of the group manager consists of $\mathcal{S}_{\mathsf{GM}} \coloneqq \mathbf{T_A}$. The algorithm outputs $(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}}, \mathcal{S}_{\mathsf{OA}})$.

Join$^{(\mathsf{GM},\mathcal{U}_i)}$**:** the group manager GM and the prospective user $\mathcal{U}_i$ run the following interactive protocol: $\langle \mathsf{J}_{\mathsf{user}}(\lambda, \mathcal{Y}), \mathsf{J}_{\mathsf{GM}}(\lambda, St, \mathcal{Y}, \mathcal{S}_{\mathsf{GM}}) \rangle$

1. $\mathcal{U}_i$ samples a discrete Gaussian vector $\mathbf{z}_i \leftarrow D_{\mathbb{Z}^{4m},\sigma}$ and computes $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i \in \mathbb{Z}_q^{4n}$. He sends the vector $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$, whose binary representation $\mathsf{bin}(\mathbf{v}_i)$ consists of $4n\lceil \log q \rceil = 2m$ bits, together with an ordinary digital signature $sig_i = \mathrm{Sign}_{\mathsf{usk}[i]}(\mathbf{v}_i)$ to GM.
2. J_{GM} verifies that $\mathbf{v}_i$ was not previously used by a registered user and that sig_i is a valid signature on $\mathbf{v}_i$ w.r.t. $\mathsf{upk}[i]$. It aborts if this is not the case. Otherwise, GM chooses a fresh ℓ-bit identifier $\mathsf{id}_i = \mathsf{id}_i[1] \ldots \mathsf{id}_i[\ell] \in \{0,1\}^{\ell}$ and uses $\mathcal{S}_{\mathsf{GM}} = \mathbf{T_A}$ to certify $\mathcal{U}_i$ as a new group member. To this end, GM defines the matrix

$$\mathbf{A}_{\mathsf{id}_i} = \left[\ \mathbf{A}\ \middle|\ \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_i[j]\mathbf{A}_j\ \right] \in \mathbb{Z}_q^{n \times 2m}. \tag{7.16}$$

Then, GM runs $\mathbf{T}'_{\mathsf{id}_i} \leftarrow \mathsf{ExtBasis}(\mathbf{A}_{\mathsf{id}_i}, \mathbf{T_A})$ to obtain a short delegated basis $\mathbf{T}'_{\mathsf{id}_i}$ of $\Lambda_q^{\perp}(\mathbf{A}_{\mathsf{id}_i}) \in \mathbb{Z}^{2m \times 2m}$. Finally, GM samples a short vector $\mathbf{s}_i \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and uses the obtained delegated basis $\mathbf{T}'_{\mathsf{id}_i}$ to compute a short vector $\mathbf{d}_i = \begin{bmatrix} \mathbf{d}_{i,1} \\ \hline \mathbf{d}_{i,2} \end{bmatrix} \in \mathbb{Z}^{2m}$ such that

$$\begin{aligned} \mathbf{A}_{\mathsf{id}_i} \cdot \mathbf{d}_i &= \left[\ \mathbf{A}\ \middle|\ \mathbf{A}_0 + \textstyle\sum_{j=1}^{\ell} \mathsf{id}_i[j]\mathbf{A}_j\ \right] \cdot \mathbf{d}_i \\ &= \mathbf{u} + \mathbf{D} \cdot \{0,1\}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i) \bmod q. \end{aligned} \tag{7.17}$$

The triple $(\mathsf{id}_i, \mathbf{d}_i, \mathbf{s}_i)$ is sent to $\mathcal{U}_i$. Then, $\mathsf{J}_{\mathsf{user}}$ verifies that the received $(\mathsf{id}_i, \mathbf{d}_i, \mathbf{s}_i)$ satisfies (7.17) and that $\|\mathbf{d}_i\|_\infty \leq \beta$, $\|\mathbf{s}_i\|_\infty \leq \beta$. If these conditions are not satisfied, $\mathsf{J}_{\mathsf{user}}$ aborts. Otherwise, $\mathsf{J}_{\mathsf{user}}$ defines the membership certificate as $\mathsf{cert}_i = (\mathsf{id}_i, \mathbf{d}_i, \mathbf{s}_i)$. The membership secret sec_i is defined to be $\mathsf{sec}_i = \mathbf{z}_i \in \mathbb{Z}^{4m}$. J_{GM} stores $\mathsf{transcript}_i = (\mathbf{v}_i, \mathsf{cert}_i, i, \mathsf{upk}[i], sig_i)$ in the database St_{trans} of joining transcripts.

Sign$(\mathcal{Y}, \mathsf{cert}_i, \mathsf{sec}_i, M)$**:** To sign $M \in \{0,1\}^*$ using $\mathsf{cert}_i = (\mathsf{id}_i, \mathbf{d}_i, \mathbf{s}_i)$, where $\mathbf{d}_i = [\mathbf{d}_{i,1}^T \mid \mathbf{d}_{i,2}^T]^T \in \mathbb{Z}^{2m}$ and $\mathbf{s}_i \in \mathbb{Z}^{2m}$, as well as the membership secret $\mathsf{sec}_i = \mathbf{z}_i \in \mathbb{Z}^{4m}$, the group member $\mathcal{U}_i$ generates a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(n)$ and conducts the following steps.

1. Compute $\mathbf{G}_0 = H_0(\mathsf{VK}) \in \mathbb{Z}_q^{n\times 2m}$ and use it as an IBE public key to encrypt $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{2m}$, where $\mathbf{v}_i = \mathbf{F}\cdot\mathbf{z}_i \in \mathbb{Z}_q^{4n}$ is the syndrome of $\mathsf{sec}_i = \mathbf{z}_i \in \mathbb{Z}^{4m}$ for the matrix $\mathbf{F}$. Namely, compute $\mathbf{c}_{\mathbf{v}_i} \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m}$ as

$$\mathbf{c}_{\mathbf{v}_i} = (\mathbf{c}_1, \mathbf{c}_2) \quad = \quad (\mathbf{B}^T\cdot\mathbf{e}_0 + \mathbf{x}_1,\ \mathbf{G}_0^T\cdot\mathbf{e}_0 + \mathbf{x}_2 + \mathsf{bin}(\mathbf{v}_i)\cdot\lfloor q/2\rfloor) \quad (7.18)$$

for randomly chosen $\mathbf{e}_0 \hookleftarrow \chi^n, \mathbf{x}_1 \hookleftarrow \chi^m, \mathbf{x}_2 \hookleftarrow \chi^{2m}$. Notice that, as in the construction of [LNW15], the columns of $\mathbf{G}_0$ can be interpreted as public keys for the multi-bit version of the dual Regev encryption scheme.

2. Run the protocol in Section 7.3.3 to prove the knowledge of $\mathsf{id}_i \in \{0,1\}^\ell$, vectors $\mathbf{s}_i \in \mathbb{Z}^{2m}, \mathbf{d}_{i,1}, \mathbf{d}_{i,2} \in \mathbb{Z}^m, \mathbf{z}_i \in \mathbb{Z}^{4m}$ with infinity norm bound β; $\mathbf{e}_0 \in \mathbb{Z}^n$, $\mathbf{x}_1 \in \mathbb{Z}^m, \mathbf{x}_2 \in \mathbb{Z}^{2m}$ with infinity norm bound B and $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{2m}, \mathbf{w}_i \in \{0,1\}^m$, that satisfy (7.18) as well as

$$\mathbf{A}\cdot\mathbf{d}_{i,1} + \mathbf{A}_0\cdot\mathbf{d}_{i,2} + \sum_{j=1}^{\ell}(\mathsf{id}_i[j]\cdot\mathbf{d}_{i,2})\cdot\mathbf{A}_j - \mathbf{D}\cdot\mathbf{w}_i = \mathbf{u} \in \mathbb{Z}_q^n \quad (7.19)$$

and

$$\begin{cases} \mathbf{H}_{2n\times m}\cdot\mathbf{w}_i = \mathbf{D}_0\cdot\mathsf{bin}(\mathbf{v}_i) + \mathbf{D}_1\cdot\mathbf{s}_i \in \mathbb{Z}_q^{2n} \\ \mathbf{F}\cdot\mathbf{z}_i = \mathbf{H}_{4n\times 2m}\cdot\mathsf{bin}(\mathbf{v}_i) \in \mathbb{Z}_q^{4n}. \end{cases} \quad (7.20)$$

The protocol is repeated $t = \omega(\log n)$ times in parallel to achieve negligible soundness error, and then made non-interactive using the Fiat-Shamir heuristic [FS86] as a triple $\pi_K = (\{\mathsf{Comm}_{K,j}\}_{j=1}^t, \mathsf{Chall}_K, \{\mathsf{Resp}_{K,j}\}_{j=1}^t)$, where $\mathsf{Chall}_K = H(M, \mathsf{vk}, \mathbf{c}_{\mathbf{v}_i}, \{\mathsf{Comm}_{K,j}\}_{j=1}^t) \in \{1,2,3\}^t$

3. Compute a one-time signature $sig = \mathcal{S}(\mathsf{SK}, (\mathbf{c}_{\mathbf{v}_i}, \pi_K))$.

Output the signature that consists of

$$\Sigma = (\mathsf{VK}, \mathbf{c}_{\mathbf{v}_i}, \pi_K, sig). \quad (7.21)$$

Verify$(\mathcal{Y}, M, \Sigma)$**:** Parse the signature Σ as in (7.21). Then, return 1 if and only if: (i) $\mathcal{V}(\mathsf{VK}, (\mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{s}_i}, \mathbf{c}_{\mathsf{id}}, \pi_K), sig) = 1$; (ii) The proof π_K properly verifies.

Open$(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \Sigma)$**:** Parse $\mathcal{S}_{\mathsf{OA}}$ as $\mathbf{T}_{\mathbf{B}} \in \mathbb{Z}^{m\times m}$ and Σ as in (7.21).

1. Compute $\mathbf{G}_0 = H_0(\mathsf{VK}) \in \mathbb{Z}_q^{n\times 2m}$. Then, using $\mathbf{T}_{\mathbf{B}}$ to compute a small-norm matrix $\mathbf{E}_{0,\mathsf{VK}} \in \mathbb{Z}^{m\times 2m}$ such that $\mathbf{B}\cdot\mathbf{E}_{0,\mathsf{VK}} = \mathbf{G}_0 \bmod q$.
2. Using $\mathbf{E}_{0,\mathsf{VK}}$, decrypt $\mathbf{c}_{\mathbf{v}_i}$ to obtain a string $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{2m}$ (i.e., by computing $\lfloor(\mathbf{c}_2 - \mathbf{E}_{0,\mathsf{VK}}^T\cdot\mathbf{c}_1)/(q/2)\rceil$).
3. Determine whether the $\mathsf{bin}(\mathbf{v}) \in \{0,1\}^{2m}$ obtained at step 2 corresponds to a vector $\mathbf{v} = \mathbf{H}_{4n\times 2m}\cdot\mathsf{bin}(\mathbf{v}) \bmod q$ that appears in a record $\mathsf{transcript}_i = (\mathbf{v}, \mathsf{cert}_i, i, \mathsf{upk}[i], sig_i)$ of the database St_{trans} for some i. If so, output the corresponding i (and, optionally, $\mathsf{upk}[i]$). Otherwise, output $\perp$.

We remark that the scheme readily extends to provide a mechanism whereby the opening authority can efficiently prove that signatures were correctly opened at each opening operation. The difference between the dynamic group signature models suggested by Kiayias and Yung [KY06] and Bellare *et al.* [BSZ05] is that, in the latter, the opening authority (OA) must be able to convince a judge that the Open algorithm was run correctly. Here, such a mechanism can be realized using the techniques of public-key encryption with non-interactive opening [DHKT08]. Namely, since $\mathsf{bin}(\mathbf{v}_i)$ is encrypted using an IBE scheme for the identity vk, the OA can simply reveal the decryption matrix $\mathbf{E}_{0,\mathsf{VK}}$, that satisfies $\mathbf{B} \cdot \mathbf{E}_{0,\mathsf{vk}} = \mathbf{G}_0 \bmod q$ (which corresponds to the verification of a GPV signature) and allows the verifier to perform step 2 of the opening algorithm himself. The resulting construction is easily seen to satisfy the notion of opening soundness of Sakai *et al.* [SSE+12].

7.2.2 Efficiency and Correctness

Efficiency. The given dynamic group signature scheme can be implemented in polynomial time. The group public key has total bit-size $\mathcal{O}(\ell nm \log q) = \widetilde{\mathcal{O}}(\lambda^2) \cdot \log N_{\mathsf{gs}}$. The secret signing key of each user consists of a small constant number of low-norm vectors, and has bit-size $\widetilde{\mathcal{O}}(\lambda)$.

The size of each group signature is largely dominated by that of the non-interactive argument π_K, which is obtained from the Stern-like protocol of Section 7.3.3. Each round of the protocol has communication cost $\widetilde{\mathcal{O}}(m \cdot \log q) \cdot \log N_{\mathsf{gs}}$. Thus, the bit-size of π_K is $t \cdot \widetilde{\mathcal{O}}(m \cdot \log q) \cdot \log N_{\mathsf{gs}} = \widetilde{\mathcal{O}}(\lambda) \cdot \log N_{\mathsf{gs}}$. This is also the asymptotic bound on the size of the group signature.

Correctness. The correctness of algorithm $\mathsf{Verify}(\mathcal{Y}, M, \Sigma)$ follows from the facts that every certified group member is able to compute valid witness vectors satisfying equations (7.18), (7.19) and (7.20), and that the underlying argument system is perfectly complete. Moreover, the scheme parameters are chosen so that the GPV IBE [GPV08] is correct, which implies that algorithm $\mathsf{Open}(\mathcal{Y}, \mathcal{S}_{\mathsf{OA}}, M, \Sigma)$ is also correct.

7.2.3 Security Analysis

Due to the fact that the number of public matrices $\{\mathbf{A}_j\}_{j=0}^{\ell}$ is only logarithmic in $N_{\mathsf{gs}} = 2^\ell$ instead of being linear in the security parameter λ, the proof of security against misidentification attacks (as defined in Section 5.3) cannot rely on the security of our signature scheme in a modular manner. The reason is that, at each run of the Join protocol, the group manager maintains a state and, instead of choosing the ℓ-bit identifier id uniformly in $\{0,1\}^\ell$, it chooses an identifier that has not been used yet. Since $\ell \ll \lambda$ (given that $N_{\mathsf{gs}} = 2^\ell$ is polynomial in λ), we thus have to prove security from scratch. However, the strategy of the reduction is exactly the same as in the security proof of the signature scheme.

Theorem 7.7. *The scheme is secure against misidentification attacks under the* $\mathsf{SIS}_{n,2m,q,\beta'}$ *assumption, for* $\beta' = \mathcal{O}(\ell\sigma^2 m^{3/2})$.

Proof. We prove that any adversary $\mathcal{A}$ with non-negligible success probability ε implies an algorithm $\mathcal{B}$ solving the SIS problem in the random oracle model.

Let $\mathcal{A}$ be such a PPT adversary. We then build a PPT reduction $\mathcal{B}$ that uses the adversary $\mathcal{A}$ to solve $\mathsf{SIS}_{n,2m,q,\beta'}$: specifically, $\mathcal{B}$ takes as input $\bar{\mathbf{A}} = \left[\bar{\mathbf{A}}_1 | \bar{\mathbf{A}}_2\right] \in \mathbb{Z}_q^{n \times 2m}$, where $\bar{\mathbf{A}}_1, \bar{\mathbf{A}}_2 \in \mathbb{Z}_q^{n \times m}$, and finds $\mathbf{w} \in \Lambda_q^{\perp}(\bar{\mathbf{A}})$ with $0 < \|\mathbf{w}\| \leq \beta'$.

Initialization. Algorithm $\mathcal{B}$ first chooses a random $coin \hookleftarrow U(\{0,1,2\})$ as a guess for the kind of misidentification attack that $\mathcal{A}$ will mount. Also, $\mathcal{B}$ chooses a random ℓ-bit string $\mathsf{id}^{\dagger} \hookleftarrow (\{0,1\}^{\ell})$. In addition, $\mathcal{B}$ samples $i^{\star} \hookleftarrow ([1, Q_a])$.
Looking ahead, $coin = 0$ corresponds to the case where, after repeated executions of $\mathcal{A}$, the knowledge extractor of the proof system reveals witnesses containing a new identifier $\mathsf{id}^{\star} \in \{0,1\}^{\ell}$ that does not belong to any user in U^a. In this case, $\mathcal{B}$ will be able to exploit $\mathcal{A}$'s forgery when $\mathsf{id}^{\star} = \mathsf{id}^{\dagger}$. The case $coin = 1$ corresponds to $\mathcal{B}$'s expectation that the knowledge extractor will obtain the identifier $\mathsf{id}^{\star} = \mathsf{id}^{\dagger}$ of a group member in U^a (i.e., a group member that was legitimately introduced at the $i^{\star}$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query, for some $i^{\star} \in \{1, \ldots, Q_a\}$, where the identifier $\mathsf{id}^{\dagger}$ is used by $\mathcal{Q}_{\mathsf{a\text{-}join}}$), but $\mathsf{bin}(\mathbf{v}^{\star}) \in \{0,1\}^{2m}$ (which is encrypted in in $\mathbf{c}^{\star}_{\mathbf{v}_i}$ as part of the forgery $\Sigma^{\star}$) and the extracted $\mathbf{s}^{\star} \in \mathbb{Z}^{2m}$ are such that $\{0,1\}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}^{\star}) + \mathbf{D}_1 \cdot \mathbf{s}^{\star}) \in \{0,1\}^m$ does not match the string $\{0,1\}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^{\star}}) + \mathbf{D}_1 \cdot \mathbf{s}_{i^{\star}}) \in \{0,1\}^{2m}$ for which user $i^{\star}$ obtained a membership certificate at the $i^{\star}$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query. When $coin = 1$, the choice of $i^{\star}$ corresponds to a guess that the knowledge extractor will reveal an ℓ-bit identifier that coincides with the identifier $\mathsf{id}^{\dagger}$ assigned to the user introduced at the $i^{\star}$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query. The last case $coin = 2$ corresponds to $\mathcal{B}$'s expectation that decrypting $\mathbf{c}^{\star}_{\mathbf{v}_i}$ (which is part of $\Sigma^{\star}$) and running the knowledge extractor on $\mathcal{A}$ will uncover vectors $\mathsf{bin}(\mathbf{v}^{\star}) \in \{0,1\}^{2m}$, $\mathbf{w}^{\star} \in \{0,1\}^m$ and $\mathbf{s}^{\star} \in \mathbb{Z}^{2m}$ such that $\mathbf{w}^{\star} = \mathsf{bin}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}^{\star}) + \mathbf{D}_1 \cdot \mathbf{s}^{\star})$ and

$$\{0,1\}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}^{\star}) + \mathbf{D}_1 \cdot \mathbf{s}^{\star}) = \{0,1\}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^{\star}}) + \mathbf{D}_1 \cdot \mathbf{s}_{i^{\star}}) \tag{7.22}$$

but $(\mathsf{bin}(\mathbf{v}^{\star}), \mathbf{s}^{\star}) \neq (\mathsf{bin}(\mathbf{v}_{i^{\star}}), \mathbf{s}_{i^{\star}})$, where $\mathbf{v}_{i^{\star}} \in \mathbb{Z}_q^{4n}$ and $\mathbf{s}_{i^{\star}} \in \mathbb{Z}^{2m}$ are the vectors involved in the $i^{\star}$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query.
Depending on $coin \in \{0,1,2\}$, the group public key $\mathcal{Y}$ is generated using different methods.

• If $coin = 0$, algorithm $\mathcal{B}$ first randomly chooses $\mathsf{id}^{\dagger} \hookleftarrow (\{0,1\}^{\ell})$ as a guess for the ℓ-bit string that will be revealed by the knowledge extractor of the proof system after repeated executions of the adversary $\mathcal{A}$. Then, it runs $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ and a basis $\mathbf{T}_{\mathbf{C}}$ of $\Lambda_q^{\perp}(\mathbf{C})$ with $\|\widetilde{\mathbf{T}_{\mathbf{C}}}\| \leq \mathcal{O}(\sqrt{n \log q})$. Then, it chooses $\ell + 2$ matrices $\mathbf{Q}_0, \ldots, \mathbf{Q}_{\ell}, \mathbf{Q}_D \in \mathbb{Z}^{m \times m}$, each matrix having its columns sampled independently from $D_{\mathbb{Z}^m, \sigma}$. Then, $\mathcal{B}$ defines the matrices $\{\mathbf{A}_i\}_{i=0}^{\ell}$ as

$$\begin{cases} \mathbf{A}_0 = \bar{\mathbf{A}}_1 \cdot \mathbf{Q}_0 + (\sum_{i=1}^{\ell} \mathsf{id}^{\dagger}[i]) \cdot \mathbf{C} \\ \mathbf{A}_j = \bar{\mathbf{A}}_1 \cdot \mathbf{Q}_i + (-1)^{\mathsf{id}^{\dagger}[j]} \cdot \mathbf{C}, \quad \text{for } j \in [1, \ell]. \\ \mathbf{D} = \bar{\mathbf{A}}_1 \cdot \mathbf{Q}_D \end{cases}$$

It also defines $\mathbf{A} = \bar{\mathbf{A}}_1$. Next, it samples a vector $\mathbf{e}_u \hookleftarrow D^m_{\mathbb{Z},\sigma}$ and computes a syndrome $\mathbf{u} = \bar{\mathbf{A}}_1 \cdot \mathbf{e}_u \in \mathbb{Z}_q^n$. It picks $\mathbf{D}_0, \mathbf{D}_1 \hookleftarrow (\mathbb{Z}_q^{2n \times 2m})$ at random and also faithfully generates the GPV master key pair $(\mathbf{B}, \mathbf{T}_{\mathbf{B}})$ as in Step 3 of the real setup algorithm. The group public key $\mathcal{Y} = (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{B}, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{F}, \mathbf{u}, \mathcal{OTS}, H, H_0)$ is finally given to $\mathcal{A}$.

Note that, for each $\mathsf{id} \neq \mathsf{id}^\dagger$, we have

$$\begin{aligned}\mathbf{A}_{\mathsf{id}} &= \left[\ \bar{\mathbf{A}}_1 \ \middle|\ \mathbf{A}_0 + \textstyle\sum_{i=1}^{\ell} \mathsf{id}[i]\mathbf{A}_i \ \right] \\ &= \left[\ \bar{\mathbf{A}}_1 \ \middle|\ \bar{\mathbf{A}}_1 \cdot (\mathbf{Q}_0 + \textstyle\sum_{i=1}^{\ell} \mathsf{id}[i]\mathbf{Q}_i) + (\sum_{i=1}^{\ell} \mathsf{id}^\dagger[i] + (-1)^{\mathsf{id}^\dagger[i]}\mathsf{id}[i]) \cdot \mathbf{C} \ \right] \\ &= \left[\ \bar{\mathbf{A}}_1 \ \middle|\ \bar{\mathbf{A}}_1 + h_{\mathsf{id}} \cdot \mathbf{C} \ \right] \end{aligned} \tag{7.23}$$

where $h_{\mathsf{id}} \in [1, \ell]$ denotes the Hamming distance between the identifiers id and $\mathsf{id}^\dagger$. Since $q > \ell$, we have $h_{\mathsf{id}_j} \neq 0 \bmod q$ whenever $\mathsf{id}_j \neq \mathsf{id}^\dagger$, so that algorithm $\mathcal{B}$ is able to compute (see [ABB10, Se. 4.2], using the basis $\mathbf{T}_{\mathbf{C}}$ of $\Lambda_q^\perp(\mathbf{C})$ and the refined GPVSample of Lemma 3.5) a basis $\mathbf{T}_{\mathsf{id}}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathsf{id}})$ with $\|\widetilde{\mathbf{T}_{\mathsf{id}}}\| \leq \Omega(\sqrt{n \log q \log n})$. In contrast, algorithm $\mathcal{B}$ lacks a trapdoor for $\mathbf{A}_{\mathsf{id}^\dagger}$ as the latter only depends on $\mathbf{A}$ and $\{\mathbf{Q}_k\}_{k=0}^{\ell}$. Observe that, since the columns of the matrices $\{\mathbf{Q}_k\}_{k=0}^{\ell}$ are sampled from $D_{\mathbb{Z}^m,\sigma}$, the matrices $\mathbf{A}_0, \ldots, \mathbf{A}_\ell$ are within statistical distance $2^{-\Omega(m)}$ of $U(\mathbb{Z}_q^{n\times m})$.

• If $coin = 1$, algorithm $\mathcal{B}$ sets up $\mathcal{Y}$ by defining $\mathbf{D} = \bar{\mathbf{A}}$. Initially, $\mathcal{B}$ chooses $Q_a - 1$ distinct strings $\mathsf{id}_1, \ldots, \mathsf{id}_{i^\star-1}, \mathsf{id}_{i^\star+1}, \ldots, \mathsf{id}_{Q_a} \in \{0,1\}^\ell$ such that, for each $i \in [1, Q_a]\backslash\{i^\star\}$, id_i will be embedded in the membership certificate returned in the i-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query. Let also $\mathsf{id}^\dagger = \mathsf{id}_{i^\star}$ be the ℓ-bit identifier that will be used in the $i^\star$-th query. The reduction $\mathcal{B}$ picks random $h_0, h_1, \ldots, h_\ell \in \mathbb{Z}_q$ under the constraints

$$\begin{aligned} h_{\mathsf{id}^\dagger} = h_0 + \sum_{j=1}^{\ell} \mathsf{id}^\dagger[j] \cdot h_j &= 0 \bmod q \\ h_{\mathsf{id}_i} = h_0 + \sum_{j=1}^{\ell} \mathsf{id}_i[j] \cdot h_j &\neq 0 \bmod q \qquad\qquad i \in \{1, \ldots, Q_a\} \setminus \{i^\dagger\} \end{aligned}$$

Next, $\mathcal{B}$ runs $(\mathbf{C}, \mathbf{T}_{\mathbf{C}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, $(\mathbf{D}_1, \mathbf{T}_{\mathbf{D}_1}) \leftarrow \mathsf{TrapGen}(1^{2n}, 1^{2m}, q)$ so as to obtain statistically random matrices $\mathbf{C} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{D}_1 \in \mathbb{Z}_q^{2n\times 2m}$ together with trapdoors $\mathbf{T}_{\mathbf{C}} \in \mathbb{Z}^{m\times m}$, $\mathbf{T}_{\mathbf{D}_1} \in \mathbb{Z}^{2m\times 2m}$ consisting of short bases of $\Lambda_q^\perp(\mathbf{C})$ and $\Lambda_q^\perp(\mathbf{D}_1)$, respectively. Then, $\mathcal{B}$ picks a random $\mathbf{D}_0 \hookleftarrow (\mathbb{Z}_q^{2n\times 2m})$ and re-randomizes $\mathbf{D} = \bar{\mathbf{A}}_1 \in \mathbb{Z}_q^{n\times m}$ using Gaussian matrices $\mathbf{S}, \mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_\ell \hookleftarrow \mathbb{Z}^{m\times m}$ whose columns are sampled from the distribution $D_{\mathbb{Z}^m,\sigma}$. Namely, from $\mathbf{D} = \bar{\mathbf{A}}_1$, $\mathcal{B}$ defines

$$\begin{aligned} \mathbf{A} &= \bar{\mathbf{A}}_1 \cdot \mathbf{S} \\ \mathbf{A}_0 &= \bar{\mathbf{A}}_1 \cdot \mathbf{S}_0 + h_0 \cdot \mathbf{C} \\ \mathbf{A}_j &= \bar{\mathbf{A}}_1 \cdot \mathbf{S}_j + h_j \cdot \mathbf{C} \qquad\qquad \forall j \in \{1, \ldots, \ell\}. \end{aligned} \tag{7.24}$$

As part of the generation of $\mathcal{Y}$, the vector $\mathbf{u} \in \mathbb{Z}_q^n$ is obtained by picking short discrete Gaussian vectors $\mathbf{d}_{i^\star,1}, \mathbf{d}_{i^\star,2} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ and computing

$$\mathbf{u} = [\mathbf{A} \ \mid \ \mathbf{A}_0 + \sum_{j=1}^{\ell} \mathsf{id}^\dagger[j]\mathbf{A}_j] \cdot \begin{bmatrix} \mathbf{d}_{i^\star,1} \\ \mathbf{d}_{i^\star,2} \end{bmatrix} - \mathbf{D} \cdot \mathsf{bin}(\mathbf{c}_M), \tag{7.25}$$

where $\mathbf{c}_M \hookleftarrow (\mathbb{Z}_q^{2n})$ is a randomly chosen vector. Observe that, since $\mathbf{A}$ is statistically uniform over $\mathbb{Z}_q^{n\times m}$ and $\mathbf{d}_{i^\star,1} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, the distribution of $\mathbf{u}$ is statistically close to $U(\mathbb{Z}_q^n)$.

• If $coin = 2$, $\mathcal{B}$ picks $\bar{\mathbf{A}}' \hookleftarrow (\mathbb{Z}_q^{n\times 2m})$ and a random matrix $\mathbf{Q} \hookleftarrow \mathbb{Z}^{2m\times 2m}$ whose columns are sampled from $D_{\mathbb{Z}^{2m},\sigma}$. These are used to define

$$\mathbf{D}_0 = \left[\frac{\bar{\mathbf{A}}}{\bar{\mathbf{A}}'}\right] \in \mathbb{Z}_q^{2n\times 2m},$$

and $\mathbf{D}_1 = \mathbf{D}_0 \cdot \mathbf{Q} \bmod q$, which is statistically close to $U(\mathbb{Z}_q^{2n\times 2m})$. All other components of $\mathcal{Y}$ are obtained by faithfully running the setup algorithm.

For each value of $coin \in \{0,1,2\}$, the group public key

$$\mathcal{Y} = (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{B}, \mathbf{D}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{F}, \mathbf{u}, \mathcal{OTS}, H, H_0)$$

has a distribution which is statistically close to that of the real scheme and $\mathcal{Y}$ is given to $\mathcal{A}$.

Queries. The reduction $\mathcal{B}$ starts interacting with the adversary $\mathcal{A}$ and the way it handles $\mathcal{A}$'s queries to the $\mathcal{Q}_{\mathsf{a\text{-}join}}$ oracle depends on the value of $coin \in \{0,1,2\}$.

• If $coin = 0$, answers $\mathcal{Q}_{\mathsf{a\text{-}join}}$-queries as follows. When $\mathcal{A}$ triggers an execution of the joining protocol, it chooses a syndrome $\mathbf{v}_i \in \mathbb{Z}_q^n$. To answer the query, $\mathcal{B}$ chooses a fresh ℓ-bit identifier $\mathsf{id}_i \in \{0,1\}^\ell$ such that $\mathsf{id}_i \neq \mathsf{id}^\dagger$. If $\mathcal{A}$ also provides a correct signature sig_i such that $\mathrm{Verify}_{\mathsf{upk}[i]}(\mathbf{v}_i, sig_i) = 1$, $\mathcal{B}$ samples $\mathbf{s}_i \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and uses the trapdoor $\mathbf{T_C}$ to compute a short vector $\mathbf{d}_i = [\mathbf{d}_{i,1}^T \mid \mathbf{d}_{i,2}^T]^T \in \mathbb{Z}^{2m}$ such that

$$\mathbf{A}_{\mathsf{id}_i} \cdot \left[\frac{\mathbf{d}_{i,1}}{\mathbf{d}_{i,2}}\right] = \mathbf{u} + \mathbf{D} \cdot \{0,1\}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i), \tag{7.26}$$

where $\mathbf{A}_{\mathsf{id}_i} \in \mathbb{Z}_q^{n\times 2m}$ is the matrix in (7.23). Note that $\mathcal{B}$ is able to compute such a vector using the SampleRight algorithm of [ABB10] (since the Hamming distance h_{id_i} between id_i and $\mathsf{id}^\star$ is non-zero). The membership certificate $\mathsf{cert}_i = (\mathsf{id}_i, \mathbf{d}_i, \mathbf{s}_i)$ is then returned to $\mathcal{A}$.

• If $coin = 1$, algorithm $\mathcal{B}$ responds each $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query depending on the index $i \in \{1,\ldots,Q_a\}$ of the query. Specifically, we distinguish two cases.

- If $i \neq i^\star$, $\mathcal{B}$ proceeds as in the previous case. Namely, it recalls the ℓ-bit identifier $\mathsf{id}_i \in \{0,1\}^\ell$ (for which $\mathsf{id}_i \neq \mathsf{id}^\dagger$) that was chosen in the setup phase and samples a short vector $\mathbf{s}_i \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$. If $\mathcal{A}$ also provides a correct signature sig_i such that $\mathrm{Verify}_{\mathsf{upk}[i]}(\mathbf{v}_i, sig_i) = 1$, generates a membership certificate cert_i for $\mathcal{A}$ as in the case $coin = 0$. Note that

$$\begin{aligned} \mathbf{A}_{\mathsf{id}_i} &= \left[\begin{array}{c|c} \bar{\mathbf{A}} \cdot \mathbf{S} & \bar{\mathbf{A}} \cdot (\mathbf{S}_0 + \sum_{j=1}^{\ell} \mathsf{id}_i[j]\mathbf{S}_j) + h_{\mathsf{id}_i}\mathbf{C} \end{array}\right] \\ &= \left[\begin{array}{c|c} \bar{\mathbf{A}} \cdot \mathbf{S} & \bar{\mathbf{A}} + h_{\mathsf{id}_i} \cdot \mathbf{C} \end{array}\right] \end{aligned} \tag{7.27}$$

  Since $h_{\mathsf{id}_i} \neq 0$, $\mathcal{B}$ can use the trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^\perp(\mathbf{C})$ to compute a short vector $\mathbf{d}_i = [\mathbf{d}_{i,1}^T \mid \mathbf{d}_{i,2}^T]^T \in \mathbb{Z}^{2m}$ such that

$$\mathbf{A}_{\mathsf{id}_i} \cdot \left[\frac{\mathbf{d}_{i,1}}{\mathbf{d}_{i,2}}\right] = \mathbf{u} + \mathbf{D} \cdot \{0,1\}(\mathbf{D}_0 \cdot (\mathsf{bin}(\mathbf{v}_i) + \mathbf{D}_1 \cdot \mathbf{s}_i),$$

  where $\mathbf{v}_i \in \mathbb{Z}_q^{4n}$ is the syndrome chosen by $\mathcal{A}$ at step 1 of the joining protocol.

- If $i = i^\star$, $\mathcal{B}$ undertakes to generate a membership certificate $\mathsf{cert}_{i^\star}$ for the ℓ-bit identifier $\mathsf{id}^\dagger \in \{0,1\}^\ell$ that was chosen at the outset of the game. To this end, $\mathcal{B}$ has to compute $\mathsf{cert}_{i^\star}$ without using the trapdoor $\mathbf{T_C}$ since the matrix $\mathbf{A}_{\mathsf{id}^\dagger}$ does no longer depend on $\mathbf{C}$ in (7.27). This can be done by recalling the vector $\mathbf{d}_{i^\star,1}, \mathbf{d}_{i^\star,2} \in \mathbb{Z}^m$ and $\mathbf{c}_M \in \mathbb{Z}_q^{2n}$ that were used to define $\mathbf{u} \in \mathbb{Z}_q^n$ in (7.25) and using $\mathbf{T}_{\mathbf{D}_1}$. If $\mathcal{A}$ provides a correct signature $sig_{i^\star}$ such that $\mathrm{Verify}_{\mathsf{upk}[i^\star]}(\mathbf{v}_{i^\star}, sig_{i^\star}) = 1$, $\mathcal{B}$ uses the trapdoor $\mathbf{T}_{\mathbf{D}_1}$ of $\Lambda_q^\perp(\mathbf{D}_1)$ to sample a short vector $\mathbf{s}_{i^\star} \in \mathbb{Z}^{2m}$ of $D_{\Lambda_q^{\mathbf{c}_{i^\star}}(\mathbf{D}_1),\sigma}$, where $\mathbf{c}_{i^\star} = \mathbf{c}_M - \mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^\star}) \bmod q$, satisfying

$$\mathbf{D}_1 \cdot \mathbf{s}_{i^\star} = \mathbf{c}_M - \mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^\star}) \mod q,$$

before returning $\mathsf{cert}_{i^\star} = (\mathsf{id}^\dagger, \mathbf{d}_{i^\star} = [\mathbf{d}_{i^\star,1}^T \mid \mathbf{d}_{i^\star,2}^T]^T, \mathbf{s}_{i^\star})$ to $\mathcal{A}$. From the definition of $\mathbf{u} \in \mathbb{Z}_q^n$ (7.25), it is easy to see that $\mathsf{cert}_{i^\star} = (\mathsf{id}^\dagger, \mathbf{d}_{i^\star}, \mathbf{s}_{i^\star})$ forms a valid membership certificate for any membership secret $\mathbf{z}_{i^\star} \in \mathbb{Z}^{4m}$ corresponding to the syndrome $\mathbf{v}_{i^\star} = \mathbf{F} \cdot \mathbf{z}_{i^\star} \bmod q$.

Regardless of the value of $coin$, queries to the random oracle H are handled by returning a uniformly chosen value in $\{1,2,3\}^t$. 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. Queries to the random oracle H_0 are answered in the usual way, by returning a uniformly random value in the appropriate range.

Forgery. When $\mathcal{A}$ halts, it outputs a signature $\Sigma^\star = (\mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \pi_K^\star, sig^\star)$ on some message $M^\star$. At this point, $\mathcal{B}$ uses the trapdoor $\mathbf{T_B}$ to decrypt $\mathbf{c}_{\mathbf{v}_i}^\star$ and obtain an m-bit string $\mathsf{bin}(\mathbf{v}^\star) \in \{0,1\}^m$.

If we parse the proof $\pi_K^\star$ as $(\{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t, \mathsf{Chall}_K^\star, \{\mathsf{Resp}_{K,j}^\star\}_{j=1}^t)$, the adversary $\mathcal{A}$ must have invoked the random oracle H on the input $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ with high probability. Otherwise, the probability that $\mathsf{Chall}_K^\star = H(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ is negligible (at most 3^{-t}).

It comes that, with probability at least $\varepsilon' := \varepsilon - 3^{-t}$, $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ coincides with the $\kappa^\star$-th random oracle query for some $\kappa^\star \leq Q_H$.

At this stage, the reduction $\mathcal{B}$ runs the adversary $\mathcal{A}$ up to $32 \cdot Q_H/(\varepsilon - 3^{-t})$ times with the *same* random tape and input as in the initial run. All queries are answered as previously with one difference in the treatment of random oracle queries. Namely, the first $\kappa^\star - 1$ random oracle 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 $\mathsf{Chall}_1, \ldots, \mathsf{Chall}_{\kappa^\star-1}$ as in the first run. This implies that the $\kappa^\star$-th query will involve exactly the same tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ as in the first run. However, from the $\kappa^\star$-th query onwards, $\mathcal{A}$ obtains fresh random oracle values $\mathsf{Chall}'_{\kappa^\star}, \ldots, \mathsf{Chall}'_{Q_H}$ at each new execution. The Improved Forking Lemma of Brickell *et al.* [BPVY00] guarantees that, with probability at least $1/2$, $\mathcal{B}$ can obtain a 3-fork involving the same tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}_i}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ with pairwise distinct answers $\mathsf{Chall}_{\kappa^\star}^{(1)}, \mathsf{Chall}_{\kappa^\star}^{(2)}, \mathsf{Chall}_{\kappa^\star}^{(3)} \in \{1,2,3\}^t$. With probability $1 - (7/9)^t$ it can be shown that there exists an index $j \in \{1, \ldots, t\}$ for which the j-th bits of $\mathsf{Chall}_{\kappa^\star}^{(1)}, \mathsf{Chall}_{\kappa^\star}^{(2)}, \mathsf{Chall}_{\kappa^\star}^{(3)}$ are $(\mathsf{Chall}_{\kappa^\star,j}^{(1)}, \mathsf{Chall}_{\kappa^\star,j}^{(2)}, \mathsf{Chall}_{\kappa^\star,j}^{(3)}) = (1,2,3)$. From the corresponding responses $(\mathsf{Resp}_{K,j}^\star{}^{(1)}, \mathsf{Resp}_{K,j}^\star{}^{(2)}, \mathsf{Resp}_{K,j}^\star{}^{(3)})$, $\mathcal{B}$ is able to extract witnesses

$(\mathbf{d}_1^\star, \mathbf{d}_2^\star) \in \mathbb{Z}^m \times \mathbb{Z}^m$, $\mathsf{id}^\star \in \{0,1\}^\ell$ and $\mathbf{w}^\star \in \{0,1\}^m$ from the proof of knowledge $\pi_K^\star$ such that

$$\begin{aligned} \mathbf{A}_{\mathsf{id}^\star} \cdot \begin{bmatrix} \mathbf{d}_1^\star \\ \hline \mathbf{d}_2^\star \end{bmatrix} &= \mathbf{u} + \mathbf{D} \cdot \mathbf{w}^\star \\ \mathbf{w}^\star &= \{0,1\}(\mathbf{D}_0 \cdot (\mathsf{bin}(\mathbf{v}^\star) + \mathbf{D}_1 \cdot \mathbf{s}^\star), \end{aligned}$$

At this point, $\mathcal{B}$ aborts and declares failure in the following situations:

- $coin = 0$ but $\mathsf{id}^\star \in \{0,1\}^\ell$ is recycled from some output of the $\mathcal{Q}_{\mathsf{a\text{-}join}}$ oracle.
- $coin = 0$ and $\mathsf{id}^\star \neq \mathsf{id}^\dagger$.
- $coin = 1$ but $\mathsf{id}^\star \in \{0,1\}^\ell$ never appeared in a membership certificate returned by the $\mathcal{Q}_{\mathsf{a\text{-}join}}$ oracle.
- $coin = 1$ and $\mathsf{id}^\star \in \{0,1\}^\ell$ belongs to some user in U^a, but this user is not the one introduced at the $i^\star$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$-query (i.e., $i^\star \neq i^\dagger$ and $\mathsf{id}^\star \neq \mathsf{id}^\dagger$).
- $coin = 1$ and the knowledge extractor revealed vectors $\mathsf{bin}(\mathbf{v}^\star) \in \{0,1\}^{2m}$ and $\mathbf{s}^\star \in \mathbb{Z}^{2m}$ satisfying the collision (7.22), where $\mathsf{bin}(\mathbf{v}_{i^\star})$ and $\mathbf{s}_{i^\star}$ are the vectors involved in the $i^\star$-th $\mathcal{Q}_{\mathsf{a\text{-}join}}$ query.
- $coin = 2$ and the knowledge extraction yields vectors $\mathsf{bin}(\mathbf{v}^\star) \in \{0,1\}^{2m}$ and $\mathbf{s}^\star \in \mathbb{Z}^{2m}$ such that the collision (7.22) does not occur.

We call fail the event that one of the above situations occurs. Given that the choices of $coin \hookleftarrow (\{0,1,2\})$ and $i^\star \hookleftarrow ([1, Q_a])$ are completely independent of $\mathcal{A}$'s view, the choice of $coin$ is correct with probability $1/3$. If $coin = 0$, $\mathcal{B}$'s choice of $\mathsf{id}^\dagger \hookleftarrow (\{0,1\}^\ell)$ is correct with probability $1/(N_{\mathsf{gs}} - Q_a) \geq 1/N_{\mathsf{gs}}$ and, when $coin = 1$, $\mathcal{B}$'s correctly guesses $i^\star \in [1, Q_a]$ with probability $1/Q_a$. We find

$$\Pr[\neg\mathsf{fail}] \geq \frac{1}{3 \cdot \max(N_{\mathsf{gs}}, Q_a)} = \frac{1}{3 \cdot N_{\mathsf{gs}}}.$$

Assuming that fail does not occur, $\mathcal{B}$ can solve the problem instance as follows.

- If $coin = 0$, we have $\mathsf{id}^\star = \mathsf{id}^\dagger$ and $\mathcal{B}$ knows a short vector $\mathbf{e}_u \in \mathbb{Z}^m$ such that $\mathbf{u} = \bar{\mathbf{A}}_1 \cdot \mathbf{e}_u \bmod q$. Hence, it can obtain a short integer vector

$$\mathbf{h} = \mathbf{d}_1^\star + (\mathbf{Q}_0 + \sum_{i=1}^{\ell} \mathsf{id}^\dagger[i]\mathbf{Q}_i) \cdot \mathbf{d}_2^\star - \mathbf{Q}_D \cdot \mathsf{bin}(\mathbf{v}^\star) - \mathbf{e}_u \in \mathbb{Z}^m$$

  such that $\bar{\mathbf{A}}_1 \cdot \mathbf{h} = \mathbf{0}^m \bmod q$. Moreover, we have $\mathbf{h} \neq \mathbf{0}^m$ w.h.p. since the syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ statistically hides $\mathbf{e}_u \in \mathbb{Z}^m$ in $\Lambda_q^{\mathbf{u}}(\bar{\mathbf{A}}_1)$. Finally, the norm of $\mathbf{h}$ is at most $\|\mathbf{h}\|_2 \leq (\ell+1)\sigma^2 m^{3/2} + \sigma m^{1/2}(m+2)$. This implies that $(\mathbf{h}^T \mid \mathbf{0}^m)^T$ is a short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$ and solves the initial SIS instance.

- If $coin = 1$, the extracted witnesses $(\mathbf{d}_1^\star, \mathbf{d}_2^\star, \mathbf{s}^\star, \mathsf{id}^\star)$ and the decrypted $\mathsf{bin}(\mathbf{v}^\star)$ satisfy $\mathsf{id}^\star = \mathsf{id}^\dagger$,

$$\mathbf{w}^\star = \mathsf{bin}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}^\star) + \mathbf{D}_1 \cdot \mathbf{s}^\star) \neq \mathsf{bin}(\mathbf{D}_0 \cdot \mathsf{bin}(\mathbf{v}_{i^\star}) + \mathbf{D}_1 \cdot \mathbf{s}_{i^\star}) = \mathbf{w}_{i^\star}$$

(since $\neg\mathsf{fail}$ implies that the collision (7.22) did not occur if $coin = 1$) and

$$\left[\begin{array}{c|c|c|c|c|c} \mathbf{A} & \mathbf{A}_0 & \mathbf{A}_1 & \dots & \mathbf{A}_\ell & -\mathbf{D} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{d}_1^\star \\ \hline \mathbf{d}_2^\star \\ \hline \mathsf{id}^\dagger[1]\mathbf{d}_2^\star \\ \hline \vdots \\ \hline \mathsf{id}^\dagger[\ell]\mathbf{d}_2^\star \\ \hline \mathbf{w}^\star \end{array}\right] = \mathbf{u} \bmod q. \tag{7.28}$$

Since $\mathcal{B}$ already knew short vectors $(\mathbf{d}_{i^\star,1}, \mathbf{d}_{i^\star,2}, \mathbf{w}_{i^\star}) \in \mathbb{Z}^m \times \mathbb{Z}^m \times \mathbb{Z}^m$ such that

$$\left[\begin{array}{c|c|c|c|c|c} \mathbf{A} & \mathbf{A}_0 & \mathbf{A}_1 & \dots & \mathbf{A}_\ell & -\mathbf{D} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{d}_{i^\star,1}^\star \\ \hline \mathbf{d}_{i^\star,2}^\star \\ \hline \mathsf{id}^\dagger[1]\mathbf{d}_{i^\star,2}^\star \\ \hline \vdots \\ \hline \mathsf{id}^\dagger[\ell]\mathbf{d}_{i^\star,2}^\star \\ \hline \mathbf{w}_{i^\star} \end{array}\right] = \mathbf{u} \bmod q, \tag{7.29}$$

by subtracting (7.29) from (7.28), we find that

$$\mathbf{h} = \mathbf{S} \cdot (\mathbf{d}_1^\star - \mathbf{d}_{i^\star,1}) + (\mathbf{S}_0 + \sum_{j=1}^{\ell} \mathsf{id}^\dagger[j]\mathbf{S}_j) \cdot (\mathbf{d}_2^\star - \mathbf{d}_{i^\star,2}) + (\mathbf{w}^\star - \mathbf{w}_{i^\star}) \tag{7.30}$$

is a small-norm vector $\mathbf{h} \in \mathbb{Z}^m$ satisfying $\bar{\mathbf{A}}_1 \cdot \mathbf{h} = \mathbf{0} \bmod q$. We claim that $\mathbf{h} \neq \mathbf{0}$ with high probability. Indeed, we know that $\mathbf{w}^\star \neq \mathbf{w}_{i^\star}$ if $\neg\mathsf{fail}$ occurs. This implies that the last term of (7.30) is non-zero, which rules out that $(\mathbf{d}_1^\star, \mathbf{d}_2^\star) = (\mathbf{d}_{i^\star,1}, \mathbf{d}_{i^\star,2})$. Since the columns of $\mathbf{S}$ and $\{\mathbf{S}_j\}_{j=0}^\ell$ have a lot of entropy conditionally on $\mathcal{Y}$, this implies that we can only have $\mathbf{h} = \mathbf{0}^m$ with negligible probability. Furthermore, the norm of $\mathbf{h}$ can be bounded by $\|\mathbf{h}\|_2 \leq 4\sigma^2 m^{3/2}(\ell + 2) + 2m^{1/2}$, so that $(\mathbf{h}^T \mid \mathbf{0}^m)^T$ solves the original SIS instance.

- If $coin = 2$, $\mathcal{B}$ is done as well since the collision (7.22) directly provides a vector

$$\mathbf{h} = \mathsf{bin}(\mathbf{v}^\star) - \mathsf{bin}(\mathbf{v}_i^\star) + \mathbf{Q} \cdot (\mathbf{s}^\star - \mathbf{s}_i^\star) \in \mathbb{Z}^{2m}$$

of $\Lambda_q^\perp(\mathbf{D}_0)$ (which is also in the lattice $\Lambda_q^\perp(\bar{\mathbf{A}})$ by construction) and has norm $\|\mathbf{h}\|_2 \leq 2(\sigma^2(2m)^{3/2} + (2m)^{1/2})$. Moreover, $\mathbf{h} \in \mathbb{Z}^{2m}$ is non-zero with overwhelming probability given that $\mathsf{bin}(\mathbf{v}^\star) \neq \mathsf{bin}(\mathbf{v}_i^\star)$ and the large amount of entropy retained by the columns $\mathbf{Q} \in \mathbb{Z}^{2m \times 2m}$ given $\mathbf{D}_1 = \mathbf{D}_0 \cdot \mathbf{Q}$.

□

Theorem 7.8. *The scheme is secure against framing attacks under the* $\mathsf{SIS}_{4n,4m,q,\beta''}$ *assumption, where* $\beta'' = 4\sigma\sqrt{m}$.

Proof. Let us assume that a PPT adversary $\mathcal{A}$ can create a forgery $(M^\star, \Sigma^\star)$ that opens to some honest user $i \in U^b$ who did not sign $M^\star$. In the random oracle model, we give a reduction $\mathcal{B}$ that uses $\mathcal{A}$ to solve an instance of the $\mathsf{SIS}_{4n,4m,q,\beta''}$ problem: $\mathcal{B}$ takes as input $\bar{\mathbf{A}} \in \mathbb{Z}_q^{4n \times 4m}$ and finds a non-zero short vector $\mathbf{w} \in \Lambda_q^\perp(\bar{\mathbf{A}})$.
Algorithm $\mathcal{B}$ generates the group public key $\mathcal{Y}$ by faithfully running the real setup algorithm with the sole difference that, at step 2 of Setup, $\mathcal{B}$ defines $\mathbf{F} = \bar{\mathbf{A}} \in \mathbb{Z}_q^{4n \times 4m}$. However, the distribution of $\mathcal{Y}$ is as in the real scheme. As a result of having generated $\mathcal{Y}$ itself, $\mathcal{B}$ knows $\mathcal{S}_{\mathsf{GM}} = \mathbf{T_A}$ and $\mathcal{S}_{\mathsf{OA}} = \mathbf{T_B}$. The adversary $\mathcal{B}$ is run on input of the group public key

$$\mathcal{Y} \coloneqq \Big(\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell},\ \mathbf{B},\ \mathbf{D},\ \mathbf{D}_0,\ \mathbf{D}_1,\ \mathbf{F} = \bar{\mathbf{A}},\ \mathbf{u},\ \Pi^{\mathsf{OTS}},\ H,\ H_0)\Big).$$

If $\mathcal{A}$ chooses to corrupt the group manager or the opening authority during the game, $\mathcal{B}$ is able to reveal $\mathcal{S}_{\mathsf{GM}} = \mathbf{T_A}$ and $\mathcal{S}_{\mathsf{OA}} = \mathbf{T_B}$. Then, $\mathcal{B}$ starts interacting with $\mathcal{A}$ as follows.

- Q_{keyGM}-queries: If $\mathcal{A}$ decides to corrupt the group manager, $\mathcal{B}$ hands the secret key $\mathcal{S}_{\mathsf{GM}} = \mathbf{T_A}$ to $\mathcal{A}$.

- $Q_{\mathsf{b\text{-}join}}$-queries: At any time $\mathcal{A}$ can act as a corrupted group manager and introduce a new honest user i in the group by invoking the $Q_{\mathsf{b\text{-}join}}$ oracle. At each $Q_{\mathsf{b\text{-}join}}$-query, $\mathcal{B}$ faithfully runs $\mathsf{J}_{\mathsf{user}}$ on behalf of the honest user in an execution of Join protocol.

- Q_{pub}-queries: These can be answered as in the real game, by having the simulator return $\mathcal{Y}$.

- Q_{sig}-queries: When the adversary $\mathcal{A}$ requests user $i \in U^b$ to sign a message M, $\mathcal{B}$ first generates a one-time key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(n)$ to compute $\mathbf{G}_0 = H_0(\mathsf{VK}) \in \mathbb{Z}_q^{n \times 2m}$. Next, $\mathcal{B}$ recalls the vector $\mathbf{z}_i \in \mathbb{Z}^{4m}$ that was chosen to define the syndrome $\mathbf{v}_i = \mathbf{F} \cdot \mathbf{z}_i$ at step 1 of the Join protocol as well as the identifier $\mathsf{id}_i \in \{0,1\}^\ell$ and the short vectors $(\mathbf{d}_{i,1}, \mathbf{d}_{i,2}, \mathbf{s}_i)$ that were supplied by $\mathcal{A}$ in an earlier $Q_{\mathsf{b\text{-}join}}$-query. It faithfully computes a signature by IBE-encrypting $\mathsf{bin}(\mathbf{v}_i) \in \{0,1\}^{2m}$ and using $(\mathbf{d}_{i,1}, \mathbf{d}_{i,2}, \mathbf{s}_i, \mathbf{z}_i, \mathbf{s}_i, \mathsf{id}_i)$ to compute a witness indistinguishable proof $\pi_K = (\{\mathsf{Comm}_{K,j}\}_{j=1}^t, \mathsf{Chall}_K, \{\mathsf{Resp}_{K,j}\}_{j=1}^t)$. Finally, $\mathcal{B}$ computes a one-time signature $sig = \mathcal{S}(\mathsf{SK}, (\mathbf{c}_{\mathbf{v}_i}, \pi_K))$ and returns the signature $\Sigma = (\mathsf{VK}, \mathbf{c}_{\mathbf{v}_i}, \pi_K, sig)$ to $\mathcal{A}$.

When $\mathcal{A}$ halts, it outputs a signature $\Sigma^\star = (\mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \pi_K^\star, sig^\star)$ for some message $M^\star$, which opens to $i^\star \in U^b$ although user $i^\star$ did not sign the message $M^\star$ at any time. Since $(M^\star, \Sigma^\star)$ supposedly frames user $i^\star$, the opening of $\Sigma^\star$ must reveal the m-bit string $\mathsf{bin}(\mathbf{v}_{i^\star}) \in \{0,1\}^m$. We note that the reduction $\mathcal{B}$ has recollection of a short vector $\mathbf{z}_{i^\star} \in \mathbb{Z}^{4m}$ (of norm $\|\mathbf{z}_{i^\star}\| < 2\sigma\sqrt{m}$) such that $\mathbf{v}_{i^\star} = \mathbf{F} \cdot \mathbf{z}_{i^\star} \bmod q$ which it chose when running $\mathsf{J}_{\mathsf{user}}$ on behalf of user $i^\star$ when this user was introduced in the group. Hence, $\mathcal{B}$ would be able to solve its given SIS instance if it had another short vector $\mathbf{z}' \in \mathbb{Z}^{4m}$ satisfying $\mathbf{v}_{i^\star} = \mathbf{F} \cdot \mathbf{z}' \bmod q$. To compute such a vector, $\mathcal{B}$ proceeds by replaying the adversary $\mathcal{A}$ sufficiently many times and applying the Improved Forking Lemma of Brickell *et al.* [BPVY00].
If we parse $\pi_K^\star$ as $(\{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t, \mathsf{Chall}_K^\star, \{\mathsf{Resp}_{K,j}^\star\}_{j=1}^t)$, with high probability, $\mathcal{A}$ must have queried H on the input $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$. Otherwise, we would only have $\mathsf{Chall}_K^\star = H(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ with negligible probability 3^{-t}. It comes that, with probability at least $\varepsilon' \coloneqq \varepsilon - 3^{-t}$, the tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_{\mathbf{v}}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ was

the input of the $\kappa^\star$-th random oracle query for some index $\kappa^\star \leq Q_H$.
At this point, the reduction $\mathcal{B}$ runs the adversary $\mathcal{A}$ up to $32 \cdot Q_H/(\varepsilon - 3^{-t})$ times with the *same* random tape and input as in the first run. All queries are answered as previously with one difference in the way to handle H-queries. Namely, the first $\kappa^\star - 1$ H-queries – which are the same as in the first execution since $\mathcal{A}$ is run with the same random tape – obtain the same answers $\mathsf{Chall}_1, \ldots, \mathsf{Chall}_{\kappa^\star-1}$ as in the original run. This implies that the $\kappa^\star$-th query will also involve exactly the same tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_\mathbf{v}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ as in the original run. From the $\kappa^\star$-th query forward, however, the adversary $\mathcal{A}$ obtains fresh random oracle outputs $\mathsf{Chall}'_{\kappa^\star}, \ldots, \mathsf{Chall}'_{Q_H}$ at each new execution. The Improved Forking Lemma of [BPVY00] ensures that, with probability $> 1/2$, $\mathcal{B}$ obtains a 3-fork involving the tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_\mathbf{v}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$ of the initial run and with pairwise distinct answers $\mathsf{Chall}_{\kappa^\star}^{(1)}, \mathsf{Chall}_{\kappa^\star}^{(2)}, \mathsf{Chall}_{\kappa^\star}^{(3)} \in \{1,2,3\}^t$. Since the forgeries of the 3-fork all correspond to the tuple $(M^\star, \mathsf{VK}^\star, \mathbf{c}_\mathbf{v}^\star, \{\mathsf{Comm}_{K,j}^\star\}_{j=1}^t)$, they open to the same m-bit string $\mathsf{bin}(\mathbf{v}_{i^\star}) \in \{0,1\}^m$ and which is uniquely determined by $\mathbf{c}_\mathbf{v}^\star$. In turn, this implies that the three forgeries all reveal the same $\mathsf{bin}(\mathbf{v}_{i^\star})$ at the second step of Open. With probability $1 - (7/9)^t$ it can be shown that there exists $j \in \{1, \ldots, t\}$ such that the j-th bits of $\mathsf{Chall}_{\kappa^\star}^{(1)}, \mathsf{Chall}_{\kappa^\star}^{(2)}, \mathsf{Chall}_{\kappa^\star}^{(3)}$ are $(\mathsf{Chall}_{\kappa^\star,j}^{(1)}, \mathsf{Chall}_{\kappa^\star,j}^{(2)}, \mathsf{Chall}_{\kappa^\star,j}^{(3)}) = (1,2,3)$. From the corresponding responses $(\mathsf{Resp}_{K,j}^\star{}^{(1)}, \mathsf{Resp}_{K,j}^\star{}^{(2)}, \mathsf{Resp}_{K,j}^\star{}^{(3)})$, $\mathcal{B}$ is able to extract a short vector $\mathbf{z}' \in \mathbb{Z}^{4m}$ such that $\mathbf{v}_{i^\star} = \mathbf{F} \cdot \mathbf{z}' \bmod q$.
Due to the statistical witness indistinguishability of the Stern-like proof of knowledge which is used to generate signature, with overwhelming probability, we have $\mathbf{z}' \neq \mathbf{z}_{i^\star}$. Indeed, from the adversary's view, the distribution of $\mathbf{z}_{i^\star}$ is $D_{\Lambda_q^{\mathbf{v}_{i^\star}}(\mathbf{F}),\sigma}$, which means that it has at least n bits of min-entropy. Hence, the difference $\mathbf{h} = \mathbf{z}' - \mathbf{z}_{i^\star} \in \mathbb{Z}^{4m}$ is a suitably short non-zero vector of $\Lambda_q^\perp(\bar{\mathbf{A}})$. □

Theorem 7.9. *In the random oracle model, the scheme provides CCA-anonymity if the* $\mathsf{LWE}_{n,q,\chi}$ *assumption holds and if* Π^{OTS} *is a strongly unforgeable one-time signature.*

Proof. We proceed as in [LNW15] and prove the result via a sequence of games which are computationally indistinguishable. The first game consists of the real anonymity experiment which is parameterized by a bit $d \in \{0,1\}$ that determines the challenger's choice in the challenge phase. The last game is the same regardless of whether $d = 0$ or $d = 1$. It follows that, under the stated assumptions, no PPT adversary can distinguish $\mathbf{Exp}_\mathcal{A}^{\text{anon}-0}$ from $\mathbf{Exp}_\mathcal{A}^{\text{anon}-1}$ with noticeable advantage.

Game$^{(d)}$ **0:** This is the real anonymity experiment $\mathbf{Exp}_\mathcal{A}^{\text{anon}-d}(\lambda)$ as described in Definition 5.5. More precisely, the challenger starts by running the $\mathsf{Setup}(1^\lambda, 1^{N_{\mathsf{gs}}})$ algorithm to obtain $(\mathcal{Y}, \mathcal{S}_{\mathsf{GM}} = \mathbf{T}_\mathbf{A} \in \mathbb{Z}^{m\times m}, \mathcal{S}_{\mathsf{OA}} = \mathbf{T}_\mathbf{B} \in \mathbb{Z}^{m\times m})$ along with state information St. The challenger next hands the public parameters $\mathcal{Y}$ and the group manager key $\mathcal{S}_{\mathsf{GM}}$ to the adversary $\mathcal{A}$. On the following adversary signature opening queries on signatures $\Sigma = (\mathsf{vk}, \mathbf{c}_{\mathbf{v}_d}, \pi_K, sig)$, the challenger uses the opening authority key $\mathbf{T}_\mathbf{A} \in \mathbb{Z}^{m\times m}$ he possesses to decrypt the GPV encryption of the signer identity $\mathbf{c}_{\mathbf{v}_d} \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{2m}$. At some point, the adversary $\mathcal{A}$ requests a challenge by outputting a target message $M^\star \in \{0,1\}^*$ and two user key pairs

$$\left(\mathsf{sec}_i^\star = \mathbf{z}_i^\star \in \mathbb{Z}^{4m}, \mathsf{cert}_i^\star \in (\mathsf{id}_i^\star, \mathbf{d}_i^\star, \mathbf{s}_i^\star) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^{2m}\right)_{i\in\{0,1\}}$$

which must be valid and distinct (otherwise, the challenger aborts the experiment). This challenge query is answered by having the challenger return a signature of the target message under the identity id_d: namely, this challenge signature is computed as $\Sigma^\star = (\mathsf{vk}^\star, \mathbf{c}^\star_{\mathbf{v}_d}, \pi^\star_K, sig^\star) \leftarrow \mathsf{Sign}(\mathcal{Y}, \mathsf{cert}^\star_d, \mathsf{sec}^\star_d, M^\star)$ for the given parameter d of the Game. Finally, $\mathcal{A}$ outputs a bit $d' \in \{0,1\}$ which is also the experiment's output.

Game$^{(d)}$ 1: In this experiment, we slightly change Game$^{(d)}$ 0 as follows. At the outset of the game, the challenger generates the one-time signature key pair $(\mathsf{vk}^\star, \mathsf{sk}^\star)$ that will be used in the challenge phase. During the game, if the adversary $\mathcal{A}$ requests the opening of a valid signature $\Sigma = (\mathsf{vk}, \mathbf{c}_{\mathbf{v}_i}, \pi_K, sig)$ where $\mathsf{vk} = \mathsf{vk}^\star$, the challenger returns a random bit and aborts. However, this event F_1 would contradict the strong unforgeability of the one-time signature Π^{OTS}. Indeed, before the challenge phase $\mathsf{vk}^\star$ is independent of $\mathcal{A}$'s view and the probability that $\mathsf{vk}^\star$ shows up in $\mathcal{A}$'s queries is negligible. After seeing the challenge signature $\Sigma^\star$, if $\mathcal{A}$ comes up with a valid signature $\Sigma = (\mathsf{vk}, \mathbf{c}_{\mathbf{v}_i}, \pi_K, sig)$ such that $\mathsf{vk} = \mathsf{vk}^\star$, then sig is a forged one-time signature, which defeats the strong unforgeability of Π^{OTS}. Therefore the probability $\Pr[F_1]$ that the challenger aborts in this experiment is negligible. From here on, we thus assume that $\mathcal{A}$'s opening queries for valid signatures do not include $\mathsf{vk}^\star$.

Game$^{(d)}$ 2: In this game, we program the random oracle H_0 in the following way: at the beginning of the game, we choose a uniformly random matrix $\mathbf{G}_0^\star \hookleftarrow (\mathbb{Z}_q^{n\times 2m})$ and set $H_0(\mathsf{vk}^\star) = \mathbf{G}_0^\star$. From the adversary's view, the distribution of $\mathbf{G}_0^\star$ is statistically close to the one in the real attack game, as in [GPV08]. As for other queries, for each fresh H_0-queries on vk, the challenger samples small-norm matrices $\mathbf{E}_{0,\mathsf{vk}} \hookleftarrow D^{2m}_{\mathbb{Z}^m,\sigma}$ and programs the oracle such that $H_0(\mathsf{vk}) = \mathbf{B}\cdot\mathbf{E}_{0,\mathsf{vk}} \bmod q$. The chosen matrices $\mathbf{E}_{0,\mathsf{vk}}$ are retained for later use. Note that the values of $H_0(\mathsf{vk})$ are statistically close to the uniform. For any query involving a previously queried vk, the challenger consistently returns the previously stored images. The adversary's view remains the same as in Game$^{(d)}$ 1, analogously to the security proof of the GPV IBE [GPV08].

Game$^{(d)}$ 3: Here, we will change the behavior of the opening algorithm. Namely, at each fresh oracle query, we still store the matrices $\mathbf{E}_{0,\mathsf{vk}} \in \mathbb{Z}_q^{m\times 2m}$ and, at the beginning of the game, the challenger samples an uniformly random $\mathbf{B}^\star \in \mathbb{Z}_q^{n\times m}$ that is later used in place of $\mathbf{B}$ to answer H_0-queries. To answer the adversary's queries of the opening of a signature $\Sigma = (\mathsf{vk}, \mathbf{c}_{\mathbf{v}_i},\ \pi_K, sig)$, the challenger recalls the small-norm matrices $\mathbf{E}_{0,\mathsf{vk}}$ which were defined when $\mathcal{A}$ first queried $H_0(\mathsf{vk})$. These matrices are used as "decryption matrices" to open Σ for the corresponding $\mathbf{G}_0 = H_0(\mathsf{vk}) \in \mathbb{Z}_q^{n\times 2m}$. For similar reasons as in the security proof of [GPV08], the distribution of $\mathbf{G}_0$ is statistically close to the uniform, which implies that Game$^{(d)}$ 2 and Game$^{(d)}$ 3 are statistically indistinguishable.

Game$^{(d)}$ 4: Instead of faithfully generating the NIZKPoK π_K of Section 7.3.3, the challenger simulates the proof without using the witness (note that this is possible since the HVZK property of the underlying proof system is preserved under parallel repetitions). This is done by running the simulator for the underlying interactive protocol for each $j \in \{1,\ldots,t\}$, and then programming the random oracle H accordingly. The challenge signature $\Sigma^\star = (\mathsf{vk}^\star, \mathbf{c}^\star_{\mathbf{v}_d}, \pi^\star_K, sig^\star)$ is statistically close to

the challenge signature of the previous game, because the proof system is statistically zero-knowledge as stated in Lemma 4.2. Consequently, $\mathsf{Game}^{(d)}$ 3 and $\mathsf{Game}^{(d)}$ 4 are indistinguishable.

$\mathsf{Game}^{(d)}$ 5: In this game, we modify the generation of the challenge ciphertext $\mathbf{c}^{\star}_{\mathbf{v}_d}$. Instead of using the real encryption algorithm of the GPV IBE to compute $\mathbf{c}^{\star}_{\mathbf{v}_d}$ as the encryption of $\mathbf{v}^{\star}_d = \mathbf{F} \cdot \mathbf{z}_d \in \mathbb{Z}_q^{4n}$, we return truly random ciphertexts. In other words, we let

$$\mathbf{c}^{\star}_{\mathbf{v}_d} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 + \mathsf{bin}(\mathbf{v}^{\star}_d)\lfloor q/2 \rfloor \end{pmatrix},$$

where $\mathbf{r}_1 \hookleftarrow (\mathbb{Z}_q^m)$, $\mathbf{r}_2 \hookleftarrow (\mathbb{Z}_q^{2m})$ are uniformly random. The hardness of the decisional $\mathsf{LWE}_{n,q,\chi}$ problem implies that $\mathbf{c}^{\star}_{\mathbf{v}_d}$ in extsfGame 4 and extsfGame 5 are computationally indistinguishable. If $\mathcal{A}$ can distinguish between these two games, it can furthermore distinguish

$$\begin{pmatrix} \mathbf{B}^T \\ \hline \mathbf{G}_0^{\star T} \end{pmatrix} \mathbf{e}_0 + \begin{pmatrix} \mathbf{x}_1 \\ \hline \mathbf{x}_2 \end{pmatrix} \text{ from } \begin{pmatrix} \mathbf{r}_1 \\ \hline \mathbf{r}_2 \end{pmatrix},$$

which would break the decisional $\mathsf{LWE}_{n,q,\chi}$ assumption.

Therefore, $\mathsf{Game}^{(d)}$ 4 and $\mathsf{Game}^{(d)}$ 5 are computationally indistinguishable.

Game 6: We finally make a conceptual modification on the previous game. Namely we sample uniformly random $\mathbf{r}'_1 \hookleftarrow (\mathbb{Z}_q^m)$, $\mathbf{r}'_2 \hookleftarrow (\mathbb{Z}_q^{2m})$ and assign

$$\mathbf{c}^{\star}_{\mathbf{v}_d} = \begin{pmatrix} \mathbf{r}'_1 \\ \mathbf{r}'_2 \end{pmatrix}.$$

Clearly, the distribution of $\mathbf{c}^{\star}_{\mathbf{v}_i}$ has not changed since $\mathsf{Game}^{(d)}$ 5. Since Game 6 does no longer depend on the challenger's bit $d \in \{0, 1\}$, the result follows. □

7.3 Subprotocols for Stern-like Argument

7.3.1 Proving the Consistency of Commitments

The argument system used in our protocol for signing a committed value in Section 7.1.3 can be summarized as follows.

Common Input: Matrices $\{\mathbf{D}_k \in \mathbb{Z}_q^{2n \times 2m}\}_{k=0}^{N}$; $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$; $\mathbf{G}_1 \in \mathbb{Z}_q^{n \times 2m}$;

vectors $\mathbf{c}_{\mathfrak{m}} \in \mathbb{Z}_q^{2n}$; $\{\mathbf{c}_{k,1} \in \mathbb{Z}_q^m\}_{k=1}^{N}$; $\{\mathbf{c}_{k,2} \in \mathbb{Z}_q^{2m}\}_{k=1}^{N}$; $\mathbf{c}_{\mathbf{s}',1} \in \mathbb{Z}_q^m$; $\mathbf{c}_{\mathbf{s}',2} \in \mathbb{Z}_q^{2m}$.

Prover's Input: $\mathfrak{m} = (\mathfrak{m}_1^T \| \ldots \| \mathfrak{m}_N^T)^T \in \mathsf{CorEnc}(mN)$;

$\{\mathbf{s}_k \in [-B, B]^n,\ \mathbf{e}_{k,1} \in [-B, B]^m;\ \mathbf{e}_{k,2} \in [-B, B]^{2m}\}_{k=1}^{N}$; $\mathbf{s}_0 \in [-B, B]^n$;

$\mathbf{e}_{0,1} \in [-B, B]^m$; $\mathbf{e}_{0,2} \in [-B, B]^{2m}$; $\mathbf{s}' \in [-(p-1), (p-1)]^{2m}$

Prover's Goal: Convince the verifier in ZK that:

$$\begin{cases} \mathbf{c}_{\mathfrak{m}} = \mathbf{D}_0 \cdot \mathbf{s}' + \sum_{k=1}^{N} \mathbf{D}_k \cdot \mathfrak{m}_k \bmod q; \\ \mathbf{c}_{\mathbf{s}',1} = \mathbf{B}^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,1} \bmod q; \;\; \mathbf{c}_{\mathbf{s}',2} = \mathbf{G}_1^T \cdot \mathbf{s}_0 + \mathbf{e}_{0,2} + \lfloor q/p \rfloor \cdot \mathbf{s}' \bmod q; \\ \forall k \in [N] : \mathbf{c}_{k,1} = \mathbf{B}^T \cdot \mathbf{s}_k + \mathbf{e}_{k,1}; \;\; \mathbf{c}_{k,2} = \mathbf{G}_1^T \cdot \mathbf{s}_k + \mathbf{e}_{k,2} + \lfloor q/2 \rfloor \cdot \mathfrak{m}_k. \end{cases} \tag{7.31}$$

We will show that the above argument system can be obtained from the one in Section 4.3.2. We proceed in two steps.

Step 1: *Transforming the equations in (7.31) into a unified one of the form* $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$*, where* $\|\mathbf{x}\|_\infty = 1$ *and* $\mathbf{x} \in \mathsf{VALID}$ *- a "specially-designed" set.*

To do so, we first form the following vectors and matrices:

$$\begin{cases} \mathbf{x}_1 = \left(\mathbf{s}_0^T \| \mathbf{e}_{0,1}^T \| \mathbf{e}_{0,2}^T \| \mathbf{s}_1^T \| \mathbf{e}_{1,1}^T \| \mathbf{e}_{1,2}^T \| \ldots \| \mathbf{s}_N^T \| \mathbf{e}_{N,1}^T \| \mathbf{e}_{N,2}^T\right)^T \in [-B, B]^{(n+3m)(N+1)}; \\ \mathbf{v} = \left(\mathbf{c}_{\mathfrak{m}}^T \| \mathbf{c}_{\mathbf{s}',1}^T \| \mathbf{c}_{\mathbf{s}',2}^T \| \mathbf{c}_{1,1}^T \| \mathbf{c}_{1,2}^T \| \ldots \| \mathbf{c}_{N,1}^T \| \mathbf{c}_{N,2}^T\right)^T \in \mathbb{Z}_q^{2n+3m(N+1)}; \\ \mathbf{P}_1 = \left(\begin{array}{c|c} \mathbf{B}^T & \mathbf{I}_{3m} \\ \hline \mathbf{G}_1^T & \end{array}\right); \quad \mathbf{Q}_2 = \left(\begin{array}{c} \mathbf{0} \\ \hline \lfloor \frac{q}{2} \rfloor \mathbf{I}_{2m} \end{array}\right); \quad \mathbf{Q}_p = \left(\begin{array}{c} \mathbf{0} \\ \hline \lfloor \frac{q}{p} \rfloor \mathbf{I}_{2m} \end{array}\right) \\ \mathbf{M}_1 = \left(\begin{array}{c} \mathbf{0} \\ \hline \mathbf{P}_1 \\ \quad \mathbf{P}_1 \\ \quad\quad \ddots \\ \quad\quad\quad \mathbf{P}_1 \end{array}\right); \quad \mathbf{M}_2 = \left(\begin{array}{c} \mathbf{D}_1 \| \ldots \| \mathbf{D}_N \\ \hline \mathbf{0} \\ \hline \mathbf{Q}_2 \\ \quad \ddots \\ \quad\quad \mathbf{Q}_2 \end{array}\right); \quad \mathbf{M}_3 = \left(\begin{array}{c} \mathbf{D}_0 \\ \hline \mathbf{Q}_p \\ \hline \\ \mathbf{0} \\ \\ \end{array}\right). \end{cases}$$

We then observe that (7.31) can be rewritten as:

$$\mathbf{M}_1 \cdot \mathbf{x}_1 + \mathbf{M}_2 \cdot \mathfrak{m} + \mathbf{M}_3 \cdot \mathbf{s}' = \mathbf{v} \in \mathbb{Z}_q^D, \tag{7.32}$$

where $D = 2n + 3m(N+1)$. Now we employ the techniques from Section 4.3.2 to convert (7.32) into the form $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$. Specifically, if we let:

$$\begin{cases} \mathsf{DecExt}_{(n+3m)(N+1),B}(\mathbf{x}_1) \to \hat{\mathbf{x}}_1 \in \mathsf{B}^3_{(n+3m)(N+1)\delta_B}; \\ \mathbf{M}_1' = \mathbf{M}_1 \cdot \widehat{\mathbf{K}}_{(n+3m)(N+1),B} \in \mathbb{Z}_q^{D \times 3(n+3m)(N+1)\delta_B}; \\ \mathsf{DecExt}_{2m,p-1}(\mathbf{s}') \to \hat{\mathbf{s}} \in \mathsf{B}^3_{2m\delta_{p-1}}; \;\; \mathbf{M}_3' = \mathbf{M}_3 \cdot \widehat{\mathbf{K}}_{2m,p-1} \in \mathbb{Z}_q^{D \times 6m\delta_{p-1}}, \end{cases}$$

$L = 3(n+3m)(N+1)\delta_B + 2mN + 6m\delta_{p-1}$, and $\mathbf{P} = [\mathbf{M}_1' | \mathbf{M}_2 | \mathbf{M}_3'] \in \mathbb{Z}_q^{D \times L}$, and $\mathbf{x} = \left(\hat{\mathbf{x}}_1^T \| \mathfrak{m}^T \| \hat{\mathbf{s}}^T\right)^T$, then we will obtain the desired equation:

$$\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q.$$

Having performed the above unification, we now define VALID as the set of all vectors $\mathbf{t} \in \{-1, 0, 1\}^L$ of the form $\mathbf{t} = \left(\mathbf{t}_1^T \| \mathbf{t}_2^T \| \mathbf{t}_3^T\right)^T$, where $\mathbf{t}_1 \in \mathsf{B}^3_{(n+3m)(N+1)\delta_B}$, $\mathbf{t}_2 \in \mathsf{CorEnc}(mN)$, and $\mathbf{t}_3 \in \mathsf{B}^3_{2m\delta_{p-1}}$. Note that $\mathbf{x} \in \mathsf{VALID}$.

Step 2: *Specifying the set* $\mathcal{S}$ *and permutations of* L *elements* $\{T_\pi : \pi \in \mathcal{S}\}$ *for which the conditions in (4.3) hold.*

- Define $\mathcal{S} := \mathcal{S}_{3(n+3m)(N+1)\delta_B} \times \{0,1\}^{mN} \times \mathcal{S}_{6m\delta_{p-1}}$.

- For $\pi = (\pi_1, \mathbf{b}, \pi_3) \in \mathcal{S}$, and for vector $\mathbf{w} = \left(\mathbf{w}_1^T \| \mathbf{w}_2^T \| \mathbf{w}_3^T\right)^T \in \mathbb{Z}_q^L$, where $\mathbf{w}_1 \in \mathbb{Z}_q^{3(n+3m)(N+1)\delta_B}$, $\mathbf{w}_2 \in \mathbb{Z}_q^{2mN}$, $\mathbf{w}_3 \in \mathbb{Z}_q^{6m\delta_{p-1}}$, we define:

$$T_\pi = \left(\pi_1(\mathbf{w}_1)^T \| E_{\mathbf{b}}(\mathbf{w}_2)^T \| \pi_3(\mathbf{w}_3)^T\right)^T.$$

By inspection, it can be seen that the properties in (4.3) are satisfied, as desired. As a result, we can obtain the required argument system by running the protocol in Section 4.3.2 with common input $(\mathbf{P}, \mathbf{v})$ and prover's input $\mathbf{x}$.

7.3.2 Proving the Possession of a Signature on a Committed Value

We now describe how to derive the protocol for proving the possession of a signature on a committed value, that is used in Section 7.1.3.

Common Input: Matrices $\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D} \in \mathbb{Z}_q^{n\times m}$; $\{\mathbf{D}_k \in \mathbb{Z}_q^{2n\times 2m}\}_{k=0}^{N}$;$\mathbf{B} \in \mathbb{Z}_q^{n\times m}$; $\mathbf{G}_1 \in \mathbb{Z}_q^{n\times 2m}$; $\mathbf{G}_0 \in \mathbb{Z}_q^{n\times \ell}$;
vectors $\{\mathbf{c}_{k,1}\}_{k=1}^{N}, \mathbf{c}_{\tau,1}, \mathbf{c}_{\mathbf{v},1}, \mathbf{c}_{s,1} \in \mathbb{Z}_q^m$; $\{\mathbf{c}_{k,2}\}_{k=1}^{N}, \mathbf{c}_{\mathbf{v},2}, \mathbf{c}_{s,2} \in \mathbb{Z}_q^{2m}$; $\mathbf{c}_{\tau,2} \in \mathbb{Z}_q^{\ell}$; $\mathbf{u} \in \mathbb{Z}_q^n$.

Prover's Input: $\mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}$, where $\mathbf{v}_1, \mathbf{v}_2 \in [-\beta, \beta]^m$ and $\beta = \sigma \cdot \omega(\log m)$ - the infinity norm bound of signatures; $\tau \in \{0,1\}^{\ell}$; $\mathbf{s} \in [-(p-1), (p-1)]^{2m}$;
$\mathfrak{m} = (\mathfrak{m}_1^T \| \ldots \| \mathfrak{m}_N^T)^T \in \mathsf{CorEnc}(mN)$; $\{\mathbf{s}_k\}_{k=1}^{N}, \mathbf{s}_{\mathbf{v}}, \mathbf{s}_0, \mathbf{s}_\tau \in [-B, B]^n$;
$\{\mathbf{e}_{k,1}\}_{k=1}^{N}, \mathbf{e}_{\mathbf{v},1}, \mathbf{e}_{0,1}, \mathbf{e}_{\tau,1} \in [-B, B]^m$; $\{\mathbf{e}_{k,2}\}_{k=1}^{N}, \mathbf{e}_{0,2}, \mathbf{e}_{\mathbf{v},2} \in [-B, B]^{2m}$;
$\mathbf{e}_{\tau,2} \in [-B, B]^{\ell}$.

Prover's Goal: Convince the verifier in ZK that:

$$\mathbf{A}\cdot\mathbf{v}_1 + \mathbf{A}_0\cdot\mathbf{v}_2 + \sum_{i=1}^{\ell}\mathbf{A}_i\cdot\tau[i]\mathbf{v}_2 - \mathbf{D}\cdot\mathsf{bin}(\mathbf{D}_0\cdot\mathbf{s} + \sum_{k=1}^{N}\mathbf{D}_i\cdot\mathfrak{m}_k) = \mathbf{u} \bmod q, \tag{7.33}$$

and that (modulo q)

$$\begin{cases} \forall k \in [N] : \mathbf{c}_{k,1} = \mathbf{B}^T\cdot\mathbf{s}_k + \mathbf{e}_{k,1}; \mathbf{c}_{k,2} = \mathbf{G}_1^T\cdot\mathbf{s}_k + \mathbf{e}_{k,2} + \lfloor q/2\rfloor\cdot\mathfrak{m}_k; \\ \mathbf{c}_{\mathbf{v},1} = \mathbf{B}^T\cdot\mathbf{s}_{\mathbf{v}} + \mathbf{e}_{\mathbf{v},1}; \\ \mathbf{c}_{\mathbf{v},2} = \mathbf{G}_1^T\cdot\mathbf{s}_{\mathbf{v}} + \mathbf{e}_{\mathbf{v},2} + \lfloor\frac{q}{p}\rfloor\cdot\mathbf{v} \\ \qquad = \mathbf{G}_1^T\cdot\mathbf{s}_{\mathbf{v}} + \mathbf{e}_{\mathbf{v},2} + \begin{pmatrix} \lfloor\frac{q}{p}\rfloor\mathbf{I}_m \\ \mathbf{0} \end{pmatrix}\cdot\mathbf{v}_1 + \begin{pmatrix} \mathbf{0} \\ \lfloor\frac{q}{p}\rfloor\mathbf{I}_m \end{pmatrix}\cdot\mathbf{v}_2; \\ \mathbf{c}_{\mathbf{s},1} = \mathbf{B}^T\cdot\mathbf{s}_0 + \mathbf{e}_{0,1};\ \ \mathbf{c}_{\mathbf{s},2} = \mathbf{G}_1^T\cdot\mathbf{s}_0 + \mathbf{e}_{0,2} + \lfloor q/p\rfloor\cdot\mathbf{s}; \\ \mathbf{c}_{\tau,1} = \mathbf{B}^T\cdot\mathbf{s}_\tau + \mathbf{e}_{\tau,1};\ \mathbf{c}_{\tau,2} = \mathbf{G}_0^T\cdot\mathbf{s}_\tau + \mathbf{e}_{\tau,2} + \lfloor q/2\rfloor\cdot\tau. \end{cases} \tag{7.34}$$

We proceed in two steps.

Step 1: *Transforming the equations in (7.33) and (7.34) into a unified one of the form* $\mathbf{P}\cdot\mathbf{x} = \mathbf{c} \bmod q$, *where* $\|\mathbf{x}\|_\infty = 1$ *and* $\mathbf{x} \in \mathsf{VALID}$ *- a "specially-designed" set.*

Note that, if we let $\mathbf{y} = \mathsf{bin}(\mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_i \cdot \mathfrak{m}_k) \in \{0,1\}^m$, then we have $\mathbf{H}_{2n\times m} \cdot \mathbf{y} = \mathbf{D}_0 \cdot \mathbf{s} + \sum_{k=1}^{N} \mathbf{D}_i \cdot \mathfrak{m}_k \bmod q$, and (7.33) can be equivalently written as:

$$\begin{pmatrix} \mathbf{A} \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{v}_1 + \begin{pmatrix} \mathbf{A}_0 \\ \mathbf{0} \end{pmatrix} \cdot \mathbf{v}_2 \;+\; \sum_{i=1}^{\ell} \begin{pmatrix} \mathbf{A}_i \\ \mathbf{0} \end{pmatrix} \cdot \tau[i]\mathbf{v}_2 + \begin{pmatrix} \mathbf{0} \\ \mathbf{D}_0 \end{pmatrix} \cdot \mathbf{s} + \begin{pmatrix} -\mathbf{D} \\ -\mathbf{H}_{2n\times m} \end{pmatrix} \cdot \mathbf{y} + \begin{pmatrix} \mathbf{0} \\ \mathbf{D}_1 | \ldots | \mathbf{D}_N \end{pmatrix} \cdot \mathfrak{m} = \begin{pmatrix} \mathbf{u} \\ \mathbf{0}^{2n} \end{pmatrix} \bmod q.$$

Next, we use linear algebra to combine this equation and (7.34) into (modulo q):

$$\mathbf{F}\cdot\mathbf{v}_1 + \mathbf{F}_0\cdot\mathbf{v}_2 + \sum_{i=1}^{\ell} \mathbf{F}_i \cdot \tau[i]\mathbf{v}_2 + \mathbf{M}_1\cdot\tau + \mathbf{M}_2\cdot\mathbf{y} + \mathbf{M}_3\cdot\mathfrak{m} + \mathbf{M}_4\cdot\mathbf{s} + \mathbf{M}_5\cdot\mathbf{e} = \mathbf{c}, \tag{7.35}$$

where, for dimensions $D = \ell + 3n + 7m + 3mN$ and $L_0 = D + nN$,

- Matrices $\mathbf{F}, \mathbf{F}_0, \mathbf{F}_1, \ldots, \mathbf{F}_\ell \in \mathbb{Z}_q^{D\times m}$, $\mathbf{M}_1 \in \mathbb{Z}_q^{D\times \ell}$, $\mathbf{M}_2 \in \mathbb{Z}_q^{D\times m}$, $\mathbf{M}_3 \in \mathbb{Z}_q^{D\times 2mN}$, $\mathbf{M}_4 \in \mathbb{Z}_q^{D\times 2m}$, $\mathbf{M}_5 \in \mathbb{Z}_q^{D\times L_0}$ and vector $\mathbf{c} \in \mathbb{Z}_q^D$ are built from the public input.
- Vector $\mathbf{e} = (\mathbf{s}_1^T \| \ldots \| \mathbf{s}_N^T \| \mathbf{s}_{\mathbf{v}}^T \| \mathbf{s}_0^T \| \mathbf{s}_\tau^T \| \mathbf{e}_{1,1}^T \| \ldots \| \mathbf{e}_{N,1}^T \| \mathbf{e}_{\mathbf{v},1}^T \| \mathbf{e}_{0,1}^T \| \mathbf{e}_{\tau,1}^T \| \| \mathbf{e}_{1,2}^T \| \ldots \| \mathbf{e}_{N,2}^T \| \mathbf{e}_{0,2}^T \| \mathbf{e}_{\mathbf{v},2}^T \| \mathbf{e}_{\tau,2}^T)^T \in [-B, B]^{L_0}$.

Now we further transform (7.35) using the techniques from Section 4.3.2. Specifically, we form the following:

$$\begin{cases} \mathsf{DecExt}_{m,\beta}(\mathbf{v}_1) \to \hat{\mathbf{v}}_1 \in \mathsf{B}^3_{m\delta_\beta};\ \mathsf{DecExt}_{m,\beta}(\mathbf{v}_2) \to \hat{\mathbf{v}}_2 \in \mathsf{B}^3_{m\delta_\beta}; \\ \mathbf{F}' = [\mathbf{F}\cdot\widehat{\mathbf{K}}_{m,\beta} | \mathbf{F}_0\cdot\widehat{\mathbf{K}}_{m,\beta} | \mathbf{F}_1\cdot\widehat{\mathbf{K}}_{m,\beta} | \ldots | \mathbf{F}_\ell\cdot\widehat{\mathbf{K}}_{m,\beta} | \mathbf{0}^{D\times 3m\delta_\beta\ell}] \in \mathbb{Z}_q^{D\times 3m\delta_\beta(2\ell+2)}; \\ \mathsf{Ext}_{2\ell}(\tau) \to \hat{\tau} = (\tau[1], \ldots, \tau[\ell], \ldots, \tau[2\ell])^T \in \mathsf{B}^2_\ell;\ \mathbf{M}'_1 = [\mathbf{M}_1 | \mathbf{0}^{D\times\ell}] \in \mathbb{Z}_q^{D\times 2\ell}; \\ \mathsf{Ext}_{2m}(\mathbf{y}) \to \hat{\mathbf{y}} \in \mathsf{B}^2_m;\ \mathbf{M}'_2 = [\mathbf{M}_2 | \mathbf{0}^{D\times m}] \in \mathbb{Z}_q^{D\times 2m}; \\ \mathsf{DecExt}_{2m,p-1}(\mathbf{s}) \to \hat{\mathbf{s}} \in \mathsf{B}^3_{2m\delta_{p-1}};\ \mathbf{M}'_4 = \mathbf{M}_4\cdot\widehat{\mathbf{K}}_{2m,p-1} \in \mathbb{Z}_q^{D\times 6m\delta_{p-1}}; \\ \mathsf{DecExt}_{L_0,B}(\mathbf{e}) \to \hat{\mathbf{e}} \in \mathsf{B}^3_{L_0\delta_B};\ \mathbf{M}'_5 = \mathbf{M}_5\cdot\widehat{\mathbf{K}}_{L_0,B} \in \mathbb{Z}_q^{D\times 3L_0\delta_B}. \end{cases}$$

Now, let $L = 3m\delta_\beta(2\ell+2) + 2\ell + 2m + 2mN + 6m\delta_{p-1} + 3L_0\delta_B$, and construct matrix $\mathbf{P} = [\mathbf{F}' | \mathbf{M}'_1 | \mathbf{M}'_2 | \mathbf{M}_3 | \mathbf{M}'_4 | \mathbf{M}'_5] \in \mathbb{Z}_q^{D\times L}$ and vector

$$\mathbf{x} = (\hat{\mathbf{v}}_1^T \| \hat{\mathbf{v}}_2^T \| \tau[1]\hat{\mathbf{v}}_2^T \| \ldots \| \tau[\ell]\hat{\mathbf{v}}_2^T \| \ldots \| \tau[2\ell]\hat{\mathbf{v}}_2^T \| \hat{\tau}^T \| \hat{\mathbf{y}}^T \| \mathfrak{m}^T \| \hat{\mathbf{s}}^T \| \hat{\mathbf{e}}^T)^T,$$

then we will obtain the equation $\mathbf{P}\cdot\mathbf{x} = \mathbf{c} \bmod q$.

Before going on, we define VALID as the set of $\mathbf{w} \in \{-1, 0, 1\}^L$ of the form:

$$\mathbf{w} = (\mathbf{w}_1^T \| \mathbf{w}_2^T \| g_1\mathbf{w}_2^T \| \ldots \| g_{2\ell}\mathbf{w}_2^T \| \mathbf{g}^T \| \mathbf{w}_3^T \| \mathbf{w}_4^T \| \mathbf{w}_5^T \| \mathbf{w}_6^T)^T$$

for some $\mathbf{w}_1, \mathbf{w}_2 \in \mathsf{B}^3_{m\delta_\beta}$, $\mathbf{g} = (g_1, \ldots, g_{2\ell}) \in \mathsf{B}_{2\ell}$, $\mathbf{w}_3 \in \mathsf{B}^2_m$, $\mathbf{w}_4 \in \mathsf{CorEnc}(mN)$, $\mathbf{w}_5 \in \mathsf{B}^3_{2m\delta_{p-1}}$, and $\mathbf{w}_6 \in \mathsf{B}^3_{L_0\delta_B}$. It can be checked that the constructed vector $\mathbf{x}$ belongs to this tailored set VALID.

Step 2: *Specifying the set $\mathcal{S}$ and permutations of L elements $\{T_\pi : \pi \in \mathcal{S}\}$ for which the conditions in (4.3) hold.*

- Define $\mathcal{S} = \mathcal{S}_{3m\delta_\beta} \times \mathcal{S}_{3m\delta_\beta} \times \mathcal{S}_{2\ell} \times \mathcal{S}_{2m} \times \{0,1\}^{mN} \times \mathcal{S}_{6m\delta_{p-1}} \times \mathcal{S}_{3L_0\delta_B}$.

- For $\pi = (\phi, \psi, \gamma, \rho, \mathbf{b}, \eta, \xi) \in \mathcal{S}$ and $\mathbf{z} = (\mathbf{z}_0^1 \| \mathbf{z}_0^2 \| \mathbf{z}_1 \| \ldots \| \mathbf{z}_{2\ell} \| \mathbf{g} \| \mathbf{t}_1 \| \mathbf{t}_2 \| \mathbf{t}_3 \| \mathbf{t}_4) \in \mathbb{Z}_q^L$, where $\mathbf{z}_0^1, \mathbf{z}_0^2, \mathbf{z}_1, \ldots, \mathbf{z}_{2\ell} \in \mathbb{Z}_q^{3m\delta_\beta}$, $\mathbf{g} \in \mathbb{Z}_q^{2\ell}$, $\mathbf{t}_1 \in \mathbb{Z}_q^{2m}$, $\mathbf{t}_2 \in \mathbb{Z}_q^{2mN}$, $\mathbf{t}_3 \in \mathbb{Z}_q^{6m\delta_{p-1}}$, and $\mathbf{t}_4 \in \mathbb{Z}_q^{3L_0\delta_B}$, we define:

$$T_\pi(\mathbf{z}) = \big(\phi(\mathbf{z}_0^1)^T \| \psi(\mathbf{z}_0^2)^T \| \psi(\mathbf{z}_{\gamma(1)})^T \| \ldots \| \psi(\mathbf{z}_{\gamma(2\ell)})^T \| \gamma(\mathbf{g})^T \| \\ \| \rho(\mathbf{t}_1)^T \| E_{\mathbf{b}}(\mathbf{t}_2)^T \| \eta(\mathbf{t}_3)^T \| \xi(\mathbf{t}_4)^T \big)^T$$

  as the permutation that transforms $\mathbf{z}$ as follows:

  1. It rearranges the order of the 2ℓ blocks $\mathbf{z}_1, \ldots, \mathbf{z}_{2\ell}$ according to γ.
  2. It then permutes block $\mathbf{z}_0^1$ according to ϕ, blocks $\mathbf{z}_0^2$, $\{\mathbf{z}_i\}_{i=1}^{2\ell}$ according to ψ, block $\mathbf{g}$ according to γ, block $\mathbf{t}_1$ according to ρ, block $\mathbf{t}_2$ according to $E_{\mathbf{b}}$, block $\mathbf{t}_3$ according to η, and block $\mathbf{t}_4$ according to ξ.

It can be check that (4.3) holds. Therefore, we can obtain a statistical ZKAoK for the given relation by running the protocol in Section 4.3.2.

7.3.3 The Underlying ZKAoK for the Group Signature Scheme

The argument system upon which our group signature scheme is built can be summarized as follows.

Common Input: Matrices $\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{B} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{D}_0, \mathbf{D}_1 \in \mathbb{Z}_q^{2n\times 2m}$, $\mathbf{F} \in \mathbb{Z}_q^{4n\times 4m}$, $\mathbf{H}_{2n\times m} \in \mathbb{Z}_q^{2n\times m}$, $\mathbf{H}_{4n\times 2m} \in \mathbb{Z}_q^{4n\times 2m}$, $\mathbf{G}_0 \in \mathbb{Z}_q^{n\times 2m}$; vectors $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{c}_1 \in \mathbb{Z}_q^m$, $\mathbf{c}_2 \in \mathbb{Z}_q^{2m}$.

Prover's Input: $\mathbf{z} \in [-\beta, \beta]^{4m}$, $\mathbf{y} \in \{0,1\}^{2m}$, $\mathbf{w} \in \{0,1\}^m$, $\mathbf{d}_1, \mathbf{d}_2 \in [-\beta, \beta]^m$, $\mathbf{s} \in [-\beta, \beta]^{2m}$, $\mathrm{id} = (\mathrm{id}[1], \ldots, \mathrm{id}[\ell])^T \in \{0,1\}^\ell$,
$\mathbf{e}_0 \in [-B, B]^n$, $\mathbf{e}_1 \in [-B, B]^m$, $\mathbf{e}_2 \in [-B, B]^{2m}$.

Prover's Goal: Convince the verifier in ZK that

$$\begin{cases} \mathbf{F} \cdot \mathbf{z} = \mathbf{H}_{4n\times 2m} \cdot \mathbf{y} \bmod q; \;\; \mathbf{H}_{2n\times m} \cdot \mathbf{w} = \mathbf{D}_0 \cdot \mathbf{y} + \mathbf{D}_1 \cdot \mathbf{s} \bmod q; \\ \mathbf{A} \cdot \mathbf{d}_1 + \mathbf{A}_0 \cdot \mathbf{d}_2 + \sum_{j=1}^{\ell} \mathbf{A}_j \cdot (\mathrm{id}[j] \cdot \mathbf{d}_2) - \mathbf{D} \cdot \mathbf{w} = \mathbf{u} \bmod q; \\ \mathbf{c}_1 = \mathbf{B}^T \cdot \mathbf{e}_0 + \mathbf{e}_1 \bmod q; \;\; \mathbf{c}_2 = \mathbf{G}_0^T \cdot \mathbf{e}_0 + \mathbf{e}_2 + \lfloor q/2 \rfloor \cdot \mathbf{y} \bmod q. \end{cases}$$

Using the same strategy as in Sections 7.3.1 and 7.3.2, we can derive a statistical ZKAoK for the above relation from the protocol in Section 4.3.2. As the transformations are similar to those in Section 7.3.2, we only sketch main points.

In the first step, we combine the given equations to an equation of the form:

$$\mathbf{M} \cdot \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{s} \\ \mathbf{z} \end{pmatrix} + \mathbf{M}_0 \cdot \mathbf{d}_2 + \sum_{j=1}^{\ell} \mathbf{M}_j(\mathrm{id}[j]\mathbf{d}_2) + \mathbf{M}' \cdot \begin{pmatrix} \mathbf{w} \\ \mathbf{y} \end{pmatrix} + \mathbf{M}'' \cdot \begin{pmatrix} \mathbf{e}_0 \\ \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix} = \mathbf{v} \bmod q,$$

where matrices $\mathbf{M}, \mathbf{M}_0, \ldots, \mathbf{M}_\ell, \mathbf{M}', \mathbf{M}''$ and vector $\mathbf{v}$ are built from the input.

We then apply the techniques of Section 4.3.2 for $\mathbf{x}_0 = (\mathbf{d}_1^T \| \mathbf{s}^T \| \mathbf{z}^T)^T \in [-\beta, \beta]^{7m}$, $\mathbf{d}_2 \in [-\beta, \beta]^m$; $\mathbf{x}_1 = (\mathbf{w}^T \| \mathbf{y}^T)^T \in \{0,1\}^{3m}$; and $\mathbf{x}_2 = (\mathbf{e}_0^T \| \mathbf{e}_1^T \| \mathbf{e}_2^T)^T \in [-B, B]^{n+3m}$. This allows us to obtain a unified equation $\mathbf{P} \cdot \mathbf{x} = \mathbf{v} \bmod q$, and to define the sets VALID, $\mathcal{S}$, and permutations $\{T_\pi : \pi \in \mathcal{S}\}$ so that the conditions in (4.3) hold, in a similar manner as in Section 7.3.2.

Part III

Group Encryption and Adaptive Oblivious Transfer

Chapter 8

Lattice-Based Group Encryption

Kiayias, Tsiounis and Yung [KTY07] presented group encryption (GE) as the encryption analogue of group signatures [CvH91], which allow users to anonymously sign messages on behalf of an entire group they belong to. While group signatures aim at hiding the source of some message within a crowd administered by some group manager, group encryption rather seeks to hide its destination within a group of legitimate receivers. In both cases, a verifier should be convinced that the anonymous signer/receiver indeed belongs to a purported population. In order to keep users accountable for their actions, an opening authority (OA) is further empowered with some information allowing it to un-anonymize signatures/ciphertexts.

Kiayias, Tsiounis and Yung [KTY07] formalized GE schemes as a primitive allowing the sender to generate publicly verifiable guarantees that: (1) The ciphertext is well-formed and intended for some registered group member who will be able to decrypt; (2) the opening authority will be able identify the receiver if necessary; (3) The plaintext satisfies certain properties such as being a witness for some public relation or the private key that underlies a given public key. In the model of Kiayias *et al.* [KTY07], the message secrecy and anonymity properties are required to withstand active adversaries, which are granted access to decryption oracles in all security experiments.

As a natural application, group encryption allows a firewall to filter all incoming encrypted emails except those intended for some certified organization member and the content of which is additionally guaranteed to satisfy certain requirements, like the absence of malware.

GE schemes are also motivated by natural privacy applications such as anonymous trusted third parties, key recovery mechanisms or oblivious retriever storage systems. In optimistic protocols, GE allows verifiably encrypting messages to *anonymous* trusted third parties which mostly remain off-line and only come into play to sort out conflicts. In order to protect privacy-sensitive information such as users' citizenship, group encryption makes it possible to hide the identity of users' preferred trusted third parties within a set of properly certified trustees.

In cloud storage services, GE enables privacy-preserving asynchronous transfers of encrypted datasets. Namely, it allows users to archive encrypted datasets on remote servers while convincing those servers that the data is indeed intended for some anonymous certified client who paid a subscription to the storage provider. Moreover, a judge should

be able to identify the archive's recipient in case a misbehaving server is found guilty of hosting suspicious transaction records or any other illegal content.

As pointed out by Kiayias *et al.* [KTY07], group encryption also implies a form of hierarchical group signatures [TW05], where signatures can only be opened by a set of eligible trustees operating in a very specific manner determiner by the signer.

The design of numerous privacy-preserving cryptographic protocols crucially relies on zero-knowledge proofs [GMR85] to prove properties about encrypted or committed values so as to enforce honest behavior on behalf of participants or protect the privacy of users. In the lattice settings, efficient zero-knowledge proofs are non-trivial to construct due to the limited amount of algebraic structure. While natural methods of proving knowledge of secret keys [MV03, Lyu08, KTX08, LNSW13] are available, they are only known to work for specific languages. When it comes to proving circuit satisfiability, the best known methods are designed for the LPN setting [JKPT12] or take advantage of the extra structure available in the ring LWE setting [XXW13, BKLP15]. Hence, these methods are not known to readily carry over to standard (i.e., non-ideal) lattices. In the standard model, the problem is even trickier as we do not have a lattice-based counterpart of Groth-Sahai proofs [GS08] and efficient non-interactive proof systems are only available for specific problems [PV08].

The difficulty of designing efficient zero-knowledge proofs for lattice-related languages makes it highly non-trivial to adapt privacy-preserving cryptographic primitives in the lattice setting. In spite of these technical hurdles, a recent body of work successfully designed anonymity-enabling mechanisms like ring signatures [KTX08, AMBB+13], blind signatures [Rü10], group signatures [GKV10, LLLS13, LLNW14, BCK+14, NZZ15, LNW15, LLNW16] or, more recently, signature schemes with companion zero-knowledge protocols [LLM+16a]. A common feature of all these works is that the zero-knowledge layer of the proposed protocols only deals with linear equations, where witnesses are only multiplied by public values.

In this chapter, motivated by the design of advanced privacy-preserving protocols in the lattice setting, we construct zero-knowledge arguments for non-linear statements among witnesses consisting of vectors and matrices. For suitable parameters $q, n, m \in \mathbb{Z}$, we consider zero-knowledge argument systems whereby a prover can demonstrate knowledge of secret matrices $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$ and vectors $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in \mathbb{Z}^m$ such that: (i) $\mathbf{e} \in \mathbb{Z}^m$ has small norm; (ii) A public vector $\mathbf{b} \in \mathbb{Z}_q^n$ equals $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$; (iii) The underlying pair $(\mathbf{X}, \mathbf{s})$ satisfies additional algebraic relations: for instance, it should be possible to prove possession of a signature on some representation of the matrix $\mathbf{X}$. In particular, our zero-knowledge argument makes it possible to prove that a given ciphertext is a well-formed LWE-based encryption with respect to some hidden, but certified public key. This protocol comes in handy in the design of *group encryption* schemes [KTY07], where such languages naturally arise. Using these advances, we thus construct, in this chapter, the first construction of group encryption under lattice assumptions.

Related work. Kiayias, Tsiounis and Yung (KTY) [KTY07] formalized the notion of group encryption and provided a modular design using zero-knowledge proofs, digital signatures, anonymous CCA-secure public-key encryption and commitment schemes. They also gave an efficient instantiation using Paillier's cryptosystem [Pai99] and Camenisch-Lysyanskaya signatures [CL02b].

Cathalo, Libert and Yung [CLY09] designed a non-interactive system in the standard model under non-interactive pairing-related assumptions. El Aimani and Joye [EAJ13] suggested various efficiency improvements with both interactive and non-interactive proofs.

Libert *et al.* [LYJP14] empowered the GE primitive with a refined traceability mechanism akin to that of traceable signatures [KTY04]. Namely, by releasing a user-specific trapdoor, the opening authority can allow anyone to publicly trace ciphertexts encrypted for this specific group member without affecting the privacy of other users. Back in 2010, Izabachène, Pointcheval and Vergnaud [IPV10] considered the problem of eliminating subliminal channels in a different form of traceable group encryption.

As a matter of fact, all existing realizations of group encryption or similar primitives rely on traditional number theoretic assumptions like the hardness of factoring or computing discrete logarithms. In particular, all of them are vulnerable to quantum attacks. For the sake of not putting all one's eggs in the same basket, it is highly desirable to have instantiations based on alternative, quantum-resistant foundations.

In the next sections, we first present the definitions of a group encryption schemes and the required building block. Then, we describe the zero-knowledge protocol we use to handle these quadratic relations before finally describing our scheme.

8.1 Syntax and Definitions of Group Encryption

We use the syntax and the security model of Kiayias, Tsiounis and Yung [KTY07]. The group encryption (GE) primitive involves a sender, a verifier, a group manager (GM) that manages the group of receivers and an opening authority (OA) which is capable of identifying ciphertexts' recipients.

In the syntax of [KTY07], a GE scheme is specified by the description of a relation R as well as a tuple $\mathsf{GE} = (\mathsf{SETUP}, \mathsf{JOIN}, \langle \mathcal{G}_r, R, \mathsf{sample}_R \rangle, \mathsf{ENC}, \mathsf{DEC}, \mathsf{OPEN}, \langle \mathcal{P}, \mathcal{V} \rangle)$ of algorithms or protocols. In details, SETUP is a set of initialization procedures that all take (implicitly or explicitly) a security parameter 1^λ as input. We call them $\mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$, $\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{par})$ and $\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{par})$. The first one of these procedures generates a set of public parameters par (like the KTY construction [KTY07], we rely on a common reference string even when using interaction between provers and verifiers). The latter two procedures are used to produce key pairs $(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}})$, $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$ for the GM and the OA. In the following, par is incorporated in the inputs of all algorithms although we sometimes omit to explicitly write it.

$\mathsf{JOIN} = (\mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}})$ is an interactive protocol between the GM and the prospective user. After the execution of JOIN, the GM stores the public key pk and its certificate $\mathsf{cert}_{\mathsf{pk}}$ in a public directory database. As in [KY05], we will restrict this protocol to have minimal interaction and consist of only two messages: the first one is the user's public key pk sent by $\mathsf{J}_{\mathsf{user}}$ to J_{GM} and the latter's response is a certificate $\mathsf{cert}_{\mathsf{pk}}$ for pk that makes the user's group membership effective. We do not require the user to prove knowledge of his private key sk or anything else about it. In our construction, valid keys will be publicly recognizable and users will not have to prove their validity. By avoiding proofs of knowledge of private keys, the security proof never has to rewind the adversary to extract those private keys, which allows supporting concurrent joins as advocated by Kiayias and Yung [KY05]. If

applications demand it, it is possible to add proofs of knowledge of private keys in a modular way but our security proofs do not require rewinding the adversary in executions of JOIN.

Algorithm sample_R allows sampling pairs $(x, w) \in R$ (made of a public value x and a witness w) using keys $(\mathsf{pk}_R, \mathsf{sk}_R)$ produced by $\mathcal{G}_r(1^\lambda)$ which samples public/secret parameters for the relation R. Depending on the relation, sk_R may be the empty string (as in the scheme [KTY07] and ours which both involve publicly samplable relations). The testing procedure $R(x, w)$ uses pk_R to return 1 whenever $(x, w) \in R$. To encrypt a witness w such that $(x, w) \in R$ for some public x, the sender fetches the pair $(\mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$ from database and runs the randomized encryption algorithm. The latter takes as input w, a label L, the receiver's pair $(\mathsf{pk}, \mathsf{cert}_{\mathsf{pk}})$ as well as public keys $\mathsf{pk}_{\mathsf{GM}}$ and $\mathsf{pk}_{\mathsf{OA}}$. Its output is a ciphertext $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$. On input of the same elements, the certificate $\mathsf{cert}_{\mathsf{pk}}$, the ciphertext Ψ and the random coins $coins_\Psi$ that were used to produce Ψ, the non-interactive algorithm PP generates a proof π_Ψ that there exists a certified receiver whose public key was registered in database and who is able to decrypt Ψ and obtain a witness w such that $(x, w) \in R$. The verification algorithm $\mathcal{V}$ takes as input Ψ, $\mathsf{pk}_{\mathsf{GM}}$, $\mathsf{pk}_{\mathsf{OA}}$, π_Ψ and the description of R and outputs 0 or 1. Given Ψ, L and the receiver's private key sk, the output of DEC is either a witness w such that $(x, w) \in R$ or a rejection symbol $\perp$. Finally, OPEN takes as input a ciphertext/label pair (Ψ, L) and the OA's secret key $\mathsf{sk}_{\mathsf{OA}}$ and returns a receiver's public key pk.

The model of [KTY07] considers four properties termed correctness, message security, anonymity and soundness. In the security definitions, stateful oracles capture the adversary's interaction with the system. In the soundness game, the KTY model requires that pk belongs to the language of valid public keys. Here, we are implicitly assuming that the space of valid public keys is dense (all matrices are valid keys, as is the case in our scheme).

In the upcoming definitions, we sometimes use the notation

$$\langle \mathsf{output}_A | \mathsf{output}_B \rangle \leftarrow \langle A(\mathsf{input}_A), B(\mathsf{input}_B) \rangle (\mathsf{common\text{-}input})$$

to denote the execution of a protocol between A and B obtaining their own outputs from their respective inputs.

Correctness. The correctness property requires that the following experiment returns 1 with overwhelming probability.

Experiment $\mathbf{Exp}^{\mathrm{correctness}}(\lambda)$

$\mathsf{param} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$; $(\mathsf{pk}_R, \mathsf{sk}_R) \leftarrow \mathcal{G}_r(\lambda)$; $(x, w) \leftarrow \mathsf{sample}_R(\mathsf{pk}_R, \mathsf{sk}_R)$;
$(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}}) \leftarrow \mathsf{SETUP}_{\mathsf{GM}}(\mathsf{param})$; $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) \leftarrow \mathsf{SETUP}_{\mathsf{OA}}(\mathsf{param})$;
$\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert}_{\mathsf{pk}} | \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}} \rangle \leftarrow \langle \mathsf{J}_{\mathsf{user}}, \mathsf{J}_{\mathsf{GM}}(\mathsf{sk}_{\mathsf{GM}}) \rangle (\mathsf{pk}_{\mathsf{GM}})$;
$\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$;
$\pi_\Psi \leftarrow \mathcal{P}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}, w, L, \Psi, coins_\Psi)$;
if $\big((w \neq \mathsf{DEC}(\mathsf{sk}, \Psi, L)) \vee (\mathsf{pk} \neq \mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \Psi, L))$
$\quad \vee (\mathcal{V}(\Psi, L, \pi_\Psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}) = 0)\big)$ **then**
  return 0
else
  return 1;

Message Secrecy. The message secrecy property is defined by an experiment where the adversary has access to oracles that may be stateful (and maintain a state across queries) or stateless:

- $\mathsf{DEC}(\mathsf{sk})$: is a stateless oracle for the user decryption function DEC. When this oracle is restricted not to decrypt a ciphertext-label pair (Ψ, L), we denote it by $\mathsf{DEC}^{\neg\langle\Psi,L\rangle}$.

- $\mathsf{CH}^b_{\mathsf{ror}}(\lambda, \mathsf{pk}, w, L)$: is a real-or-random challenge oracle which is called *once*. It returns $(\Psi, coins_\Psi)$ such that $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w, L)$ if $b = 1$ whereas, if $b = 0$, $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, w', L)$ encrypts a random plaintext of length $O(\lambda)$ uniformly sampled in the plaintext space. In both cases, $coins_\Psi$ denote the random coins used to generate Ψ.

- $\mathsf{PROVE}^b_{\mathsf{PP},\mathsf{PP}'}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}, \mathsf{cert}_{\mathsf{pk}}, \mathsf{pk}_R, x, w, \Psi, L, coins_\Psi)$: is a stateful oracle that can be invoked a polynomial number times. If $b = 1$, it replies by running the real prover PP on the inputs to create an actual proof π_Ψ. If $b = 0$, the oracle runs a simulator PP' that uses the same inputs as PP except witness $w, coins_\Psi$ and generates a simulated proof.

These oracles are used in an experiment where the adversary controls the GM, the OA and all members except the honest receiver. The adversary $\mathcal{A}$ embodies the dishonest GM that certifies the honest receiver in an execution of JOIN. It is granted access to an oracle DEC which decrypts on behalf of that receiver. In the challenge phase, it transmits a state information aux to itself and invokes the challenge oracle for a label and a pair $(x, w) \in R$ of its choice. After the challenge phase, it can also query the PROVE oracle many times and finally attempts to guess the challenger's bit b.

As pointed out in [KTY07, CLY09], designing an efficient simulator PP' (for executing $\mathsf{PROVE}^b_{\mathsf{PP},\mathsf{PP}'}(.)$ when $b = 0$) is part of the security proof.

Definition 8.1. A GE scheme satisfies *message security* if, for any PPT adversary $\mathcal{A}$, the experiment below returns 1 with probability at most $1/2 + \mathsf{negl}(\lambda)$.

Experiment $\mathbf{Exp}^{\mathrm{sec}}_{\mathcal{A}}(\lambda)$

$\mathsf{par} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda); (\mathsf{aux}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}) \leftarrow \mathcal{A}(\mathsf{par});$

$\langle \mathsf{pk}, \mathsf{sk}, \mathsf{cert}_{\mathsf{pk}} | \mathsf{aux} \rangle \leftarrow \langle \mathsf{J}_{\mathsf{user}}, \mathcal{A}(\mathsf{aux}) \rangle (\mathsf{pk}_{\mathsf{GM}});$

$(\mathsf{aux}, x, w, L, \mathsf{pk}_R) \leftarrow \mathcal{A}^{\mathsf{DEC}(\mathsf{sk},.)}(\mathsf{aux});$

if $(x, w) \notin R$ **then**

  return 0;

$b \hookleftarrow \{0, 1\}; (\Psi, coins_\Psi) \leftarrow \mathsf{CH}^b_{\mathsf{ror}}(\lambda, \mathsf{pk}, w, L);$

$b' \leftarrow \mathcal{A}^{\mathsf{PROVE}^b_{\mathsf{PP},\mathsf{PP}'}(\mathsf{pk}_{\mathsf{GM}},\mathsf{pk}_{\mathsf{OA}},\mathsf{pk},\mathsf{cert}_{\mathsf{pk}},\mathsf{pk}_R,x,w,\Psi,L,coins_\Psi),\mathsf{DEC}^{\neg\langle\Psi,L\rangle}(\mathsf{sk},.)}(\mathsf{aux}, \Psi);$

if $b = b'$ **then**

  return 1

else

  return 0;

Anonymity. In the experiment modeling the anonymity property, the adversary controls the entire system except the opening authority and two well-behaved users. The challenger thus introduces two honest users' public keys $\mathsf{pk}_0, \mathsf{pk}_1$ in database and thus obtains certificate for both $\mathsf{pk}_0, \mathsf{pk}_1$ from the adversarially-controlled GM. For a pair $(x, w) \in R$ of its choice, the adversary obtains an encryption of w under pk_b for some $b \in \{0, 1\}$ chosen by the challenger. The adversary is provided with decryption oracles w.r.t. both keys $\mathsf{pk}_0, \mathsf{pk}_1$. In addition, it has the following oracles at disposal:

- $\mathsf{CH}^b_{\mathsf{anon}}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_0, \mathsf{pk}_1, w, L)$: is a challenge oracle that is only queried once by the adversary. It returns a pair $(\Psi, coins_\Psi)$ consisting of a ciphertext $\Psi \leftarrow \mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_b, \mathsf{cert}_{\mathsf{pk}_b}, w, L)$ and the coin tosses $coins_\Psi$ that were used to generate Ψ.

- $\mathsf{USER}(\mathsf{pk}_{\mathsf{GM}})$: is a stateful oracle that obtains certificates from the adversary by simulating two executions of $\mathsf{J}_{\mathsf{user}}$ to introduce two honest users in the group. It uses a string keys where the outputs $(\mathsf{pk}_0, \mathsf{sk}_0, \mathsf{cert}_{\mathsf{pk}_0})$, $(\mathsf{pk}_1, \mathsf{sk}_1, \mathsf{cert}_{\mathsf{pk}_1})$ of honest users are written as long as the adversarially-supplied certificates $\{\mathsf{cert}_{\mathsf{pk}_d}\}_{d=0}^1$ are valid w.r.t. $\mathsf{pk}_{\mathsf{GM}}$ (i.e., invalid certificates are ignored by the oracle and no entry is introduced in keys for them).

- $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, .)$: is a stateless oracle that simulates the opening algorithm and, on input of a GE ciphertext, returns the receiver's public key.

The reason why the USER oracle is needed is that both honest users' public keys $\mathsf{pk}_0, \mathsf{pk}_1$ must have been properly certified by the adversarially-controlled GM before the challenge phase because the adversary subsequently obtains proofs generated using $(\mathsf{pk}_b, \mathsf{cert}_{\mathsf{pk}_b})$.

Definition 8.2. A GE scheme satisfies *anonymity* if, for any PPT adversary $\mathcal{A}$, the experiment below returns 1 with a probability not exceeding $1/2 + \mathsf{negl}(\lambda)$.

Experiment $\mathbf{Exp}^{\mathrm{anon}}_{\mathcal{A}}(\lambda)$

$\mathsf{par} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$; $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) \leftarrow \mathsf{SETUP}_{\mathsf{OA}}(\mathsf{par})$;
$(\mathsf{aux}, \mathsf{pk}_{\mathsf{GM}}) \leftarrow \mathcal{A}(\mathsf{par}, \mathsf{pk}_{\mathsf{OA}})$; $\mathsf{aux} \leftarrow \mathcal{A}^{\mathsf{USER}(\mathsf{pk}_{\mathsf{GM}}), \mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, .)}(\mathsf{aux})$;
if $\mathsf{keys} \neq (\mathsf{pk}_0, \mathsf{sk}_0, \mathsf{cert}_{\mathsf{pk}_0}, \mathsf{pk}_1, \mathsf{sk}_1, \mathsf{cert}_{\mathsf{pk}_1})(\mathsf{aux})$ **then**
return 0;
$(\mathsf{aux}, x, w, L, \mathsf{pk}_R) \leftarrow \mathcal{A}^{\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, .), \mathsf{DEC}(\mathsf{sk}_0, .), \mathsf{DEC}(\mathsf{sk}_1, .)}(\mathsf{aux})$;
if $(x, w) \notin R$ **then**
  return 0;
$b \hookleftarrow \{0, 1\}$; $(\Psi, coins_\Psi) \leftarrow \mathsf{CH}^b_{\mathsf{anon}}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_0, \mathsf{pk}_1, w, L)$;
$b' \leftarrow \mathcal{A}^{\mathcal{P}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_b, \mathsf{cert}_{\mathsf{pk}_b}, x, w, \Psi, L, coins_\Psi,}$
$\mathsf{OPEN}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_{\mathsf{OA}}, .), \mathsf{DEC}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_0, .), \mathsf{DEC}^{\neg\langle\Psi, L\rangle}(\mathsf{sk}_1, .))}(\mathsf{aux}, \Psi)$;
if $b = b'$ **then**
  return 1
else
  return 0;

Soundness. Here, the adversary creates the group of receivers by interacting with the honest GM. Its goal is to produce a ciphertext Ψ and a convincing proof that Ψ is valid w.r.t. a relation R of its choice but either: (1) The opening of Ψ reveals a receiver's public key pk that does not belong to any group member; (2) The output pk of OPEN is not a valid public key (*i.e.*, $\mathsf{pk} \notin \mathcal{PK}$, where $\mathcal{PK}$ is the language of valid public keys); (3) The ciphertext C is not in the space $\mathcal{C}^{x,L,\mathsf{pk}_R,\mathsf{pk}_{\mathsf{GM}},\mathsf{pk}_{\mathsf{OA}},\mathsf{pk}}$ of valid ciphertexts. This notion is formalized by a game where the adversary is given access to a user registration oracle $\mathsf{REG}(\mathsf{sk}_{\mathsf{GM}}, .)$ that simulates J_{GM}. This oracle maintains a list $\mathsf{database}$ where registered public keys and their certificates are stored.

Definition 8.3. A GE scheme is *sound* if, for any PPT adversary $\mathcal{A}$, the experiment below returns 1 with negligible probability.

Experiment $\mathbf{Exp}^{\text{soundness}}_{\mathcal{A}}(\lambda)$

$\mathsf{par} \leftarrow \mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$; $(\mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}}) \leftarrow \mathsf{SETUP}_{\mathsf{OA}}(\mathsf{par})$;
$(\mathsf{pk}_{\mathsf{GM}}, \mathsf{sk}_{\mathsf{GM}}) \leftarrow \mathsf{SETUP}_{\mathsf{GM}}(\mathsf{par})$;
$(\mathsf{pk}_R, x, \Psi, \pi_\Psi, L, \mathsf{aux}) \leftarrow \mathcal{A}^{\mathsf{REG}(\mathsf{sk}_{\mathsf{GM}},.)}(\mathsf{par}, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{sk}_{\mathsf{OA}})$;
if $\mathcal{V}(\Psi, L, \pi_\Psi, \mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}) = 0$ **then**
  return 0;
$\mathsf{pk} \leftarrow \mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \Psi, L)$;
if $\big((\mathsf{pk} \notin \mathsf{database}) \vee (\mathsf{pk} \notin \mathcal{PK}) \vee (\Psi \notin \mathcal{C}^{x,L,\mathsf{pk}_R,\mathsf{pk}_{\mathsf{GM}},\mathsf{pk}_{\mathsf{OA}},\mathsf{pk}})\big)$ **then**
  return 1
else
  return 0;

The model of Kiayias *et al.* [KTY07] requires that pk belongs to the language of valid public keys, so that the adversary is considered to defeat the soundness property when (Ψ, L) opens to a key outside the language (i.e., $\mathsf{pk} \notin \mathcal{PK}$). In our scheme, we will assume that the space of valid public keys is dense in that all matrices of a given dimension are valid public keys, which have an underlying private key. We nevertheless use the same definition as [KTY07] in order to emphasize that we are not relaxing the model in any way.

8.2 Building Blocks

8.2.1 The Agrawal-Boneh-Boyen IBE Scheme

8.2.1.1 Identity-Based Encryption.

An IBE scheme is a tuple of efficient algorithms $(\mathsf{Setup}, \mathsf{Extract}_{\mathsf{PP}}, \mathsf{Encrypt}_{\mathsf{PP}}, \mathsf{Decrypt}_{\mathsf{PP}})$ such that

Setup(1^λ)**:** On security parameter λ, this algorithm outputs public parameters PP and a master secret key msk.

Extract$_{\mathsf{PP}}$**(msk**, ID**):** Takes as input a master secret key msk and an identity ID and outputs a secret key $\mathsf{sk}_{\mathsf{ID}}$.

Encrypt$_{\mathsf{PP}}(\mathsf{ID}, M)$**:** Given an identity ID and a message M, it outputs a ciphertext C.

Experiment $\mathbf{Exp}^{\mathrm{ROR}}_{\mathcal{A}}(\lambda)$

$\mathsf{ID}^\star \leftarrow \mathcal{A}(\mathsf{id}, \lambda)$; $(\mathsf{PP}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$;
$M \leftarrow \mathcal{A}^{\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk},\cdot)}_{\mathsf{Ch}}(\mathsf{PP})$;
$b \hookleftarrow \mathcal{U}(\{0,1\})$;
if $b = 1$ **then**
  $C^\star \leftarrow \mathsf{Encrypt}_{\mathsf{PP}}(M, \mathsf{ID}^\star)$
else
  $C^\star \hookleftarrow \mathcal{U}(\mathcal{C})$;
$b' \leftarrow \mathcal{A}^{\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk},\cdot)}(\mathsf{guess}, C^\star)$;
if $b = b'$ **then**
  return 1
else
  return 0

Figure 8.1 – Security experiment for the pseudo-random-ciphertext property for an IBE

$\mathsf{Decrypt}_{\mathsf{PP}}(\mathsf{sk}_{\mathsf{ID}}, C)$: Given a secret key $\mathsf{sk}_{\mathsf{ID}}$ and a ciphertext C, outputs either a decryption error symbol $\perp$, or a message M.

Correctness requires that, for any pair $(\mathsf{PP}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$, any ID and any message M, we have $\mathsf{Decrypt}_{\mathsf{PP}}(\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk}, \mathsf{ID}), \mathsf{Encrypt}_{\mathsf{PP}}(\mathsf{ID}, M)) = M$. Our proofs rely on the semantic security of the scheme against selective adversaries (IND-sID-CPA) but also on the stronger property of ciphertext pseudo-randomness. Informally, this notions demands that the adversary be unable to distinguish an encryption of a message of its choice from a random element of the ciphertext space $\mathcal{C}$. Notice that this property implies IND-sID-CPA security.

Definition 8.4. An IBE scheme has pseudo-random-ciphertexts if no PPT adversary $\mathcal{A}$ with access to private key extraction oracle $\mathsf{Extract}_{\mathsf{PP}}(\mathsf{msk}, \cdot)$ has non-negligible advantage $\mathrm{Adv}^{\mathrm{ROR}}_{\mathcal{A}}\lambda = |\Pr[\mathbf{Expt}^{\mathrm{ROR}}_{\mathcal{A}} = 1] - \frac{1}{2}|$ in the game described in Figure 8.1

8.2.1.2 The ABB System.

Agrawal, Boneh and Boyen described [ABB10] a compact IBE scheme in the standard model which allows encrypting multi-bit messages.

$\mathsf{Setup}(1^\lambda)$: Given a security parameter λ, choose parameters q, n, σ, α and define $k = \lfloor \log q \rfloor$, $\bar{m} = nk$, $m = 2\bar{m}$ and choose a noise distribution χ for LWE.

1. Compute $(\bar{\mathbf{A}}, \mathbf{T}_{\bar{\mathbf{A}}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$.
2. Define $\mathbf{G} = \mathbf{I}_n \otimes [1|2|\ldots|2^{k-1}] \in \mathbb{Z}_q^{n\times\bar{m}}$. Sample matrices $\mathbf{B} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times\bar{m}})$, $\mathbf{U} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$.
3. Let $\mathsf{FRD} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ be the full-rank difference mapping from [ABB10].

Output $\mathsf{PP} = (\bar{\mathbf{A}}, \mathbf{B}, \mathbf{U})$ and $\mathsf{msk} = \mathbf{T}_{\bar{\mathbf{A}}}$.

Extract$_{\mathsf{PP}}(\mathbf{msk}, \mathsf{ID})$: Given $\mathsf{msk} = \mathbf{T}_{\bar{\mathbf{A}}}$ and an identity $\mathsf{ID} \in \mathbb{Z}_q^n$, do as follows:

1. Define the matrix $\mathbf{B}_{\mathsf{ID}} = \mathbf{B} + \mathsf{FRD}(\mathsf{ID}) \cdot \mathbf{G} \in \mathbb{Z}_q^{n \times \bar{m}}$.
2. Let $\mathbf{B}_{\mathbf{A},\mathsf{ID}} = [\mathbf{A} \mid \mathbf{B}_{\mathsf{ID}}] \in \mathbb{Z}_q^{n \times (m+\bar{m})}$, use $\mathbf{T}_A$ to compute a delegated basis $\mathbf{T}_{\mathsf{ID}}$ for the lattice $\Lambda^{\perp}(\mathbf{B}_{\mathbf{A},\mathsf{ID}})$.
3. Use $\mathbf{T}_{\mathsf{ID}}$ to sample a small-norm matrix $\mathbf{E}_{\mathsf{ID}} \in \mathbb{Z}^{(m+\bar{m}) \times m}$ satisfying the equality $\mathbf{B}_{\mathbf{A},\mathsf{ID}} \cdot \mathbf{E}_{\mathsf{ID}} = \mathbf{U} \bmod q$.
4. Output $\mathsf{sk}_{\mathsf{ID}} = \mathbf{E}_{\mathsf{ID}} \in \mathbb{Z}^{(m+\bar{m}) \times m}$.

Encrypt$_{\mathsf{PP}}(\mathsf{ID}, \mathbf{m})$: Given an identity ID and a message $\mathbf{m} \in \{0,1\}^m$,

1. Compute the matrix $\mathbf{B}_{\mathsf{ID}} = \mathbf{B} + \mathsf{FRD}(\mathsf{ID}) \cdot \mathbf{G} \in \mathbb{Z}_q^{n \times \bar{m}}$. Sample vectors $\mathbf{s} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^n), \mathbf{x}, \mathbf{y} \hookleftarrow \chi^m, \mathbf{R} \hookleftarrow D_{\mathbb{Z},\sigma}^{m \times \bar{m}}$ and compute $\mathbf{z} = \mathbf{R}^T \cdot \mathbf{y} \in \mathbb{Z}^m$.
2. Compute
$$\begin{cases} \mathbf{c}^{(1)} = \bar{\mathbf{A}}^T \cdot \mathbf{s} + \mathbf{y} \bmod q, \\ \mathbf{c}^{(2)} = \mathbf{B}_{\mathsf{ID}}^T \cdot \mathbf{s} + \mathbf{z} \bmod q, \\ \mathbf{c}^{(3)} = \mathbf{U}^T \cdot \mathbf{s} + \mathbf{x} + \mathbf{m} \cdot \left\lfloor \frac{q}{2} \right\rfloor. \end{cases} \tag{8.1}$$
3. Output $\mathbf{c} = (\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \mathbf{c}^{(3)}) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^m$.

Decrypt$_{\mathsf{PP}}(\mathsf{sk}_{\mathsf{ID}}, \mathbf{c})$: Given $\mathsf{sk}_{\mathsf{ID}} = \mathbf{E}_{\mathsf{ID}}$ and $\mathbf{c} = (\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \mathbf{c}^{(3)}) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^m$, compute and output $\mathbf{m}' = \left\lfloor \left(\mathbf{c}^{(3)} - \mathbf{E}_{\mathsf{ID}} \cdot \begin{bmatrix} \mathbf{c}^{(1)} \\ \mathbf{c}^{(2)} \end{bmatrix} \right) \cdot \left\lfloor \frac{q}{2} \right\rfloor^{-1} \right\rceil \in \{0,1\}^m$.

Theorem 8.1 ([ABB10, Th. 23])**.** *The ABB IBE scheme has pseudo-random ciphertexts if the* $\mathsf{LWE}_{n,q,\chi}$ *assumption holds.*

8.3 Warm-up: Decompositions, Extensions, Permutations

This section introduces the notations and techniques that will be used throughout the chapter. It details Stern-like protocols that have been introduced in Section 4.3. The techniques that will be employed for handling quadratic relations (double-bit extension $\mathsf{ext}(\cdot,\cdot)$, expansion $\mathsf{expand}^{\otimes}(\cdot,\cdot)$ of matrix-vector product and the associated permuting mechanisms) are novel contributions.

8.3.1 Decompositions

For any $B \in \mathbb{Z}_+$, define the number $\delta_B := \lfloor \log_2 B \rfloor + 1 = \lceil \log_2(B+1) \rceil$ and the sequence $B_1, \ldots, B_{\delta_B}$, where $B_j = \lfloor \frac{B+2^{j-1}}{2^j} \rfloor, \forall j \in [1, \delta_B]$. As observed in [LNSW13], the sequence satisfies $\sum_{j=1}^{\delta_B} B_j = B$ and any integer $v \in [0, B]$ can be decomposed into a binary vector $\mathsf{idec}_B(v) = (v^{(1)}, \ldots, v^{(\delta_B)})^T \in \{0,1\}^{\delta_B}$ such that $\sum_{j=1}^{\delta_B} B_j \cdot v^{(j)} = v$. We describe this decomposition procedure in a deterministic manner:

1. $v' := v$

2. For $j = 1$ to δ_B do:
   (i) If $v' \geq B_j$ then $v^{(j)} := 1$, else $v^{(j)} := 0$;
   (ii) $v' := v' - B_j \cdot v^{(j)}$.
3. Output $\mathsf{idec}_B(v) = (v^{(1)}, \ldots, v^{(\delta_B)})^T$.

Next, for any positive integers $\mathfrak{m}, B$, we define the decomposition matrix:

$$\mathbf{H}_{\mathfrak{m},B} := \begin{bmatrix} B_1 \ldots B_{\delta_B} & & & \\ & B_1 \ldots B_{\delta_B} & & \\ & & \ddots & \\ & & & B_1 \ldots B_{\delta_B} \end{bmatrix} \in \mathbb{Z}^{\mathfrak{m} \times \mathfrak{m}\delta_B}, \tag{8.2}$$

and the following injective functions:

(i) $\mathsf{vdec}_{\mathfrak{m},B} : [0,B]^{\mathfrak{m}} \to \{0,1\}^{\mathfrak{m}\delta_B}$ that maps vector $\mathbf{v} = (v_1, \ldots, v_{\mathfrak{m}})^T$ to vector $\left(\mathsf{idec}_B(v_1)^T \| \ldots \| \mathsf{idec}_B(v_{\mathfrak{m}})^T\right)^T$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}_{\mathfrak{m},B}(\mathbf{v}) = \mathbf{v}$.

(ii) $\mathsf{vdec}'_{\mathfrak{m},B} : [-B,B]^{\mathfrak{m}} \to \{-1,0,1\}^{\mathfrak{m}\delta_B}$ that maps vector $\mathbf{w} = (w_1, \ldots, w_{\mathfrak{m}})^T$ to vector $\left(\sigma(w_1) \cdot \mathsf{idec}_B(w_1)^T \| \ldots \| \sigma(w_{\mathfrak{m}}) \cdot \mathsf{idec}_B(w_{\mathfrak{m}})^T\right)^T$, where for each $i = 1, \ldots, \mathfrak{m}$: $\sigma(w_i) = 0$ if $w_i = 0$; $\sigma(w_i) = -1$ if $w_i < 0$; $\sigma(w_i) = 1$ if $w_i > 0$. Note that $\mathbf{H}_{\mathfrak{m},B} \cdot \mathsf{vdec}'_{\mathfrak{m},B}(\mathbf{w}) = \mathbf{w}$.

We also define the following matrix decomposition procedure. For positive integers n, m, q, define the injective function $\mathsf{mdec}_{n,m,q} : \mathbb{Z}_q^{m \times n} \to \{0,1\}^{mn\delta_{q-1}}$ that maps matrix $\mathbf{X} = [\mathbf{x}_1 | \ldots | \mathbf{x}_n] \in \mathbb{Z}_q^{m \times n}$, where $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{Z}_q^m$, to vector

$$\begin{aligned} \mathsf{mdec}_{n,m,q}(\mathbf{X}) &= \left(\mathsf{vdec}_{m,q-1}(\mathbf{x}_1)^T \| \ldots \| \mathsf{vdec}_{m,q-1}(\mathbf{x}_n)^T\right)^T \\ &= (x_{1,1}, \ldots, x_{1,mk}, x_{2,1}, \ldots, x_{2,mk}, \ldots, x_{n,1}, \ldots, x_{n,mk})^T \\ &\qquad \in \{0,1\}^{nm\delta_{q-1}}, \end{aligned}$$

where, for each $(i,j) \in [n] \times [m\delta_{q-1}]$, $x_{i,j} \in \{0,1\}$ denotes the j-th bit of the decomposition of the i-th column of $\mathbf{X}$.
Looking ahead, when proving knowledge of witnesses $(\mathbf{X}, \mathbf{s}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n$ satisfying $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, we will have to consider terms of the form $x_{i,j} \cdot s_{i,t}$, where $\mathbf{s} = (s_1, \ldots, s_n)^T \in \mathbb{Z}_q^n$ and $(s_{i,1}, \ldots, s_{i,\delta_{q-1}})^T = \mathsf{idec}_{q-1}(s_i)$ for each $i \in [n]$.

8.3.2 Extensions and Permutations

We now introduce the extensions and permutations which will be essential for proving quadratic relations.

- For each $c \in \{0,1\}$, denote by $\overline{c}$ the bit $1 - c \in \{0,1\}$.
- For $c_1, c_2 \in \{0,1\}$, define the vector

$$\mathsf{ext}(c_1, c_2) = (\overline{c}_1 \cdot \overline{c}_2, \overline{c}_1 \cdot c_2, c_1 \cdot \overline{c}_2, c_1 \cdot c_2)^T \in \{0,1\}^4.$$

- For $b_1, b_2 \in \{0,1\}$, define the permutation T_{b_1,b_2} that transforms vector $\mathbf{v} = (v_{0,0}, v_{0,1}, v_{1,0}, v_{1,1})^T \in \mathbb{Z}_q^4$ to vector $(v_{b_1,b_2}, v_{b_1,\bar{b}_2}, v_{\bar{b}_1,b_2}, v_{\bar{b}_1,\bar{b}_2})^T$.

  Note that, for all $c_1, c_2, b_1, b_2 \in \{0,1\}$, we have the following:

$$\mathbf{z} = \mathsf{ext}(c_1, c_2) \iff T_{b_1,b_2}(\mathbf{z}) = \mathsf{ext}(c_1 \oplus b_1, c_2 \oplus b_2), \tag{8.3}$$

where $\oplus$ denotes the bit-wise addition modulo 2.

Now, for positive integers n, m, k, and for vectors

$$\mathbf{x} = (x_{1,1}, \ldots, x_{1,mk}, x_{2,1}, \ldots, x_{2,mk}, \ldots, x_{n,1}, x_{n,mk})^T \in \{0,1\}^{nmk}$$

and $\mathbf{s}_0 = (s_{1,1}, \ldots, s_{1,k}, s_{2,1}, \ldots, s_{2,k}, \ldots, s_{n,1}, \ldots, s_{n,k})^T \in \{0,1\}^{nk}$, we define the vector $\mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0) \in \{0,1\}^{4nmk^2}$ as

$$\begin{aligned}
\mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0) = &(\mathsf{ext}^T(x_{1,1}, s_{1,1}) \| \mathsf{ext}^T(x_{1,1}, s_{1,2}) \| \ldots \| \mathsf{ext}^T(x_{1,1}, s_{1,k}) \| \\
&\| \mathsf{ext}^T(x_{1,2}, s_{1,1}) \| \mathsf{ext}^T(x_{1,2}, s_{1,2}) \| \ldots \| \mathsf{ext}^T(x_{1,2}, s_{1,k}) \| \ldots \\
&\| \mathsf{ext}^T(x_{1,mk}, s_{1,1}) \| \mathsf{ext}^T(x_{1,mk}, s_{1,2}) \| \ldots \| \mathsf{ext}^T(x_{1,mk}, s_{1,k}) \| \\
&\| \mathsf{ext}^T(x_{2,1}, s_{2,1}) \| \mathsf{ext}^T(x_{2,1}, s_{2,2}) \| \ldots \| \mathsf{ext}^T(x_{2,1}, s_{2,k}) \| \ldots \\
&\| \mathsf{ext}^T(x_{2,mk}, s_{2,1}) \| \mathsf{ext}^T(x_{2,mk}, s_{2,2}) \| \ldots \| \mathsf{ext}^T(x_{2,mk}, s_{2,k}) \| \ldots \\
&\| \mathsf{ext}^T(x_{n,1}, s_{n,1}) \| \mathsf{ext}^T(x_{n,1}, s_{n,2}) \| \ldots \| \mathsf{ext}^T(x_{n,1}, s_{n,k}) \| \ldots \\
&\| \mathsf{ext}^T(x_{n,mk}, s_{n,1}) \| \mathsf{ext}^T(x_{n,mk}, s_{n,2}) \| \ldots \| \mathsf{ext}^T(x_{n,mk}, s_{n,k}))^T.
\end{aligned}$$

That is, $\mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0)$ is obtained by applying ext to all pairs of the form $(x_{i,j}, s_{i,t})$ for $(i,j,t) \in [n] \times [mk] \times [k]$.

Now, for $\mathbf{b} = (b_{1,1}, \ldots, b_{1,mk}, b_{2,1}, \ldots, b_{2,mk}, \ldots, b_{n,1}, b_{n,mk})^T \in \{0,1\}^{nmk}$ and $\mathbf{d} = (d_{1,1}, \ldots, d_{1,k}, d_{2,1}, \ldots, d_{2,k}, \ldots, d_{n,1}, \ldots, d_{n,k})^T \in \{0,1\}^{nk}$, we define the permutation $P_{\mathbf{b},\mathbf{d}}$ that transforms vector

$$\begin{aligned}
\mathbf{v} = &((\mathbf{v}_{1,1,1}^T \| \ldots \| \mathbf{v}_{1,1,k}^T) \| (\mathbf{v}_{1,2,1}^T \| \ldots \| \mathbf{v}_{1,2,k}^T) \| \ldots \| (\mathbf{v}_{1,mk,1}^T \| \ldots \| \mathbf{v}_{1,mk,k}^T) \| \\
&(\mathbf{v}_{2,1,1}^T \| \ldots \| \mathbf{v}_{2,1,k}^T) \| (\mathbf{v}_{2,2,1}^T \| \ldots \| \mathbf{v}_{2,2,k}^T) \| \ldots \| (\mathbf{v}_{2,mk,1}^T \| \ldots \| \mathbf{v}_{2,mk,k}^T) \| \\
&(\mathbf{v}_{n,1,1}^T \| \ldots \| \mathbf{v}_{n,1,k}^T) \| (\mathbf{v}_{n,2,1}^T \| \ldots \| \mathbf{v}_{n,2,k}^T) \| \ldots \| (\mathbf{v}_{n,mk,1}^T \| \ldots \| \mathbf{v}_{n,mk,k}^T))^T \in \mathbb{Z}^{4nmk^2},
\end{aligned}$$

consisting of nmk^2 blocks of length 4, to the vector $P_{\mathbf{b},\mathbf{d}}(\mathbf{v})$ of the form

$$\begin{aligned}
(&(\mathbf{w}_{1,1,1}^T \| \ldots \| \mathbf{w}_{1,1,k}^T) \| (\mathbf{w}_{1,2,1}^T \| \ldots \| \mathbf{w}_{1,2,k}^T) \| \ldots \| (\mathbf{w}_{1,mk,1}^T \| \ldots \| \mathbf{w}_{1,mk,k}^T) \| \\
&(\mathbf{w}_{2,1,1}^T \| \ldots \| \mathbf{w}_{2,1,k}^T) \| (\mathbf{w}_{2,2,1}^T \| \ldots \| \mathbf{w}_{2,2,k}^T) \| \ldots \| (\mathbf{w}_{2,mk,1}^T \| \ldots \| \mathbf{w}_{2,mk,k}^T) \| \\
&(\mathbf{w}_{n,1,1}^T \| \ldots \| \mathbf{w}_{n,1,k}^T) \| (\mathbf{w}_{n,2,1}^T \| \ldots \| \mathbf{w}_{n,2,k}^T) \| \ldots \| (\mathbf{w}_{n,mk,1}^T \| \ldots \| \mathbf{w}_{n,mk,k}^T))^T,
\end{aligned}$$

where for each $(i,j,t) \in [n] \times [mk] \times [k]$: $\mathbf{w}_{i,j,t} = T_{b_{i,j},d_{i,t}}(\mathbf{v}_{i,j,t})$.

Observe that, for all $\mathbf{b} \in \{0,1\}^{nmk}, \mathbf{d} \in \{0,1\}^{nk}$, we have:

$$\mathbf{z} = \mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0) \iff P_{\mathbf{b},\mathbf{d}}(\mathbf{z}) = \mathsf{expand}^{\otimes}(\mathbf{x} \oplus \mathbf{b}, \mathbf{s}_0 \oplus \mathbf{d}). \tag{8.4}$$

Next, we recall the notations, extensions and permutations used in previous Stern-like protocols [LNSW13, LNW15, ELL+15, LLM+16a] for proving linear relations.

For any positive integer t, denote by $\mathfrak{S}_t$ the symmetric group of all permutations of t elements, by B_{2t} the set of all vectors in $\{0,1\}^{2t}$ having Hamming weight t, and by B_{3t} the set of all vectors in $\{-1,0,1\}^{3t}$ having exactly t coordinates equal to j, for each $j \in \{-1,0,1\}$. Note that for any $\phi \in \mathfrak{S}_{2t}$ and $\psi \in \mathfrak{S}_{3t}$, we have the following equivalences:

$$\mathbf{x} \in \mathsf{B}_{2t} \iff \phi(\mathbf{x}) \in \mathsf{B}_{2t} \quad \text{and} \quad \mathbf{y} \in \mathsf{B}_{3t} \iff \psi(\mathbf{y}) \in \mathsf{B}_{3t}. \tag{8.5}$$

The following extending procedures are defined for any positive integers t.

- $\mathsf{ExtendTwo}_t : \{0,1\}^t \to \mathsf{B}_{2t}$. On input vector $\mathbf{x}$ with Hamming weight w, it outputs
$$\mathbf{x}' = (\mathbf{x}^T \| \mathbf{1}^{t-w} \| \mathbf{0}^w)^T.$$
- $\mathsf{ExtendThree}_t : \{-1,0,1\}^t \to \mathsf{B}_{3t}$. On input vector $\mathbf{y}$ containing n_j coordinates equal to j for $j \in \{-1,0,1\}$, this procedure outputs the vector
$$\mathbf{y}' = (\mathbf{y}^T \| \mathbf{1}^{t-n_1} \| \mathbf{0}^{t-n_0} \| (-\mathbf{1})^{t-n_{-1}}).$$

We also use the following encoding and permutation to achieve fine-grained control over coordinates of binary witness-vectors.

- For any positive integer t, define the function encode_t that encodes vector $\mathbf{x} = (x_1, \ldots, x_t)^T \in \{0,1\}^t$ to vector $\mathsf{encode}_t(\mathbf{x}) = (\bar{x}_1, x_1, \ldots, \bar{x}_t, x_t)^T \in \{0,1\}^{2t}$.
- For any positive integer t and any vector $\mathbf{c} = (c_1, \ldots, c_t)^T \in \{0,1\}^t$, define the permutation $F_{\mathbf{c}}^{(t)}$ that transforms vector $\mathbf{v} = (v_1^{(0)}, v_1^{(1)}, \ldots, v_t^{(0)}, v_t^{(1)})^T \in \mathbb{Z}^{2t}$ into vector $F_{\mathbf{c}}^{(t)}(\mathbf{v}) = (v_1^{(c_1)}, v_1^{(\bar{c}_1)}, \ldots, v_t^{(c_t)}, v_t^{(\bar{c}_t)})^T$.

Note that the following equivalence holds for all $t, \mathbf{c}$:

$$\mathbf{y} = \mathsf{encode}_t(\mathbf{x}) \iff F_{\mathbf{c}}^{(t)}(\mathbf{y}) = \mathsf{encode}_t(\mathbf{x} \oplus \mathbf{c}). \tag{8.6}$$

To close this warm-up section, we remark that the equivalences observed in (8.4), (8.5) and (8.6) will play crucial roles in our zero-knowledge layer.

8.4 The Supporting Zero-Knowledge Layer

In this section, we first demonstrate how to prove in zero-knowledge that a given vector $\mathbf{b}$ is a correct LWE evaluation, i.e., $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$, where the hidden matrix $\mathbf{X}$ and vector $\mathbf{s}$ may satisfy additional conditions.

This sub-protocol, which we believe will have other applications, is one of the major challenges in our road towards the design of lattice-based group encryption. We then plug this building block into the big picture as described in Section 4.3, and construct the supporting zero-knowledge argument of knowledge (ZKAoK) for our group encryption scheme (Section 8.5).

8.4.1 Proving the LWE Relation With Hidden Matrices

Let n, m, q, β be positive integers where $\beta \ll q$, and let $k = \delta_{q-1} = \lceil \log_2 q \rceil$. We identify $\mathbb{Z}_q$ as the set $\{0, 1, \ldots, q-1\}$. We consider a zero-knowledge argument system allowing prover $\mathcal{P}$ to convince verifier $\mathcal{V}$ on input $\mathbf{b} \in \mathbb{Z}_q^m$ that $\mathcal{P}$ knows secret matrix $\mathbf{X} \in \mathbb{Z}_q^{m \times n}$, and vectors $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in [-\beta, \beta]^m$ such that:

$$\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q. \tag{8.7}$$

Moreover, the argument system should be readily extended to proving that $\mathbf{X}$ and $\mathbf{s}$ satisfy additional conditions, such as:

- The bits representing $\mathbf{X}$ are certified by an authority, and the prover also knows that secret signature-certificate.
- The (secret) hash of $\mathbf{X}$ is correctly encrypted to a given ciphertext.
- The LWE secret $\mathbf{s}$ is involved in other linear equations.

Let $q_1, \ldots, q_k \in \mathbb{Z}_q$ be the sequence of integers obtained by decomposing $q-1$ using the technique recalled in Section 8.3.1, and define the row vector $\mathbf{g} = (q_1, \ldots, q_k)$. Let $\mathbf{X} = [\mathbf{x}_1 | \ldots | \mathbf{x}_n] \in \mathbb{Z}_q^{m \times n}$ and $\mathbf{s} = (s_1, \ldots, s_n)^T$. For each index $i \in [n]$, let us consider $\mathsf{vdec}_{m,q-1}(\mathbf{x}_i) = (x_{i,1}, \ldots, x_{i,mk})^T \in \{0,1\}^{mk}$. Let

$$\mathsf{vdec}_{n,q-1}(\mathbf{s}) = (s_{1,1}, \ldots, s_{1,k}, s_{2,1}, \ldots, s_{2,k}, \ldots, s_{n,1}, \ldots s_{n,k})^T \in \{0,1\}^{nk}$$

and observe that $s_i = \mathbf{g} \cdot \mathsf{idec}_{q-1}(s_i) = \mathbf{g} \cdot (s_{i,1}, \ldots, s_{i,k})^T$ for each $i \in [n]$. We have:

$$\begin{aligned} \mathbf{X} \cdot \mathbf{s} &= \sum_{i=1}^{n} \mathbf{x}_i \cdot s_i = \sum_{i=1}^{n} \mathbf{H}_{m,q-1} \cdot \mathsf{vdec}_{m,q-1}(\mathbf{x}_i) \cdot s_i \\ &= \mathbf{H}_{m,q-1} \cdot \Big(\sum_{i=1}^{n} (x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^T \Big) \bmod q. \end{aligned}$$

Observe that, for each $i \in [n]$ and each $j \in [mk]$, we have

$$\begin{aligned} x_{i,j} \cdot s_i &= x_{i,j} \cdot \mathbf{g} \cdot (s_{i,1}, \ldots, s_{i,k})^T \\ &= (q_1, \ldots, q_k) \cdot (x_{i,j} \cdot s_{i,1}, \ldots, x_{i,j} \cdot s_{i,k})^T. \end{aligned}$$

We now extend vector $(q_1, q_2, \ldots, q_k)$ to $\mathbf{g}' = (0,0,0,q_1,0,0,0,q_2,\ldots,0,0,0,q_k) \in \mathbb{Z}_q^{4k}$. For all $(i,j) \in [n] \times [mk]$, we have:

$$x_{i,j} \cdot s_i = \mathbf{g}' \cdot (\mathsf{ext}^T(x_{i,j}, s_{i,1}) \| \ldots \| \mathsf{ext}^T(x_{i,j}, s_{i,k}))^T.$$

Let us define the matrices

$$\mathbf{Q}_0 := \mathbf{I}_{mk} \otimes \mathbf{g}' = \begin{bmatrix} \mathbf{g}' & & & \\ & \mathbf{g}' & & \\ & & \ddots & \\ & & & \mathbf{g}' \end{bmatrix} \in \mathbb{Z}_q^{mk \times 4mk^2}, \tag{8.8}$$

and $\widehat{\mathbf{Q}} = [\overbrace{\mathbf{Q}_0|\ldots|\mathbf{Q}_0}^{n \text{ times}}] \in \mathbb{Z}_q^{mk \times 4nmk^2}$. For each $i \in [n]$, define

$$\mathbf{y}_i = (\mathsf{ext}^T(x_{i,1}, s_{i,1}) \| \ldots \| \mathsf{ext}^T(x_{i,1}, s_{i,k}))^T \| \mathsf{ext}^T(x_{i,2}, s_{i,1}) \| \ldots \| \mathsf{ext}^T(x_{i,2}, s_{i,k}) \\ \| \ldots \| \mathsf{ext}^T(x_{i,mk}, s_{i,1} \| \ldots \| \mathsf{ext}^T(x_{i,mk}, s_{i,k}))^T \in \{0,1\}^{4mk^2}.$$

Then, for all $i \in [n]$, we have: $(x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^T = \mathbf{Q}_0 \cdot \mathbf{y}_i$. Now, we note that

$$(\mathbf{y}_1^T \| \ldots \| \mathbf{y}_n^T)^T = \mathsf{expand}^{\otimes}(\mathsf{mdec}_{n,m,q}(\mathbf{X}), \mathsf{vdec}_{n,q-1}(\mathbf{s})),$$

and

$$\sum_{i=1}^{n} (x_{i,1} \cdot s_i, \ldots, x_{i,mk} \cdot s_i)^T \\ = \sum_{i=1}^{n} \mathbf{Q}_0 \cdot \mathbf{y}_i = \widehat{\mathbf{Q}} \cdot \mathsf{expand}^{\otimes}(\mathsf{mdec}_{n,m,q}(\mathbf{X}), \mathsf{vdec}_{n,q-1}(\mathbf{s})). \tag{8.9}$$

Letting $\mathbf{Q} = \mathbf{H}_{m,q-1} \cdot \widehat{\mathbf{Q}} \in \mathbb{Z}_q^{m \times 4nmk^2}$ and left-multiplying (8.9) by $\mathbf{H}_{m,q-1}$, we obtain the equation:

$$\mathbf{X} \cdot \mathbf{s} = \mathbf{Q} \cdot \mathsf{expand}^{\otimes}(\mathsf{mdec}_{n,m,q}(\mathbf{X}), \mathsf{vdec}_{n,q-1}(\mathbf{s})) \bmod q.$$

This means that the task of proving knowledge of $(\mathbf{X}, \mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^n \times [-\beta, \beta]^m$ such that $\mathbf{b} = \mathbf{X} \cdot \mathbf{s} + \mathbf{e} \bmod q$ boils down to proving knowledge of $\mathbf{z} \in \{0,1\}^{4nmk^2}$, $\mathbf{x} \in \{0,1\}^{nmk}$, $\mathbf{s}_0 \in \{0,1\}^{nk}$ and a short $\mathbf{e} \in \mathbb{Z}^m$ such that

$$\mathbf{b} = \mathbf{Q} \cdot \mathbf{z} + \mathbf{I}_m \cdot \mathbf{e} \bmod q \qquad \text{and} \qquad \mathbf{z} = \mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0).$$

As the knowledge of small-norm vector $\mathbf{e}$ can easily be proven with Stern-like protocol (e.g., [LNSW13]), the challenging part is to prove in zero-knowledge the constraint "$\mathbf{z} = \mathsf{expand}^{\otimes}(\mathbf{x}, \mathbf{s}_0)$". To this end, we will use the following permuting technique inspired by the equivalence of equation (8.4). We sample uniformly random $\mathbf{d}_x \in \{0,1\}^{nmk}$ and $\mathbf{d}_s \in \{0,1\}^{nk}$, send $\mathbf{x}' = \mathbf{x} \oplus \mathbf{d}_x$ and $\mathbf{s}' = \mathbf{s}_0 \oplus \mathbf{d}_s$ to the verifier, and let the latter check that $P_{\mathbf{d}_x,\mathbf{d}_s}(\mathbf{z}) = \mathsf{expand}^{\otimes}(\mathbf{x}', \mathbf{s}')$. This will be sufficient to convince the verifier that the original vector $\mathbf{z}$ satisfies the required constraint. The crucial point is that no additional information about $\mathbf{x}$ and $\mathbf{s}_0$ is leaked, since these binary vectors are perfectly hidden under the "one-time pad" $\mathbf{d}_x$ and $\mathbf{d}_s$, respectively.

In the framework of Stern's protocol, the idea of using "one-time-pad" permutations further allows us to prove that $\mathbf{x}$ and $\mathbf{s}_0$ satisfy additional conditions, i.e., they appear in other equations. This is done by first setting up an equivalence similar to (8.4) in the places where these objects appear, and then, using the same "one-time pad" for each of them in all appearances. We will explain in detail how this technique can be realized in the next subsection.

8.5 Our Lattice-Based Group Encryption Scheme

To build a GE scheme using our zero-knowledge argument system, we need to choose a specific key-private CCA2-secure encryption scheme. The first idea is to use the CCA2-secure public-key cryptosystem which is implied by the Agrawal-Boneh-Boyen identity-based encryption (IBE) scheme [ABB10] (which is recalled in Section 8.2.1) via the Canetti-Halevi-Katz (CHK) transformation [CHK04]. The ABB scheme is a natural choice since it

has pseudo-random ciphertexts (which implies the key-privacy [BBDP01] when the CHK paradigm is applied) and provides one of the most efficient CCA2 cryptosystem based on the hardness of LWE in the standard model. One difficulty is that the Kiayias-Tsiounis-Yung model [KTY07] requires that certified public keys be valid public keys (i.e., which have a matching secret key). When new group members join the system and request a certificate for their public key $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$, a direct use of the ABB/CHK technique would incur of proof of existence of a GPV trapdoor [GPV08] corresponding to $\mathbf{B}_{\mathsf{U}}$ (i.e., a small-norm matrix $\mathbf{T}_{\mathbf{B}_{\mathsf{U}}} \in \mathbb{Z}^{\bar{m} \times \bar{m}}$ s.t. $\mathbf{B} \cdot \mathbf{T}_{\mathbf{B}_{\mathsf{U}}} = \mathbf{0}^n \bmod q$). While the techniques of Peikert and Vaikuntanathan [PV08] would provide a solution to this problem (as they allow proving that $\mathbf{T}_{\mathbf{B}_{\mathsf{U}}} \in \mathbb{Z}^{\bar{m} \times \bar{m}}$ has full-rank), we found it simpler to rely on the trapdoor mechanism of Micciancio and Peikert [MP12].

If we assume public parameters containing a random matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$, each user's public key can consist of a matrix $\mathbf{B}_{\mathsf{U}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$, where $\mathbf{T}_{\mathsf{U}} \in \mathbb{Z}^{m \times \bar{m}}$ is a small-norm matrix whose calms are sampled from a discrete Gaussian distribution. Note that, if $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$ is uniformly distributed, then [GPV08, Lemma 5.1] ensures that, with overwhelming probability, any matrix $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ has an underlying small-norm matrix satisfying $\mathbf{B}_{\mathsf{U}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{U}} \bmod q$. This simplifies the joining procedure by eliminating the need for proofs of public key validity.

In the encryption algorithm, the sender computes a dual Regev encryption [GPV08] of the witness $\mathbf{w} \in \{0,1\}^m$ using a matrix $[\bar{\mathbf{A}} \mid \mathbf{B}_{\mathsf{U}} + \mathsf{FRD}(\mathsf{vk}) \cdot \mathbf{G}] \in \mathbb{Z}_q^{n \times (m+\bar{m})}$ such that: (i) $\mathsf{vk} \in \mathbb{Z}_q^n$ is the verification key of a one-time signature; (ii) $\mathsf{FRD} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n \times n}$ is the full-rank difference[1] function of [ABB10]; (iii) $\mathbf{G} = \mathbf{I}_n \otimes [1|2|\ldots|2^{k-1}] \in \mathbb{Z}_q^{n \times \bar{m}}$ is the gadget matrix of [MP12]. Given that $\mathbf{G}$ has a publicly known trapdoor allowing to sample short vectors in $\Lambda_q^{\perp}(\mathbf{G})$, the user can use his private key $\mathbf{T}_{\mathsf{U}} \in \mathbb{Z}^{m \times \bar{m}}$ to decrypt by running the SampleRight algorithm of Lemma 3.7.

Having encrypted the witness $\mathbf{w} \in \{0,1\}^m$ by running the ABB encryption algorithm, the sender proceeds by encrypting a hash value of $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ under the public key $\mathbf{B}_{\mathsf{OA}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{OA}} \in \mathbb{Z}_q^{n \times \bar{m}}$ of the opening authority. The latter hash value is obtained as a bit-wise decomposition of $\mathbf{F} \cdot \mathsf{mdec}_{n,m,q}(\mathbf{B}_{\mathsf{U}}^T) \in \mathbb{Z}_q^{2n}$, where $\mathbf{F} \in \mathbb{Z}_q^{2n \times n\bar{m}\lceil \log q \rceil}$ is a random public matrix and $\mathsf{mdec}_{n,m,q}(\mathbf{B}_{\mathsf{U}}^T) \in \{0,1\}^{n\bar{m}\lceil \log q \rceil}$ denotes an entry-wise binary decomposition of the matrix $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$.

By combining our new argument for quadratic relations and the extensions of Stern's protocol suggested in [LNW15, LLM+16a], we are able to prove that some component of the ciphertext is of the form $\mathbf{c} = \mathbf{B}_{\mathsf{U}}^T \cdot \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^{\bar{m}}$, for some $\mathbf{s} \in \mathbb{Z}_q^n$ and a small-norm $\mathbf{e} \in \mathbb{Z}^{\bar{m}}$ while also arguing possession of a signature on the binary decomposition $\mathsf{mdec}_{n,m,q}(\mathbf{B}_{\mathsf{U}}^T) \in \{0,1\}^{n\bar{m}\lceil \log q \rceil}$ of $\mathbf{B}_{\mathsf{U}}^T$. For this purpose, we use a variant of a signature scheme due to Böhl *et al.*'s signature [BHJ+15] which was described in Chapter 7 (and of which a description is given in Section 7.1). At the same time, the prover $\mathcal{P}$ can also argue that a hash value of $\mathsf{mdec}_{n,m,q}(\mathbf{B}_{\mathsf{U}}^T)$ is properly encrypted under the OA's public key using the ABB encryption scheme.

[1] This means that, for any two distinct one-time verification keys $\mathsf{vk}, \mathsf{vk}' \in \mathbb{Z}_q^n$, the difference $\mathsf{FRD}(\mathsf{vk}) - \mathsf{FRD}(\mathsf{vk}') \in \mathbb{Z}_q^{n \times n}$ is invertible over $\mathbb{Z}_q$.

8.5.1 Description of the Scheme

Our GE scheme allows encrypting witnesses for the ISIS relation (as in Definition 3.8) $\mathrm{R}_{\mathsf{ISIS}}(n, m, q, 1)$, which consists of pairs $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}) \in (\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n) \times \{0,1\}^m$ satisfying $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$. This relation is in the same spirit as the one of Kiayias, Tsiounis and Yung [KTY07], who consider the verifiable encryption of discrete logarithms. While the construction of [KTY07] allow verifiably encrypting discrete-logarithm-type secret keys under the public key of some anonymous trusted third party, our construction makes it possible to encrypt GPV-type secret keys [GPV08].

$\mathsf{SETUP}_{\mathsf{init}}(1^\lambda)$**:** This algorithm performs the following:

1. Choose integers $n = \mathcal{O}(\lambda)$, prime $q = \widetilde{\mathcal{O}}(n^4)$, and let $k = \lceil \log_2 q \rceil$, $\bar{m} = nk$ and $m = 2\bar{m} = 2nk$. Choose a B-bounded distribution χ over $\mathbb{Z}$ for some $B = \sqrt{n}\omega(\log n)$.
2. Choose a Gaussian parameter $\sigma = \Omega(\sqrt{n \log q} \log n)$. Let $\beta = \sigma\omega(\log n)$ be the upper bound of samples from $D_{\mathbb{Z},\sigma}$.
3. Select integers $\ell = \ell(\lambda)$ which determines the maximum expected group size 2^ℓ, and $\kappa = \omega(\log \lambda)$ (the number of protocol repetitions).
4. Select a strongly unforgeable one-time signature $\mathcal{OTS} = (\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$. We assume that the verification keys live in $\mathbb{Z}_q^n$.
5. Select public parameters $\mathsf{COM}_{\mathsf{par}}$ for a statistically-hiding commitment scheme like [KTX08]. This commitment will serve as a building block for the zero-knowledge argument system used in $\langle \mathcal{P}, \mathcal{V} \rangle$.
6. Let $\mathsf{FRD} : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n\times n}$ be the full-rank difference mapping from [ABB10].
7. Pick a random matrix $\mathbf{F} \leftarrow \mathbb{Z}_q^{2n\times n\bar{m}k}$, which will be used to hash users' public keys from $\mathbb{Z}_q^{n\times\bar{m}}$ to $\mathbb{Z}_q^n$.
8. Let $\mathbf{G} \in \mathbb{Z}_q^{n\times\bar{m}}$ be the gadget matrix $\mathbf{G} = \mathbf{I}_n \otimes \begin{bmatrix} 1 & 2 & \dots & 2^{k-1} \end{bmatrix}$ of [MP12]. Pick matrices $\bar{\mathbf{A}}, \mathbf{U} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$ and $\mathbf{V} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$. Looking ahead, $\mathbf{U}$ will be used to encrypt for the receiver while $\mathbf{V}$ will be used to encrypt the user's public key under the OA's public key. As for $\bar{\mathbf{A}}$, it will be used in two instances of the ABB encryption scheme [ABB10].

Output

$$\mathsf{par} \quad = \quad \{\lambda, n, q, k, m, B, \chi, \sigma, \beta, \ell, \kappa, \mathcal{OTS}, \mathsf{COM}_{\mathsf{par}}, \mathsf{FRD}, \bar{\mathbf{A}}, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{V}\}.$$

$\langle \mathcal{G}_r, \mathsf{sample}_R \rangle$**:** Algorithm $\mathcal{G}_r(1^\lambda, 1^n, 1^m)$ proceeds by sampling a random matrix $\mathbf{A}_R \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$ and outputting $(\mathsf{pk}_R, \mathsf{sk}_R) = (\mathbf{A}_R, \varepsilon)$. On input of a public key $\mathsf{pk}_R = \mathbf{A}_R \in \mathbb{Z}_q^{n\times m}$ for the relation $\mathrm{R}_{\mathsf{ISIS}}$, algorithm sample_R picks $\mathbf{w} \hookleftarrow \mathcal{U}(\{0,1\}^m)$ and outputs a pair $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w})$, where $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \in \mathbb{Z}_q^n$.

$\mathsf{SETUP}_{\mathsf{GM}}(\mathsf{par})$**:** The GM generates $(\mathsf{sk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{GM}}) \leftarrow \mathsf{Keygen}(1^\lambda, q, n, m, \ell, \sigma)$ as a key pair for the SIS-based signature scheme of [LLM+16a] (as recalled in Section 7.1). This key pair consists of $\mathsf{sk}_{\mathsf{GM}} := \mathbf{T}_{\mathbf{A}}$ and

$$\mathsf{pk}_{\mathsf{GM}} := \Big(\mathbf{A}, \mathbf{A}_0, \dots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n\times m},\ \mathbf{D}_0, \mathbf{D}_1 \in \mathbb{Z}_q^{n\times m}, \mathbf{D} \in \mathbb{Z}_q^{n\times\bar{m}}, \mathbf{u} \in \mathbb{Z}_q^n\Big). \tag{8.10}$$

$\mathsf{SETUP}_{\mathsf{OA}}(\mathsf{par})$**:** The OA samples a small-norm matrix $\mathbf{T}_{\mathsf{OA}} \leftarrow D^{\bar{m}}_{\mathbb{Z}^m,\sigma}$ in $\mathbb{Z}^{m\times\bar{m}}$ to obtain a statistically uniform $\mathbf{B}_{\mathsf{OA}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{OA}} \in \mathbb{Z}_q^{n\times\bar{m}}$. The OA's key pair consists of $(\mathsf{sk}_{\mathsf{OA}}, \mathsf{pk}_{\mathsf{OA}}) = (\mathbf{T}_{\mathsf{OA}}, \mathbf{B}_{\mathsf{OA}})$.

JOIN: The prospective user U and the GM interact in the following protocol.

1. U first samples $\mathbf{T}_{\mathsf{U}} \leftarrow D^{\bar{m}}_{\mathbb{Z}^m,\sigma}$ in $\mathbb{Z}^{m\times\bar{m}}$ to compute a statistically uniform matrix $\mathbf{B}_{\mathsf{U}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$. The prospective user defines his key pair as $(\mathsf{pk}_{\mathsf{U}}, \mathsf{sk}_{\mathsf{U}}) = (\mathbf{B}_{\mathsf{U}}, \mathbf{T}_{\mathsf{U}})$ and sends $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}}$ to the GM.
2. Upon receiving a public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ from the user, the GM certifies pk_U via the following steps:
   a. Compute $\mathbf{h}_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U}}^T) \in \mathbb{Z}_q^{2n}$ as a hash value of the public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$.
   b. Use the trapdoor $\mathsf{sk}_{\mathsf{GM}} = \mathbf{T}_{\mathbf{A}}$ to generate a signature

$$\mathsf{cert}_{\mathsf{U}} = (\tau, \mathbf{d}, \mathbf{r}) \in \{0,1\}^{\ell} \times [-\beta,\beta]^{2m} \times [-\beta,\beta]^{m}, \tag{8.11}$$

   satisfying

$$\begin{aligned}&[\mathbf{A} \mid \sum_{j=1}^{\ell} \tau[j]\mathbf{A}_j] \cdot \mathbf{d} \\ &\quad = \mathbf{u} + \mathbf{D} \cdot \mathsf{vdec}_{n,q-1}(\mathbf{D}_0 \cdot \mathbf{r} + \mathbf{D}_1 \cdot \mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U}})) \bmod q,\end{aligned} \tag{8.12}$$

   where $\tau = \tau[1]\ldots\tau[\ell] \in \{0,1\}^{\ell}$, as in the scheme of Section 7.1.

   U verifies that $\mathsf{cert}_{\mathsf{U}}$ is tuple of the form (8.11) satisfying (8.12) and returns $\perp$ if it is not the case. The GM stores $(\mathsf{pk}_{\mathsf{U}}, \mathsf{cert}_{\mathsf{U}})$ in the user database database and returns the certificate $\mathsf{cert}_{\mathsf{U}}$ to the new user U.

$\mathsf{ENC}(\mathsf{pk}_{\mathsf{GM}}, \mathsf{pk}_{\mathsf{OA}}, \mathsf{pk}_{\mathsf{U}}, \mathsf{cert}_{\mathsf{U}}, \mathbf{w}, L)$**:** To encrypt a witness $\mathbf{w} \in \{0,1\}^m$ for $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w})$ in relation $\mathrm{R}_{\mathsf{ISIS}}(n, m, q, 1)$ (i.e., $\mathbf{A}_R \cdot \mathbf{w} = \mathbf{u}_R \bmod q$), parse $\mathsf{pk}_{\mathsf{GM}}$ as in (8.10), $\mathsf{pk}_{\mathsf{OA}}$ as $\mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n\times\bar{m}}$, pk_{U} as $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ and $\mathsf{cert}_{\mathsf{U}}$ as in (8.11).

1. Generate a one-time key-pair $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{Gen}(1^{\lambda})$, where $\mathsf{vk} \in \mathbb{Z}_q^n$.
2. Compute a full-rank-difference hash $\mathbf{H}_{\mathsf{vk}} = \mathsf{FRD}(\mathsf{vk}) \in \mathbb{Z}_q^{n\times n}$ of the one-time verification key $\mathsf{vk} \in \mathbb{Z}_q^n$.
3. Encrypt the witness $\mathbf{w} \in \{0,1\}^m$ under U's public key $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ using the tag vk by taking the following steps:
   a. Sample $\mathbf{s}_{\mathsf{rec}} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, $\mathbf{R}_{\mathsf{rec}} \leftarrow D^{m\times\bar{m}}_{\mathbb{Z},\sigma}$ and $\mathbf{x}_{\mathsf{rec}}, \mathbf{y}_{\mathsf{rec}} \leftarrow \chi^m$. Compute $\mathbf{z}_{\mathsf{rec}} = \mathbf{R}_{\mathsf{rec}}^T \cdot \mathbf{y}_{\mathsf{rec}} \in \mathbb{Z}^{\bar{m}}$.
   b. Compute

$$\begin{cases} \mathbf{c}_{\mathsf{rec}}^{(1)} = \bar{\mathbf{A}}^T \cdot \mathbf{s}_{\mathsf{rec}} + \mathbf{y}_{\mathsf{rec}} \bmod q \\ \mathbf{c}_{\mathsf{rec}}^{(2)} = (\mathbf{B}_{\mathsf{U}} + \mathbf{H}_{\mathsf{vk}} \cdot \mathbf{G})^T \cdot \mathbf{s}_{\mathsf{rec}} + \mathbf{z}_{\mathsf{rec}} \bmod q; \\ \mathbf{c}_{\mathsf{rec}}^{(3)} = \mathbf{U}^T \cdot \mathbf{s}_{\mathsf{rec}} + \mathbf{x}_{\mathsf{rec}} + \mathbf{w} \cdot \left\lfloor \frac{q}{2} \right\rfloor, \end{cases} \tag{8.13}$$

and let $\mathbf{c}_{\mathsf{rec}} = (\mathbf{c}_{\mathsf{rec}}^{(1)}, \mathbf{c}_{\mathsf{rec}}^{(2)}, \mathbf{c}_{\mathsf{rec}}^{(3)}) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^m$, which forms an ABB ciphertext [ABB10] for the tag $\mathsf{vk} \in \mathbb{Z}_q^n$.

4. Encrypt the decomposition $\mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U}}) \in \{0,1\}^m$ of the hashed pk_{U} under the OA's public key $\mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n \times \bar{m}}$ w.r.t. the tag $\mathsf{vk} \in \mathbb{Z}_q^n$. Namely, conduct the following steps:
   a. Sample $\mathbf{s}_{\mathsf{oa}} \leftarrow \mathcal{U}(\mathbb{Z}_q^n)$, $\mathbf{R}_{\mathsf{oa}} \leftarrow D_{\mathbb{Z},\sigma}^{m \times \bar{m}}$, $\mathbf{x}_{\mathsf{oa}} \leftarrow \chi^m, \mathbf{y}_{\mathsf{oa}} \leftarrow \chi^m$. Set $\mathbf{z}_{\mathsf{oa}} = \mathbf{R}_{\mathsf{oa}}^T \cdot \mathbf{y}_{\mathsf{oa}} \in \mathbb{Z}^{\bar{m}}$.
   b. Compute

$$\begin{cases} \mathbf{c}_{\mathsf{oa}}^{(1)} = \bar{\mathbf{A}}^T \cdot \mathbf{s}_{\mathsf{oa}} + \mathbf{y}_{\mathsf{oa}} \bmod q; \\ \mathbf{c}_{\mathsf{oa}}^{(2)} = (\mathbf{B}_{\mathsf{OA}} + \mathbf{H}_{\mathsf{vk}} \cdot \mathbf{G})^T \cdot \mathbf{s}_{\mathsf{oa}} + \mathbf{z}_{\mathsf{oa}} \bmod q; \\ \mathbf{c}_{\mathsf{oa}}^{(3)} = \mathbf{V}^T \cdot \mathbf{s}_{\mathsf{oa}} + \mathbf{x}_{\mathsf{oa}} + \mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U}}) \cdot \left\lfloor \frac{q}{2} \right\rfloor, \end{cases} \tag{8.14}$$

   and let $\mathbf{c}_{\mathsf{oa}} = (\mathbf{c}_{\mathsf{oa}}^{(1)}, \mathbf{c}_{\mathsf{oa}}^{(2)}, \mathbf{c}_{\mathsf{oa}}^{(3)}) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{\bar{m}} \times \mathbb{Z}_q^m$.
5. Compute a one-time signature $\Sigma = \mathsf{Sig}(\mathsf{sk}, (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, L))$.

Output the ciphertext

$$\boldsymbol{\Psi} = (\mathsf{vk}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, \Sigma). \tag{8.15}$$

and the state information $coins_{\boldsymbol{\Psi}} = (\mathbf{s}_{\mathsf{rec}}, \mathbf{R}_{\mathsf{rec}}, \mathbf{x}_{\mathsf{rec}}, \mathbf{y}_{\mathsf{rec}}, \mathbf{s}_{\mathsf{oa}}, \mathbf{R}_{\mathsf{oa}}, \mathbf{x}_{\mathsf{oa}}, \mathbf{y}_{\mathsf{oa}})$.

$\mathsf{DEC}(\mathsf{sk}_{\mathsf{U}}, \boldsymbol{\Psi}, L)$: The decryption algorithm proceeds as follows:

1. If $\mathsf{Ver}(\mathsf{vk}, \Sigma, (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, L)) = 0$, return $\perp$. Otherwise, parse the secret key sk_{U} as $\mathbf{T}_{\mathsf{U}} \in \mathbb{Z}^{m \times \bar{m}}$ and the ciphertext $\boldsymbol{\Psi}$ as in (8.15). Define the matrix $\mathbf{B}_{\mathsf{vk}} = \mathbf{B}_{\mathsf{U}} + \mathsf{FRD}(\mathsf{vk}) \cdot \mathbf{G} \in \mathbb{Z}_q^{n \times \bar{m}}$.
2. Decrypt $\mathbf{c}_{\mathsf{rec}}$ using a decryption key for the tag $\mathsf{vk} \in \mathbb{Z}^n$. Namely,
   a. Define $\mathbf{B}_{\mathsf{U},\mathsf{vk}} = [\bar{\mathbf{A}} | \mathbf{B}_{\mathsf{vk}}] = [\bar{\mathbf{A}} | \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{U}} + \mathsf{FRD}(\mathsf{vk}) \cdot \mathbf{G}] \in \mathbb{Z}_q^{n \times (m+\bar{m})}$. Using $\mathbf{T}_{\mathsf{U}}$ and the publicly known trapdoor $\mathbf{T}_{\mathbf{G}}$ of $\mathbf{G}$, compute a small-norm matrix $\mathbf{E}_{\mathsf{vk}} \in \mathbb{Z}^{(m+\bar{m}) \times m}$ such that $\mathbf{B}_{\mathsf{U},\mathsf{vk}} \cdot \mathbf{E}_{\mathsf{vk}} = \mathbf{U} \bmod q$ by running the SampleRight algorithm of Lemma 3.7.
   b. Compute

$$\mathbf{w} = \left\lfloor \left(\mathbf{c}_{\mathsf{rec}}^{(3)} - \mathbf{E}_{\mathsf{vk}}^T \cdot \begin{bmatrix} \mathbf{c}_{\mathsf{rec}}^{(1)} \\ \mathbf{c}_{\mathsf{rec}}^{(2)} \end{bmatrix} \right) / \left\lfloor \frac{q}{2} \right\rfloor \right\rceil \in \mathbb{Z}^m$$

   and return the obtained $\mathbf{w} \in \{0,1\}^m$.

$\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \boldsymbol{\Psi}, L)$: The opening algorithm proceeds as follows:

1. If $\mathsf{Ver}(\mathsf{vk}, \Sigma, (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}), L) = 0$, then return $\perp$. Otherwise, parse $\mathsf{sk}_{\mathsf{OA}}$ as $\mathbf{T}_{\mathsf{OA}} \in \mathbb{Z}^{m \times \bar{m}}$ and the ciphertext $\boldsymbol{\Psi}$ as in (8.15).

2. Decrypt $\mathbf{c}_{\mathsf{oa}}$ using a decryption key for the tag $\mathsf{vk} \in \mathbb{Z}_q^n$ in the same way as in the decryption algorithm. That is, do the following:

   a. Define the matrix $\mathbf{B}_{\mathsf{OA},\mathsf{vk}} = [\bar{\mathbf{A}}|\mathbf{B}_{\mathsf{OA}} + \mathsf{FRD}(\mathsf{vk}) \cdot \mathbf{G}] \in \mathbb{Z}_q^{n \times (m+\bar{m})}$. Use $\mathbf{T}_{\mathsf{OA}}$ to compute a small-norm $\mathbf{E}_{\mathsf{OA},\mathsf{vk}} \in \mathbb{Z}^{(m+\bar{m}) \times m}$ satisfying $\mathbf{B}_{\mathsf{OA},\mathsf{vk}} \cdot \mathbf{E}_{\mathsf{OA},\mathsf{vk}} = \mathbf{V} \bmod q$.

   b. Compute

$$\mathbf{h} = \left\lfloor \left(\mathbf{c}_{\mathsf{oa}}^{(3)} - \mathbf{E}_{\mathsf{OA},\mathsf{vk}}^T \cdot \begin{bmatrix} \mathbf{c}_{\mathsf{oa}}^{(1)} \\ \mathbf{c}_{\mathsf{oa}}^{(2)} \end{bmatrix} \right) / \left\lfloor \frac{q}{2} \right\rfloor \right\rceil \in \{0,1\}^m$$

   and $\mathbf{h}'_{\mathsf{U}} = \mathbf{H}_{2n,q-1} \cdot \mathbf{h} \in \mathbb{Z}_q^{2n}$.

3. Look up database to find a public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ that hashes to $\mathbf{h}'_{\mathsf{U}} \in \mathbb{Z}_q^{2n}$ (i.e., such that $\mathbf{h}'_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U}}^T)$). If more than one such key exists, return $\perp$. If only one key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ satisfies $\mathbf{h}'_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U}}^T)$, return that key pk_{U}. In any other situation, return $\perp$.

$\langle \mathcal{P}, \mathcal{V} \rangle$**:** The common input consists of par and $\mathsf{pk}_{\mathsf{GM}}$ as specified above, and $(\mathbf{A}_R, \mathbf{u}_R)$ in $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$, $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n \times \bar{m}}$, and a ciphertext $\boldsymbol{\Psi}$ as in (8.15). Both parties compute $\mathbf{B}_{\mathsf{OA},\mathsf{vk}} = [\bar{\mathbf{A}}|\mathbf{B}_{\mathsf{OA}} + \mathsf{FRD}(\mathsf{vk}) \cdot \mathbf{G}]$ as specified above. The prover's secret input consists of a witness $\mathbf{w} \in \{0,1\}^m$, $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}}$, $\mathsf{cert}_{\mathsf{U}} = (\tau, \mathbf{d}, \mathbf{r}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$, and the random coins $coins_{\boldsymbol{\Psi}} = (\mathbf{s}_{\mathsf{rec}}, \mathbf{R}_{\mathsf{rec}}, \mathbf{x}_{\mathsf{rec}}, \mathbf{y}_{\mathsf{rec}}, \mathbf{s}_{\mathsf{oa}}, \mathbf{R}_{\mathsf{oa}}, \mathbf{x}_{\mathsf{oa}}, \mathbf{y}_{\mathsf{oa}})$ used to generate $\boldsymbol{\Psi}$.

The prover's goal is to convince the verifier in zero-knowledge that his secret input satisfies the following:

1. $\mathbf{A}_R \cdot \mathbf{w} = \mathbf{u}_R \bmod q$.
2. $\mathbf{h}_{\mathbf{M}} = \mathbf{F} \cdot \mathsf{mdec}_{n,m,q}(\mathbf{M}) \bmod q$.
3. Conditions (8.11) and (8.12) hold.
4. Vectors $\mathbf{x}_{\mathsf{rec}}, \mathbf{y}_{\mathsf{rec}}, \mathbf{x}_{\mathsf{oa}}, \mathbf{y}_{\mathsf{oa}}$ have infinity norms bounded by B, and vectors $\mathbf{z}_{\mathsf{rec}}, \mathbf{z}_{\mathsf{oa}}$ have infinity norms bounded by $\beta m B$.
5. Equations in (8.13) and (8.14) hold.

To this end $\mathcal{P}$ conducts the following steps.

1. Decompose the matrix $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ into $\mathbf{b}_{\mathsf{U}} = \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U}}^T) \in \{0,1\}^{n\bar{m}k}$ and the vectors $\mathbf{s}_{\mathsf{rec}}, \mathbf{s}_{\mathsf{oa}} \in \mathbb{Z}_q^n$ into $\mathbf{s}_{0,\mathsf{rec}} = \mathsf{vdec}_{n,q-1}(\mathbf{s}_{\mathsf{rec}}) \in \{0,1\}^{nk}$ and $\mathbf{s}_{0,\mathsf{oa}} = \mathsf{vdec}_{n,q-1}(\mathbf{s}_{\mathsf{oa}}) \in \{0,1\}^{nk}$. Combine the first two binary vectors into $\mathbf{z}_{\boldsymbol{\Psi}} = \mathsf{expand}^{\otimes}(\mathbf{b}_{\mathsf{U}}, \mathbf{s}_{0,\mathsf{rec}}) \in \{0,1\}^{4n\bar{m}k^2}$. Define

$$\mathbf{Q} = \mathbf{H}_{\bar{m},q-1} \cdot [\overbrace{\mathbf{Q}_0|\ldots|\mathbf{Q}_0}^{n \text{ times}}] \in \mathbb{Z}_q^{\bar{m} \times 4n\bar{m}k^2},$$

where $\mathbf{Q}_0 = \mathbf{I}_{\bar{m}k} \otimes \mathbf{g}' \in \mathbb{Z}_q^{\bar{m}k \times 4\bar{m}k^2}$ is the matrix defined as in (8.8).

2. Generate a zero-knowledge argument of knowledge of

$$\begin{cases} \tau \in \{0,1\}^{\ell},\ \mathbf{d} = [\mathbf{d}_1^T | \mathbf{d}_2^T]^T \in [-\beta,\beta]^{2m},\ \mathbf{r} \in [-\beta,\beta]^m \\ \mathbf{t}_{\mathsf{U}} \in \{0,1\}^m,\ \mathbf{w}_{\mathsf{U}} \in \{0,1\}^{\bar{m}} \\ \mathbf{b}_{\mathsf{U}} \in \{0,1\}^{n\bar{m}k},\ \mathbf{s}_{0,\mathsf{rec}} \in \{0,1\}^{nk},\ \mathbf{z}_{\boldsymbol{\Psi}} = \mathsf{expand}^{\otimes}(\mathbf{b}_{\mathsf{U}}, \mathbf{s}_{0,\mathsf{rec}}) \\ \mathbf{x}_{\mathsf{rec}},\ \mathbf{y}_{\mathsf{rec}} \in [-B,B]^m,\ \mathbf{z}_{\mathsf{rec}} \in [-\beta mB, \beta mB]^{\bar{m}},\ \mathbf{w} \in \{0,1\}^m, \\ \mathbf{s}_{0,\mathsf{oa}} \in \{0,1\}^{nk},\ \mathbf{x}_{\mathsf{oa}},\ \mathbf{y}_{\mathsf{oa}} \in [-B,B]^m,\ \mathbf{z}_{\mathsf{oa}} \in [-\beta mB, \beta mB]^{\bar{m}} \end{cases}$$

such that the following system of 10 equations holds:

$$\begin{cases} \mathbf{u} = [\mathbf{A}|\mathbf{A}_0|\mathbf{A}_1|\ldots|\mathbf{A}_{\ell}] \cdot \begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \\ \tau[1]\cdot\mathbf{d}_2 \\ \vdots \\ \tau[\ell]\cdot\mathbf{d}_2 \end{pmatrix} + (-\mathbf{D})\cdot\mathbf{w}_{\mathsf{U}} \bmod q, \\ \mathbf{0} = \mathbf{H}_{n,q-1}\cdot\mathbf{w}_{\mathsf{U}} + (-\mathbf{D}_0)\cdot\mathbf{r} + (-\mathbf{D}_1)\cdot\mathbf{t}_{\mathsf{U}} \bmod q, \\ \mathbf{0} = \mathbf{H}_{2n,q-1}\cdot\mathbf{t}_{\mathsf{U}} + (-\mathbf{F})\cdot\mathbf{b}_{\mathsf{U}} \bmod q, \\ \mathbf{c}_{\mathsf{rec}}^{(1)} = (\bar{\mathbf{A}}^T\cdot\mathbf{H}_{n,q-1})\cdot\mathbf{s}_{0,\mathsf{rec}} + \mathbf{I}_m\cdot\mathbf{y}_{\mathsf{rec}} \bmod q, \\ \mathbf{c}_{\mathsf{rec}}^{(2)} = \mathbf{Q}\cdot\mathbf{z}_{\boldsymbol{\Psi}} + (\mathbf{G}^T\cdot\mathbf{H}_{\mathsf{vk}}^T\cdot\mathbf{H}_{n,q-1})\cdot\mathbf{s}_{0,\mathsf{rec}} + \mathbf{I}_{\bar{m}}\cdot\mathbf{z}_{\mathsf{rec}} \bmod q, \\ \mathbf{c}_{\mathsf{rec}}^{(3)} = (\mathbf{U}^T\cdot\mathbf{H}_{n,q-1})\cdot\mathbf{s}_{0,\mathsf{rec}} + \mathbf{I}_m\cdot\mathbf{x}_{\mathsf{rec}} + (\lfloor\frac{q}{2}\rfloor\cdot\mathbf{I}_m)\cdot\mathbf{w} \bmod q, \\ \mathbf{u}_R = \mathbf{A}_R\cdot\mathbf{w} \bmod q, \\ \mathbf{c}_{\mathsf{oa}}^{(1)} = (\bar{\mathbf{A}}^T\cdot\mathbf{H}_{n,q-1})\cdot\mathbf{s}_{0,\mathsf{oa}} + \mathbf{I}_m\cdot\mathbf{y}_{\mathsf{oa}} \bmod q, \\ \mathbf{c}_{\mathsf{oa}}^{(2)} = [(\mathbf{B}_{\mathsf{OA}} + \mathbf{H}_{\mathsf{vk}}\cdot\mathbf{G})^T\cdot\mathbf{H}_{n,q-1}]\cdot\mathbf{s}_{0,\mathsf{oa}} + \mathbf{I}_{\bar{m}}\cdot\mathbf{z}_{\mathsf{oa}} \bmod q, \\ \mathbf{c}_{\mathsf{oa}}^{(3)} = (\mathbf{V}^T\cdot\mathbf{H}_{n,q-1})\cdot\mathbf{s}_{0,\mathsf{oa}} + \mathbf{I}_m\cdot\mathbf{x}_{\mathsf{oa}} + (\lfloor\frac{q}{2}\rfloor\cdot\mathbf{I}_m)\cdot\mathbf{t}_{\mathsf{U}} \bmod q. \end{cases} \tag{8.16}$$

Let $\mathbf{w}_1 = \mathbf{b}_{\mathsf{U}}$, $\mathbf{w}_2 = \mathbf{s}_{0,\mathsf{rec}}$, $\mathbf{w}_3 = \mathbf{z}_{\boldsymbol{\Psi}} = \mathsf{expand}^{\otimes}(\mathbf{b}_{\mathsf{U}}, \mathbf{s}_{0,\mathsf{rec}})$, $\mathbf{w}_4 = \mathbf{w}_{\mathsf{U}}$, $\mathbf{w}_5 = \mathbf{t}_{\mathsf{U}}$, $\mathbf{w}_6 = \mathbf{s}_{0,\mathsf{oa}}$, $\mathbf{w}_7 = \mathbf{w}$, $\mathbf{w}_8 = \mathbf{x}_{\mathsf{rec}}$, $\mathbf{w}_9 = \mathbf{y}_{\mathsf{rec}}$, $\mathbf{w}_{10} = \mathbf{z}_{\mathsf{rec}}$, $\mathbf{w}_{11} = \mathbf{r}$, $\mathbf{w}_{12} = \mathbf{x}_{\mathsf{oa}}$, $\mathbf{w}_{13} = \mathbf{y}_{\mathsf{oa}}$, $\mathbf{w}_{14} = \mathbf{z}_{\mathsf{oa}}$ and

$$\mathbf{w}_{15} = \left(\mathbf{d}_1^T \,\|\, \mathbf{d}_2^T \,\|\, \tau[1]\cdot\mathbf{d}_2^T \,\|\ldots\|\, \tau[\ell]\cdot\mathbf{d}_2^T\right)^T.$$

Then system (8.16) can be rewritten as:

$$\begin{cases} \mathbf{v}_1 = \mathbf{M}_{1,1}\cdot\mathbf{w}_1 + \mathbf{M}_{1,2}\cdot\mathbf{w}_2 + \ldots + \mathbf{M}_{1,15}\cdot\mathbf{w}_{15} \bmod q, \\ \mathbf{v}_2 = \mathbf{M}_{2,1}\cdot\mathbf{w}_1 + \mathbf{M}_{2,2}\cdot\mathbf{w}_2 + \ldots + \mathbf{M}_{2,15}\cdot\mathbf{w}_{15} \bmod q, \\ \qquad\qquad\qquad\qquad \vdots \\ \mathbf{v}_{10} = \mathbf{M}_{10,1}\cdot\mathbf{w}_1 + \mathbf{M}_{10,2}\cdot\mathbf{w}_2 + \ldots + \mathbf{M}_{10,15}\cdot\mathbf{w}_{15} \bmod q, \end{cases} \tag{8.17}$$

where $\{\mathbf{M}_{i,j}\}_{(i,j)\in[10]\times[15]}$, $\{\mathbf{v}_i\}_{i\in[10]}$ are public matrices and vectors (which are possibly zero).

The argument system is obtained by invoking the protocol from Section 4.3. The protocol is repeated κ times to make the soundness error negligibly small.

8.5.2 Efficiency and Correctness

Efficiency. It can be seen that the given group encryption scheme can be implemented in polynomial time. We now will evaluate the bit-sizes of keys and ciphertext, as well as the communication cost of the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$.

- The public key of GM, as in (8.10), has bit-size $\mathcal{O}(\ell n^2 \log^2 q) = \widetilde{\mathcal{O}}(\ell \lambda^2)$.
- The public keys of OA and each user both have bit-size $n\bar{m}\lceil \log_2 q \rceil = \widetilde{\mathcal{O}}(\lambda^2)$.
- The secret key of each party in the scheme is a trapdoor of bit-size $\widetilde{\mathcal{O}}(\lambda^2)$. The user's certificate $\mathsf{cert}_{\mathsf{U}}$ has bit-size $\widetilde{\mathcal{O}}(\lambda)$.
- The ciphertext $\boldsymbol{\Psi}$ consists of $\mathsf{vk} \in \mathbb{Z}_q^n$, two ABB ciphertexts of total size $2(2m + \bar{m})\lceil \log_2 q \rceil$ and a one-time signature Σ. Thus, its bit-size is $\widetilde{\mathcal{O}}(\lambda) + |\Sigma|$.
- The communication cost of the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$ is largely dominated by the bit-size of the witness $\mathbf{z}_{\boldsymbol{\Psi}} = \mathsf{expand}^{\otimes}(\mathbf{b}_{\mathsf{U}}, \mathbf{s}_{0,\mathsf{rec}}) \in \{0,1\}^{4n\bar{m}k^2}$. The total cost is $\kappa \cdot \mathcal{O}(n^2 \log^4 q) = \widetilde{\mathcal{O}}(\lambda^2)$ bits.

Correctness. The given group encryption scheme is correct with overwhelming probability. We first remark that the scheme parameters are set up so that the two instances of the ABB identity-based encryption [ABB10] are correct. Indeed, during the decryption procedure of $\mathsf{DEC}(\mathsf{sk}_{\mathsf{U}}, \boldsymbol{\Psi}, L)$, we have:

$$\mathbf{c}_{\mathsf{rec}}^{(3)} - \mathbf{E}_{\mathsf{vk}}^T \cdot \begin{bmatrix} \mathbf{c}_{\mathsf{rec}}^{(1)} \\ \mathbf{c}_{\mathsf{rec}}^{(2)} \end{bmatrix} = \mathbf{x}_{\mathsf{rec}} - \mathbf{E}_{\mathsf{vk}}^T \cdot \begin{bmatrix} \mathbf{y}_{\mathsf{rec}} \\ \mathbf{z}_{\mathsf{rec}} \end{bmatrix} + \mathbf{w} \cdot \left\lfloor \frac{q}{2} \right\rfloor .$$

Note that $\|\mathbf{x}_{\mathsf{rec}}\|_\infty$ and $\|\mathbf{y}_{\mathsf{rec}}\|_\infty$ are bounded by B, and $\|\mathbf{z}_{\mathsf{rec}}\|_\infty = \|\mathbf{R}_{\mathsf{rec}}^T \cdot \mathbf{y}_{\mathsf{rec}}\|_\infty \leq \beta m B = \widetilde{\mathcal{O}}(n^2)$. Furthermore, the entries of the discrete Gaussian matrix $\mathbf{E}_{\mathsf{vk}}^T$ are bounded by $\widetilde{\mathcal{O}}(\sqrt{n})$. Hence, the error term $\mathbf{x}_{\mathsf{rec}} - \mathbf{E}_{\mathsf{vk}}^T \cdot \begin{bmatrix} \mathbf{y}_{\mathsf{rec}} \\ \mathbf{z}_{\mathsf{rec}} \end{bmatrix}$ is bounded by $\widetilde{\mathcal{O}}(n^{3.5})$ which is much smaller than $q/4 = \widetilde{\mathcal{O}}(n^4)$. As a result, the decryption algorithm returns $\mathbf{w}$ with overwhelming probability. The correctness of algorithm $\mathsf{OPEN}(\mathsf{sk}_{\mathsf{OA}}, \boldsymbol{\Psi}, L)$ also follows from a similar argument.

Finally, we note that if a certified group user honestly follows all the prescribed algorithms, then he should be able to compute valid witness-vectors to be used in the protocol $\langle \mathcal{P}, \mathcal{V} \rangle$, and he should be accepted by the verifier, thanks to the perfect completeness of the argument system in Section 4.3.

8.5.3 Security

Our scheme is proven secure under the SIS and LWE assumptions using classical reduction techniques. The security results are explicited in the following theorems.

8.5.3.1 Anonymity

Theorem 8.2. *The scheme provides anonymity if the* LWE *assumption holds and if $\mathcal{OTS}$ is a strongly unforgeable one-time signature.*

Proof. We consider a sequence of games where the first game is the real experiment of definition 8.2 while, in the final game, the adversary $\mathcal{A}$ is essentially an adversary against the anonymity of the Agrawal-Boneh-Boyen IBE scheme [ABB10]. In Game i, we call W_i the event that the challenger outputs 1.

Game 1**:** The challenger $\mathcal{B}$ generates public parameters par, which include matrices $\bar{\mathbf{A}}, \mathbf{U}, \mathbf{V} \in \mathbb{Z}_q^{n\times m}$ and $\mathbf{F} \in \mathbb{Z}_q^{2n\times n\bar{m}k}$. The opening authority's public key $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n\times\bar{m}}$ is given to $\mathcal{A}$ who generates a group manager's public key $\mathsf{pk}_{\mathsf{GM}}$ of its own. By invoking the USER oracle, $\mathcal{A}$ registers two distinct receivers' public keys $\mathsf{pk}_{\mathsf{U},0} = \mathbf{B}_{\mathsf{U},0} \in \mathbb{Z}_q^{n\times\bar{m}}$, $\mathsf{pk}_{\mathsf{U},1} = \mathbf{B}_{\mathsf{U},1} \in \mathbb{Z}_q^{n\times\bar{m}}$ chosen by the challenger. It also makes a number of opening queries and decryption queries, which the challenger handles using $\mathsf{sk}_{\mathsf{OA}} = \mathbf{T}_{\mathsf{OA}}$ and $\mathsf{sk}_{\mathsf{U},0} = \mathbf{T}_{\mathsf{U},0}$, $\mathsf{sk}_{\mathsf{U},1} = \mathbf{T}_{\mathsf{U},1}$, respectively. After a while, the adversary outputs $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}, L)$ such that $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$, with $\mathbf{A}_R \in \mathbb{Z}_p^{n\times m}$, $\mathbf{u}_R \in \mathbb{Z}_q^n$ and $\mathbf{w} \in \{0,1\}^m$. In return, $\mathcal{A}$ obtains, as a challenge, a group encryption $\Psi^\star = (\mathsf{vk}^\star, \mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{oa}}, \Sigma^\star)$. of the witness $\mathbf{w}$ under $\mathsf{pk}_{\mathsf{U},b} = \mathbf{B}_{\mathsf{U},b}$, for some random bit $b \leftarrow \mathcal{U}(\{0,1\})$ of the challenger's choice. Then, the adversary obtains proofs $\pi^\star_{\Psi^\star}$ for $\Psi^\star$ and makes further opening and decryption queries under the natural restrictions of Definition 8.2. When the adversary $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$ and the challenger outputs 1 if and only if $b' = b$.

Game 2**:** This game is like Game 1 except the challenger aborts in the event that the adversary $\mathcal{A}$ queries the opening of a ciphertext $\Psi = (\mathsf{vk}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, \Sigma)$ such that $\mathsf{vk} = \mathsf{vk}^\star$ and σ is valid (we assume w.l.o.g. that $\mathsf{vk}^\star$ is generated ahead of time). If this event occurs, the adversary $\mathcal{A}$ is necessarily able to break the strong unforgeability of $\mathcal{OTS}$ (note that, if the query occurs before the challenge phase, it means that $\mathcal{A}$ has forged a signature without seeing a signature at all). There thus exist a one-time signature forger $\mathcal{B}$ such that $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathrm{ots}}_{\mathcal{B}}(\lambda)$, which means that Game 2 is identical to Game 1 so long as $\mathcal{OTS}$ is a strongly unforgeable one-time signature.

Game 3**:** In this game, we modify the generation of proofs $\pi^\star_{\Psi^\star}$: instead of generating proofs using the real witnesses, we appeal to the zero-knowledge simulator of the argument system of Section 4.3.2 at each invocation of $\mathcal{P}$ after the challenge phase. Note that, since we assume public parameters generated by a trusted party, the statistical ZK simulator is allowed to use a trapdoor embedded in par to generate simulated proofs (using, e.g., Damgård's technique [Dam00]). The statistical zero-knowledge property of the argument system ensures that $\mathcal{A}$'s view remains statistically close to that of Game 2: we have $|\Pr[W_3] - \Pr[W_2]| \leq \mathsf{negl}(\lambda)$.

Game 4**:** We now modify the generation of the challenge ciphertext $\Psi^\star$. In this game, the challenger computes the ciphertext $\mathbf{c}^\star_{\mathsf{oa}}$ as an ABB encryption under the identity $\mathsf{vk}^\star$ of a random m-bit string instead of a decomposition $\mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U},b}) \in \{0,1\}^m$ of $\mathbf{h}_{\mathsf{U},b} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}^T_{\mathsf{U},b}) \in \mathbb{Z}_q^{2n}$. Since the random encryption coins $\mathbf{s}^\star_{\mathsf{oa}}, \mathbf{R}^\star_{\mathsf{oa}}, \mathbf{x}^\star_{\mathsf{oa}}, \mathbf{y}^\star_{\mathsf{oa}}$ are no longer used to generate proofs $\pi_{\Psi^\star}$, we can show that any noticeable change in $\mathcal{A}$'s output distribution implies a selective adversary against the ABB IBE, as established by Lemma 8.3, which would contradict the LWE assumption. The result of Agrawal *et al.* [ABB10, Theorem 23] (recalled in Theorem 8.1) indeed implies $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$.

In Game 4, we can show that, if the adversary $\mathcal{A}$ has noticeable advantage in the anonymity game, we can break the anonymity of the ABB IBE system, as shown in the proof of Lemma 8.4. From the result of [ABB10, Theorem 23], we deduce that $|\Pr[W_4] - 1/2| \leq \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$, which implies the announced result. □

Lemma 8.3. *Any PPT adversary such that* $\Pr[W_4]$ *is noticeably different from* $\Pr[W_3]$ *implies a selective adversary against the ABB IBE scheme.*

Proof. Let $\mathcal{A}$ be a PPT adversary for which $|\Pr[W_4] - \Pr[W_3]| = \varepsilon$ is non-negligible. We use $\mathcal{A}$ to build a selective adversary against the ABB IBE system.

At the outset of the game, the reduction $\mathcal{B}$ generates a one-time signature key pair $(\mathsf{vk}^\star, \mathsf{sk}^\star)$ and declares $\mathsf{vk}^\star$ as the target identity to its challenger for the selective security game, and obtains in return the IBE public parameters

$$\mathsf{PP} = (\bar{\mathbf{A}}, \mathbf{B}, \mathbf{V}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times \bar{m}} \times \mathbb{Z}_q^{n\times m}.$$

Next, the reduction runs the appropriate steps of the actual $\mathsf{SETUP}_{\mathsf{init}}$ algorithm to obtain $\mathsf{COM}_{\mathsf{par}}$, $\mathbf{F} \in \mathbb{Z}_q^{2n\times n\bar{m}k}$ and $\mathbf{U} \in \mathbb{Z}_q^{n\times m}$. Namely, $\mathcal{B}$ samples $\mathbf{F} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{2m\times n\bar{m}k})$ and $\mathbf{U} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$ like in the $\mathsf{SETUP}_{\mathsf{init}}$ algorithm and sends

$$\mathsf{par} = \{\lambda, n, q, k, m, B, \chi, \sigma, \beta, \ell, \kappa, \mathcal{OTS}, \mathsf{COM}_{\mathsf{par}}, \mathsf{FRD}, \bar{\mathbf{A}}, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{V}\}$$

along with $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B} \in \mathbb{Z}_q^{n\times \bar{m}}$ to the adversary $\mathcal{A}$.

In return, the adversary $\mathcal{A}$ chooses $\mathsf{pk}_{\mathsf{GM}}$, which allows it to enroll two users for whom $\mathcal{B}$ faithfully generates $(\mathsf{pk}_{\mathsf{U},i}, \mathsf{sk}_{\mathsf{U},i})_{i\in\{0,1\}}$. Knowing both private keys $\{\mathsf{sk}_{\mathsf{U},i} = \mathbf{T}_{\mathsf{U},i}\}_{i\in\{0,1\}}$, $\mathcal{B}$ is able to perfectly simulate the $\mathsf{DEC}(\cdot)$ oracle.

Open Queries. To answer opening queries for ciphertexts $\mathbf{\Psi} = (\mathsf{vk}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{OA}}, \Sigma)$ and labels L, $\mathcal{B}$ first checks that $\mathsf{Ver}(\mathsf{vk}, \Sigma, (\mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{OA}}, L)) = 1$. If this test fails, $\mathcal{B}$ returns $\perp$. Otherwise, $\mathcal{B}$ queries its IBE challenger to obtain a IBE private key $\mathbf{T}_{\mathsf{OA},\mathsf{vk}} \in \mathbb{Z}^{(m+\bar{m})\times m}$ for identity $\mathsf{vk} \neq \mathsf{vk}^\star$. The IBE challenger's response allows $\mathcal{B}$ to decrypt $\mathbf{c}_{\mathsf{OA}}$ and figure out the identity of the receiver by looking up database. The result of the opening operation is then returned to $\mathcal{A}$.

After a number of queries, $\mathcal{A}$ decides to move to the challenge phase and sends a challenge query $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}^\star, L^\star)$ such that $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w}^\star \bmod q$. The reduction handles this query by requesting a challenge ciphertext for the IBE security game with the messages $\mathbf{m}_0 = \mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U},b})$, for some random bit $b \hookleftarrow \mathcal{U}(\{0,1\})$ and $\mathbf{m}_1 \hookleftarrow \mathcal{U}(\{0,1\}^m)$. In return, $\mathcal{B}$ obtains a challenge ciphertext $\mathbf{c}^\star_{\mathsf{OA}}$ under identity $\mathsf{vk}^\star$, which is embedded in $\mathcal{A}$'s challenge ciphertext. Namely, $\mathbf{\Psi}^\star = (\mathsf{vk}^\star, \mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{OA}}, \Sigma^\star)$ is obtained by computing $\mathbf{c}^\star_{\mathsf{rec}}$ as an ABB encryption of the witness $\mathbf{w}^\star$ using the matrix $\mathbf{B}_{\mathsf{U},b} \in \mathbb{Z}_q^{n\times\bar{m}}$ as in (8.13) and $\Sigma^\star = \mathsf{Sign}(\mathsf{sk}^\star, (\mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{OA}}, L^\star))$. All queries to the proving oracle $\mathcal{P}$ are replied by returning a simulated ZK argument as in Game 3.

When $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$. If $b = b'$, $\mathcal{B}$ returns the bit 0 as a guess that the selective security challenger encrypted $\mathbf{m}_0 = \mathsf{vdec}_{n,q-1}(\mathbf{h}_{\mathsf{U},b})$. Otherwise, $\mathcal{B}$ outputs 1 meaning that the IBE challenger chose to encrypt $\mathbf{m}_1$, which was chosen independently of the value of $b \in \{0,1\}$. If we call Random (resp. Real) the event that the IBE challenger chooses to encrypt $\mathbf{m}_1$ (resp. $\mathbf{m}_0$), we can assess the advantage of the reduction $\mathcal{B}$ as

$$\begin{aligned}\mathbf{Adv}^{\text{sID-CPA}}_{\mathcal{B}}(\lambda) &= \big|\Pr[b = b' \mid \mathsf{Random}] - \Pr[b = b' \mid \mathsf{Real}]\big| \\ &= |\Pr[W_4] - \Pr[W_3]| \\ &= \varepsilon,\end{aligned}$$

which proves the result. □

Lemma 8.4. *In Game 4, the adversary's advantage is negligible assuming that the ABB IBE has pseudo-random ciphertexts.*

Proof. Let us assume the existence of a PPT adversary $\mathcal{A}$ with non negligible advantage ε in Game 4. From $\mathcal{A}$, we construct a selective adversary $\mathcal{B}$ that can distinguish ABB ciphertexts from random elements of the ciphertext space with non-negligible advantage in the game described in Definition 8.4.

First, $\mathcal{B}$ generates $(\mathsf{sk}^\star, \mathsf{vk}^\star)$ via the key generation algorithm of the one-time-signature OTS and hands $\mathsf{vk}^\star$ to its pseudo-randomness challenger. In return, $\mathcal{B}$ receives

$$\mathsf{PP} = (\bar{\mathbf{A}}, \mathbf{B}, \mathbf{U}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times \bar{m}} \times \mathbb{Z}_q^{n\times m}$$

from its real-or-random (ROR) challenger.

Our reduction uses PP to compute public parameters for our GE scheme. To this end, it samples $\mathbf{F} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{2n\times n\bar{m}k})$, $\mathbf{V} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$ as in the real $\mathsf{SETUP}_{\mathsf{init}}$ algorithm. The reduction $\mathcal{B}$ also computes $\mathbf{B}_{\mathsf{OA}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{OA}} \bmod q$, where the small-norm matrix $\mathbf{T}_{\mathsf{OA}}$ is sampled from $D_{\mathbb{Z},\sigma}^{m\times\bar{m}}$, and sends $\mathcal{A}$ the parameters

$$\mathsf{param} = \{\lambda, n, q, k, m, \sigma, \beta, \ell, \kappa, \mathcal{OTS}, \mathsf{COM}_{\mathsf{par}}, \mathsf{FRD}, \bar{\mathbf{A}}, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{V}\},$$

where $\bar{\mathbf{A}}$ is taken from PP, along with $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}}$. The rest of the keys are generated as in Game 4.

The reduction $\mathcal{B}$ then tosses a coin $b \hookleftarrow \mathcal{U}(\{0,1\})$. When the adversary $\mathcal{A}$ triggers an execution of the join protocol, $\mathcal{B}$ generates the public keys $(\mathsf{pk}_i)_{i\in\{0,1\}}$ by defining $\mathsf{pk}_{\mathsf{U},b} = \mathbf{B}$ using the matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times\bar{m}}$ supplied by the ROR challenger as part of PP and generates $(\mathsf{pk}_{\mathsf{U},1-b}, \mathsf{sk}_{1-b}) = (\mathbf{B}_{\mathsf{U},1-b} = \bar{\mathbf{A}} \cdot \mathbf{T}_{1-b}, \mathbf{T}_{1-b})$ for a secret key $\mathbf{T}_{1-b} \hookleftarrow D_{\mathbb{Z}^m,\sigma}^{\bar{m}}$ of its own. The two public keys $(\mathsf{pk}_{\mathsf{U},i})_{i\in\{0,1\}}$ are then certified by the adversarially-controlled GM. Notice that in the adversary's view, both public keys $\mathsf{pk}_{\mathsf{U},b}$ and $\mathsf{pk}_{\mathsf{U},1-b}$ are identically distributed.

To answer decryption queries $(\mathbf{\Psi} = (\mathsf{vk}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{OA}}, \Sigma), L)$, for any query pertaining to $\mathsf{pk}_{\mathsf{U},b}$, the reduction invokes its ROR challenger to obtain an IBE private key for the identity $\mathsf{vk} \neq \mathsf{vk}^\star$ and uses the result to decrypt $\mathbf{c}_{\mathsf{rec}}$. For any decryption query involving $\mathsf{pk}_{\mathsf{U},1-b}$, the reduction can faithfully run the actual decryption algorithm using its trapdoor $\mathbf{T}_{1-b}$. Open queries are answered using $\mathbf{T}_{\mathsf{OA}}$ as in the real Open algorithm.

When the adversary $\mathcal{A}$ decides to do so, it queries a challenge for a triple $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}, L)$ of its choice subject to the constraint $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w}$. At this point, $\mathcal{B}$ queries a challenge to its own challenger for the message $\mathbf{w}$ and obtains a ciphertext $\mathbf{c}$, which is embedded in $\mathbf{\Psi}^\star = (\mathsf{vk}^\star, \mathbf{c}, \mathbf{c}_{\mathsf{OA}}^\star, \Sigma^\star)$ while $\mathbf{c}_{\mathsf{OA}}^\star$ and $\Sigma^\star$ are generated as in Game 3 (in particular, $\mathbf{c}_{\mathsf{OA}}^\star$ encrypts a random string instead of a hash value of $\mathsf{pk}_{\mathsf{U},b}$). After the challenge phase, all queries to the proving oracle $\mathcal{P}$ are replied by returning a simulated ZK argument as in Game 3.

When $\mathcal{A}$ ends, it outputs a bit $b' \in \{0,1\}$. If $b' = b$, the reduction outputs Real. Otherwise, it outputs Random. Indeed, if the ROR challenger is playing the real game, we are exactly in Game 4: we have $\Pr[b' = b|\mathsf{Real}] = \Pr[W_4]$. Otherwise, the challenge ciphertext $\mathbf{\Psi}^\star$ is completely independent of $b \in \{0,1\}$ so that we can only have $b' = b$ with probability $\Pr[b' = b|\mathsf{Random}] = 1/2$. It follows that $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{ROR}}(\lambda) \geq |\Pr[W_4] - 1/2|$. □

8.5.3.2 Message Secrecy

Theorem 8.5. *The scheme provides message secrecy assuming that the* LWE *assumption holds and that $\mathcal{OTS}$ is a strongly unforgeable one-time signature.*

Proof. We proceed via a sequence of games. The first one corresponds to the experiment of Definition 8.1 when the challenger's bit b is 1 and the adversary obtains an actual encryption of the witness $\mathbf{w} \in \{0,1\}^m$ and real proofs at each invocation of the $\mathsf{PROVE}(.)$ oracle. In the last game, the adversary $\mathcal{A}$ is given an encryption of some random plaintext whereas $\mathsf{PROVE}(.)$ returns simulated zero-knowledge arguments which are generated a simulator $\mathcal{P}'$ that does not use any witness. In Game i, W_i stands for the event that the adversary $\mathcal{A}$ outputs the bit $b' = 1$.

Game 1: This is the real game, where the challenger feeds $\mathcal{A}$ with public parameters par containing $\bar{\mathbf{A}}, \mathbf{U}, \mathbf{V} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{F} \in \mathbb{Z}_q^{2n \times n\bar{m}k}$. The adversary produces public keys $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n \times \bar{m}}$ and $\mathsf{pk}_{\mathsf{GM}} = (\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{D}, \mathbf{u})$ on behalf of the opening authority and the group manager which are both under its control. The challenger and $\mathcal{A}$ run an execution of the JOIN protocol which allows $\mathcal{A}$ to register and certify the public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n \times \bar{m}}$ of some honest receiver chosen by the challenger. Then, the adversary $\mathcal{A}$ makes a polynomial number of decryption queries which the challenger faithfully handles using the private key $\mathsf{sk}_{\mathsf{U}} = \mathbf{T}_{\mathsf{U}} \in \mathbb{Z}^{m \times \bar{m}}$ for which $\mathbf{B}_{\mathsf{U}} \cdot \mathbf{T}_{\mathsf{U}} = \mathbf{0}^{n \times \bar{m}}$. At some point, the adversary $\mathcal{A}$ outputs a triple $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}, L)$ such that $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$, with $\mathbf{A}_R \in \mathbb{Z}_p^{n \times m}$, $\mathbf{u}_R \in \mathbb{Z}_q^n$ and $\mathbf{w} \in \{0,1\}^m$. At this point, the challenger generates a challenge ciphertext $\Psi^\star = (\mathsf{vk}^\star, \mathbf{c}_{\mathsf{rec}}^\star, \mathbf{c}_{\mathsf{oa}}^\star, \Sigma^\star)$ consisting of a group encryption of the real witness $\mathbf{w}$ under $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}}$. Then, the adversary obtains a polynomial number of proofs $\pi_{\Psi^\star}^\star$ related to the challenge ciphertext $\Psi^\star$ and is granted further access to the decryption oracle under the obvious restrictions. When $\mathcal{A}$ halts, it outputs a bit $b' \in \{0,1\}$.

Game 2: In this game, we modify the $\mathsf{DEC}(.)$ oracle and have the challenger reject any ciphertext of the form $\Psi = (\mathsf{vk}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{oa}}, \Sigma)$ such that $\mathsf{vk} = \mathsf{vk}^\star$ (note that $\mathsf{vk}^\star$ can be generated at the outset of the game w.l.o.g.). Clearly Game 2 is identical to Game 1 until the event that the challenger rejects a ciphertext that would not have been rejected in Game 1. This can only occur if $\mathcal{A}$ is able to break the strong unforgeability of the one-time signature $\mathcal{OTS}$. As in the proof of Theorem 8.2, we have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda)$, which is negligible if $\mathcal{OTS}$ is strongly unforgeable.

Game 3: We now modify the generation of proofs $\pi_{\Psi^\star}^\star$. Instead of generating them using the witnesses used in the generation of $\Psi^\star$, we rely on the zero-knowledge simulator of the argument system of Section 4.3.2 at each invocation of $\mathsf{PROVE}_{\mathcal{P},\mathcal{P}'}^b$ after the challenge phase (note that, since we assume trusted public parameters, the simulator can use techniques [Dam00] to achieve statistically perfect simulation without increasing the number of rounds). The statistical ZK property of the argument system ensures that this change will remain unnoticed, even in the view of an all powerful adversary: we have $|\Pr[W_3] - \Pr[W_2]| \in \mathsf{negl}(\lambda)$. From now onwards, the random coins $coins_{\mathbf{\Psi}}^\star = (\mathbf{s}_{\mathsf{rec}}^\star, \mathbf{R}_{\mathsf{rec}}^\star, \mathbf{x}_{\mathsf{rec}}^\star, \mathbf{y}_{\mathsf{rec}}^\star, \mathbf{s}_{\mathsf{oa}}^\star, \mathbf{R}_{\mathsf{oa}}^\star, \mathbf{x}_{\mathsf{oa}}^\star, \mathbf{y}_{\mathsf{oa}}^\star)$ are no longer used by the PROVE oracle.

Game 4: In the generation of $\Psi^\star$, we set $\mathbf{c}_{\mathsf{rec}}^\star$ as an encryption of a random element of $\mathbb{Z}_p^m$. Since the random encryption coins $\mathbf{s}_{\mathsf{rec}}^\star, \mathbf{R}_{\mathsf{rec}}^\star, \mathbf{x}_{\mathsf{rec}}^\star, \mathbf{y}_{\mathsf{rec}}^\star$ are not used in Game 3,

Lemma 8.6 gives a simple reduction showing that any significant change in $\mathcal{A}$'s behavior would imply a selective adversary against the ABB identity-based encryption scheme. The result of [ABB10] tells us that, under the LWE assumption, Game 4 is computationally indistinguishable from Game 3 in the adversary's view: we have $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{LWE}}(\lambda)$.

Game 5: We bring a last modification to the $\mathsf{DEC}(.)$ oracle and now refrain from applying the rejection rule of Game 2. If $\mathcal{OTS}$ is strongly unforgeable, the distance $|\Pr[W_5] - \Pr[W_4]| \leq \mathbf{Adv}^{\mathrm{ots}}(\lambda)$ must be negligible.

In the last game, the oracle $\mathsf{PROVE}(.)$ does not need to know any witness. It thus mirrors the experiment of Definition 8.1 where the challenger's bit is $b = 0$. Putting everything altogether, we get $|\Pr[W_5] - \Pr[W_1]| \in \mathsf{negl}(\lambda)$, which yields the claimed result. □

Lemma 8.6. *Any PPT adversary that can distinguish Game* 4 *from Game* 3 *implies a selective adversary against the ABB IBE scheme.*

Proof. Let us assume a PPT adversary $\mathcal{A}$ such that $\varepsilon = |\Pr[W_4] - \Pr[W_3]|$ is noticeable. We use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ that breaks the IND-sID-CPA security of the ABB scheme, which would contradict the LWE assumption, as established in [ABB10, Th. 23].

At the very beginning of the IND-sID-CPA game, the reduction $\mathcal{B}$ generates a one-time signature key pair $(\mathsf{sk}^\star, \mathsf{vk}^\star)$ and hands $\mathsf{vk}^\star$ to its selective security challenger as the target identity under which the challenge ciphertext will later be computed. In response, $\mathcal{B}$ receives the public parameters

$$\mathsf{PP} = (\bar{\mathbf{A}}, \mathbf{B}, \mathbf{U}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times \bar{m}} \times \mathbb{Z}_q^{n\times m}$$

from its IBE challenger.

The reduction then runs the missing steps of the actual $\mathsf{Setup}_{\mathsf{init}}$ algorithm: namely, $\mathcal{B}$ samples $\mathbf{F} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{2m\times n\bar{m}k})$, $\mathbf{V} \hookleftarrow \mathcal{U}(\mathbb{Z}_q^{n\times m})$ and generates $\mathsf{COM}_{\mathsf{par}}$ before sending the common public parameters

$$\mathsf{par} = \{\lambda, n, q, k, m, B, \chi, \sigma, \beta, \ell, \kappa, \mathcal{OTS}, \mathsf{COM}_{\mathsf{par}}, \mathsf{FRD}, \bar{\mathbf{A}}, \mathbf{G}, \mathbf{F}, \mathbf{U}, \mathbf{V}\}$$

to the adversary $\mathcal{A}$.

At this point, the adversary $\mathcal{A}$ chooses the public keys $\mathsf{pk}_{\mathsf{OA}} = \mathbf{B}_{\mathsf{OA}} \in \mathbb{Z}_q^{n\times\bar{m}}$ and $\mathsf{pk}_{\mathsf{GM}} = (\mathbf{A}, \{\mathbf{A}_i\}_{i=0}^{\ell}, \mathbf{D}_0, \mathbf{D}_1, \mathbf{D}, \mathbf{u})$ on behalf of the opening authority and the group manager. It also starts an execution of the joining protocol in which the reduction $\mathcal{B}$ defines $\mathsf{pk}_{\mathsf{U}} = \mathbf{B} \in \mathbb{Z}_q^{n\times\bar{m}}$ as the honest receiver's public key, where $\mathbf{B} \in \mathbb{Z}_q^{n\times\bar{m}}$ is taken from the public parameters PP supplied by its IBE challenger. Note that $\mathsf{pk} = \mathbf{B} \in \mathbb{Z}_q^{n\times\bar{m}}$ is distributed as a real key in $\mathcal{A}$'s view. This public key is certified by $\mathbf{A}$ which controls the GM.

In the next stage, $\mathcal{A}$ makes a number of decryption queries for ciphertexts of the form $\boldsymbol{\Psi} = (\mathsf{vk}, \mathbf{c}_{\mathsf{rec}}, \mathbf{c}_{\mathsf{OA}}, \Sigma)$. To answer these, the reduction invokes its IBE challenger so as to obtain an IBE private key $\mathbf{E}_{\mathsf{vk}} \in \mathbb{Z}^{(m+\bar{m})\times m}$ for the identity $\mathsf{vk} \neq \mathsf{vk}^\star$. The resulting $\mathbf{E}_{\mathsf{vk}}$ is used it to IBE-decrypt $\mathbf{c}_{\mathsf{rec}}$ and return the corresponding witness $\mathbf{w}$ to $\mathbf{A}$.

At some point, the adversary $\mathcal{A}$ queries a challenge ciphertext by outputting a triple $((\mathbf{A}_R, \mathbf{u}_R), \mathbf{w}, L)$ such that $\mathbf{w} \in \{0,1\}^m$ satisfies $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$. Then, the reduction

$\mathcal{B}$ requests a challenge ciphertext $\mathbf{c}^\star_{\mathsf{rec}}$ to its IBE challenger by sending it the messages $\mathbf{m}_1 = \mathbf{w} \in \{0,1\}^m$ and $\mathbf{m}_0 \hookleftarrow \mathcal{U}(\{0,1\}^m)$. The resulting ciphertext $\mathbf{c}^\star_{\mathsf{rec}}$ is embedded in $\mathbf{\Psi}^\star = (\mathsf{vk}^\star, \mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{OA}}, \Sigma^\star)$ by faithfully computing $\mathbf{c}^\star_{\mathsf{OA}}$ and $\Sigma^\star$ as in the actual Enc algorithm.

After the challenge phase, $\mathcal{A}$ keeps sending decryption queries for ciphertexts $\Psi^\star$ containing one-time verification keys $\mathsf{vk} \neq \mathsf{vk}^\star$ and these decryption queries are answered as before. In addition, $\mathcal{A}$ is granted access to the stateful oracle $\mathsf{PROVE}^b_{\mathcal{P},\mathcal{P}'}$. Recall that, from Game 3 onwards, all these queries are answered by returning simulated zero-knowledge arguments. Eventually $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is also returned by $\mathcal{B}$ to its own challenger.

If the IBE challenger provides a challenge $\mathbf{c}^\star_{\mathsf{rec}}$ that encrypts a random message (i.e., by encrypting $\mathbf{m}_0$), then we are exactly in the setting of Game 4. In the even that $\mathbf{c}^\star_{\mathsf{rec}}$ rather encrypts $\mathbf{m}_1 = \mathbf{w} \in \{0,1\}^m$, $\mathcal{A}$'s view is exactly the same as in Game 3. If we denote by Random (resp. Real) the event that the IBE challenger chooses to encrypt $\mathbf{m}_0$ (resp. $\mathbf{m}_1$), the advantage of the reduction $\mathcal{B}$ as an IND-sID-CPA adversary is

$$\begin{aligned}\mathrm{Adv}^{\text{sID-CPA}}_{\mathcal{B}}(\lambda) &= \big|\Pr[b'=1|\mathsf{Real}] - \Pr[b'=1|\mathsf{Random}]\big| = |\Pr[W_3] - \Pr[W_4]| \\ &= \varepsilon,\end{aligned}$$

which concludes our proof. □

8.5.3.3 Soundness

Theorem 8.7. *The scheme provides soundness under the* SIS *assumption.*

Proof. To prove the result, we observe that, in order to break the soundness property, the adversary must come up with a relation $\mathsf{pk}_{\mathcal{R}} = (\mathbf{A}_R, \mathbf{u}_R) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$, a ciphertext $\Psi^\star = (\mathsf{vk}^\star, \mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{oa}}, \Sigma^\star)$, a label L and produce a convincing proof $\pi_{\Psi^\star}$ such that either

1. $\mathbf{c}^\star_{\mathsf{oa}}$ does not decrypt to a string $\mathbf{h} \in \{0,1\}^m$ such that $\mathbf{h}_{\mathsf{U}} = \mathbf{H}_{2n,q-1} \cdot \mathbf{h} \in \mathbb{Z}_q^{2n}$ coincides with $\mathbf{h}_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,m,q}(\mathbf{B}_{\mathsf{U}}^T)$ for some $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ appearing in database.

2. $\mathbf{c}^\star_{\mathsf{oa}}$ opens to a certified public key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$, which belongs to database (and for which a certificate was issued), but $\mathbf{B}_{\mathsf{U}}$ is outside the language $\mathcal{PK}$ of valid public keys. This case is immediately ruled out by the density of the public key space. Namely, all matrices $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ are potentially valid public keys as there always exist a small-norm matrix $\mathbf{T}_{\mathsf{U}} \in \mathbb{Z}^{m\times\bar{m}}$ such that $\mathbf{B}_{\mathsf{U}} = \bar{\mathbf{A}} \cdot \mathbf{T}_{\mathsf{U}} \bmod q$.

3. $\mathbf{c}^\star_{\mathsf{oa}}$ opens to a certified key $\mathsf{pk}_{\mathsf{U}} = \mathbf{B}_{\mathsf{U}}$ for which $\Psi^\star = (\mathsf{vk}^\star, \mathbf{c}^\star_{\mathsf{rec}}, \mathbf{c}^\star_{\mathsf{oa}}, \Sigma^\star)$ is not a valid encryption of $\mathbf{w} \in \{0,1\}^m$ such that $\mathbf{u}_R = \mathbf{A}_R \cdot \mathbf{w} \bmod q$.

4. The opening algorithm fails to uniquely identify the receiver. This occurs if $\mathbf{c}^\star_{\mathsf{oa}}$ decrypts to a string $\mathbf{h} \in \{0,1\}^m$ such that $\mathbf{h}'_{\mathsf{U}} = \mathbf{H}_{2n,q-1} \cdot \mathbf{h} \in \mathbb{Z}_q^{2n}$ corresponds to at least two distinct public keys $\mathbf{B}_{\mathsf{U},0}, \mathbf{B}_{\mathsf{U},1} \in \mathbb{Z}_q^{n\times\bar{m}}$ which satisfy

$$\mathbf{h}'_{\mathsf{U}} = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},0}^T) \bmod q = \mathbf{F} \cdot \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},1}^T) \bmod q.$$

Since $\mathsf{mdec}_{n,\bar{m},q}(.) : \mathbb{Z}_q^{\bar{m}\times n} \rightarrow \{0,1\}^{n\bar{m}k}$ is an injective function, the above equality necessarily implies a collision for the SIS-based hash function built upon $\mathbf{F} \in \mathbb{Z}_q^{2n\times n\bar{m}k}$: namely,

$$\mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},0}^T) - \mathsf{mdec}_{n,\bar{m},q}(\mathbf{B}_{\mathsf{U},1}^T) \quad \in \{-1,0,1\}^{n\bar{m}k}$$

is a short non-zero vector of $\Lambda_q^{\perp}(\mathbf{F})$.

Having shown that cases *b* and *d* cannot occur if the SIS assumption holds, we only need to consider cases *a* and *c*. The computational soundness of the argument system ensures that, by replaying the soundness adversary a sufficient number of times, the knowledge extractor will be able to extract either: (i) A breach in the computational soundness of the argument system and thus the binding property of the commitment scheme COM (which relies on the SIS assumption with the commitment scheme of [KTX08]). Note that this situation covers case (*c.*) above. (ii) A set of witnesses

$$\begin{cases} \tau \in \{0,1\}^{\ell},\ \mathbf{d} = [\mathbf{d}_1^T|\mathbf{d}_2^T]^T \in [-\beta,\beta]^{2m},\ \mathbf{r} \in [-\beta,\beta]^m \\ \mathbf{t}_{\mathsf{U}} \in \{0,1\}^m,\ \mathbf{w}_{\mathsf{U}} \in \{0,1\}^{\bar{m}} \\ \mathbf{b}_{\mathsf{U}} \in \{0,1\}^{n\bar{m}k},\ \mathbf{s}_{0,\mathsf{rec}} \in \{0,1\}^{nk},\ \mathbf{z}_{\mathbf{\Psi}} \in \{0,1\}^{4n\bar{m}k^2} \\ \mathbf{x}_{\mathsf{rec}},\ \mathbf{y}_{\mathsf{rec}} \in [-B,B]^m,\ \mathbf{z}_{\mathsf{rec}} \in [-\beta mB,\beta mB]^{\bar{m}},\ \mathbf{w} \in \{0,1\}^m, \\ \mathbf{s}_{\mathsf{oa}} \in \{0,1\}^{nk},\ \mathbf{x}_{\mathsf{oa}},\ \mathbf{y}_{\mathsf{oa}} \in [-B,B]^m,\ \mathbf{z}_{\mathsf{oa}} \in [-\beta mB,\beta mB]^{\bar{m}} \end{cases}$$

satisfying relations (8.16). Given that witnesses $\tau \in \{0,1\}^{\ell}$, $\mathbf{d} \in [-\beta,\beta]^{2m}$, $\mathbf{r} \in [-\beta,\beta]^m$ and $\mathbf{t}_{\mathsf{U}} \in \{0,1\}^m$ satisfy (8.16), it comes that $(\tau,\mathbf{d},\mathbf{r})$ form a valid signature for the message $\mathbf{t}_{\mathsf{U}} \in \{0,1\}^m$. At this point, case *a* implies that no matrix $\mathbf{B}_{\mathsf{U}} \in \mathbb{Z}_q^{n\times\bar{m}}$ of database decomposes to a string $\mathbf{h}_{\mathsf{U}} \in \{0,1\}^{n\bar{m}k}$ such that $\mathbf{t}_{\mathsf{U}} = \mathsf{vdec}_{n,q-1}(\mathbf{F}\cdot\mathbf{h}_{\mathsf{U}} \bmod q)$ was signed by the reduction during an execution of JOIN. This implies that the pair $(\mathbf{t}_{\mathsf{U}},(\tau,\mathbf{d},\mathbf{r}))$ forms a forgery for the SIS-based signature scheme of Section 7.1. The reduction is straightforward and omitted. □

Chapter 9

Lattice-Based Oblivious Transfer with Access Control

Oblivious transfer (OT) is a central cryptographic primitive coined by Rabin [Rab81] and extended by Even *et al.* [EGL85]. It involves a sender S with a database of messages $M_1, \ldots, M_N$ and a receiver R with an index $\rho \in \{1, \ldots, N\}$. The protocol allows R to retrieve the ρ-th entry M_ρ from S without letting S infer anything on R's choice ρ. Moreover, R only obtains M_ρ learns nothing about $\{M_i\}_{i \neq \rho}$.

In its adaptive flavor [NP99], OT allows the receiver to interact k times with S to retrieve $M_{\rho_1}, \ldots, M_{\rho_k}$ in such a way that, for each index $i \in \{2, \ldots, k\}$, the i-th index ρ_i may depend on the messages $M_{\rho_1}, \ldots, M_{\rho_{i-1}}$ previously obtained by R.

OT is known to be a complete building block for cryptography (as for example, [GMW87]) in that, if it can be realized, then any secure multiparty computation can be. In its adaptive variant, OT is motivated by applications in privacy-preserving access to sensitive databases (e.g., medical records or financial data) stored in encrypted form on remote servers, oblivious searches or location-based services.

As far as efficiency goes, adaptive OT protocols should be designed in such a way that, after an inevitable initialization phase with linear communication complexity in N and the security parameter λ, the complexity of each transfer is at most poly-logarithmic in N. At the same time, this asymptotic efficiency should not come at the expense of sacrificing ideal security properties. The most efficient adaptive OT protocols that satisfy the latter criterion stem from the work of Camenisch, Neven and shelat [CNs07] and its follow-ups [GH07, GH08, GH11].

In its basic form, (adaptive) OT does not restrict in any way the population of users who can obtain specific records. In many sensitive databases (e.g., DNA databases or patients' medical history), however, not all users should be able to download all records: it is vital access to certain entries be conditioned on the receiver holding suitable credentials delivered by authorities. At the same time, privacy protection mandates that authorized users be able to query database records while leaking as little as possible about their interests or activities. In medical datasets, for example, the specific entries retrieved by a given doctor could reveal which disease his patients are suffering from. In financial or patent datasets, the access pattern of a company could betray its investment strategy or the invention it is developing. In order to combine user-privacy and fine-grained database security, it is thus

desirable to enrich adaptive OT protocols with refined access control mechanisms in many of their natural use cases.

This motivated Camenisch, Dubovitskaya and Neven [CDN09] to introduce a variant named *oblivious transfer with access control* (OT-AC), where each database record is protected by a different access control policy $P : \{0,1\}^* \to \{0,1\}$. Based on their attributes, users can obtain credentials generated by pre-determined authorities, which entitle them to anonymously retrieve database records of which the access policy accepts their certified attributes: in other words, the user can only download the records for which he has a valid credential Cred_x for an attribute string $x \in \{0,1\}^*$ such that $P(x) = 1$. During the transfer phase, the user demonstrates possession of a pair (Cred_x, x) and simultaneously convinces the sender that he is querying some record M_ρ associated with a policy P such that $P(x) = 1$. The only information that the database holder eventually learns is that some user retrieved some record which he was authorized to obtain.

Camenisch *et al.* formalized the OT-AC primitive and provided a construction in groups with a bilinear map [CDN09]. While efficient, their solution "only" supports access policies consisting of conjunctions: each policy P is specified by a list of attributes that a given user should obtain a credential for in order to complete the transfer. Several subsequent works [ZAW+10, CDNZ11, CDEN12] considered more expressive access policies while even hiding the access policies in some cases [CDNZ11, CDEN12]. Unfortunately, all of them rely on non-standard assumptions (known as "q-type assumptions" as described in Chapter 2) in groups with a bilinear maps. For the sake of not putting all one's eggs in the same basket, a primitive as powerful as OT-AC ought to have alternative realizations based on firmer foundations.

In this chapter, we propose a solution based on lattice assumptions where access policies consist of any branching program of width 5, which is known [Bar86] to suffice for the realization of any access policy in NC1. As a result of independent interest, we provide protocols for proving the correct evaluation of a committed branching program. More precisely, we give zero-knowledge arguments for demonstrating possession of a secret input $\mathbf{x} \in \{0,1\}^\kappa$ and a secret (and possibly certified) branching program BP such that $\mathsf{BP}(\mathbf{x}) = 1$.

Related Work. Oblivious transfer with adaptive queries dates back to the work of Naor and Pinkas [NP99], which requires $O(\log N)$ interaction rounds per transfer. Naor and Pinkas [NP05] also gave generic constructions of (adaptive) k-out-of-N OT from private information retrieval (PIR) [CGKS95]. The constructions of [NP99, NP05], however, are only secure in the half-simulation model, where simulation-based security is only considered for one of the two parties (receiver security being formalized in terms of a game-based definition). Moreover, the constructions of Adaptive OT from PIR [NP05] requires a complexity $O(N^{1/2})$ at each transfer where Adaptive OT allows for $O(\log N)$ cost. Before 2007, many OT protocols (e.g., [NP01, AIR01, TK05]) were analyzed in terms of half-simulation.

While several efficient fully simulatable protocols appeared the last 15 years (e.g., [DN03, Lin08, PVW08] and references therein), full simulatability remained elusive in the adaptive k-out-of-N setting [NP99] until the work [CNs07] of Camenisch, Neven and shelat, who introduced the "assisted decryption" paradigm. The latter consists in having the sender obliviously decrypt a re-randomized version of one of the original ciphertexts contained in the database. This technique served as a blueprint for many subsequent protocols

[GH07, GH08, GH11, JL09], including those with access control [CDN09, CDNZ11, CDEN12, ACDN13] and those presented in this chapter. In the adaptive k-out-of-N setting (which we denote as $\mathcal{OT}^N_{k\times 1}$), the difficulty is to achieve full simulatability without having to transmit a $O(N)$ bits at each transfer. To our knowledge, except the oblivious-PRF-based approach of Jarecki and Liu [JL09], all known fully simulatable $\mathcal{OT}^N_{k\times 1}$ protocols rely on bilinear maps[1]. A recent work of Döttling *et al.* [DFKS16] uses non-black-box techniques to realize LWE-based 2-round oblivious PRF (OPRF) protocols [FIPR05]. However, while fully simulatable OPRFs imply [JL09] fully simulatable adaptive OT, the OPRF construction of [DFKS16] does not satisfy the standard notion of full simulation-based security against malicious adversaries (which is impossible to achieve in two rounds). It also relies on the full power of homomorphic encryption, which we do not require.

A number of works introduced various forms of access control in OT. Priced OT [AIR01] assigns variable prices to all database records. In conditional OT [DCOR99], access to a record is made contingent on the user's secret satisfying some predicate. Restricted OT [Her11] explicitly protects each record with an independent access policy. Still, none of these OT flavors aims at protecting the anonymity of users. The model of Coull, Green and Hohenberger [CGH09] does consider user anonymity via stateful credentials. For the applications of OT-AC, it would nevertheless require re-issuing user credentials at each transfer.

While efficient, the initial OT-AC protocol of Camenisch *et al.* [CDN09] relies on non-standard assumptions in groups with a bilinear map and only realizes access policies made of conjunctions. Abe *et al.* [ACDN13] gave a different protocol which they proved secure under more standard assumptions in the universal composability framework [Can01]. Their policies, however, remain limited to conjunctions. It was mentioned in [CDN09, ACDN13] that disjunctions and DNF formulas can be handled by duplicating database entries. Unfortunately, this approach rapidly becomes prohibitively expensive in the case of (t, n)-threshold policies with $t \approx n/2$. Moreover, securing the protocol against malicious senders requires them to prove that all duplicates encrypt the same message. More expressive policies were considered by Zhang *et al.* [ZAW+10] who gave a construction based on attribute-based encryption [SW05] that extends to access policies expressed by any Boolean formulas (and thus NC1 circuits). Camenisch, Dubovitskaya, Neven and Zaverucha [CDNZ11] generalized the OT-AC functionality so as to hide the access policies. In [CDEN12], Camenisch *et al.* gave a more efficient construction with hidden policies based on the attribute-based encryption scheme of [NYO08]. At the expense of a proof in the generic group model, [CDEN12] improves upon the expressiveness of [CDNZ11] in that its policies extend into CNF formulas. While the solutions of [CDNZ11, CDEN12] both hide the access policies to users (and the successful termination of transfers to the database), their policies can only live in a proper subset of NC1. As of now, threshold policies can only be efficiently handled by the ABE-based construction of Zhang *et al.* [ZAW+10], which requires *ad hoc* assumptions in groups with a bilinear map.

In the forthcoming sections, we first present the adaptive oblivious transfer scheme and its access control flavour, then we present the needed building blocks, in particular a simpler

[1] Several pairing-free candidates were suggested in [KPN10, KPN11] but, as pointed out in [GH11], they cannot achieve full simulatability in the sense of [CNs07]. In particular, the sender can detect if the receiver fetches the same record in two distinct transfers.

version of the signature scheme presented in Section 7.1. We next present our constructions and the zero-knowledge protocol to guarantee the correct execution of a branching program. Finally, we close this chapter with the description of a shift of our scheme from the standard model to the random oracle model to reduce the communication complexity cost, and a comparison table between the different existing solutions.

9.1 Adaptive Oblivious Transfer

In the syntax of [CNs07], an adaptive k-out-of-N OT scheme OT_k^N is a tuple of stateful PPT algorithms $(\mathsf{S}_\mathsf{I}, \mathsf{R}_\mathsf{I}, \mathsf{S}_\mathsf{T}, \mathsf{R}_\mathsf{T})$. The sender $\mathsf{S} = (\mathsf{S}_\mathsf{I}, \mathsf{S}_\mathsf{T})$ consists of two interactive algorithms S_I and S_T and the receiver has a similar representation as algorithms R_I and R_T. In the *initialization phase*, the sender and the receiver run interactive algorithms S_I and R_I, respectively, where S_I takes as input messages $M_1, \ldots, M_N$ while R_I has no input. This phase ends with the two algorithms S_I and R_I outputting their state information S_0 and R_0 respectively.

During the i-th *transfer*, $1 \leq i \leq k$, both parties run an interactive protocol via the R_T and S_T algorithms. The sender starts runs $\mathsf{S}_\mathsf{T}(S_{i-1})$ to obtain its updated state information S_i while the receiver runs $\mathsf{R}_\mathsf{T}(R_{i-1}, \rho_i)$ on input of its previous state R_{i-1} and the index $\rho_i \in \{1, \ldots, N\}$ of the message it wishes to retrieve. At the end, R_T outputs an updated state R_i and a message M'_{ρ_i}.

Correctness mandates that, for all $M_1, \ldots, M_N$, for all $\rho_1, \ldots, \rho_k \in [N]$ and all coin tosses ϖ of the (honestly run) algorithms, we have $M'_{\rho_i} = M_{\rho_i}$ for all i.

We consider protocols that are secure (against static corruptions) in the sense of simulation-based definitions. The security properties against a cheating sender and a cheating receiver are formalized via the "real-world/ideal-world" paradigm. The security definitions of [CNs07] are recalled in the following Section.

9.1.1 Security Definitions for Adaptive k-out-of-N Oblivious Transfer

Security is defined via the "real-world/ideal-world" paradigm which was first introduced in the Universal Composability (UC) framework [Can01]. Like [CNs07, CDN09], however, we do not incorporate all the formalities of the UC framework. We define two experiments: the **Real** experiment, where the two parties run the actual protocol, and the **Ideal** experiment wherein a *trusted third party* assumes the role of the functionality.

The model of [CNs07] formalizes two security notions called *sender security* and *receiver security*. The former considers the security of honest senders against cheating senders whereas the latter considers the security of honest receivers interacting with malicious senders.

For an adaptive OT protocol OT_k^N comprised of algorithms $(\mathsf{S}_\mathsf{I}, \mathsf{S}_\mathsf{T}, \mathsf{R}_\mathsf{I}, \mathsf{R}_\mathsf{T})$, we denote define the honest sender S as the algorithm that runs $\mathsf{S}_\mathsf{I}(M_1, \ldots, M_N)$ during the initialization phase, runs S_T at each transfer and eventually returns $S_k = \epsilon$ as its final output. Similarly, the honest receiver R is the algorithm that runs R_I in the initialization phase, runs $\mathsf{R}_\mathsf{T}(R_{i-1}, \rho_i)$ during the i-th transfer and eventually returns $R_k = (M'_{\rho_1}, \ldots, M'_{\rho_k})$ as its final output.

Real Experiment. Here, a sender $\widehat{\mathsf{S}}$ and a receiver $\widehat{\mathsf{R}}$ which proceed as follows for experiment $\mathbf{Real}_{\widehat{\mathsf{S}},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$.
The sender $\widehat{\mathsf{S}}$ is given messages $M_1, \ldots, M_N$ and interacts with $\widehat{\mathsf{R}}$ which does not have any input in the initialization phase. At end of the latter, $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$ output their initial states S_0 and R_0 respectively. Then, $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$ start k sequential interactions: for $i \in [k]$, in the i-th transfer, the sender $\widehat{\mathsf{S}}$ and the receiver $\widehat{\mathsf{R}}$ run $S_i \leftarrow \widehat{\mathsf{S}}(S_{i-1})$ and $(R_i, M'_{\rho_i}) \leftarrow \widehat{\mathsf{R}}(R_{i-1}, \rho_i)$, where $\rho_i \in [N]$ is a message index and (S_i, R_i) denote updated states for $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$, respectively. Note that M'_{ρ_i} may be different from M_{ρ_i} if one of the participant deviates from the protocol. At the end of the k-th interaction, $\widehat{\mathsf{S}}$ and $\widehat{\mathsf{R}}$ output strings S_k and R_k respectively. The output of $\mathbf{Real}_{\widehat{\mathsf{S}},\widehat{\mathsf{R}}}$ is the pair (S_k, R_k).

The honest sender S is the algorithm that runs $\mathsf{S}(M_1, \ldots, M_N)$ as in the initialization phase, runs S_T in all subsequent interactions and always outputs $S_k = \varepsilon$. The honest receiver R is the algorithm that runs R_I in the initialization phase, runs $\mathsf{R}_\mathsf{T}(\mathsf{R}_{i-1}, \rho_i)$ at the i-th transfer and returns the list of received messages $R_k = (M'_{\rho_1}, \ldots, M'_{\rho_k})$ as its final output.

Ideal Experiment. We define the experiment $\mathbf{Ideal}_{\widehat{\mathsf{S}}',\widehat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ as follows.
The (possibly malicious) algorithm $\widehat{\mathsf{S}}'(M_1, \ldots, M_N)$ generates messages $M'_1, \ldots, M'_N$ which are given to the trusted party T. In each of the k transfers, T obtains a bit b_i from the sender $\widehat{\mathsf{S}}'$ and an index ρ'_i from the (possibly malicious) receiver $\widehat{\mathsf{R}}'(\rho_i)$. If $b_i = 1$, and $\rho'_i \in [N]$, then T reveals M'_{ρ_i} to the receiver $\widehat{\mathsf{R}}'$. Otherwise, $\widehat{\mathsf{R}}'$ receives $\perp$ from T. At the end of the k-th transfer, $\widehat{\mathsf{S}}'$ and $\widehat{\mathsf{R}}'$ output a string S_k and R_k and the output of the experiment is the pair (S_k, R_k).

The ideal sender $\mathsf{S}'(M_1, \ldots, M_N)$ is defined the be the sender that sends $(M_1, \ldots, M_N)$ which sends the messages $(M_1, \ldots, M_N)$ to T in the initialization phase, sends $b_i = 1$ in each transfer and outputs the final state $S_k = \varepsilon$. The honest ideal receiver R' is defined to be the algorithm that sends T the real selection index ρ_i at each transfer and eventually outputs the list of all received messages $R_k = (M'_{\rho_1}, \ldots, M'_{\rho_k})$ as its final state.

The bit b_i sent by $\widehat{\mathsf{S}}'$ at each transfer models its capability of making the transfer fail. By forcing $\widehat{\mathsf{S}}'$ to choose b_i without seeing ρ_i, the definition prevents the cheating sender $\widehat{\mathsf{S}}'$ from deciding to cause a failure of the transfer for specific values of ρ_i.

Definition 9.1 (Sender Security)**.** An OT_k^N protocol is *sender-secure* if, for any PPT real-world cheating receiver $\widehat{\mathsf{R}}$, there exists a PPT ideal-world receiver $\widehat{\mathsf{R}}'$ such that, for any polynomial $N_m(\lambda)$, any $N \in [N_m(\lambda)]$, any $k \in [N]$, any messages $M_1, \ldots, M_N$, and any indices $\rho_1, \ldots, \rho_k \in [N]$, no PPT distinguisher can separate the two following distributions with noticeable advantage:

$$\mathbf{Real}_{\mathsf{S},\widehat{\mathsf{R}}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$$

and

$$\mathbf{Ideal}_{\mathsf{S}',\widehat{\mathsf{R}}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k).$$

Definition 9.2 (Receiver Security)**.** An OT_k^N protocol is *receiver-secure* if, for any PPT real-world cheating sender $\widehat{\mathsf{S}}$, there exists a PPT ideal-world sender $\widehat{\mathsf{S}}'$ such that, for any polynomial $N_m(\lambda)$, any $N \in [N_m(\lambda)]$, any $k \in [N]$, any messages $M_1, \ldots, M_N$, and any indices $\rho_1, \ldots, \rho_k \in [N]$, no PPT distinguisher can tell apart the two following distributions with non-negligible advantage:

$$\mathbf{Real}_{\widehat{\mathsf{S}},\mathsf{R}}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$$

and

$$\mathbf{Ideal}_{\widehat{\mathsf{S}}',\mathsf{R}'}(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k).$$

9.1.2 Adaptive Oblivious Transfer with Access Control

Camenisch *et al.* [CDN09] define oblivious transfer with access control (OT-AC) as a tuple of PPT algorithms/protocols $(\mathsf{ISetup}, \mathsf{Issue}, \mathsf{DBSetup}, \mathsf{Transfer})$ such that:

ISetup: takes as inputs public parameters p specifying a set $\mathcal{P}$ of access policies and generates a key pair (PK_I, SK_I) for the issuer.

Issue: is an interactive protocol between the issuer I and a stateful user U under common input (p, x), where x is an attribute string. The issuer I takes as inputs its key pair (PK_I, SK_I) and a user pseudonym P_{U}. The user takes as inputs its state information st_{U}. The user U outputs either an error symbol $\perp$ or a credential $\mathsf{Cred}_{\mathsf{U}}$, and an updated state st'_{U}.

DBSetup: is an algorithm that takes as input the issuer's public key PK_I, a database $DB = (M_i, \mathsf{AP}_i)_{i=1}^N$ containing records M_i whose access is restricted by an access policy AP_i and outputs a database public key PK_{DB}, an encryption of the records $(ER_i)_{i=1}^N$ and a database secret key SK_{DB}.

Transfer: is a protocol between the database DB and a user U with common inputs (PK_I, PK_{DB}). DB inputs SK_{DB} and U inputs $(\rho, st_{\mathsf{U}}, ER_\rho, \mathsf{AP}_\rho)$, where $\rho \in [N]$ is a record index to which U is requesting access. The interaction ends with U outputting $\perp$ or a string $M_{\rho'}$ and an updated state st'_{U}.

We assume private communication links, so that communications between a user and the issuer are authenticated, and those between a user and the database are anonymized: otherwise, anonymizing the Transfer protocol is impossible.

The security definitions formalize two properties called *user anonymity* and *database security*. The former captures that the database should be unable to tell which honest user is making a query and neither can tell which records are being accessed. This should remain true even if the database colludes with corrupted users and the issuer. As for database security, the intuition is that a cheating user cannot access a record for which it does not have the required credentials, even when colluding with other dishonest users. In case the issuer is colluding with these cheating users, they cannot obtain more records from the database than they retrieve.

Similarly to the $\mathcal{OT}_{k\times 1}^N$ case, security is defined by requiring that any PPT real-world adversary $\mathcal{A}$ and any environment $\mathcal{E}$, there exists a PPT adversary $\mathcal{A}'$ which controls the

same parties and such that no environment $\mathcal{E}$ can tell if it is running in the real world interacting with the real $\mathcal{A}$ or in the ideal-world interacting with $\mathcal{A}'$. The distribution of outputs of the environment in the different settings is denoted by $\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)$ and $\mathbf{Ideal}_{\mathcal{E},\mathcal{A}'}(\lambda)$ for real-world adversary $\mathcal{A}$ and ideal-world adversary $\mathcal{A}'$, respectively.

Definition 9.3. An AC-OT protocol is said to securely implement the functionality if for any real-world adversary $\mathcal{A}$ and any real world environment $\mathcal{E}$, there exists an ideal-world simulator $\mathcal{A}'$ controlling the same parties in the ideal-world as $\mathcal{A}$ does in the real-world, such that

$$|\mathbf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathbf{Ideal}_{\mathcal{E},\mathcal{A}}(\lambda)| \leq \mathsf{negl}(n)(\lambda).$$

Real World. We describe the way that real-world algorithms interact when all participants (i.e., the real-world users $\mathsf{U}_1, \ldots, \mathsf{U}_U$, the database DB and the issuer I) are honest. The issuer starts by generating a key pair $(PK_I, SK_I) \leftarrow \mathsf{ISetup}(\mathsf{p})$, and sends PK_I to all users $\{\mathsf{U}_i\}_{i=1}^{U}$ and the database DB.

When $\mathcal{E}$ sends a message $(\texttt{initdb}, \mathrm{DB} = (M_i, \mathsf{AP}_i)_{i=1}^{N})$ to the database DB, the latter encrypts the database DB by running $\mathsf{DBSetup}$ and sends the encrypted records to all users.

When $\mathcal{E}$ sends a message $(\texttt{issue}, x)$ to user U_i, this user starts an Issue protocol with the issuer on common input x, at the end of which it returns 1 to the environment if the protocol succeeded or 0 otherwise.

When $\mathcal{E}$ sends a message $(\texttt{transfer}, \rho)$ to user U_i, this user first checks if its credentials $\mathsf{Cred}_{\mathsf{U}}$ are sufficient to access the record M_ρ. If it is the case, it engages in a $\mathsf{Transfer}$ protocol with the database DB, at the end of which it receives either the message M_ρ, or an error symbol $\perp$. If it failed at any steps, the user returns 0 to $\mathcal{E}$, or 1 if it succeeded.

Notice that in this setting, neither the database nor the issuer return any outputs to the environment.

Ideal World. In the ideal world, participants only communicate via the trusted party T which implements the functionality of the protocol. We describe how T proceeds when receiving inputs from the ideal-world users $\{\mathsf{U}'_i\}_{i=1}^{U}$, issuer I' and database DB'. T maintains an initially empty set C_i for each user U'_i and sets $\mathrm{DB} \leftarrow \perp$. It handles the queries of the different parties as follows:

- When receiving a message $(\texttt{initdb}, \mathrm{DB} = (M_i, \mathsf{AP}_i)_{i=1}^{N})$ from DB', T sets $\mathrm{DB} = (M_i, \mathsf{AP}_i)_{i=1}^{N}$.
- When receiving $(\texttt{issue}, x)$ from U'_i, T sends $(\texttt{issue}, \mathsf{U}'_i, x)$ to I' which replies with a bit b. If $b = 1$, then T adds x to C_i. In any cases, T sends b to U'_i.
- When receiving $(\texttt{transfer}, \rho)$ from U'_i, the trusted party T acts as follows. If U'_i previously sent a message of the form $(\texttt{transfer}, .)$, T defines $f_{\mathsf{U}',DB} = 1$. Otherwise, it sets $f_{\mathsf{U}',DB} = 0$. If $\mathrm{DB} \neq \perp$, it sends $(\texttt{transfer}, f_{\mathsf{U}',DB})$ to DB', who sends a bit b. If $b = 1$ and if st_i contains a vector $\mathbf{x}$ such that $\mathsf{AP}_i(x) = 1$, then it sends the record to U'_i. In any other cases, it sends $\perp$ to U'_i.

In other words, the ideal-world users, database and issuer relay inputs and outputs between the environment $\mathcal{E}$ and the trusted party T.

Note that, like [CDN09], the ideal functionality allows the database to learn whether a given user interacts with the database for the first time or not. The reason is that, like the protocol of [CDN09], our basic OT-AC scheme requires the database to provide a particular interactive zero-knowledge proof at the very first time each user queries the database. In protocols where the database generates such an interactive proof, it is inevitable for U to reveal his state bit f_{DB} to DB. In constructions where the zero-knowledge proof is made non-interactive and made publicly available at the same time as the database itself, this can be avoided and we can prevent DB from learning the state bit f_{DB}. In this case, T does not send $f_{\mathsf{U}',DB}$ to DB' in the ideal-world experiment.

The ideal world thus implies the following security properties.

User Anonymity. The database cannot tell which user a given query comes from and neither can it tell which record is being accessed. It only learns whether the user previously queried the database or not. Otherwise, two transfers involving the same users are unlinkable.

Database Security. A single cheating user cannot access a record for which he does not have a certified authorized attribute string. Colluding users cannot pool their credentials to gain access to a record which none of them can individually access. Moreover, if the issuer colludes with some users, the protocol still provides the equivalent of sender security in the $\mathcal{OT}^N_{k\times 1}$ functionality.

9.2 Building Blocks

We will use two distinct signature schemes because one of them only needs to be secure in the sense of a weaker security notion and can be more efficient. This weaker notion is sufficient to sign the database entries and allows a better efficiency in the scheme of Section 9.3. In particular, by making it stateful (which also suffices since all database entries are signed at once), we can reduce the public key size to $\log N$ matrices if N is the number of database entries. The second scheme must be stateful and secure in the standard EUF-CMA sense since the issuer uses it to certify users' attributes. The signature scheme of Section 7.1 is only used in the OT-AC protocol of Section 9.3 while the scheme of Section 9.2.1 is used in the adaptive OT protocol of Section 9.4 as well.

We first use the signature scheme described in Section 7.1 which extends the the Böhl *et al.* signature [BHJ+15] in order to sign messages comprised of multiple blocks while keeping the scheme compatible with zero-knowledge proofs.

9.2.1 A Simpler Variant with Bounded-Message Security and Security Against Non-Adaptive Chosen-Message Attacks

We consider a stateful variant of the scheme in Section 7.1 where a bound $Q \in \mathsf{poly}(n)$ on the number of signed messages is fixed at key generation time. In the context of $\mathcal{OT}^N_{k\times 1}$, this is sufficient and leads to efficiency improvements. In the modified scheme hereunder,

the string $\tau \in \{0,1\}^\ell$ is an ℓ-bit counter maintained by the signer to keep track of the number of previously signed messages.

This simplified variant resembles the SIS-based signature scheme of Böhl *et al.* [BHJ+15].

In this version, the message space is $\{0,1\}^{n\lceil \log q \rceil}$ so that vectors of $\mathbb{Z}_q^n$ can be signed by first decomposing them using $\mathsf{vdec}_{n,q-1}(.)$.

Keygen$(1^\lambda, 1^Q)$**:** Given $\lambda > 0$ and the maximal number $Q \in \mathsf{poly}(\lambda)$ of signatures, choose $n = \mathcal{O}(\lambda)$, a prime $q = \widetilde{\mathcal{O}}(Q \cdot n^4)$, $m = 2n\lceil \log q \rceil$, an integer $\ell = \lceil \log Q \rceil$ and Gaussian parameters $\sigma = \Omega(\sqrt{n \log q} \log n)$. The message space is $\{0,1\}^{m_d}$, for some $m_d \in \mathsf{poly}(\lambda)$ with $m_d \geq m$.

1. Run $\mathsf{TrapGen}(1^n, 1^m, q)$ to get $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T_A}$ of $\Lambda_q^\perp(\mathbf{A})$, which allows sampling short vectors in $\Lambda_q^\perp(\mathbf{A})$ with a Gaussian parameter σ. Next, choose $\ell + 1$ random $\mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \hookleftarrow U(\mathbb{Z}_q^{n \times m})$.
2. Choose $\mathbf{D} \hookleftarrow U(\mathbb{Z}_q^{n \times m_d})$ as well as a random vector $\mathbf{u} \hookleftarrow U(\mathbb{Z}_q^n)$.

The counter τ is initialized to $\tau = 0$. The private key consists of $SK := \mathbf{T_A}$ and the public key is $PK := (\mathbf{A},\ \{\mathbf{A}_j\}_{j=0}^\ell,\ \mathbf{D},\ \mathbf{u})$.

Sign$(SK, \tau, \mathfrak{m})$**:** To sign a message $\mathfrak{m} \in \{0,1\}^{m_d}$,

1. Increment the counter by setting $\tau := \tau + 1$ and interpret it as a string $\tau \in \{0,1\}^\ell$. Then, using $SK := \mathbf{T_A}$, compute a short delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$ for the matrix $\mathbf{A}_\tau = [\mathbf{A} \mid \mathbf{A}_0 + \sum_{j=1}^\ell \tau[j]\mathbf{A}_j] \in \mathbb{Z}_q^{n \times 2m}$.
2. Compute the vector $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m} \in \mathbb{Z}_q^n$. Then, using the delegated basis $\mathbf{T}_\tau \in \mathbb{Z}^{2m \times 2m}$, sample a short vector $\mathbf{v} \in \mathbb{Z}^{2m}$ in $D_{\Lambda_q^{\mathbf{u}_M}(\mathbf{A}_\tau), \sigma}$.

Output the signature $sig = (\tau, \mathbf{v}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m}$.

Verify$(PK, \mathfrak{m}, sig)$**:** Given PK, $\mathfrak{m} \in \{0,1\}^{m_d}$ and a signature $sig = (\tau, \mathbf{v}) \in \{0,1\}^\ell \times \mathbb{Z}^{2m}$, return 1 if $\|\mathbf{v}\| < \sigma\sqrt{2m}$ and $\mathbf{A}_\tau \cdot \mathbf{v} = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m} \bmod q$.

For our purposes, the scheme only needs to satisfy a notion of bounded-message security under non-adaptive chosen-message attack. In this relaxed model, the adversary only obtains a bounded number of signatures for messages that are chosen non-adaptively (i.e., all at once and before seeing the public key) by the adversary. This security notion is sufficient for signing the N database entries. Note that the queries are non-adaptive but the adversary can adaptively choose its forgery message.

Theorem 9.1. *The scheme is bounded message secure under non-adaptive chosen-message attacks if the* SIS *assumption holds.*

Proof. We show that the scheme presented in Section 9.2.1 is secure against non-adaptive chosen-message attacks (na-CMA) under the SIS assumption. The shape of the proof is similar to the security proof of the signature scheme of Section 7.1. Namely, to prove the security, we distinguish two kinds of attacks:

Type I attacks, where in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star)$, $\tau^\star$ did not appear in any outputs of the signing oracle.

Type II attacks, where in the adversary's forgery $sig^\star = (\tau^\star, \mathbf{v}^\star)$, $\tau^\star$ has been recycled from an output $sig^{(i^\star)} = (\tau^{(i^\star)}, \mathbf{v}^{(i^\star)})$ of the signing oracle for some query $i^\star \in \{1, \ldots, Q\}$.

Lemma 9.2 states that the signature scheme is secure against Type I forgery using the same technique as is [ABB10, Boy10, MP12]. Lemma 9.3 claims that the signature scheme resists Type II attacks, with a proof that is very similar to the one of Lemma 9.2. Both security proofs assume the computational hardness of the SIS problem. □

Lemma 9.2. *The signature scheme of Section 9.2.1 is secure against Type I attacks if the* $\mathsf{SIS}_{n,m,q,\beta'}$ *assumption holds, with* $\beta' = \sigma^2 m^{3/2}(\ell + 2) + \sigma m^{1/2}$.

Proof. Let $\mathcal{A}$ be a PPT adversary against the na-CMA security of our scheme that mounts Type I attacks with non negligible success probability ε. We construct a PPT algorithm $\mathcal{B}$ using $\mathcal{A}$ to break the $\mathsf{SIS}_{n,m,q,\beta'}$ assumption. Our reduction $\mathcal{B}$ takes as input a target matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$ and computes $\mathbf{v} \in \Lambda_q^\perp(\bar{\mathbf{A}})$ satisfying $0 < \|\mathbf{v}\| \leq \beta'$.

At first, $\mathcal{B}$ calls $\mathcal{A}$ to obtain the messages to be queried: $\mathfrak{m}^{(1)}, \ldots, \mathfrak{m}^{(Q)}$. For the sake of readability, let us define $\tau^{(i)} = i$, viewed as a bit-string, to be the tag corresponding to the i-th signature in our scheme.

Setup. As in [HW09], the reduction guesses the shortest prefix such that the string $\tau^\star$ embedded in $\mathcal{A}$'s forgery differs from all prefixes to $\{\tau^{(1)}, \ldots, \tau^{(Q)}\}$. To achieve this, $\mathcal{B}$ chooses at random $i^\dagger \hookleftarrow U(\{1, \ldots, Q\})$ and $t^\dagger \hookleftarrow U(\{1, \ldots, \ell\})$. Then, with probability $1/(Q \cdot \ell)$, the longest common prefix between $\tau^\star$ and one of the tags $\{\tau^{(i)}\}_{i=1}^Q$ is the string $\tau^\star[1] \cdots \tau^\star[t^\dagger - 1] \in \{0,1\}^{t^\dagger - 1}$: the first $(t^\dagger - 1)$-th bits of $\tau^\star$. Let us define $\tau^\dagger = \tau^\star_{|t^\dagger}$, where $s_{|i}$ denotes the i-th prefix for a string s. By construction $\tau^\dagger \notin \{\tau^{(1)}_{|t^\dagger}, \ldots, \tau^{(Q)}_{|t^\dagger}\}$ with probability $1/(Q \cdot \ell)$.

Next, the reduction $\mathcal{B}$ runs $\mathsf{TrapGen}(1^n, 1^m, q)$ to obtain matrices $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ and a short basis $\mathbf{T}_\mathbf{C} \in \mathbb{Z}^{m \times m}$ of $\Lambda_q^\perp(\mathbf{C})$, which will be useful to answer the following opening oracle queries. The reduction $\mathcal{B}$ continues by picking $\ell + 1$ matrices $\mathbf{Q}_0, \ldots, \mathbf{Q}_\ell \in \mathbb{Z}^{m \times m}$ where each matrix $\mathbf{Q}_i$ has its column independently sampled from $D_{\mathbb{Z}^m, \sigma}$, and $\mathcal{B}$defines the matrices $\mathbf{A} = \bar{\mathbf{A}}$ and $\{\mathbf{A}_j\}_{j=0}^\ell$ as follows

$$\begin{cases} \mathbf{A}_0 = \bar{\mathbf{A}} \cdot \mathbf{Q}_0 + \left(\sum_{j=1}^{t^\dagger} \tau^\star[j]\right) \cdot \mathbf{C} & \\ \mathbf{A}_j = \bar{\mathbf{A}} \cdot \mathbf{Q}_j + (-1)^{\tau^\star[j]} \cdot \mathbf{C} & \text{for } j \in [1, t^\dagger] \\ \mathbf{A}_j = \bar{\mathbf{A}} \cdot \mathbf{Q}_j & \text{for } j \in [t^\dagger + 1, \ell] \end{cases}.$$

We can notice that

$$\begin{aligned}\mathbf{A}_{\tau^{(i)}} &= \Big[\mathbf{A} \;\Big|\; \mathbf{A}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j]\mathbf{A}_j\Big] \\ &= \Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot (\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot \mathbf{Q}_j) + (\sum_{j=1}^{t^\dagger} \tau^\star[j] + (-1)^{\tau^\star[j]} \cdot \tau^{(i)}[j]) \cdot \mathbf{C}\Big] \\ &= \Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot (\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot \mathbf{Q}_j) + h_{\tau^{(i)}} \cdot \mathbf{C}\Big],\end{aligned}$$

where $h_{\tau^{(i)}}$ denotes the hamming distance between $\tau^{(i)}_{|t^\dagger}$ and $\tau^\dagger$. With probability $1/(Q \cdot \ell)$, and as $\ell > q$, it holds that $h_{\tau^{(i)}} \neq 0 \bmod q$ whenever $\tau^{(i)}_{|t^\dagger} \neq \tau^\star_{|t^\dagger}$.

The reduction then picks a random short matrix $\mathbf{R} \hookleftarrow \mathbb{Z}^{m \times m_d}$ which has its m_d columns independently sampled from $D_{\mathbb{Z}^m,\sigma}$, and $\mathcal{B}$computes

$$\mathbf{D} = \bar{\mathbf{A}} \cdot \mathbf{R} \in \mathbb{Z}_q^{n \times m_d}.$$

To finish, $\mathcal{B}$ samples a short vector $\mathbf{e}_u \in D_{\mathbb{Z}^m,\sigma}$ and computes the vector $\mathbf{u} = \bar{\mathbf{A}} \cdot \mathbf{e}_u$. The following public key is finally given to $\mathcal{A}$:

$$PK := (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u}).$$

Signing queries. To handle signature queries, the reduction $\mathcal{B}$ uses the trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ to generate a signature. To this end, $\mathcal{B}$ starts by computing the vector $\mathbf{u}_M = \mathbf{u} + \mathbf{D} \cdot \mathfrak{m}^{(i)}$. Then $\mathcal{B}$ can use $\mathbf{T_C}$ with the algorithm SampleRight from Lemma 3.7 to compute a short vector $\mathbf{v}^{(i)}$ in $D^{\mathbf{u}_M}_{\Lambda^\perp(\mathbf{A}_{\tau^{(i)}}),\sigma}$, distributed like a valid signature and satisfying the verification equation (7.2).

Output. At some point, the attacker $\mathcal{A}$ halts and outputs a *valid* signature $sig^\star = (\tau^\star, \mathbf{v}^\star)$ for a message $\mathfrak{m}^\star \notin \{\mathfrak{m}^{(1)}, \ldots, \mathfrak{m}^{(Q)}\}$. Since the signature is valid, it satisfies $\|\mathbf{v}^\star\| \leq \sigma\sqrt{2m}$.

Parsing $\mathbf{v}^\star$ as $[\mathbf{v}_1^\star \mid \mathbf{v}_2^\star]$ with $\mathbf{v}_1^\star, \mathbf{v}_2^\star \in \mathbb{Z}^m$ and injecting it in (7.2) give:

$$\begin{aligned}\Big[\bar{\mathbf{A}} \;\Big|\; \bar{\mathbf{A}} \cdot (\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^\star[j] \cdot \mathbf{Q}_j)\Big] \cdot \begin{bmatrix}\mathbf{v}_1^\star \\ \mathbf{v}_2^\star\end{bmatrix} &= \mathbf{u} + \mathbf{D} \cdot \mathfrak{m}^\star \mod q \\ &= \bar{\mathbf{A}} \cdot (\mathbf{e}_u + \mathbf{R} \cdot \mathfrak{m}^\star) \mod q\end{aligned}$$

Thus, the vector

$$\mathbf{v}' = \mathbf{v}_1^\star + (\mathbf{Q}_0 + \sum_{j=1}^{\ell} \tau^\star[j] \cdot \mathbf{Q}_j) \cdot \mathbf{v}_2^\star - \mathbf{e}_u - \mathbf{R} \cdot \mathfrak{m}^\star$$

is in $\Lambda^\perp(\bar{\mathbf{A}})$, and $\mathbf{v}'$ is non-zero with overwhelming probabilities, since in $\mathcal{A}$'s view, the distribution of $\mathbf{e}_u$ is $D_{\Lambda^{\mathbf{u}}_q(\mathbf{A}),\sigma}$, which guarantees that $\mathbf{e}_u$ is statistically hidden by the syndrome $\mathbf{u} = \bar{\mathbf{A}} \cdot \mathbf{e}_u$. Finally, the norm of $\mathbf{v}'$ is upper bounded by $\beta' = \sigma^2 m^{3/2}(\ell + 2) + 2\sigma m^{1/2}$. □

Lemma 9.3. *The signature scheme of Section 9.2.1 is secure against Type II attacks if* $\mathsf{SIS}_{n,m,q,\beta''}$ *holds, with* $\beta'' = \sqrt{2}(\ell+2)\sigma m^{3/2} + m^{1/2}$.

Proof. We will prove this result using techniques analogous to the previous proof. We show that given an adversary $\mathcal{A}$ that comes out with a Type II signature in the na-CMA game with non negligible probability ε, we can construct a PPT $\mathcal{B}$ that breaks the SIS assumption with advantage ε/Q using $\mathcal{A}$.

Firstly, the reduction $\mathcal{B}$ is given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m_d}$ as input and has to output an integer vector $\mathbf{v} \in \mathbb{Z}^{m_d}$ in $\Lambda_q^{\perp}(\mathbf{A})$ such that $0 < \|\mathbf{v}\| \leq \beta''$. Next, $\mathcal{B}$ receives from $\mathcal{A}$ the messages $\mathfrak{m}^{(1)}, \ldots, \mathfrak{m}^{(Q)}$ for which $\mathcal{A}$ will further ask signature queries.

To compute the public key, at the outset of the game, the reduction $\mathcal{B}$ starts by sampling $i^{\dagger} \hookleftarrow U(\{1, \ldots, Q\})$ corresponding to the guess that $\mathcal{A}$'s forgery will recycle $\tau^{(i\dagger)}$. This is independent of $\mathcal{A}$'s view, and the guess will be correct with probability $1/Q$. Using this guess to compute PK, the reduction $\mathcal{B}$ picks $h_0, \ldots, h_\ell \in \mathbb{Z}_q$ subject to the constraints

$$\begin{cases} h_0 + \sum_{j=1}^{\ell} \tau^{(i^{\dagger})}[j] \cdot h_j = 0 \mod q \\ h_0 + \sum_{j=1}^{\ell} \tau^{(i)}[j] \cdot h_j \neq 0 \mod q \quad \forall i \in \{1, \ldots, Q\} \backslash \{i^{\dagger}\} \end{cases} \tag{9.1}$$

$\mathcal{B}$then runs $(\mathbf{C}, \mathbf{T_C}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. The resulting matrix $\mathbf{C} \in \mathbb{Z}_q^{n \times m}$ is statistically random, and the trapdoor $\mathbf{T_C} \in \mathbb{Z}^{m \times m}$ is a short basis of $\Lambda_q^{\perp}(\mathbf{C})$. Next $\mathcal{B}$re-randomize $\mathbf{A}$ using short matrices $\mathbf{S}, \mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_\ell \in \mathbb{Z}^{m_d \times m}$ which are obtained by sampling their columns from the distribution $D_{\mathbb{Z}^{m_d}, \sigma}$. The challenger $\mathcal{B}$ then uses these matrices to define:

$$\begin{aligned} \mathbf{A} &= \mathbf{A} \cdot \mathbf{S} \\ \mathbf{A}_0 &= \mathbf{A} \cdot \mathbf{S}_0 + h_0 \cdot \mathbf{C} \\ \mathbf{A}_j &= \mathbf{A} \cdot \mathbf{S}_j + h_j \cdot \mathbf{C} \qquad\qquad j \in \{1, \ldots, \ell\} \end{aligned}$$

and sets $\mathbf{D} = \mathbf{A} \in \mathbb{Z}_q^{n \times m_d}$. Observe that matrices $\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}$ are all statistically uniform over $\mathbb{Z}_q^{n \times m}$. Then, $\mathcal{B}$ samples short vectors $\mathbf{v}_1^{\dagger}, \mathbf{v}_2^{\dagger} \hookleftarrow D_{\mathbb{Z}^m, \sigma}$ and computes $\mathbf{u} \in \mathbb{Z}_q^n$ as

$$\mathbf{u} = \mathbf{A}_{\tau^{(i^{\dagger})}} \cdot \begin{bmatrix} \mathbf{v}_1^{\dagger} \\ \mathbf{v}_2^{\dagger} \end{bmatrix} - \mathbf{A} \cdot \mathfrak{m}^{(i^{\dagger})} \mod q. \tag{9.2}$$

Finally, $\mathcal{B}$ sends to $\mathcal{A}$ the public key

$$PK := (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u})$$

which is distributed as the PK of the real scheme.

To answer signing queries, the challenger $\mathcal{B}$ do as follows.

- If the query is not the $i^{\dagger}$-th, we have:

$$\begin{aligned} \mathbf{A}_{\tau^{(i)}} &= \left[\mathbf{A} \;\middle|\; \mathbf{A}_0 + \sum_{j=0}^{\ell} \tau^{(i)}[j] \cdot \mathbf{A}_j \right] \\ &= \left[\mathbf{A} \cdot \mathbf{S} \;\middle|\; \mathbf{A} \cdot (\mathbf{S}_0 + \sum_{j=0}^{\ell} \tau^{(i)}[j] \cdot \mathbf{S}_j) + h_{\tau^{(i)}} \cdot \mathbf{C} \right], \end{aligned}$$

with $h_{\tau^{(i)}} = h_0 + \sum \tau^{(i)}[j] \cdot h_j \neq 0$ due to the first constraint of (9.1). Thus, using the same technique as in the previous proof from [MP12], the challenger $\mathcal{B}$ can use the trapdoor $\mathbf{T_C}$ along with SampleRight algorithm to sample a short vector in $\Lambda_q^{\mathbf{u}_M}(\mathbf{A}_{\tau^{(i)}})$ satisfying (7.2).

- At the $i^\dagger$-th query, thanks to the second constraint of (9.1), we have:

$$\begin{aligned}\mathbf{A}_{\tau^{(i^\dagger)}} &= \left[\mathbf{A} \;\middle|\; \mathbf{A}_0 + \sum_{j=0}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{A}_j\right] \\ &= \left[\mathbf{A}\cdot\mathbf{S} \;\middle|\; \mathbf{A}\cdot(\mathbf{S}_0 + \sum_{j=0}^{\ell} \tau^{(i^\dagger)}[j] \cdot \mathbf{S}_j)\right].\end{aligned}$$

  To answer this specific query, the challenger $\mathcal{B}$ returns $sig^{(i^\dagger)} = (\tau^{(i^\dagger)}, \mathbf{v}^{(i^\dagger)})$ where $\mathbf{v}^{(i^\dagger)} = (\mathbf{v}_1^{\dagger T} \mid \mathbf{v}_2^{\dagger T})^T$ verifying (9.2), which furthermore implies that $sig^{(i^\dagger)}$ verifies (7.2).

Thus we claim that $\mathcal{B}$ can solve the SIS problem using the Type II forgery provided by $\mathcal{A}$. At the end of the game, the adversary outputs a valid signature $sig^\star = (\tau^{(i^\star)}, \mathbf{v}^\star)$ on a message $\mathfrak{m}^\star$ with $\|\mathbf{v}^\star\| \leq \sigma\sqrt{2m}$. In the event that $\tau^{(i^\star)} \neq \tau^{i^\dagger}$, the reduction aborts. The latter event happens with probability $1 - 1/Q$. If we parse $\mathbf{v}^\star$ as $(\mathbf{v}_1^{\star,T} \mid \mathbf{v}_2^{\star T})^T \in \mathbb{Z}^{2m}$, with $\mathbf{v}_1^\star, \mathbf{v}_2^\star \in \mathbb{Z}^m$, it holds that:

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \begin{bmatrix}\mathbf{v}_1^\star \\ \hline \mathbf{v}_2^\star\end{bmatrix} = \mathbf{u} + \mathbf{A}\cdot\mathfrak{m}^\star \mod q. \tag{9.3}$$

According to the way $\mathbf{u}$ was defined at the beginning of the game, we also have a vector $\mathbf{v}^\dagger = (\mathbf{v}_1^{\dagger T} \mid \mathbf{v}_2^{\dagger T})^T$ such that

$$\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \begin{bmatrix}\mathbf{v}_1^\dagger \\ \hline \mathbf{v}_2^\dagger\end{bmatrix} = \mathbf{u} + \mathbf{A}\cdot\mathfrak{m}^\dagger \mod q. \tag{9.4}$$

As $sig^\star$ is a valid forgery for the dn-CMA game, it follows that $m^\dagger \neq m^\star$. And we get by subtracting (9.3) and (9.4)

$$\begin{aligned}\mathbf{A}_{\tau^{(i^\dagger)}} \cdot \begin{bmatrix}\mathbf{v}_1^\star - \mathbf{v}_1^\dagger \\ \hline \mathbf{v}_2^\star - \mathbf{v}_2^\dagger\end{bmatrix} &= \mathbf{A}\cdot\left(\mathfrak{m}^\star - \mathfrak{m}^\dagger\right) \mod q, \\ \left[\mathbf{A}\cdot\mathbf{S} \;\middle|\; \mathbf{A}\cdot(\mathbf{S}_0 + \sum_{j=0}^{\ell} \tau^{(i^\dagger)}[j]\cdot\mathbf{S}_j)\right] \cdot \begin{bmatrix}\mathbf{v}_1^\star - \mathbf{v}_1^\dagger \\ \hline \mathbf{v}_2^\star - \mathbf{v}_2^\dagger\end{bmatrix} &= \mathbf{A}\cdot\left(\mathfrak{m}^\star - \mathfrak{m}^\dagger\right) \mod q.\end{aligned}$$

Leading us to the fact that

$$\mathbf{v}' = \underbrace{\mathbf{S}\cdot(\mathbf{v}_1^\star - \mathbf{v}_2^\dagger) + \left(\mathbf{S}_0 + \sum_{j=1}^{\ell} \tau^{(i^\dagger)}[j]\cdot\mathbf{S}_j\right)\cdot(\mathbf{v}_2^\star - \mathbf{v}_2^\dagger)}_{(a)} + \underbrace{\mathfrak{m}^\dagger - \mathfrak{m}^\star}_{-(b)} \tag{9.5}$$

is an integer vector of $\Lambda_q^\perp(\mathbf{A})$, with norm bounded by $\|\mathbf{v}'\| \leq \sqrt{2}(\ell+2)\sigma m^{3/2} + m^{1/2} = \beta''$. Furthermore, if $\mathbf{v}'$ was zero, it implies that $(a) = (b)$ in Equation (9.5). And as $sig^\star \neq sig^\dagger$,

we have that either $\mathbf{v}_1^\star \neq \mathbf{v}_1^\dagger$ or $\mathbf{v}_2^\star \neq \mathbf{v}_2^\dagger$. As a consequence, (a) is information theoretically unpredictable for $\mathcal{A}$ since the columns of $\mathbf{S}, \mathbf{S}_0, \ldots \mathbf{S}_\ell$ are statistically hidden from $\mathcal{A}$, as shown in [MP12] for instance: conditionally on the public key, each column of $\mathbf{S}$ and $\{\mathbf{S}_j\}_{j=0}^{\ell}$ has at least n bits of min-entropy. □

9.3 A Fully Simulatable Adaptive OT Protocol

Our basic $\mathcal{OT}_{k\times 1}^N$ protocol builds on the "assisted decryption" technique [CNs07]. The databases holder encrypts all entries using a multi-bit variant [PVW08] of Regev's cryptosystem [Reg05] and proves the well-formedness of its public key and all ciphertexts. In addition, all ciphertexts are signed using a signature scheme. At each transfer, the receiver statistically re-randomizes a blinded version of the desired ciphertext, where the blinding is done via the additive homomorphism of Regev. Then, the receiver provides a witness indistinguishable (WI) argument that the modified ciphertext (which is submitted for oblivious decryption) is a transformation of one of the original ciphertexts by arguing knowledge of a signature on this hidden ciphertext. In response, the sender obliviously decrypts the modified ciphertext and argues in zero-knowledge that the response is correct.

Adapting the technique of [CNs07] to the lattice setting requires the following building blocks: (i) A signature scheme allowing to sign ciphertexts while remaining compatible with ZK proofs; (ii) A ZK protocol allowing to prove knowledge of a signature on some hidden ciphertext which belongs to a public set and was transformed into a given ciphertext; (iii) A protocol for proving the correct decryption of a ciphertext; (iv) A method of statistically re-randomizing an LWE-encrypted ciphertext in a way that enables oblivious decryption. The first three ingredients can be obtained from Chapter 7. Since component (i) only needs to be secure against random-message attacks as long as the adversary obtains at most N signatures, we use the simplified SIS-based signature scheme of Section 9.2.1. The statistical re-randomization of Regev ciphertexts is handled via the noise flooding technique [AJLA+12], which consists in drowning the initial noise with a sub-exponentially larger noise. While recent results [DS16, BdPMW16] provide potentially more efficient alternatives, we chose the flooding technique for simplicity because it does not require the use of FHE (and also because the known multi-bit version [HAO15] of the GSW FHE [GSW13] incurs an *ad hoc* circular security assumption).

9.3.1 Description

Our scheme works with security parameter λ, modulus q, lattice dimensions $n = \mathcal{O}(\lambda)$ and $m = 2n\lceil \log q \rceil$. Let $B_\chi = \widetilde{\mathcal{O}}(\sqrt{n})$, and let χ be a B_χ-bounded distribution. We also define an integer B as a randomization parameter such that $B = n^{\omega(1)} \cdot (m+1)B_\chi$ and $B + (m+1)B_\chi \leq q/5$ (to ensure decryption correctness). Our basic $\mathcal{OT}_{k\times 1}^N$ protocol goes as follows.

Initialization $(\mathsf{S}_\mathsf{I}(1^\lambda, \mathsf{DB}), \mathsf{R}_\mathsf{I}(1^\lambda))$**:** In this protocol, the sender S_I has a database $\mathsf{DB} = (M_1, \ldots, M_N)$ of N messages, where $M_i \in \{0,1\}^t$ for each $i \in [N]$, for some $t \in \mathsf{poly}(\lambda)$. It interacts with the receiver R_I as follows.

1. Generate a key pair for the signature scheme of Section 9.2.1 in order to sign $Q = N$ messages of length $m_d = (n+t)\cdot\lceil \log q \rceil$ each. This key pair consists of

$SK_{sig} = \mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ and $PK_{sig} := (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u})$, where $\ell = \log N$ and $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in U(\mathbb{Z}_q^{n \times m})$, $\mathbf{D} \in U(\mathbb{Z}_q^{n \times m_d})$. The counter is initialized to $\tau = 0$.

2. Choose $\mathbf{S} \hookleftarrow \chi^{n \times t}$ that will serve as a secret key for an LWE-based encryption scheme. Then, sample $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$, $\mathbf{E} \hookleftarrow \chi^{m \times t}$ and compute

$$\mathbf{P} = [\mathbf{p}_1 | \ldots | \mathbf{p}_t] = \mathbf{F}^T \cdot \mathbf{S} + \mathbf{E} \ \in \mathbb{Z}_q^{m \times t}, \tag{9.6}$$

so that $(\mathbf{F}, \mathbf{P}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{m \times t}$ forms a public key for a t-bit variant of Regev's encryption scheme [Reg05].

3. Sample vectors $\mathbf{a}_1, \ldots, \mathbf{a}_N \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{x}_1, \ldots, \mathbf{x}_N \hookleftarrow \chi^t$ to compute

$$(\mathbf{a}_i, \mathbf{b}_i) = \left(\mathbf{a}_i, \ \mathbf{S}^T \cdot \mathbf{a}_i + \mathbf{x}_i + M_i \cdot \lfloor q/2 \rfloor\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t \quad \forall i \in [N]. \tag{9.7}$$

4. For each $i \in [N]$, generate a signature $(\tau_i, \mathbf{v}_i) \leftarrow \mathsf{Sign}(SK_{sig}, \tau, \mathfrak{m}_i)$ on the decomposition $\mathfrak{m}_i = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_i^T | \mathbf{b}_i^T)^T \in \{0,1\}^{m_d}$.
5. S_I sends $R_0 = \left(PK_{sig}, \ (\mathbf{F}, \mathbf{P}), \ \{(\mathbf{a}_i, \mathbf{b}_i), (\tau_i, \mathbf{v}_i)\}_{i=1}^N\right)$ to R_I and interactively proves knowledge of small-norm $\mathbf{S} \in \mathbb{Z}^{n \times t}$, $\mathbf{E} \in \mathbb{Z}^{m \times t}$, short vectors $\{\mathbf{x}_i\}_{i=1}^N$ and t-bit messages $\{M_i\}_{i=1}^N$, for which (9.6) and (9.7) hold. To this end, S_I plays the role of the prover in the ZK argument system described in Section 9.5.2. If the argument of knowledge does not verify or if there exists $i \in [N]$ such that $(\tau_i, \mathbf{v}_i)$ is an invalid signature on the message $\mathfrak{m}_i = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_i^T | \mathbf{b}_i^T)^T$ w.r.t. PK_{sig}, then R_I aborts.
6. Finally S_I defines $S_0 = \big((\mathbf{S}, \mathbf{E}), (\mathbf{F}, \mathbf{P}), PK_{sig}\big)$, which it keeps to itself.

Transfer$\big(\mathsf{S}_\mathsf{T}(S_{i-1}), \mathsf{R}_\mathsf{T}(R_{i-1}, \rho_i)\big)$**:** At the i-th transfer, the receiver R_T has state R_{i-1} and an index $\rho_i \in [1, N]$. It interacts as follows with the sender S_T that has state S_{i-1} in order to obtain M_{ρ_i} from DB.

1. R_T samples vectors $\mathbf{e} \hookleftarrow U(\{-1, 0, 1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and a random $\nu \hookleftarrow U([-B, B]^t)$ to compute

$$(\mathbf{c}_0, \mathbf{c}_1) = \left(\mathbf{a}_{\rho_i} + \mathbf{F} \cdot \mathbf{e}, \ \mathbf{b}_{\rho_i} + \mathbf{P}^T \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu\right) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t, \tag{9.8}$$

which is a re-randomization of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i} + \mu \cdot \lfloor q/2 \rfloor)$. The ciphertext $(\mathbf{c}_0, \mathbf{c}_1)$ is sent to S_T. In addition, R_T provides an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is indeed a transformation of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i})$ for some $\rho_i \in [N]$, and R_T knows a signature on $\mathfrak{m} = \mathsf{vdec}_{n+1,q-1}(\mathbf{a}_{\rho_i}^T | \mathbf{b}_{\rho_i}^T)^T \in \{0,1\}^{m_d}$. To this end, R_T runs the prover in the ZK argument system in Section 9.5.4.

2. If the argument of step 1 verifies, S_T uses $\mathbf{S}$ to decrypt $(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ and obtain $M' = \lfloor (\mathbf{c}_1 - \mathbf{S}^T \cdot \mathbf{c}_0)/(q/2) \rceil \in \{0,1\}^t$, which is sent back to R_T. In addition, S_T provides a zero-knowledge argument of knowledge of vector $\mathbf{y} = \mathbf{c}_1 - \mathbf{S}^T \cdot \mathbf{c}_0 - M' \cdot \lfloor q/2 \rfloor \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ and small-norm matrices $\mathbf{E} \in \mathbb{Z}^{m \times t}$, $\mathbf{S} \in \mathbb{Z}^{n \times t}$ satisfying (modulo q)

$$\mathbf{P} = \mathbf{F}^T \cdot \mathbf{S} + \mathbf{E} \qquad \mathbf{c}_0^T \cdot \mathbf{S} + \mathbf{y}^T = \mathbf{c}_1^T - M'^T \cdot \lfloor q/2 \rfloor. \tag{9.9}$$

To this end, S_T runs the prover in the ZK argument system in Section 9.5.3.

3. If the ZK argument produced by S_T does not properly verify at step 2, R_T halts and outputs $\perp$. Otherwise, R_T recalls the random string $\mu \in \{0,1\}^t$ that was chosen at step 1 and computes $M_{\rho_i} = M' \oplus \mu$. The transfer ends with S_T and R_T outputting $S_i = S_{i-1}$ and $R_i = R_{i-1}$, respectively.

In the initialization phase, the sender has to repeat step 5 with each receiver to prove that $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ are well-formed. Using the Fiat-Shamir heuristic [FS86], we can decrease this initialization cost from $O(N \cdot U)$ to $O(N)$ (regardless of the number of users U) by making the proof non-interactive. This modification also reduces each transfer to 5 communication rounds since, even in the transfer phase, the sender's ZK arguments can be non-interactive and the receiver's arguments only need to be WI, which is preserved when the basic ZK protocol (which has a ternary challenge space) is repeated $\omega(\log n)$ times in parallel. To keep the security proof simple, we derive the matrix $\mathbf{F} \in \mathbb{Z}_q^{n \times m}$ from a second random oracle. Knowing a short basis of $\Lambda_q^\perp(\mathbf{F})$, the simulator can extract the columns of $\mathbf{S}$ from the public key $\mathbf{P} \in \mathbb{Z}_q^{n \times m}$. Details are given in Appendix 9.6.

9.3.2 Security

The security of the above $\mathcal{OT}_{k\times 1}^N$ protocol against static corruptions is stated by the following theorems.

Theorem 9.4. *The $\mathcal{OT}_{k\times 1}^N$ protocol provides receiver security under the* SIS *assumption.*

Proof. We prove that any real-world cheating sender $\hat{\mathsf{S}}$ implies an ideal-world cheating sender $\hat{\mathsf{S}}'$ such that, under the SIS assumption, the two distributions $\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}$ and $\mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}$ with common inputs $(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ are indistinguishable to any PPT distinguisher $\mathcal{D}$.

To this end, we consider a sequence of hybrid experiments with binary outputs. In each experiment Exp_i, a distinguisher $\mathcal{D}$ takes as input the states (S_k, R_k) produced by $\hat{\mathsf{S}}$ and R' at the end of the experiment and outputs a bit. We define W_i as the event that the output of experiment Exp_i is 1. The first experiment outputs whatever the distinguisher $\mathcal{D}$ outputs and corresponds to the real interaction between the cheating sender $\hat{\mathsf{S}}$ and the receiver R.

Exp_0: This experiment involves a real execution of $\hat{\mathsf{S}}$ in interaction with a honest receiver R which queries the index $\rho_i \in [N]$ at the i-th transfer for each $i \in [k]$. The output of Exp_0 is exactly the output of the distinguisher $\mathcal{D}$ on input of $X = (S_k, R_k) \leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$, so that we have

$$\Pr[W_0] = \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}].$$

Exp_1: This experiment is like Exp_0 except that, at step 5 of the initialization phase, the knowledge extractor of the argument system is used to extract the witnesses $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$, $\bar{\mathbf{x}}_j \in \chi^N$, $\bar{M}_j \in \{0,1\}^N$, for each $j \in [t]$, from the sender's argument. In the event that the knowledge extractor fails to extract valid witnesses, the experiment aborts and outputs $\perp$. We know that the zero-knowledge argument system is computationally sound as long as the underlying commitment is computationally binding. If the perfectly hiding commitment of [KTX08] is used, the binding property

is in turn implied by the SIS assumption. Under the SIS assumption, it follows that Exp_1 returns 1 with about the same probability as Exp_0. Specifically, there exists a SIS solver $\mathcal{B}$ such that $|\Pr[W_1] - \Pr[W_0]| \leq \mathbf{Adv}^{\mathsf{SIS}}_{\mathcal{B}}(\lambda)$.

Exp_2: This experiment is identical to Exp_1 except that the receiver R' makes use of the matrix $\mathbf{S} \in \chi^{n \times t}$, which underlies $\mathbf{P} \in \mathbb{Z}_q^{m \times t}$ in (9.6) and was extracted at step 5 of the initialization phase. Namely, at step 2 of each transfer, R' uses $\mathbf{S}$ to determine if the ZK argument sent by $\hat{\mathsf{S}}$ is really an argument for a true statement or if $\hat{\mathsf{S}}$ somehow managed to break the soundness of the argument system. Namely, upon receiving the response $M' \in \{0,1\}^t$ of $\hat{\mathsf{S}}$ at step 2, R' uses the previously extracted $\mathbf{S} \in \chi^{n \times t}$ to determine whether there exists a vector $\mathbf{y} \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ such that

$$\mathbf{c}_0^T \cdot \mathbf{S} + \mathbf{y}^T = \mathbf{c}_1^T - M'^T \cdot \lfloor q/2 \rfloor. \tag{9.10}$$

If no such vector $\mathbf{y}$ exists, R' infers that $\hat{\mathsf{S}}$ broke the soundness of the argument system. In this case, $\hat{\mathsf{S}}$ can be rewound so as to break the binding property of the statistically hiding commitment scheme used by the ZK argument system, which in turn contradicts the SIS assumption. We thus have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathsf{SIS}}_{\mathcal{B}}(\lambda)$ for some efficient algorithm $\mathcal{B}$ which is given rewinding access to $\hat{\mathsf{S}}$.

Exp_3: This experiment is like Exp_2 with the difference that, at each transfer, the receiver R' chooses the index $\rho_i = 1$ and thus always requests the first message of the encrypted database. In more details, at each transfer, R' samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and $\nu \hookleftarrow U([-B,B]^t)$ to compute and send

$$(\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{a}_1 + \mathbf{F} \cdot \mathbf{e},\ \mathbf{b}_1 + \mathbf{P}^T \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t,$$

which is a re-randomization of $(\mathbf{a}_1, \mathbf{b}_1 + \mu \cdot \lfloor q/2 \rfloor)$. Moreover, R'_{T} uses the witness $\rho_i = 1$ to faithfully generate an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i})$. It thus generates a WI argument of knowledge of vectors $\mathfrak{m} = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_1|\mathbf{b}_1) \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $(\mathbf{v}_1^T|\mathbf{v}_2^T)^T \in \mathbb{Z}^{2m}$ satisfying relations (9.22). By the statistically WI of the interactive argument system, this modification has no noticeable impact on the output distribution of a cheating sender $\hat{\mathsf{S}}$. Indeed, since we chose B as a randomization parameter such that $(m+1)\alpha q/B$ is negligible, the result of [DS16, Section 4.1] implies that always re-randomizing $(\mathbf{a}_1, \mathbf{b}_1 + \mu \cdot \lfloor q/2 \rfloor)$ leaves the view of $\hat{\mathsf{S}}$ statistically unchanged. We have $|\Pr[W_2] - \Pr[W_1]| \leq \mathsf{negl}(\lambda)$.

In Exp_3, we can define the ideal-world cheating sender $\hat{\mathsf{S}}'$ which emulates the honest receiver R' interacting with $\hat{\mathsf{S}}$. At the initialization phase, $\hat{\mathsf{S}}'$ appeals to the knowledge extractor of the argument system so as to extract the small-norm matrices $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t] \in \chi^{n \times t}$ and $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m \times t}$ satisfying (9.6). Armed with the decryption key $\mathbf{E} \in \chi^{m \times t}$ of the cryptosystem, $\hat{\mathsf{S}}'$ can decrypt $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ and obtain the messages $M_1, \ldots, M_N \in \{0,1\}^N$ that were encrypted in (9.7) by $\hat{\mathsf{S}}$. It then submits $M_1, \ldots, M_N \in \{0,1\}^N$ to the trusted party T. As in Exp_2, during each transfer phase, $\hat{\mathsf{S}}'$ computes $(\mathbf{c}_0, \mathbf{c}_1)$ as a re-randomization of $(\mathbf{a}_1, \mathbf{b}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ and faithfully generates the receiver's argument of knowledge using the witness $\rho_i = 1$ at step 1. At step 2 of each transfer, $\hat{\mathsf{S}}'$ plays the role of the verifier on behalf of R' in the interactive zero-knowledge argument generated by $\hat{\mathsf{S}}$.

If $\hat{\mathsf{S}}'$ detects that $\hat{\mathsf{S}}$ creates a verifying argument for a false statement (which $\hat{\mathsf{S}}'$ can detect using the extracted matrix $\mathbf{S} \in \mathbb{Z}^{n \times t}$, by applying the test (9.10)), it aborts the interaction as in Exp_3. If the ZK argument involves a true statement, $\hat{\mathsf{S}}'$ sends 1 to the trusted party T so as to authorize the transfer in the ideal world. Otherwise, $\hat{\mathsf{S}}'$ sends 0 to T. At the end of the k-th transfer phase, $\hat{\mathsf{S}}'$ outputs whatever $\hat{\mathsf{S}}$ outputs as its final state S_k.

In Exp_3, it is easy to see that

$$\Pr[W_3] = \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}].$$

When putting the above altogether, we find that there exists a PPT SIS solver $\mathcal{B}$ such that

$$\begin{aligned} |\Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}] \\ - \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}]| \leq 2 \cdot \mathbf{Adv}^{\mathsf{SIS}}_{\mathcal{B}}(\lambda) + \mathsf{negl}(\lambda), \end{aligned}$$

which proves the result. □

Theorem 9.5. *The $\mathcal{OT}^N_{k\times 1}$ protocol provides sender security under the* SIS *and* LWE *assumptions.*

Proof. Given a real malicious receiver $\hat{\mathsf{R}}$, we construct a cheating receiver $\hat{\mathsf{R}}'$ in the ideal world such that, under the SIS and LWE assumption, no PPT distinguisher $\mathcal{D}$ can tell apart the distributions $\mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$ and $\mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}$ under common inputs: N, k, $M_1, \ldots, M_N$, $\rho_1, \ldots, \rho_k$.

To do this, we proceed again via a sequence of hybrid experiments with binary outputs. For each i, we consider the probability that a distinguisher $\mathcal{D}$ outputs 1 on input of the states (S_k, R_k) that constitute the outcome of experiment Exp_i. We also define W_i to be the event that experiment Exp_i outputs 1.

Exp_0: This experiment corresponds to a real execution of $\hat{\mathsf{R}}$ in interaction with a honest sender $\mathsf{S}(M_1, \ldots, M_N)$. The output of the experiment is identical to that of the distinguisher $\mathcal{D}$ on input of $X = (S_k, R_k) \leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$. We have

$$\Pr[W_0] = \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}].$$

Exp_1: This experiment departs from Exp_0 in that, when the dishonest receiver $\hat{\mathsf{R}}_{\mathsf{T}}$ sends the ciphertext $(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ at step 1 of each transfer, the knowledge extractor of the argument system is used to extract the witnesses $\mathfrak{m} \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1, 0, 1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B, B]^t$, $\tau \in \{0,1\}^\ell$ and $\mathbf{v} = (\mathbf{v}_1^T | \mathbf{v}_2^T)^T \in \mathbb{Z}^{2m}$ which satisfy (9.22). If the knowledge extractor fails to produce valid witnesses at some transfer, the experiment aborts and outputs $\perp$. Recall that the zero-knowledge argument system is computationally sound if the underlying commitment is binding, which is equivalent to the SIS assumption if the perfectly hiding commitment of [KTX08] is used. Under the SIS assumption, experiment Exp_1 returns 1 with about the same probability as Exp_0. There thus exists a SIS solver $\mathcal{B}$ such that $|\Pr[W_1] - \Pr[W_0]| \leq k \cdot \mathbf{Adv}^{\mathsf{SIS}}_{\mathcal{B}}(\lambda)$, where k is the number of transfers.

Exp_2: This experiment is identical to Exp_1 except that, at step 1 of each transfer, the experiment aborts if the extracted witnesses $\mathfrak{m} \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $\mathbf{v} = (\mathbf{v}_1^T|\mathbf{v}_2^T)^T \in \mathbb{Z}^{2m}$ are such that the product

$$\left[\frac{\mathbf{a}_\mathfrak{m}}{\mathbf{b}_\mathfrak{m}}\right] = \left[\frac{\mathbf{H}_{n,q-1}}{\mathbf{H}_{t,q-1}}\right] \cdot \mathfrak{m} \in \mathbb{Z}_q^{n+t}$$

does not match any ciphertext $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ appearing in R_0 (namely, we have $(\mathbf{a}_\mathfrak{m}, \mathbf{b}_\mathfrak{m}) \neq (\mathbf{a}_i, \mathbf{b}_i)$ for each $i \in [N]$). We claim that such an event implies a breach in the bounded message security of the signature scheme:

Lemma 9.6. *Under the* SIS *assumption, experiments* Exp_2 *and* Exp_1 *are computationally indistinguishable: there exists a PPT algorithm* $\mathcal{B}$ *such that* $|\Pr[W_2] - \Pr[W_1]| \leq N \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathsf{SIS}}(\lambda)$.

Exp_3: This experiment is like Exp_2 except that, at step 5 of the initialization phase, the zero-knowledge argument of knowledge of $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$, $\bar{\mathbf{x}}_j \in \chi^N$, $\bar{M}_j \in \{0,1\}^N$ such that

$$\left[\begin{array}{c|c|c|c} \mathbf{F}^T & \mathbf{I}_m & & \\ \hline \mathbf{A}_{\mathsf{DB}}^T & & \mathbf{I}_N & \lfloor q/2 \rfloor \cdot \mathbf{I}_N \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{s}_j \\ \hline \mathbf{e}_j \\ \hline \bar{\mathbf{x}}_j \\ \hline \bar{M}_j \end{array}\right] = \left[\begin{array}{c} \mathbf{p}_j \\ \hline \mathbf{b}_j \end{array}\right] \qquad \forall j \in [t]$$

is replaced by a simulated interactive argument and so is the ZK argument of knowledge of $\{(\mathbf{s}_j, \mathbf{e}_j, \mathbf{y}[j])\}_{j=1}^t$ satisfying (9.20) at step 2 of each transfer protocol. From this experiment on, we notice that the small-norm matrices $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t] \in \mathbb{Z}^{n\times t}$, $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m\times t}$ satisfying (9.6) are no longer used by the sender S. Yet, the statistical ZK property of the zero-knowledge argument system ensures that $|\Pr[W_3] - \Pr[W_2]| \leq \mathsf{negl}(\lambda)$.

Exp_4: This experiment is like Exp_3 with the difference that, at step 2 of the initialization phase, each column $\mathbf{p}_i$ of the Regev's encryption public key matrix $\mathbf{P} = [\mathbf{p}_1 \mid \ldots \mid \mathbf{p}_t] = \mathbf{F}^T \cdot \mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{m\times t}$ is traded for a uniformly random vector $\mathbf{p}_i \leftarrow U(\mathbb{Z}_q^m)$. At the same time, each $\mathbf{b}_i = \mathbf{S}^T \cdot \mathbf{a}_i + \mathbf{x}_i + M_i \lfloor \frac{q}{2} \rfloor \in \mathbb{Z}_q^t$ is replaced by a truly uniform random vector in $\mathbb{Z}_q^t$. Therefore, $\mathbf{P}$ is a uniformly distributed matrix in $\mathbb{Z}_q^{m\times t}$, and the $(\mathbf{b}_i)_{i=1}^N$ are distributed as uniform vectors in $(\mathbb{Z}_q^t)^N$. Now, at step 5 of the initialization phase and step 2 of each transfer, the sender's zero-knowledge arguments are simulated arguments for false statements. However, a straightforward reduction shows that, under the LWE assumption over $t \cdot (m+N)$ samples, these changes should remain unnoticed to the malicious receiver $\hat{\mathsf{R}}$ and have no impact on the distinguisher's output: we have $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{LWE}}(\lambda)$.

The ideal-world receiver $\hat{\mathsf{R}}'$ is defined as follows. It assumes the role of the sender S' in interaction with the real-world receiver $\hat{\mathsf{R}}$ in Exp_4. This implies that, in the initialization phase, the matrices $(\mathbf{F}, \mathbf{P})$ are chosen as uniformly random matrices $(\mathbf{F}, \mathbf{P}) \leftarrow U(\mathbb{Z}_q^{n\times m} \times$

$\mathbb{Z}_q^{m\times t}$) and while, at step 3, $(\mathbf{a}_i, \mathbf{b}_i) \leftarrow U(\mathbb{Z}_q^n \times \mathbb{Z}_q^t)$ is chosen at random for each $i \in [N]$. The randomly generated pairs $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ are faithfully signed using $SK_{sig} = \mathbf{T}_\mathbf{A}$ at step 4. In step 5 of the initialization phase, $\hat{\mathsf{R}}'$ appeals to the simulator of the ZK argument. At the i-th transfer, when $\hat{\mathsf{R}}$ sends $(\mathbf{c}_0, \mathbf{c}_1)$ and argues knowledge of $(\mathfrak{m}, \mathbf{e}, \mu, \nu, \tau, \mathbf{v}_1, \mathbf{v}_2)$ at step 1, $\hat{\mathsf{R}}'$ uses the knowledge extractor of the argument system to extract the witnesses $(\mathfrak{m}, \mathbf{e}, \mu, \nu, \tau, \mathbf{v}_1, \mathbf{v}_2) \in \{0,1\}^{m_d} \times \{-1, 0, 1\}^t \times \{0,1\}^t \times [-B, B]^t \times \{0,1\}^\ell$ and determine the index $\rho_i \in [N]$ such that

$$\left[\frac{\mathbf{a}_{\rho_i}}{\mathbf{b}_{\rho_i}}\right] = \left[\frac{\mathbf{H}_{n,q-1}}{\mathbf{H}_{t,q-1}}\right] \cdot \mathfrak{m} \ \in \mathbb{Z}_q^{n+t}.$$

Note that, by Lemma 9.6, such an index must exist unless $\hat{\mathsf{R}}$ can forge a signature. Having determined the index $\rho_i \in [N]$ of the queried database entry, $\hat{\mathsf{R}}'$ sends ρ_i to the trusted party T which returns the message $M_{\rho_i} \in \{0,1\}^t$. The latter is used together with the extracted witness $\mu \in \{0,1\}^t$ to define the response $M' = M_{\rho_i} \oplus \mu \in \{0,1\}^t$ that $\hat{\mathsf{R}}'$ generates on behalf of the sender $\hat{\mathsf{S}}'$ at step 2 of the transfer. In addition, the ideal-world dishonest receiver $\hat{\mathsf{R}}'$ appeals to the simulator of the zero-knowledge argument system to simulate an argument of knowledge of $\{(\mathbf{s}_j, \mathbf{e}_j, \mathbf{y}[j])\}_{j=1}^t$ for the statement (9.20).

It is easy to see that, when $\hat{\mathsf{R}}$ interacts with the simulator $\hat{\mathsf{R}}'$ that emulates the real-world sender S', its view is identical to that of Exp_4: we have

$$\Pr[W_4] = \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}].$$

When combining the above, we conclude that there exist PPT algorithms $\mathcal{B}$ and $\mathcal{B}'$ such that

$$\begin{aligned}&|\Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}] \\ &\quad - \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\mathsf{S}',\hat{\mathsf{R}}'}]| \leq 2 \cdot \mathbf{Adv}^{\mathsf{SIS}}_{\mathcal{B}}(\lambda) + \mathbf{Adv}^{\mathsf{LWE}}_{\mathcal{B}'}(\lambda) + \mathsf{negl}(\lambda).\end{aligned}$$

This proves the sender security under the SIS and LWE assumptions. □

9.4 OT with Access Control for Branching Programs

In this section, we extend our protocol of Section 9.3 into a protocol where database entries can be protected by access control policies consisting of branching programs. In a nutshell, the construction goes as follows.

When the database is set up, the sender signs (a binary representation of) each database entry $(\mathbf{a}_i, \mathbf{b}_i)$ together with a hash value $\mathbf{h}_{\mathsf{BP},i} \in \mathbb{Z}_q^n$ of the corresponding branching program. For each possessed attribute $\mathbf{x} \in \{0,1\}^\kappa$, the user U obtains a credential $\mathsf{Cred}_{\mathsf{U},\mathbf{x}}$ from the issuer.

If U has a credential $\mathsf{Cred}_{\mathsf{U},\mathbf{x}}$ for an attribute $\mathbf{x}$ satisfying the ρ-th branching program, U can re-randomize $(\mathbf{a}_\rho, \mathbf{b}_\rho)$ into $(\mathbf{c}_0, \mathbf{c}_1)$, which is given to the sender, while proving that: (i) He knows a signature $(\tau, \mathbf{v})$ on some message $(\mathbf{a}_\rho, \mathbf{b}_\rho, \mathbf{h}_{\mathsf{BP},\rho})$ such that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of $(\mathbf{a}_\rho, \mathbf{b}_\rho)$; (ii) The corresponding $\mathbf{h}_{\mathsf{BP},\rho}$ is the hash value of (the binary representation of) a branching program BP_ρ that accepts an attribute $\mathbf{x} \in \{0,1\}^\kappa$ for which he has a valid credential $\mathsf{Cred}_{\mathsf{U},\mathbf{x}}$ (i.e., $\mathsf{BP}_\rho(\mathbf{x}) = 1$).

While statement (i) can be proved as in Section 9.3, handling (ii) requires a method of proving the possession of a (committed) branching program BP and a (committed) input $\mathbf{x} \in \{0,1\}^\kappa$ such that $\mathsf{BP}(\mathbf{x}) = 1$ while demonstrating possession of a credential for $\mathbf{x}$.

Recall that a branching program BP of length L, input space $\{0,1\}^\kappa$ and width 5 is specified by L tuples of the form $(\mathsf{var}(\theta), \pi_{\theta,0}, \pi_{\theta,1})$ where

- $\mathsf{var} : [L] \to [0, \kappa - 1]$ is a function that associates the θ-th tuple with the coordinate $x_{\mathsf{var}(\theta)} \in \{0,1\}$ of the input $\mathbf{x} = (x_0, \ldots, x_{\kappa-1})^T$.
- $\pi_{\theta,0}, \pi_{\theta,1} : \{0,1,2,3,4\} \to \{0,1,2,3,4\}$ are permutations that determine the θ-th step of the evaluation.

On input $\mathbf{x} = (x_0, \ldots, x_{\kappa-1})^T$, BP computes its output as follows. For each bit $b \in \{0,1\}$, let $\bar{b}$ denote the bit $1 - b$. Let η_θ denote the state of computation at step θ. The initial state is $\eta_0 = 0$ and, for $\theta \in [1, L]$, the state η_θ is computed as

$$\eta_\theta = \pi_{\theta, x_{\mathrm{var}(\theta)}}(\eta_{\theta-1}) = \pi_{\theta,0}(\eta_{\theta-1}) \cdot \bar{x}_{\mathrm{var}(\theta)} + \pi_{\theta,1}(\eta_{\theta-1}) \cdot x_{\mathrm{var}(\theta)}.$$

Finally, the output of evaluation is $\mathsf{BP}(\mathbf{x}) = 1$ if $\eta_L = 0$, otherwise $\mathsf{BP}(\mathbf{x}) = 0$.

We now let $\delta_\kappa = \lceil \log_2 \kappa \rceil$ and note that each integer in $[0, \kappa - 1]$ can be determined by δ_κ bits. In particular, for each $\theta \in [L]$, let $d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa}$ be the bits representing $\mathrm{var}(\theta)$. Then, we consider the following representation of BP:

$$\begin{aligned}\mathbf{z}_{\mathsf{BP}} = \big(&d_{1,1}, \ldots, d_{1,\delta_\kappa}, \ldots, d_{L,1}, \ldots, d_{L,\delta_\kappa}, \pi_{1,0}(0), \ldots, \pi_{1,0}(4), \pi_{1,1}(0), \ldots, \\ &\pi_{1,1}(4), \ldots, \pi_{L,0}(0), \ldots, \pi_{L,0}(4), \pi_{L,1}(0), \ldots, \pi_{L,1}(4)\big)^T \in [0,4]^\zeta, \end{aligned} \tag{9.11}$$

where $\zeta = L(\delta_\kappa + 10)$.

9.4.1 The OT-AC Protocol

We assume public parameters p consisting of a modulus q, integers n, m such that $m = 2n\lceil \log q \rceil$, a public matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times m}$, the maximal length $L \in \mathsf{poly}(n)$ of branching programs and their desired input length $\kappa \in \mathsf{poly}(n)$.

$\mathsf{ISetup}(\mathsf{p})$: Given public parameters $\mathsf{p} = \{q, n, m, \bar{\mathbf{A}}, L, \kappa\}$, first generate a key pair $(PK_I, SK_I) \leftarrow \mathsf{Keygen}(\mathsf{p}, 1)$ for the signature scheme in Section 7.1 in order to sign single-block messages (i.e., $N_b = 1$) of length $m_I = n \cdot \lceil \log q \rceil + \kappa$. Letting $\ell_I = \mathcal{O}(n)$, this key pair contains $SK_I = \mathbf{T}_{\mathbf{A}_I} \in \mathbb{Z}^{m \times m}$ and

$$PK_I := (\mathbf{A}_I,\ \{\mathbf{A}_{I,j}\}_{j=0}^{\ell_I},\ \mathbf{D}_I,\ \{\mathbf{D}_{I,0}, \mathbf{D}_{I,1}\},\ \mathbf{u}_I).$$

$\mathsf{Issue}(\mathsf{I}(\mathsf{p}, SK_I, PK_I, P_\mathsf{U}, \mathbf{x}) \leftrightarrow \mathsf{U}(\mathsf{p}, \mathbf{x}, st_\mathsf{U}))$: On common input $\mathbf{x} \in \{0,1\}^\kappa$, the issuer I and the user U interact in the following way:

1. If $st_\mathsf{U} = \emptyset$, U creates a pseudonym $P_\mathsf{U} = \bar{\mathbf{A}} \cdot \mathbf{e}_\mathsf{U} \in \mathbb{Z}_q^n$, for a randomly chosen $\mathbf{e}_\mathsf{U} \hookleftarrow U(\{0,1\}^m)$, which is sent to I. It sets $st_\mathsf{U} = (\mathbf{e}_\mathsf{U}, P_\mathsf{U}, 0, \emptyset, \emptyset)$. Otherwise, U parses its state st_U as $(\mathbf{e}_\mathsf{U}, P_\mathsf{U}, f_{DB}, C_\mathsf{U}, \mathsf{Cred}_\mathsf{U})$.

2. The issuer I defines the message $\mathfrak{m}_{\mathsf{U},\mathbf{x}} = (\mathsf{vdec}_{n,q-1}(P_{\mathsf{U}})^T|\mathbf{x}^T)^T \in \{0,1\}^{m_I}$. Then, it runs the signing algorithm of Section 7.1 to obtain and return $\mathsf{cert}_{\mathsf{U},\mathbf{x}} = (\tau_{\mathsf{U}}, \mathbf{v}_{\mathsf{U}}, \mathbf{r}_{\mathsf{U}}) \leftarrow \mathsf{Sign}(SK_I, \mathfrak{m}_{\mathsf{U},\mathbf{x}}) \in \{0,1\}^{\ell_I} \times \mathbb{Z}^{2m} \times \mathbb{Z}^m$, which binds U's pseudonym P_{U} to the attribute string $\mathbf{x} \in \{0,1\}^{\kappa}$.
3. U checks that $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$ satisfies (7.2) and that $\|\mathbf{v}_{\mathsf{U}}\| \leq \sigma\sqrt{2m}, \mathbf{r}_{\mathsf{U}} \leq \sigma\sqrt{m}$. If so, U sets $C_{\mathsf{U}} := C_{\mathsf{U}} \cup \{\mathbf{x}\}$, $\mathsf{Cred}_{\mathsf{U}} := \mathsf{Cred}_{\mathsf{U}} \cup \{\mathsf{cert}_{\mathsf{U},\mathbf{x}}\}$ and updates its state $st_{\mathsf{U}} = (\mathbf{e}_{\mathsf{U}}, P_{\mathsf{U}}, f_{DB}, C_{\mathsf{U}}, \mathsf{Cred}_{\mathsf{U}})$. If $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$ does not properly verify, U aborts the interaction and leaves st_{U} unchanged.

DBSetup$(PK_I, \mathsf{DB} = \{(M_i, \mathsf{BP}_i)\}_{i=1}^N)$**:** The sender has $\mathsf{DB} = \{(M_i, \mathsf{BP}_i)\}_{i=1}^N$ which is a database of N pairs made of a message $M_i \in \{0,1\}^t$ and a policy realized by a length-L branching program $\mathsf{BP}_i = \{\mathsf{var}_i(\theta), \pi_{i,\theta,0}, \pi_{i,\theta,1}\}_{\theta=1}^L$.

1. Choose a random matrix $\mathbf{A}_{\mathrm{HBP}} \hookleftarrow U(\mathbb{Z}_q^{n\times\zeta})$ which will be used to hash the description of branching programs.
2. Generate a key pair for the signature scheme of Section 9.2.1 in order to sign $Q = N$ messages of length $m_d = (2n+t)\cdot\lceil\log q\rceil$ each. This key pair consists of $SK_{sig} = \mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m\times m}$ and $PK_{sig} := (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u})$, where $\ell = \lceil\log N\rceil$ and $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in U(\mathbb{Z}_q^{n\times m})$, $\mathbf{D} \in U(\mathbb{Z}_q^{n\times m_d})$ with $m = 2n\lceil\log q\rceil$, $m_d = (2n+t)\lceil\log q\rceil$. The counter is initialized to $\tau = 0$.
3. Sample $\mathbf{S} \hookleftarrow \chi^{n\times t}$ which will serve as a secret key for an LWE-based encryption scheme. Then, sample $\mathbf{F} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$, $\mathbf{E} \hookleftarrow \chi^{m\times t}$ to compute
$$\mathbf{P} = [\mathbf{p}_1|\ldots|\mathbf{p}_t] = \mathbf{F}^T\cdot\mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{m\times t} \tag{9.12}$$
so that $(\mathbf{F}, \mathbf{P})$ forms a public key for a t-bit variant of Regev's system.
4. Sample vectors $\mathbf{a}_1, \ldots, \mathbf{a}_N \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{x}_1, \ldots, \mathbf{x}_N \hookleftarrow \chi^t$ to compute
$$(\mathbf{a}_i, \mathbf{b}_i) = (\mathbf{a}_i,\ \mathbf{a}_i^T\cdot\mathbf{S} + \mathbf{x}_i + M_i\cdot\lfloor q/2\rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t \qquad \forall i \in [N] \tag{9.13}$$
5. For each $i = 1$ to N, $(\mathbf{a}_i, \mathbf{b}_i)$ is bound to BP_i as follows.
   a. Let $\mathbf{z}_{\mathsf{BP},i} \in [0,4]^{\zeta}$ be the binary representation of the branching program. Compute its digest $\mathbf{h}_{\mathsf{BP},i} = \mathbf{A}_{\mathrm{HBP}}\cdot\mathbf{z}_{\mathsf{BP},i} \in \mathbb{Z}_q^n$.
   b. Using SK_{sig}, generate a signature $(\tau_i, \mathbf{v}_i) \leftarrow \mathsf{Sign}(SK_{sig}, \tau, \mathfrak{m}_i)$ on the message $\mathfrak{m}_i = \mathsf{vdec}_{2n+t,q-1}(\mathbf{a}_i|\mathbf{b}_i|\mathbf{h}_{\mathsf{BP},i}) \in \{0,1\}^{m_d}$ obtained by decomposing $(\mathbf{a}_i^T|\mathbf{b}_i^T|\mathbf{h}_{\mathsf{BP},i}^T)^T \in \mathbb{Z}_q^{2n+t}$.
6. The database's public key is defined as $PK_{\mathrm{DB}} = (PK_{sig},\ (\mathbf{F}, \mathbf{P}),\ \mathbf{A}_{\mathrm{HBP}})$ while the encrypted database is $\{ER_i = (\mathbf{a}_i, \mathbf{b}_i, (\tau_i, \mathbf{v}_i)), \mathsf{BP}_i\}_{i=1}^N$. The sender DB outputs $(PK_{\mathrm{DB}}, \{ER_i, \mathsf{BP}_i\}_{i=1}^N)$ and keeps $SK_{\mathsf{DB}} = (SK_{sig}, \mathbf{S})$.

Transfer$(\mathsf{DB}(SK_{\mathsf{DB}}, PK_{\mathsf{DB}}, PK_I), \mathsf{U}(\rho, st_{\mathsf{U}}, PK_I, PK_{\mathsf{DB}}, ER_\rho, \mathsf{BP}_\rho))$**:** From an index $\rho \in [N]$, a record $ER_\rho = (\mathbf{a}_\rho, \mathbf{b}_\rho, (\tau_\rho, \mathbf{v}_\rho))$ and a policy BP_ρ, the user U parses st_{U} as $(\mathbf{e}_{\mathsf{U}}, P_{\mathsf{U}}, f_{DB}, C_{\mathsf{U}}, \mathsf{Cred}_{\mathsf{U}})$. If C_{U} does not contain any $\mathbf{x} \in \{0,1\}^{\kappa}$ s.t. $\mathsf{BP}_\rho(\mathbf{x}) = 1$ and $\mathsf{Cred}_{\mathsf{U}}$ contains the corresponding $\mathsf{cert}_{\mathsf{U},\mathbf{x}}$, U outputs $\perp$. Otherwise, he selects such a pair $(\mathbf{x}, \mathsf{cert}_{\mathsf{U},\mathbf{x}})$ and interacts with DB:

1. If $f_{DB} = 0$, U interacts with DB for the first time and requires DB to prove knowledge of small-norm $\mathbf{S} \in \mathbb{Z}^{n \times t}$, $\mathbf{E} \in \mathbb{Z}^{m \times t}$, $\{\mathbf{x}_i\}_{i=1}^N$ and t-bit messages $\{M_i\}_{i=1}^N$ satisfying (9.12)-(9.13). To do this, DB uses the ZK argument in Section 9.5.2. If there exists $i \in [N]$ such that $(\tau_i, \mathbf{v}_i)$ is an invalid signature on $\mathsf{vdec}_{2n+t,q-1}(\mathbf{a}_i^T|\mathbf{b}_i^T|\mathbf{h}_{\mathsf{BP},i}^T)^T$ or if the ZK argument does not verify, U aborts. Otherwise, U updates st_{U} and sets $f_{DB} = 1$.

2. U re-randomizes the pair $(\mathbf{a}_\rho, \mathbf{b}_\rho)$ contained in ER_ρ. It samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and $\nu \hookleftarrow U([-B,B]^t)$ to compute

$$(\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{a}_\rho + \mathbf{F} \cdot \mathbf{e},\ \mathbf{b}_\rho + \mathbf{P}^T \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t, \tag{9.14}$$

which is sent to DB as a re-randomization of $(\mathbf{a}_\rho, \mathbf{b}_\rho + \mu \cdot \lfloor q/2 \rfloor)$. Then, U provides an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of some $(\mathbf{a}_\rho, \mathbf{b}_\rho)$ associated with a policy BP_ρ for which U has a credential $\mathsf{cert}_{\mathsf{U},x}$ for some $\mathbf{x} \in \{0,1\}^\kappa$ such that $\mathsf{BP}_\rho(\mathbf{x}) = 1$. In addition, U demonstrates possession of: (i) a preimage $\mathbf{z}_{\mathsf{BP},\rho} \in [0,4]^\zeta$ of $\mathbf{h}_{\mathsf{BP},\rho} = \mathbf{A}_{\mathrm{HBP}} \cdot \mathbf{z}_{\mathsf{BP},\rho} \in \mathbb{Z}_q^n$; (ii) a credential $\mathsf{Cred}_{\mathsf{U},\mathbf{x}}$ for the corresponding $\mathbf{x} \in \{0,1\}^\kappa$ and the private key $\mathbf{e}_{\mathsf{U}} \in \{0,1\}^m$ for the pseudonym P_{U} to which $\mathbf{x}$ is bound; (iii) the coins leading to the randomization of some $(\mathbf{a}_\rho, \mathbf{b}_\rho)$. Then entire step is conducted by arguing knowledge of

$$\left\{\begin{array}{l}
\mathbf{e}_{\mathsf{U}} \in \{0,1\}^m, \mathfrak{m}_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{m_I},\ \mathbf{x} \in \{0,1\}^\kappa,\ \widehat{\mathfrak{m}}_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{m/2} \\
\tau_{\mathsf{U}} \in \{0,1\}^{\ell_I},\ \mathbf{v}_{\mathsf{U}} = (\mathbf{v}_{\mathsf{U},1}^T|\mathbf{v}_{\mathsf{U},2}^T)^T \in [-\beta,\beta]^{2m},\ \mathbf{r}_{\mathsf{U}} \in [-\beta,\beta]^m \\
\qquad\qquad \text{// signature on } \mathfrak{m}_{\mathsf{U},\mathbf{x}} = (\mathsf{vdec}_{n,q-1}(P_{\mathsf{U}})^T|\mathbf{x}^T)^T \\
\mathbf{z}_{\mathsf{BP},\rho} \in [0,4]^\zeta \qquad \text{// representation of } \mathsf{BP}_\rho \\
\mathfrak{m} \in \{0,1\}^{m_d},\ \tau \in \{0,1\}^\ell,\ \mathbf{v} = (\mathbf{v}_1^T|\mathbf{v}_2^T)^T \in \mathbb{Z}^{2m} \\
\qquad\qquad \text{// signature on } \mathfrak{m} = \mathsf{vdec}_{2n+t,q-1}(\mathbf{a}_i^T|\mathbf{b}_i^T|\mathbf{h}_{\mathsf{BP},\rho}^T)^T \\
\mathbf{e} \in \{-1,0,1\}^t,\ \mu \in \{0,1\}^t,\ \nu \in [-B,B]^t, \\
\qquad\qquad \text{// coins allowing the re-randomization of } (\mathbf{a}_\rho, \mathbf{b}_\rho)
\end{array}\right.$$

satisfying the relations (modulo q)

$$\left\{\begin{array}{l}
\mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m} + \left[\begin{array}{c|c|c|c} \mathbf{F} & & & \\ \hline \mathbf{P}^T & \mathbf{I}_t \cdot \lfloor q/2 \rfloor & \mathbf{I}_t & \\ \hline & & & -\mathbf{A}_{\mathrm{HBP}} \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{e} \\ \hline \mu \\ \hline \nu \\ \hline \mathbf{z}_{\mathsf{BP},\rho} \end{array}\right] = \left[\begin{array}{c} \mathbf{c}_0 \\ \hline \mathbf{c}_1 \\ \hline \mathbf{0}^n \end{array}\right] \\
\qquad \text{// (recall that } (\mathbf{a}_\rho^T|\mathbf{b}_\rho^T|\mathbf{h}_{\mathsf{BP},\rho}^T)^T = \mathbf{H}_{2n+t,q-1} \cdot \mathfrak{m}) \\
\mathbf{A} \cdot \mathbf{v}_1 + \mathbf{A}_0 \cdot \mathbf{v}_2 + \sum_{j=1}^{\ell} \mathbf{A}_j \cdot (\tau[j] \cdot \mathbf{v}_2) - \mathbf{D} \cdot \mathfrak{m} = \mathbf{u} \\
\mathbf{A}_I \cdot \mathbf{v}_{\mathsf{U},1} + \mathbf{A}_{I,0} \cdot \mathbf{v}_{\mathsf{U},2} + \sum_{j=1}^{\ell_I} \mathbf{A}_{I,j} \cdot (\tau_{\mathsf{U}}[j] \cdot \mathbf{v}_{\mathsf{U},2}) - \mathbf{D}_I \cdot \widehat{\mathfrak{m}}_{\mathsf{U},\mathbf{x}} = \mathbf{u}_I \\
\mathbf{D}_{I,0} \cdot \mathbf{r}_{\mathsf{U}} + \mathbf{D}_{I,1} \cdot \mathfrak{m}_{\mathsf{U},\mathbf{x}} - \mathbf{H}_{n,q-1} \cdot \widehat{\mathfrak{m}}_{\mathsf{U},\mathbf{x}} = \mathbf{0} \\
\left[\begin{array}{c|c} \mathbf{H}_{n,q-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}_\kappa \end{array}\right] \cdot \mathfrak{m}_{\mathsf{U},\mathbf{x}} + \left[\begin{array}{c} -\bar{\mathbf{A}} \\ \hline \mathbf{0} \end{array}\right] \cdot \mathbf{e}_{\mathsf{U}} + \left[\begin{array}{c} \mathbf{0} \\ \hline -\mathbf{I}_\kappa \end{array}\right] \cdot \mathbf{x} = \mathbf{0}
\end{array}\right. \tag{9.15}$$

and such that $\mathbf{z}_{\mathsf{BP},\rho} \in [0,4]^\zeta$ encodes BP_ρ such that $\mathsf{BP}_\rho(\mathbf{x}) = 1$. This is done by running the argument system described in Section 9.5.5.

3. If the ZK argument of step 2 verifies, DB decrypts $(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ to obtain $M' = \lfloor (\mathbf{c}_1 - \mathbf{S}^T \cdot \mathbf{c}_0)/(q/2) \rceil \in \{0,1\}^t$, which is returned to U. Then, DB

argues knowledge of $\mathbf{y} = \mathbf{c}_1 - \mathbf{S}^T \cdot \mathbf{c}_0 - M' \cdot \lfloor q/2 \rfloor \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ and small-norm $\mathbf{E} \in \mathbb{Z}^{m \times t}$, $\mathbf{S} \in \mathbb{Z}^{n \times t}$ satisfying (modulo q)

$$\mathbf{P} \quad = \quad \mathbf{F}^T \cdot \mathbf{S} + \mathbf{E}\,, \qquad \mathbf{c}_0^T \cdot \mathbf{S} + \mathbf{y}^T = \mathbf{c}_1^T - {M'}^T \cdot \lfloor q/2 \rfloor.$$

To this end, DB uses the ZK argument system of Section 9.5.3.

4. If the ZK argument produced by DB does not verify, U outputs $\perp$. Otherwise, U recalls the string $\mu \in \{0,1\}^t$ and outputs $M_{\rho_i} = M' \oplus \mu$.

Like our construction of Section 9.3, the above protocol requires the DB to repeat a ZK proof of communication complexity $\Omega(N)$ with each user U during the initialization phase. By applying the Fiat-Shamir heuristic as in Appendix 9.6, the cost of the initialization phase can be made independent of the number of users: the sender can publicize $(PK_{\text{DB}}, \{ER_i, \mathsf{BP}_i\}_{i=1}^N)$ along with a with a universally verifiable non-interactive proof of well-formedness.

The security of the above protocol against static corruptions is proved in [LLM+17], under the SIS and LWE assumptions and is similar to the previous proofs.

9.5 Zero-Knowledge Subprotocols for Stern Protocol

9.5.1 Our Strategy and Basic Techniques, In a Nutshell

Before going into the details of our protocols, we first summarize our governing strategy and the techniques that will be used in the next subsections.

In each protocol, we prove knowledge of (possibly one-dimensional) integer vectors $\{\mathbf{w}_i\}_i$ that have various constraints (e.g., smallness, special arrangements of coordinates, or correlation with one another) and satisfy a system

$$\Big\{ \sum_i \mathbf{M}_{i,j} \cdot \mathbf{w}_i = \mathbf{v}_j \Big\}_j, \tag{9.16}$$

where $\{\mathbf{M}_{i,j}\}_{i,j}$, $\{\mathbf{v}_j\}_j$ are public matrices (which are possibly zero or identity matrices) and vectors. Our strategy consists in transforming this entire system into one equivalent equation $\mathbf{M} \cdot \mathbf{w} = \mathbf{v}$, where matrix $\mathbf{M}$ and vector $\mathbf{v}$ are public, while the constraints of the secret vector $\mathbf{w}$ capture those of witnesses $\{\mathbf{w}_i\}_i$ and they are provable in zero-knowledge via random permutations. For this purpose, the Stern-like protocol from Section 4.3 comes in handy.

A typical transformation step is of the form $\mathbf{w}_i \to \bar{\mathbf{w}}_i$, where there exists public matrix $\mathbf{P}_{i,j}$ such that $\mathbf{P}_{i,j} \cdot \bar{\mathbf{w}}_i = \mathbf{w}_i$. This subsumes the decomposition and extension mechanisms which first appeared in [LNSW13].

- **Decomposition:** Used when $\mathbf{w}_i$ has infinity norm bound larger than 1 and we want to work more conveniently with $\bar{\mathbf{w}}_i$ whose norm bound is exactly 1. In this case, $\mathbf{P}_{i,j}$ is a decomposition matrix.
- **Extension:** Used when we insert "dummy" coordinates to $\mathbf{w}_i$ to obtain $\bar{\mathbf{w}}_i$ whose coordinates are somewhat balanced. In this case, $\mathbf{P}_{i,j}$ is a $\{0,1\}$-matrix with zero-columns corresponding to positions of insertions.

Such a step transforms the term $\mathbf{M}_{i,j} \cdot \mathbf{w}_i$ into $\overline{\mathbf{M}}_{i,j} \cdot \bar{\mathbf{w}}_i$, where $\overline{\mathbf{M}}_{i,j} = \mathbf{M}_{i,j} \cdot \mathbf{P}_{i,j}$ is a public matrix. Also, using the commutativity property of addition, we often group together secret vectors having the same constraints.

After a number of transformations, we will reach a system equivalent to (9.16):

$$\begin{cases} \mathbf{M}'_{1,1} \cdot \mathbf{w}'_1 + \mathbf{M}'_{1,2} \cdot \mathbf{w}'_2 + \cdots + \mathbf{M}'_{1,k} \cdot \mathbf{w}'_k = \mathbf{v}_1, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \mathbf{M}'_{t,1} \cdot \mathbf{w}'_1 + \mathbf{M}'_{t,2} \cdot \mathbf{w}'_2 + \cdots + \mathbf{M}'_{t,k} \cdot \mathbf{w}'_k = \mathbf{v}_t, \end{cases} \tag{9.17}$$

where integers t, k and matrices $\mathbf{M}'_{i,j}$ are public. Defining

$$\mathbf{M} = \left(\begin{array}{c|c|c|c} \mathbf{M}'_{1,1} & \mathbf{M}'_{1,2} & \cdots & \mathbf{M}'_{1,k} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \mathbf{M}'_{t,1} & \mathbf{M}'_{t,2} & \cdots & \mathbf{M}'_{t,k} \end{array}\right); \quad \mathbf{w} = \begin{pmatrix} \mathbf{w}'_1 \\ \vdots \\ \mathbf{w}'_k \end{pmatrix}; \quad \mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_t \end{pmatrix},$$

we obtain the unified equation $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$. At this stage, we will use a properly defined composition of random permutations to prove the constraints of $\mathbf{w}$. We remark that the crucial aspect of the above process is in fact the manipulation of witness vectors, while the transformations of public matrices/vectors just follow accordingly. To ease the presentation of the next subsections, we will thus only focus on the secret vectors.

In the process, we will employ various extending and permuting techniques which require introducing some notations. The most frequently used ones are given in Table 9.1. Some of these techniques appeared (in slightly different forms) in previous works [LNSW13, LNW15, LLNW16, LLM+16a, LLM+16b]. The last three parts of the table summarizes newly-introduced techniques that will enable the treatment of secret-and-correlated objects involved in the evaluation of hidden branching programs.

In particular, the intriguing technique of the last row will be used for proving knowledge of secret integer z of the form $z = x \cdot y$ for some $(x, y) \in [0, 4] \times \{0, 1\}$ satisfying other relations. The following example illustrates how it works.

Example. Let $(x, y) = (2, 1)$ and $(c, b) = (4, 1)$. Then we have:

$$\begin{aligned} \mathsf{ext}_{5\times 2}(2,1) &= (0,\ 1,\ 0,\ 0,\ 0,\ 4,\ 0,\ 3,\ 0,\ 2)^T \\ T_{5\times 2}[4,1]\big(\mathsf{ext}_{5\times 2}(2,1)\big) &= (0,\ 0,\ 4,\ 0,\ 3,\ 0,\ 2,\ 0,\ 1,\ 0)^T \end{aligned}$$

Note that: $T_{5\times 2}[4,1]\big(\mathsf{ext}_{5\times 2}(2,1)\big) = \mathsf{ext}_{5\times 2}(1,0) = \mathsf{ext}_{5\times 2}(2 + 4 \bmod 5, 1 \oplus 1)$.

9.5.2 Protocol 1

Let n, m, q, N, t, B_χ be the parameters defined in Section 9.3. The protocol allows the prover to prove knowledge of LWE secrets and the well-formedness of ciphertexts. It is summarized as follows.

Common input: $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{P} \in \mathbb{Z}_q^{m\times t}$; $\{\mathbf{a}_i \in \mathbb{Z}_q^n,\ \mathbf{b}_i \in \mathbb{Z}_q^t\}_{i=1}^N$.

Notation	Meaning/Property/Usage/Technique
$\mathsf{B}^2_{\mathfrak{m}}$	• The set of vectors in $\{0,1\}^{2\mathfrak{m}}$ with Hamming weight $\mathfrak{m}$. • $\forall \phi \in \mathcal{S}_{2\mathfrak{m}}, \mathbf{x}' \in \mathbb{Z}^{2\mathfrak{m}} : \mathbf{x}' \in \mathsf{B}^2_{\mathfrak{m}} \Leftrightarrow \phi(\mathbf{x}') \in \mathsf{B}^2_{\mathfrak{m}}$. • To prove $\mathbf{x} \in \{0,1\}^{\mathfrak{m}}$: Extend $\mathbf{x}$ to $\mathbf{x}' \in \mathsf{B}^2_{\mathfrak{m}}$, then permute $\mathbf{x}'$.
$\mathsf{B}^3_{\mathfrak{m}}$	• The set of vectors in $\{-1,0,1\}^{3\mathfrak{m}}$ that have exactly $\mathfrak{m}$ coordinates equal to j, for every $j \in \{-1,0,1\}$. • $\forall \phi \in \mathcal{S}_{3\mathfrak{m}}, \mathbf{x}' \in \mathbb{Z}^{3\mathfrak{m}} : \mathbf{x}' \in \mathsf{B}^3_{\mathfrak{m}} \Leftrightarrow \phi(\mathbf{x}') \in \mathsf{B}^3_{\mathfrak{m}}$. • To prove $\mathbf{x} \in \{-1,0,1\}^{\mathfrak{m}}$: Extend $\mathbf{x}$ to $\mathbf{x}' \in \mathsf{B}^3_{\mathfrak{m}}$, then permute $\mathbf{x}'$.
$\mathsf{ext}_2(\cdot)$ and $T_2[\cdot](\cdot)$	• For $c \in \{0,1\} : \mathsf{ext}_2(c) = (\bar{c}, c)^T \in \{0,1\}^2$. • For $b \in \{0,1\}$ and $\mathbf{x} = (x_0, x_1)^T \in \mathbb{Z}^2$: $T_2[b](\mathbf{x}) = (x_b, x_{\bar{b}})^T$. • Property: $\mathbf{x} = \mathsf{ext}_2(c) \Leftrightarrow T_2[b](\mathbf{x}) = \mathsf{ext}_2(c \oplus b)$. • To prove $c \in \{0,1\}$ simultaneously satisfies many relations: Extend it to $\mathbf{x} = \mathsf{ext}_2(c)$, then permute and use the *same* b at all appearances.
$\mathsf{expand}(\cdot,\cdot)$ and $T_{\mathsf{exp}}[\cdot,\cdot](\cdot)$	• For $c \in \{0,1\}$ and $\mathbf{x} \in \mathbb{Z}^{\mathfrak{m}}$: $\mathsf{expand}(c, \mathbf{x}) = (\bar{c} \cdot \mathbf{x}^T \mid c \cdot \mathbf{x}^T)^T \in \mathbb{Z}^{2\mathfrak{m}}$. • For $b \in \{0,1\}, \phi \in \mathcal{S}_m$, $\mathbf{v} = \begin{pmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \end{pmatrix} \in \mathbb{Z}^{2\mathfrak{m}}$: $T_{\mathsf{exp}}[b,\phi](\mathbf{v}) = \begin{pmatrix} \phi(\mathbf{v}_b) \\ \phi(\mathbf{v}_{\bar{b}}) \end{pmatrix}$. • Property: $\mathbf{v} = \mathsf{expand}(c, \mathbf{x}) \Leftrightarrow T_{\mathsf{exp}}[b,\phi](\mathbf{v}) = \mathsf{expand}(c \oplus b, \phi(\mathbf{x}))$.
$[\cdot]_5$	For $k \in \mathbb{Z}$: $[k]_5$ denotes the integer $t \in \{0,1,2,3,4\}$, s.t. $t = k \bmod 5$.
$\mathsf{ext}_5(\cdot)$ and $T_5[\cdot](\cdot)$	• For $x \in [0,4] : \mathsf{ext}_5(x) = ([x+4]_5, [x+3]_5, [x+2]_5, [x+1]_5, x)^T \in [0,4]^5$. • For $c \in [0,4]$ and $\mathbf{v} = (v_0, v_1, v_2, v_3, v_4)^T \in \mathbb{Z}^5$: $T_5[c](\mathbf{v}) = \left(v_{[-c]_5}, v_{[-c+1]_5}, v_{[-c+2]_5}, v_{[-c+3]_5}, v_{[-c+4]_5}\right)^T$. • Property: $\mathbf{v} = \mathsf{ext}_5(x) \Leftrightarrow T_5[c](\mathbf{v}) = \mathsf{ext}_5(x + c \bmod 5)$. • To prove $x \in [0,4]$ simultaneously satisfies many relations: Extend it to $\mathbf{v} = \mathsf{ext}_5(x)$, then permute and use the *same* c at all appearances.
unit_x	• $\forall x \in [0,4]$: unit_x is the 5-dim unit vector $(v_0, \ldots, v_4)^T$ with $v_x = 1$. • For $c \in [0,4], \mathbf{v} \in \mathbb{Z}^5$: $\mathbf{v} = \mathsf{unit}_x \Leftrightarrow T_5[c](\mathbf{v}) = \mathsf{unit}_{x+c \bmod 5}$. $\rightarrow$ Allow proving $\mathbf{v} = \mathsf{unit}_x$ for some $x \in [0,4]$ satisfying other relations.
$\mathsf{ext}_{5\times2}(\cdot,\cdot)$ and $T_{5\times2}[\cdot,\cdot](\cdot)$	• For $x \in [0,4]$ and $y \in \{0,1\}$: $\mathsf{ext}_{5\times2}(x,y) = ([x+4]_5 \cdot \bar{y}, [x+4]_5 \cdot y, [x+3]_5 \cdot \bar{y}, [x+3]_5 \cdot y, [x+2]_5 \cdot \bar{y}, [x+2]_5 \cdot y, [x+1]_5 \cdot \bar{y}, [x+1]_5 \cdot y, x \cdot \bar{y}, x \cdot y)^T \in [0,4]^{10}$ • For $(c,b) \in [0,4] \times \{0,1\}$ and $\mathbf{v} = (v_{0,0}, v_{0,1}, \ldots, v_{4,0}, v_{4,1})^T \in \mathbb{Z}^{10}$: $T_{5\times2}[c,b](\mathbf{v}) = \left(v_{[-c]_5,b}, v_{[-c]_5,\bar{b}}, v_{[-c+1]_5,b}, v_{[-c+1]_5,\bar{b}}, v_{[-c+2]_5,b}, v_{[-c+2]_5,\bar{b}}, v_{[-c+3]_5,b}, v_{[-c+3]_5,\bar{b}}, v_{[-c+4]_5,b}, v_{[-c+4]_5,\bar{b}}\right)^T$. • Property: $\mathbf{v} = \mathsf{ext}_{5\times2}(x,y) \Leftrightarrow T_{5\times2}[c,b](\mathbf{v}) = \mathsf{ext}_{5\times2}(x + c \bmod 5, y \oplus b)$. $\rightarrow$ Allow proving $z = x \cdot y$ for some $(x,y) \in [0,4] \times \{0,1\}$ satisfying other relations: Extend z to $\mathbf{v} = \mathsf{ext}_{5\times2}(x,y)$, then permute and use the *same* c, b at all appearances of x, y, respectively.

Table 9.1 – Basic notations and extending/permuting techniques used in our protocols.

Prover's goal is to prove knowledge of $\mathbf{S} \in [-B_\chi, B_\chi]^{n\times t}$, $\mathbf{E} \in [-B_\chi, B_\chi]^{m\times t}$, $\{\mathbf{x}_i \in [-B_\chi, B_\chi]^t, M_i \in \{0,1\}^t\}_{i=1}^N$ such that the following equations hold:

$$\begin{cases} \mathbf{F}^T \cdot \mathbf{S} + \mathbf{E} = \mathbf{P} \bmod q \\ \forall i \in [N] : \mathbf{S}^T \cdot \mathbf{a}_i + \mathbf{x}_i + \lfloor q/2 \rfloor \cdot M_i = \mathbf{b}_i \bmod q. \end{cases} \tag{9.18}$$

For each $j \in [t]$, let $\mathbf{p}_j, \mathbf{s}_j, \mathbf{e}_j$ be the j-th column of matrices $\mathbf{P}, \mathbf{S}, \mathbf{E}$, respectively. For each $(i,j) \in [N] \times [t]$, let $\mathbf{b}_i[j], \mathbf{x}_i[j], M_i[j]$ denote the j-th coordinate of vectors $\mathbf{b}_i, \mathbf{x}_i, M_i$, respectively. Then, observe that (9.18) can be rewritten as:

$$\begin{cases} \forall j \in [t] : \mathbf{F}^T \cdot \mathbf{s}_j + \mathbf{I}_m \cdot \mathbf{e}_j = \mathbf{p}_j \bmod q \\ \forall (i,j) \in [N] \times [t] : \mathbf{a}_i^T \cdot \mathbf{s}_j + 1 \cdot \mathbf{x}_i[j] + \lfloor q/2 \rfloor \cdot M_i[j] = \mathbf{b}_i[j] \bmod q. \end{cases} \tag{9.19}$$

Then, we form the following vectors:

$$\begin{aligned} \mathbf{w}_1 &= (\mathbf{s}_1^T \mid \ldots \mid \mathbf{s}_t^T \mid \mathbf{e}_1^T \mid \ldots \mid \mathbf{e}_t^T \mid (\mathbf{x}_1[1], \ldots, \mathbf{x}_N[t]))^T \in [-B_\chi, B_\chi]^{(n+m+N)t}; \\ \mathbf{w}_2 &= (M_1[1], \ldots, M_N[t])^T \in \{0,1\}^{Nt}. \end{aligned}$$

Next, we run $\mathsf{vdec}'_{(n+m+N)t, B_\chi}$ to decompose $\mathbf{w}_1$ into $\bar{\mathbf{w}}_1$ and then extend $\bar{\mathbf{w}}_1$ to $\mathbf{w}_1^* \in \mathsf{B}^3_{(n+m+N)t\delta_{B_\chi}}$. We also extend $\mathbf{w}_2$ into $\mathbf{w}_2^* \in \mathsf{B}^2_{Nt}$ and we then form $\mathbf{w} = ((\mathbf{w}_1^*)^T \mid (\mathbf{w}_2^*)^T)^T \in \{-1,0,1\}^D$, where $D = 3(n+m+N)t\delta_{B_\chi} + 2Nt$.

Observe that relations (9.19) can be transformed into *one* equivalent equation of the form $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$, where $\mathbf{M}$ and $\mathbf{v}$ are built from the common input.

Having performed the above unification, we now define VALID as the set of all vectors $\mathbf{t} = (\mathbf{t}_1^T \mid \mathbf{t}_2^T)^T \in \{-1,0,1\}^D$, where $\mathbf{t}_1 \in \mathsf{B}^3_{(n+m+N)t\delta_{B_\chi}}$ and $\mathbf{t}_2 \in \mathsf{B}^2_{Nt}$. Clearly, our vector $\mathbf{w}$ belongs to the set VALID.

Next, we specify the set $\mathcal{S}$ and permutations of D elements $\{\Gamma_\phi : \phi \in \mathcal{S}\}$, for which the conditions in (4.3) hold.

- $\mathcal{S} := \mathcal{S}_{3(n+m+N)t\delta_{B_\chi}} \times \mathcal{S}_{2Nt}$.
- For $\phi = (\phi_1, \phi_2) \in \mathcal{S}$ and for $\mathbf{t} = (\mathbf{t}_1^T \mid \mathbf{t}_2^T)^T \in \mathbb{Z}^D$, where $\mathbf{t}_1 \in \mathbb{Z}^{3(n+m+N)t\delta_{B_\chi}}$ and $\mathbf{t}_2 \in \mathbb{Z}^{2Nt}$, we define $\Gamma_\phi(\mathbf{t}) = (\phi_1(\mathbf{t}_1)^T \mid \phi_2(\mathbf{t}_2)^T)^T$.

By inspection, it can be seen that the desired properties in (4.3) are satisfied. As a result, we can obtain the required ZKAoK by running the protocol from Section 4.3.2 with common input $(\mathbf{M}, \mathbf{v})$ and prover's input $\mathbf{w}$. The protocol has communication cost $\mathcal{O}(D \log q) = \widetilde{\mathcal{O}}(\lambda) \cdot \mathcal{O}(Nt)$ bits.

While this protocol has linear complexity in N, it is only used in the initialization phase, where $\Omega(N)$ bits inevitably have to be transmitted anyway.

9.5.3 Protocol 2

Let n, m, q, N, t, B be system parameters. The protocol allows the prover to prove knowledge of LWE secrets and the correctness of decryption.

Common input: $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{P} \in \mathbb{Z}_q^{m\times t}$; $\mathbf{c}_0 \in \mathbb{Z}_q^n$, $\mathbf{c}_1 \in \mathbb{Z}_q^t$, $M' \in \{0,1\}^t$.

Prover's goal is to prove knowledge of $\mathbf{S} \in [-B_\chi, B_\chi]^{n\times t}$, $\mathbf{E} \in [-B_\chi, B_\chi]^{m\times t}$ and $\mathbf{y} \in [-q/5, q/5]^t$ such that the following equations hold:

$$\mathbf{F}^T \cdot \mathbf{S} + \mathbf{E} = \mathbf{P} \bmod q; \quad \mathbf{c}_0^T \cdot \mathbf{S} + \mathbf{y}^T = \mathbf{c}_1^T - M'^T \cdot \lfloor q/2 \rfloor \bmod q. \tag{9.20}$$

For each $j \in [t]$, let $\mathbf{p}_j, \mathbf{s}_j, \mathbf{e}_j$ be the j-th column of matrices $\mathbf{P}, \mathbf{S}, \mathbf{E}$, respectively; and let $\mathbf{y}[j], \mathbf{c}_1[j], M'[j]$ be the j-th entry of vectors $\mathbf{y}, \mathbf{c}_1, M'$, respectively. Then, observe that (9.20) can be re-written as:

$$\forall j \in [t] : \begin{cases} \mathbf{F}^T \cdot \mathbf{s}_j + \mathbf{I}_m \cdot \mathbf{e}_j = \mathbf{p}_j \bmod q \\ \mathbf{c}_0^T \cdot \mathbf{s}_j + 1 \cdot \mathbf{y}[j] = \mathbf{c}_1[j] - M'[j] \cdot \lfloor q/2 \rfloor \bmod q. \end{cases} \tag{9.21}$$

Next, we form vector $\mathbf{w}_1 = (\mathbf{s}_1^T \mid \ldots \mid \mathbf{s}_t^T \mid \mathbf{e}_1^T \mid \ldots \mid \mathbf{e}_t^T)^T \in [-B_\chi, B_\chi]^{(n+m)t}$, then decompose it to $\bar{\mathbf{w}}_1 \in \{-1,0,1\}^{(n+m)t\delta_{B_\chi}}$, and extend $\bar{\mathbf{w}}_1$ to $\mathbf{w}_1^* \in \mathsf{B}^3_{(n+m)t\delta_{B_\chi}}$.

At the same time, we decompose vector $\mathbf{y} = (\mathbf{y}[1], \ldots, \mathbf{y}[t])^T \in [-q/5, q/5]^t$ to $\bar{\mathbf{y}} \in \{-1,0,1\}^{t\delta_{q/5}}$, and then extend $\bar{\mathbf{y}}$ to $\mathbf{y}^* \in \mathsf{B}^3_{t\delta_{q/5}}$.

Defining the ternary vector $\mathbf{w} = ((\mathbf{w}_1^*)^T \mid (\mathbf{y}^*)^T)^T \in \{-1,0,1\}^D$ of dimension $D = 3(n+m)t\delta_{B_\chi} + 3t\delta_{q/5}$, we finally obtain the equation $\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q$, for public matrix $\mathbf{M}$ and public vector $\mathbf{v}$. Using similar arguments as in Section 9.5.2, we can obtain the desired zero-knowledge argument system. The protocol has communication cost $\mathcal{O}(D \log q) = \widetilde{\mathcal{O}}(\lambda) \cdot \mathcal{O}(t)$ bits.

9.5.4 Protocol 3

Let n, m, m_d, q, t, ℓ, B be the parameters defined in Section 9.3. The protocol allows the prover to argue that a given ciphertext is a correct randomization of some hidden ciphertext and that he knows a valid signature on that ciphertext. Let β be the infinity norm bound of these valid signatures.

Common input: It consists of matrices $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$, $\mathbf{P} \in \mathbb{Z}_q^{m\times t}$, $\mathbf{A}, \mathbf{A}_0, \mathbf{A}_1, \ldots, \mathbf{A}_\ell \in \mathbb{Z}_q^{n\times m}$, $\mathbf{D} \in \mathbb{Z}_q^{n\times m_d}$ and vectors $\mathbf{c}_0 \in \mathbb{Z}_q^n$, $\mathbf{c}_1 \in \mathbb{Z}_q^t$, $\mathbf{u} \in \mathbb{Z}_q^n$.

Prover's goal is to prove knowledge of $\mathfrak{m} \in \{0,1\}^{m_d}$, $\mu \in \{0,1\}^t$, $\mathbf{e} \in \{-1,0,1\}^t$, $\nu \in [-B, B]^t$, $\tau = (\tau[1], \ldots, \tau[\ell])^T \in \{0,1\}^\ell$, $\mathbf{v}_1, \mathbf{v}_2 \in [-\beta, \beta]^m$ such that the following equations hold:

$$\begin{cases} \mathbf{A} \cdot \mathbf{v}_1 + \mathbf{A}_0 \cdot \mathbf{v}_2 + \sum_{j=1}^{\ell} \mathbf{A}_j \cdot (\tau[j] \cdot \mathbf{v}_2) - \mathbf{D} \cdot \mathfrak{m} = \mathbf{u} \bmod q; \\ \mathbf{H}_{n+t,q-1} \cdot \mathfrak{m} + \begin{pmatrix} \mathbf{F} \\ \mathbf{P}^T \end{pmatrix} \cdot \mathbf{e} + \begin{pmatrix} \mathbf{0}^{n\times t} \\ \lfloor \frac{q}{2} \rfloor \cdot \mathbf{I}_t \end{pmatrix} \cdot \mu + \begin{pmatrix} \mathbf{0}^{n\times t} \\ \mathbf{I}_t \end{pmatrix} \cdot \nu = \begin{pmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{pmatrix} \bmod q. \end{cases} \tag{9.22}$$

For this purpose, we perform the following transformations on the witnesses.

Decompositions. Decompose vectors $\mathbf{v}_1, \mathbf{v}_2, \nu$ to vectors $\bar{\mathbf{v}}_1 \in \{-1,0,1\}^{m\delta_\beta}$, $\bar{\mathbf{v}}_2 \in \{-1,0,1\}^{m\delta_\beta}$, $\bar{\nu} \in \{-1,0,1\}^{t\delta_B}$, respectively.

Extensions/Combinations.

- Let $\mathbf{w}_1 = (\mathfrak{m}^T \mid \mu^T)^T \in \{0,1\}^{m_d+t}$ and extend it into $\mathbf{w}_1^* \in \mathsf{B}^2_{m_d+t}$.
- Let $\mathbf{w}_2 = (\bar{\mathbf{v}}_1^T \mid \bar{\nu}^T \mid \mathbf{e}^T)^T \in \{-1,0,1\}^{m\delta_\beta+t\delta_B+t}$ and extend it into the vector $\mathbf{w}_2^* \in \mathsf{B}^3_{m\delta_\beta+t\delta_B+t}$.
- Extend $\bar{\mathbf{v}}_2$ into $\mathbf{s}_0 \in \mathsf{B}^3_{m\delta_\beta}$. Then, for each $j \in [\ell]$, define $\mathbf{s}_j = \mathsf{expand}(\tau[j], \mathbf{s}_0)$. (We refer to Table 9.1 for details about $\mathsf{expand}(\cdot,\cdot)$.)

Now, we form vector $\mathbf{w} = \left(\mathbf{w}_1^{*T} \mid \mathbf{w}_2^{*T} \mid \mathbf{s}_0^T \mid \mathbf{s}_1^T \mid \ldots \mid \mathbf{s}_\ell^T\right)^T \in \{-1,0,1\}^D$, where $D = (2\ell+2)3m\delta_\beta + 3t\delta_B + 3t + 2(m_d+t)$. At this point, we observe that the equations in (9.22) can be equivalently transformed into $\mathbf{M}\cdot\mathbf{w} = \mathbf{v} \bmod q$, where the matrix $\mathbf{M}$ and the vector $\mathbf{v}$ are built from the public input.

Having performed the above transformations, we now define VALID as the set of all vectors $\mathbf{t} = (\mathbf{t}_1^T \mid \mathbf{t}_2^T \mid \mathbf{t}_{3,0}^T \mid \mathbf{t}_{3,1}^T \mid \ldots \mid \mathbf{t}_{3,\ell}^T)^T \in \{-1,0,1\}^D$ for which there exists $\tau = (\tau[1], \ldots, \tau[\ell])^T \in \{0,1\}^\ell$ such that:

$$\mathbf{t}_1 \in \mathsf{B}^2_{m_d+t};\ \mathbf{t}_2 \in \mathsf{B}^3_{m\delta_\beta+t\delta_B+t};\ \mathbf{t}_{3,0} \in \mathsf{B}^3_{m\delta_\beta};\ \forall j \in [\ell] : \mathbf{t}_{3,j} = \mathsf{expand}(\tau[j], \mathbf{t}_{3,0}).$$

It can be seen that $\mathbf{w}$ belongs to this tailored set. Now, let us specify the set $\mathcal{S}$ and permutations of D elements $\{\Gamma_\phi : \phi \in \mathcal{S}\}$ satisfying the conditions in (4.3).

- $\mathcal{S} := \mathcal{S}_{2(m_d+t)} \times \mathcal{S}_{3(m\delta_\beta+t\delta_B+t)} \times \mathcal{S}_{3m\delta_\beta} \times \{0,1\}^\ell$.
- For $\phi = (\phi_1, \phi_2, \phi_3, (b[1], \ldots, b[\ell])^T) \in \mathcal{S}$, we define the permutation Γ_ϕ that transforms vector $\mathbf{t} = (\mathbf{t}_1^T \mid \mathbf{t}_2^T \mid \mathbf{t}_{3,0}^T \mid \mathbf{t}_{3,1}^T \mid \ldots \mid \mathbf{t}_{3,\ell}^T)^T \in \mathbb{Z}^D$ as follows:

$$\Gamma_\phi(\mathbf{t}) = \big(\phi_1(\mathbf{t}_1)^T \mid \quad \phi_2(\mathbf{t}_2)^T \mid \phi_3(\mathbf{t}_{3,0})^T \mid \\ T_{\mathsf{exp}}[b[1],\phi_3](\mathbf{t}_{3,1})^T \mid \ldots \mid T_{\mathsf{exp}}[b[\ell],\phi_3](\mathbf{t}_{3,\ell})^T\big)^T.$$

By inspection, it can be seen that the properties in (4.3) are indeed satisfied. As a result, we can obtain the required argument of knowledge by running the protocol from Section 4.3.2 with common input $(\mathbf{M}, \mathbf{v})$ and prover's input $\mathbf{w}$. The protocol has communication cost $\mathcal{O}(D \log q) = \widetilde{\mathcal{O}}(\lambda) \cdot \mathcal{O}(\log N + t)$ bits.

9.5.5 Protocol 4: A Treatment of Hidden Branching Programs

We now present the proof system run by the user in the OT-AC system of Section 9.4. It allows arguing knowledge of an input $\mathbf{x} = (x_0, \ldots, x_{\kappa-1})^T \in \{0,1\}^\kappa$ satisfying a hidden branching program $\mathsf{BP} = \{(\mathsf{var}(\theta), \pi_{\theta,0}, \pi_{\theta,1})\}_{\theta=1}^L$ of length for $L \in \mathsf{poly}(\lambda)$. The prover should additionally demonstrate that: (i) He has a valid credential for $\mathbf{x}$; (ii) The hashed encoding of BP is associated with some hidden ciphertext of the database (and he knows

a signature guaranteeing this link); (iii) A given ciphertext is a re-randomization of that hidden ciphertext.

Recall that, at each step $\theta \in [L]$ of the evaluation of $\mathsf{BP}(\mathbf{x})$, we have to look up the value $x_{\mathsf{var}(\theta)}$ in $\mathbf{x} = (x_0, \ldots, x_{\kappa-1})^T$ to compute the θ-th state η_θ as per

$$\eta_\theta = \pi_{\theta, x_{\mathrm{var}(\theta)}}(\eta_{\theta-1}) = \pi_{\theta,0}(\eta_{\theta-1}) \cdot \bar{x}_{\mathrm{var}(\theta)} + \pi_{\theta,1}(\eta_{\theta-1}) \cdot x_{\mathrm{var}(\theta)}. \tag{9.23}$$

To prove that each step is done correctly, it is necessary to provide evidence that the corresponding search is honestly carried out without revealing $x_{\mathrm{var}(\theta)}$, $\mathrm{var}(\theta)$ nor $\{\pi_{\theta,b}\}_{b=0}^{1}$. To this end, a first idea is to perform a simple left-to-right search on $(x_0, \ldots, x_{\kappa-1})$: namely, (9.23) is expressed in terms of a matrix-vector relation where η_θ is encoded as a unit vector of dimension 5; $\{\pi_{\theta,b}\}_{b=0}^{1}$ are represented as permutation matrices; and $\mathbf{x}_{\mathrm{var}(\theta)} = \mathbf{M}_{\mathrm{var}(\theta)} \cdot \mathbf{x}$ is computed using a matrix $\mathbf{M}_{\mathrm{var}(\theta)} \in \{0,1\}^{\kappa \times \kappa}$ containing exactly one 1 per row. While this approach can be handled using proofs for matrix-vector relations using the techniques of [LLM+16b], the expected complexity is $\mathcal{O}(\kappa)$ for each step, so that the total complexity becomes $\mathcal{O}(L\kappa)$. Fortunately, a better complexity can be achieved.

If we instead perform a dichotomic search on $\mathbf{x} = (x_0, \ldots, x_{\kappa-1})^T$, we can reduce the complexity of each step to $\mathcal{O}(\log \kappa)$. To this end, we need to prove a statement "I performed a correct dichotomic search on my secret array $\mathbf{x}$".

In order to solve this problem, we will employ two existing lattice-based tools.

(i) A variant of the SIS-based computationally binding and statistically hiding commitment scheme from [KTX08], which allows to commit to one-bit messages;

(ii) The SIS-based Merkle hash tree proposed in [LLNW16].

Let $\bar{\mathbf{A}} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$ and $\mathbf{a}_{\mathsf{com}} \hookleftarrow U(\mathbb{Z}_q^n)$. For each $i \in [0, \kappa-1]$, we let the receiver commit to $x_i \in \{0,1\}$ as $\mathsf{com}_i = \mathbf{a}_{\mathsf{com}} \cdot x_i + \bar{\mathbf{A}} \cdot \mathbf{r}_{\mathsf{com},i}$, with $\mathbf{r}_{\mathsf{com},i} \hookleftarrow U(\{0,1\}^m)$, and reveal $\mathsf{com}_1, \ldots, \mathsf{com}_{\kappa-1}$ to the sender. We build a Merkle tree of depth $\delta_\kappa = \lceil \log \kappa \rceil$ on top of the leaves $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1}$ using the SIS-based hash function $h_{\bar{\mathbf{A}}} : \{0,1\}^{n\lceil \log q \rceil} \times \{0,1\}^{n\lceil \log q \rceil} \to \{0,1\}^{n\lceil \log q \rceil}$ of [LLNW16]. Our use of Merkle trees is reminiscent of [LLNW16] in that the content of the leaves is public. The Merkle tree will actually serve as a "bridge" ensuring that: (i) The same string $\mathbf{x}$ is used in all steps while enabling dichotomic searches; (ii) At each step, the prover indeed uses some coordinate of $\mathbf{x}$ (without revealing which one), the choice of which is dictated by a path in the tree determined by $\mathsf{var}(\theta)$.

Since $\{\mathsf{com}_i\}_{i=0}^{\kappa-1}$ are public, both parties can deterministically compute the root $\mathbf{u}_{\mathsf{tree}}$ of the Merkle tree. For each $\theta \in [L]$, we consider the binary representation $d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa}$ of $\mathsf{var}(\theta)$, which is part of the encoding of BP defined in (9.11). We then prove knowledge of a bit y_θ satisfying the statement "From the root $\mathbf{u}_{\mathsf{tree}} \in \{0,1\}^{n\lceil \log q \rceil}$ of the tree, the path determined by the bits $d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa}$ leads to the leaf associated with the commitment opened to y_θ." If the Merkle tree and the commitment scheme are both secure, it should hold that $y_\theta = x_{\mathrm{var}(\theta)}$. Said otherwise, we can provably perform a "dichotomic search" for $x_{\mathrm{var}(\theta)} = y_\theta$. Moreover, the techniques from [LLNW16] can be adapted to do this in zero-knowledge manner, i.e., without revealing the path nor the reached leaf.

Now, our task can be divided into 3 steps: (i) Proving that the searches on Merkle tree yield $y_1, \ldots, y_L$; (ii) Proving that the branching program evaluates to $\mathsf{BP}(\mathbf{x}) = 1$ if $y_1, \ldots, y_L$

are used in the evaluation; (iii) Proving all the other relations mentioned above, as well as the consistency of $\{\mathsf{com}_i\}_{i=0}^{\kappa-1}$ and the fact that they open to a certified $\mathbf{x} \in \{0,1\}^\kappa$. Thanks to dichotomic searches, the communication cost drops to $\mathcal{O}(L\delta_\kappa + \kappa)$. These steps can be treated as explained below.

9.5.5.1 The Merkle Tree Step.

At each step $\theta \in [L]$, the prover demonstrates knowledge of a path consisting of δ_κ nodes $\mathbf{g}_{\theta,1}, \ldots, \mathbf{g}_{\theta,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil}$ determined by $d_{\theta,1}, \ldots, d_{\theta,\delta_\kappa}$, as well as their sibling nodes $\mathbf{t}_{\theta,1}, \ldots, \mathbf{t}_{\theta,\delta_\kappa} \in \{0,1\}^{n\lceil \log q \rceil}$. Also, the prover argues knowledge of an opening $(y_\theta, \mathbf{r}_\theta) \in \{0,1\} \times \{0,1\}^m$ for the commitment of which $\mathbf{g}_{\theta,\delta_\kappa}$ is a binary decomposition. As shown in Section 4.3, it suffices to prove the following relations (mod q):

$$\forall \theta \in [L] \begin{cases} \bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,1}, \mathbf{g}_{\theta,1}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,1}, \mathbf{t}_{\theta,1}) = \mathbf{H}_{n,q-1} \cdot \mathbf{u}_{\mathsf{tree}}, \\ \bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,2}, \mathbf{g}_{\theta,2}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,2}, \mathbf{t}_{\theta,2}) \\ \qquad\qquad - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,1} = \mathbf{0}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \bar{\mathbf{A}} \cdot \mathsf{expand}(d_{\theta,\delta_\kappa}, \mathbf{g}_{\theta,\delta_\kappa}) + \bar{\mathbf{A}} \cdot \mathsf{expand}(\bar{d}_{\theta,\kappa}, \mathbf{t}_{\theta,\kappa}) \\ \qquad\qquad - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa - 1} = \mathbf{0}, \\ \mathbf{a}_{\mathsf{com}} \cdot y_\theta + \bar{\mathbf{A}} \cdot \mathbf{r}_\theta - \mathbf{H}_{n,q-1} \cdot \mathbf{g}_{\theta,\delta_\kappa} = \mathbf{0}, \end{cases} \tag{9.24}$$

where $\mathsf{expand}(\cdot,\cdot)$ is defined in Table 9.1.

Extending.

- For each $(\theta, i) \in [L] \times [\delta_\kappa]$: Extend $\mathbf{g}_{\theta,i}, \mathbf{t}_{\theta,i} \in \{0,1\}^{m/2}$ to $\widetilde{\mathbf{g}}_{\theta,i}, \widetilde{\mathbf{t}}_{\theta,i} \in \mathsf{B}^2_{m/2}$, respectively. Then, let $\widehat{\mathbf{g}}_{\theta,i} = \mathsf{expand}(d_{\theta,i}, \widetilde{\mathbf{g}}_{\theta,i})$ and $\widehat{\mathbf{t}}_{\theta,i} = \mathsf{expand}(\bar{d}_{\theta,i}, \widetilde{\mathbf{t}}_{\theta,i})$.
- For each $\theta \in [L]$, extend the bit y_θ into the vector $\mathbf{y}_\theta = \mathsf{ext}_2(y_\theta) \in \{0,1\}^2$.
- Let $\widetilde{\mathbf{r}} = (\mathbf{r}_1^T \mid \ldots \mid \mathbf{r}_L^T)^T \in \{0,1\}^{mL}$, then extend it into the vector $\widehat{\mathbf{r}} \in \mathsf{B}^2_{mL}$.

Combining. Next, we let $D_{\mathsf{tree}} = 5mL\delta_\kappa + 2L + 2mL$ and define

$$\begin{aligned} \mathbf{w}_{\mathsf{tree}} = \big(&\widetilde{\mathbf{g}}_{1,1}^T \mid \widehat{\mathbf{g}}_{1,1}^T \mid \widehat{\mathbf{t}}_{1,1}^T \mid \ldots \mid \widetilde{\mathbf{g}}_{1,\delta_\kappa}^T \mid \widehat{\mathbf{g}}_{1,\delta_\kappa}^T \mid \widehat{\mathbf{t}}_{1,\delta_\kappa}^T \mid \ldots \mid \widetilde{\mathbf{g}}_{L,1}^T \mid \widehat{\mathbf{g}}_{L,1}^T \mid \widehat{\mathbf{t}}_{L,1}^T \\ &\mid \ldots \mid \widetilde{\mathbf{g}}_{L,\delta_\kappa}^T \mid \widehat{\mathbf{g}}_{L,\delta_\kappa}^T \mid \widehat{\mathbf{t}}_{L,\delta_\kappa}^T \mid \mathbf{y}_1^T \mid \ldots \mid \mathbf{y}_L^T \mid \widehat{\mathbf{r}}^T\big)^T \in \{0,1\}^{D_{\mathsf{tree}}}. \end{aligned} \tag{9.25}$$

Then, observe that, the above $L(\delta_\kappa + 1)$ equations can be combined into one:

$$\mathbf{M}_{\mathsf{tree}} \cdot \mathbf{w}_{\mathsf{tree}} = \mathbf{v}_{\mathsf{tree}} \bmod q, \tag{9.26}$$

where matrix $\mathbf{M}_{\mathsf{tree}}$ and vector $\mathbf{v}_{\mathsf{tree}}$ are built from the public input.

9.5.5.2 The Branching Program Step.

The last three parts of Table 9.1 describe the vector transformations that will be used to handle the secret vectors appearing in the evaluation of BP. The following equations emulate the evaluation process. In particular, for each $\theta \in [2, L]$, we introduce an extra vector $\mathbf{e}_\theta = (c_{\theta,0}, \ldots, c_{\theta,4}) \in \{0,1\}^5$ to enable the extraction of the values $\pi_{\theta,0}(\eta_{\theta-1})$, and $\pi_{\theta,1}(\eta_{\theta-1})$.

$$\begin{cases} \pi_{1,0}(0) \cdot \bar{y}_1 + \pi_{1,1}(0) \cdot y_1 - \eta_1 = 0, & \text{// computing } \eta_1 \text{ with } \eta_0 = 0 \\ \mathbf{e}_2 - \sum_{i=0}^{4} \mathsf{unit}_i \cdot c_{2,i} = (0,0,0,0,0)^T, & \text{// we will also prove } \mathbf{e}_2 = \mathsf{unit}_{\eta_1} \\ f_{2,0} - \sum_{i=0}^{4} \pi_{2,0}(i) \cdot c_{2,i} = 0, & \text{// meaning: } f_{2,0} = \pi_{2,0}(\eta_1) \\ f_{2,1} - \sum_{i=0}^{4} \pi_{2,1}(i) \cdot c_{2,i} = 0, & \text{// meaning: } f_{2,1} = \pi_{2,1}(\eta_1) \\ f_{2,0} \cdot \bar{y}_2 + f_{2,1} \cdot y_2 - \eta_2 = 0, & \text{// computing } \eta_2 \\ \vdots & \\ \mathbf{e}_L - \sum_{i=0}^{4} \mathsf{unit}_i \cdot c_{L,i} = (0,0,0,0,0)^T, & \text{// we will also prove } \mathbf{e}_L = \mathsf{unit}_{\eta_{L-1}} \\ f_{L,0} - \sum_{i=0}^{4} \pi_{L,0}(i) \cdot c_{L,i} = 0, & \text{// meaning: } f_{L,0} = \pi_{L,0}(\eta_{L-1}) \\ f_{L,1} - \sum_{i=0}^{4} \pi_{L,1}(i) \cdot c_{L,i} = 0, & \text{// meaning: } f_{L,1} = \pi_{L,1}(\eta_{L-1}) \\ f_{L,0} \cdot \bar{y}_L + f_{L,1} \cdot y_L = 0. & \text{// final state } \eta_L = 0 \end{cases} \tag{9.27}$$

Extending.

- For each $\theta \in [L-1]$, extend $\eta_\theta \in [0,4]$ to 5-dimensional vector $\mathbf{s}_\theta = \mathsf{ext}_5(\eta_\theta)$.
- For each $(\theta, j) \in [2, L] \times \{0,1\}$, extend $f_{\theta,j} \in [0,4]$ to $\mathbf{f}_{\theta,j} = \mathsf{ext}_5(f_{\theta,j})$.
- For each $(\theta, i) \in [2, L] \times [0,4]$, extend $c_{\theta,i} \in \{0,1\}$ to $\mathbf{c}_{\theta,i} = \mathsf{ext}_2(c_{\theta,i})$.
- Extend the products $\pi_{1,0}(0) \cdot \bar{y}_1$ and $\pi_{1,1}(0) \cdot y_1$ into 10-dimensional vectors $\mathbf{h}_{1,0} = \mathsf{ext}_{5\times 2}(\pi_{1,0}(0), \bar{y}_1)$ and $\mathbf{h}_{1,1} = \mathsf{ext}_{5\times 2}(\pi_{1,1}(0), y_1)$, respectively.
- For each $\theta \in [2, L]$, extend the products $f_{\theta,0} \cdot \bar{y}_\theta$ and $f_{\theta,1} \cdot y_\theta$ into 10-dimensional vectors $\mathbf{h}_{\theta,0} = \mathsf{ext}_{5\times 2}(f_{\theta,0}, \bar{y}_\theta)$ and $\mathbf{h}_{\theta,1} = \mathsf{ext}_{5\times 2}(f_{\theta,1}, y_\theta)$.
- For $(\theta, i) \in [2, L] \times [0,4]$, extend the products $\pi_{\theta,0}(i) \cdot c_{\theta,i}$ and $\pi_{\theta,1}(i) \cdot c_{\theta,i}$ into $\mathbf{z}_{\theta,0,i} = \mathsf{ext}_{5\times 2}(\pi_{\theta,0}(i), c_{\theta,i})$ and $\mathbf{z}_{\theta,1,i} = \mathsf{ext}_{5\times 2}(\pi_{\theta,1}(i), c_{\theta,i})$, respectively.

Combining. Let $D_{\mathsf{BP}} = 150L - 130$, and form $\mathbf{w}_{\mathsf{BP}} \in [0,4]^{D_{\mathsf{BP}}}$ of the form:

$$\begin{aligned} (\mathbf{s}_1^T \mid \ldots \mid \quad & \mathbf{s}_{L-1}^T \mid \mathbf{e}_2^T \mid \ldots \mid \mathbf{e}_L^T \mid \mathbf{c}_{2,0}^T \mid \ldots \mid \mathbf{c}_{L,4}^T \mid \mathbf{z}_{2,0,0}^T \mid \ldots \mid \mathbf{z}_{L,1,4}^T \mid \\ & \mathbf{f}_{2,0}^T \mid \ldots \mid \mathbf{f}_{L,1}^T \mid \mathbf{h}_{1,0}^T \mid \mathbf{h}_{1,1}^T \mid \mathbf{h}_{2,0}^T \mid \mathbf{h}_{2,1}^T \mid \ldots \mid \mathbf{h}_{L,0}^T \mid \mathbf{h}_{L,1}^T)^T. \end{aligned} \tag{9.28}$$

Then, observe that the vector $\mathbf{w}_{\mathsf{BP}}$ of (9.28) satisfies *one* equation of the form:

$$\mathbf{M}_{\mathsf{BP}} \cdot \mathbf{w}_{\mathsf{BP}} = \mathbf{v}_{\mathsf{BP}}, \tag{9.29}$$

where matrix $\mathbf{M}_{\mathsf{BP}}$ and vector $\mathbf{v}_{\mathsf{BP}}$ are obtained from the common input. Note that we work with integers in $[0,4]$, which are much smaller than q. As a result,

$$\mathbf{M}_{\mathsf{BP}} \cdot \mathbf{w}_{\mathsf{BP}} = \mathbf{v}_{\mathsf{BP}} \bmod q. \tag{9.30}$$

Conversely, if we can prove that (9.30) holds for a well-formed vector $\mathbf{w}_{\mathsf{BP}}$, then that vector should also satisfy (9.29).

9.5.5.3 The Third Step.

In the third layer, we have to prove knowledge of:

$$\begin{cases} d_{1,1},\ldots,d_{L,\delta_\kappa} \in \{0,1\}, \pi_{1,0}(0),\ldots,\pi_{L,1}(4) \in [0,4],\ \mathfrak{m} \in \{0,1\}^{m_d}, \\ \mathbf{x} = (x_0,\ldots,x_{\kappa-1})^T \in \{0,1\}^\kappa,\ \mathfrak{m}_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{\frac{m}{2}+\kappa},\ \widehat{\mathfrak{m}}_{\mathsf{U},\mathbf{x}} \in \{0,1\}^{\frac{m}{2}}, \\ \mathbf{e}_{\mathsf{U}} \in \{0,1\}^m,\ \mathbf{r}_{\mathsf{com},0},\ldots,\mathbf{r}_{\mathsf{com},\kappa-1} \in \{0,1\}^m,\ \mu \in \{0,1\}^t,\ \tau \in \{0,1\}^\ell, \\ \tau_{\mathsf{U}} \in \{0,1\}^{\ell_I}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_{\mathsf{U},1}, \mathbf{v}_{\mathsf{U},2}, \mathbf{r}_{\mathsf{U}} \in [-\beta,\beta]^m, \mathbf{e} \in \{-1,0,1\}^t, \nu \in [-B,B]^t, \end{cases} \tag{9.31}$$

which satisfy the equations of (9.15) for $\mathbf{z}_{\mathsf{BP},\rho} = (d_{1,1},\ldots,d_{L,\delta_\kappa},\pi_{1,0}(0),\ldots,\pi_{L,1}(4))^T$ and, $\forall i \in [0,\kappa-1]$, the bit x_i is committed in com_i with randomness $\mathbf{r}_{\mathsf{com},i}$:

$$\begin{bmatrix} \mathbf{a}_{\mathsf{com}} & & \\ & \ddots & \\ & & \mathbf{a}_{\mathsf{com}} \end{bmatrix} \cdot \mathbf{x} + \begin{bmatrix} \bar{\mathbf{A}} & & \\ & \ddots & \\ & & \bar{\mathbf{A}} \end{bmatrix} \cdot \begin{pmatrix} \mathbf{r}_{\mathsf{com},0} \\ \vdots \\ \mathbf{r}_{\mathsf{com},\kappa-1} \end{pmatrix} = \begin{pmatrix} \mathsf{com}_0 \\ \vdots \\ \mathsf{com}_{\kappa-1} \end{pmatrix} \bmod q.$$

Decomposing. We use $\mathsf{vdec}'_{m,\beta}(\cdot)$ to decompose $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_{\mathsf{U},1}, \mathbf{v}_{\mathsf{U},2}, \mathbf{r}_{\mathsf{U}} \in [-\beta,\beta]^m$ into $\bar{\mathbf{v}}_1, \bar{\mathbf{v}}_2, \bar{\mathbf{v}}_{\mathsf{U},1}, \bar{\mathbf{v}}_{\mathsf{U},2}, \bar{\mathbf{r}}_{\mathsf{U}} \in \{-1,0,1\}^{m\delta_\beta}$, respectively. Similarly, we decompose vector $\nu \in [-B,B]^t$ into vector $\bar{\nu} = \mathsf{vdec}'_{t,B}(\nu) \in \{-1,0,1\}^{t\delta_B}$.

Extending and Combining. Next, we perform the following steps:

- For each $(\theta,i) \in [L] \times [\delta_\kappa]$, extend $d_{\theta,i}$ to $\mathbf{d}_{\theta,i} = \mathsf{ext}_2(d_{\theta,i})$.
- For each $(\theta,j,i) \in [L] \times \{0,1\} \times [0,4]$, extend $\pi_{\theta,j}(i)$ to $\Pi_{\theta,j,i} = \mathsf{ext}_5(\pi_{\theta,j}(i))$.
- Let $\overline{\mathbf{w}}_{3,1} = (\mathbf{x}^T|\mathbf{r}^T_{\mathsf{com},0}|\ldots|\mathbf{r}^T_{\mathsf{com},\kappa-1}|\mathfrak{m}^T_{\mathsf{U},\mathbf{x}}|\widehat{\mathfrak{m}}^T_{\mathsf{U},\mathbf{x}}|\mathfrak{m}^T \mid \mathbf{e}^T_{\mathsf{U}}|\mu^T)^T \in \{0,1\}^{D_{3,1}}$, where $D_{3,1} = \kappa(m+2)+2m+m_d+t$. Then extend $\overline{\mathbf{w}}_{3,1}$ to $\mathbf{w}_{3,1} \in \mathsf{B}^2_{D_{3,1}}$.
- Define the vector $\overline{\mathbf{w}}_{3,2} = (\bar{\mathbf{v}}^T_1|\bar{\mathbf{v}}^T_{\mathsf{U},1}|\bar{\mathbf{r}}^T_{\mathsf{U}}|\bar{\nu}^T|\mathbf{e}^T)^T \in \{-1,0,1\}^{D_{3,2}}$ of dimension $D_{3,2} = 3m\delta_\beta + t(\delta_B+1)$ and extend it into $\mathbf{w}_{3,2} \in \mathsf{B}^3_{D_{3,2}}$.
- Extend $\bar{\mathbf{v}}_2$ to $\mathbf{s}_0 \in \mathsf{B}^3_{m\delta_\beta}$. Then for $j \in [\ell]$, form vector $\mathsf{s}_j = \mathsf{expand}(\tau[j], \mathbf{s}_0)$.
- Extend $\bar{\mathbf{v}}_{\mathsf{U},2}$ to $\mathbf{s}_{\mathsf{U},0} \in \mathsf{B}^3_{m\delta_\beta}$. Then for $j \in [\ell_I]$, form $\mathsf{s}_{\mathsf{U},j} = \mathsf{expand}(\tau_{\mathsf{U}}[j], \mathbf{s}_{\mathsf{U},0})$.

Given the above transformations, let $D_3 = 2L(\delta_\kappa+25) + 2D_{3,1} + 3D_{3,2} + 3m\delta_\beta(2\ell+1) + 3m\delta_\beta(2\ell_I+1)$ and construct vector $\mathbf{w}_3 \in [-1,4]^{D_3}$ of the form:

$$(\mathbf{d}^T_{1,1} \mid \ldots \mid \mathbf{d}^T_{L,\delta_\kappa} \mid \Pi^T_{1,0,0} \mid \ldots \mid \Pi^T_{L,1,4} \mid \mathbf{w}^T_{3,1} \mid \mathbf{w}^T_{3,2} \mid \mathbf{s}^T_0 \mid \mathbf{s}^T_1 \mid \ldots \mid \mathbf{s}^T_\ell \mid \mathbf{s}^T_{\mathsf{U},0} \mid \mathbf{s}^T_{\mathsf{U},1} \mid \ldots \mid \mathbf{s}^T_{\mathsf{U},\ell_I} \mid)^T. \tag{9.32}$$

Observe that the given five equations can be combined into one of the form:

$$\mathbf{M}_3 \cdot \mathbf{w}_3 = \mathbf{v}_3 \bmod q, \tag{9.33}$$

where matrix $\mathbf{M}_3$ and vector $\mathbf{v}_3$ can be built from the public input.

9.5.5.4 Putting Pieces Altogether.

At the final stage of the process, we connect the three aforementioned steps. Indeed, all the equations involved in our process are captured by (9.26), (9.30), and (9.33) - which in turn can be combined into:

$$\mathbf{M} \cdot \mathbf{w} = \mathbf{v} \bmod q, \tag{9.34}$$

where $\mathbf{w} = (\mathbf{w}_{\mathsf{tree}}^T \mid \mathbf{w}_{\mathsf{BP}}^T \mid \mathbf{w}_3^T)^T \in [-1, 4]^D$, for

$$D = D_{\mathsf{tree}} + D_{\mathsf{BP}} + D_3 = \widetilde{\mathcal{O}}(\lambda) \cdot (L \cdot \log \kappa + \kappa) + \widetilde{\mathcal{O}}(\lambda) \cdot (\log N + \lambda) + \widetilde{\mathcal{O}}(1) \cdot t.$$

The components of $\mathbf{w}$ all have constraints listed in Table 9.1. By construction, these blocks either belong to the special sets $\mathsf{B}_{\mathsf{m}}^2$, $\mathsf{B}_{\mathsf{m}}^3$ or they have the special forms $\mathsf{expand}(\cdot,\cdot)$, $\mathsf{ext}_2(\cdot)$, $\mathsf{ext}_5(\cdot)$, $\mathsf{ext}_{5\times 2}(\cdot,\cdot)$, which are invariant under the permutations defined in Table 9.1. As a result, we can specify suitable sets VALID, $\mathcal{S}$ and permutations of D elements $\{\Gamma_\phi : \phi \in \mathcal{S}\}$, for which the conditions of (4.3) are satisfied.The description of the elements VALID, $\mathcal{S}$ and Γ_ϕ is detailed as follows.

Let VALID be the set of all vectors in $[-1, 4]^D$ having the form $(\mathbf{w}_{\mathsf{tree}}^T \| \mathbf{w}_{\mathsf{BP}}^T \| \mathbf{w}_3^T)^T$, where $\mathbf{w}_{\mathsf{tree}}, \mathbf{w}_{\mathsf{BP}}, \mathbf{w}_3$ have the form (9.25), (9.28), and (9.32), respectively, and the following conditions hold:

- For each $(\theta, i) \in [L] \times [\delta_\kappa]$: There exists $d_{\theta,i} \in \{0, 1\}$ and $\widetilde{\mathbf{g}}_{\theta,i}, \widetilde{\mathbf{t}}_{\theta,i} \in \mathsf{B}_{m/2}^2$ such that $\mathbf{d}_{\theta,i} = \mathsf{ext}_2(d_{\theta,i})$ and

$$\widehat{\mathbf{g}}_{\theta,i} = \mathsf{expand}(d_{\theta,i}, \widetilde{\mathbf{g}}_{\theta,i}); \qquad \widehat{\mathbf{t}}_{\theta,i} = \mathsf{expand}(\bar{d}_{\theta,i}, \widetilde{\mathbf{t}}_{\theta,i}).$$

- There exist $y_1 \in \{0, 1\}$ and $\pi_{1,0}(0), \pi_{1,1}(0) \in [0, 4]$ such that

$$\begin{cases} \mathbf{y}_1 = \mathsf{ext}_2(y_1); \quad \mathbf{h}_{1,0} = \mathsf{ext}_{5\times 2}(\pi_{1,0}(0), \bar{y}_1); \quad \mathbf{h}_{1,1} = \mathsf{ext}_{5\times 2}(\pi_{1,1}(0), y_1); \\ \Pi_{1,0,0} = \mathsf{ext}_5(\pi_{1,0}(0)); \quad \Pi_{1,1,0} = \mathsf{ext}_5(\pi_{1,1}(0)). \end{cases}$$

- For all $(j, i) \in \{0, 1\} \times [1, 4]$: $\Pi_{1,j,i} = \mathsf{ext}_5(\pi_{1,j}(i))$, for *some* $\pi_{1,j}(i) \in [0, 4]$. (Note that these $\pi_{1,j}(i)$ do *not* participate in the evaluation of the BP.)

- For $\theta \in [2, L]$: There exist $y_\theta \in \{0, 1\}$, $f_{\theta,0}, f_{\theta,1} \in [0, 4]$ such that $\mathbf{y}_\theta = \mathsf{ext}_2(y_\theta)$ and

$$\begin{cases} \mathbf{f}_{\theta,0} = \mathsf{ext}_5(f_{\theta,0}); & \mathbf{f}_{\theta,1} = \mathsf{ext}_5(f_{\theta,1}); \\ \mathbf{h}_{\theta,0} = \mathsf{ext}_{5\times 2}(f_{\theta,0}, \bar{y}_\theta); & \mathbf{h}_{\theta,1} = \mathsf{ext}_{5\times 2}(f_{\theta,1}, y_\theta). \end{cases}$$

- For $(\theta, j, i) \in [2, L] \times \{0, 1\} \times [0, 4]$: there exist $\pi_{\theta,j}(i) \in [0, 4]$, $c_{\theta,i} \in \{0, 1\}$ such that

$$\Pi_{\theta,j,i} = \mathsf{ext}_5(\pi_{\theta,j}(i)); \qquad \mathbf{c}_{\theta,i} = \mathsf{ext}_2(c_{\theta,i}); \qquad \mathbf{z}_{\theta,j,i} = \mathsf{ext}_{5\times 2}(\pi_{\theta,j}(i), c_{\theta,i}).$$

- There exist $\eta_1, \ldots, \eta_{L-1} \in [0, 4]$ such that the following hold.
  1. For all $\theta = 1, \ldots, L-1$: $\mathbf{s}_\theta = \mathsf{ext}_5(\eta_\theta)$.
  2. For all $\theta = 2, \ldots, L$: $\mathbf{e}_\theta = \mathsf{unit}_{\eta_{\theta-1}}$.
- $\widehat{\mathbf{r}} \in \mathsf{B}^2_{mL}$; $\quad \mathbf{w}_{3,1} \in \mathsf{B}^2_{D_{3,1}}$; $\quad \mathbf{w}_{3,2} \in \mathsf{B}^3_{D_{3,2}}$;
- $\mathbf{s}_0 \in \mathsf{B}^3_{m\delta_\beta}$, and there exists $\tau = (\tau[1], \ldots, \tau[\ell])^T \in \{0,1\}^\ell$ such that for all $j \in [\ell]$: $\mathsf{s}_j = \mathsf{expand}(\tau[j], \mathbf{s}_0)$.
- $\mathbf{s}_{\mathsf{U},0} \in \mathsf{B}^3_{m\delta_\beta}$, and there exists $\tau_\mathsf{U} = (\tau_\mathsf{U}[1], \ldots, \tau_\mathsf{U}[\ell_I])^T \in \{0,1\}^{\ell_I}$ such that for all $j \in [\ell_I]$: $\mathsf{s}_{\mathsf{U},j} = \mathsf{expand}(\tau_\mathsf{U}[j], \mathbf{s}_{\mathsf{U},0})$.

By construction, we have $\mathbf{w} \in \mathsf{VALID}$. Let us now specify the set $\mathcal{S}$ and permutations of D elements $\{\Gamma_\phi : \phi \in \mathcal{S}\}$, for which the conditions in (4.3) hold. Again, we refer to the notations and techniques from Table 9.1, which we will apply here. Let

$$\mathcal{S} = \{0,1\}^{L\delta_\kappa} \times \{0,1\}^L \times [0,4]^{10L} \times [0,4]^{L-1} \times \{0,1\}^{5(L-1)} \times [0,4]^{2(L-1)} \times \big((\mathcal{S}_m)^{2L\delta_\kappa} \times \mathcal{S}_{2mL}\big) \times (\mathcal{S}_{2D_{3,1}} \times \mathcal{S}_{3D_{3,2}} \times (\mathcal{S}_{3m\delta_\beta} \times \{0,1\}^\ell) \times (\mathcal{S}_{3m\delta_\beta} \times \{0,1\}^{\ell_I})).$$

For each $\phi = (\mathbf{b}_d, \mathbf{b}_y, \mathbf{b}_p, \mathbf{b}_c, \mathbf{b}_f, \Sigma_{\mathsf{tree}}, \Sigma_3) \in \mathcal{S}$, where:

$$\begin{cases} \mathbf{b}_d = (b_{d,1,1}, \ldots, b_{d,L,\delta_\kappa})^T, \quad \mathbf{b}_y = (b_{y,1}, \ldots, b_{y,L})^T, \\ \mathbf{b}_\eta = (b_{\eta,1}, \ldots, b_{\eta,L-1})^T, \ \ \mathbf{b}_c = (b_{c,2,0}, \ldots, b_{c,L,4})^T, \ \ \mathbf{b}_f = (b_{f,2,0}, \ldots, b_{f,L,1})^T, \\ \mathbf{b}_p = (b_{\pi,1,0,0}, \ldots, b_{\pi,L,1,4})^T, \ \ \Sigma_{\mathsf{tree}} = (\sigma_{g,1,1}, \ldots, \sigma_{g,\theta,\delta_\kappa}, \sigma_{t,1,1}, \ldots, \sigma_{t,\theta,\delta_\kappa}, \sigma_r), \\ \Sigma_3 = (\sigma_{3,1},\ \sigma_{3,2},\ \sigma_{3,3,1},\ b_\tau[1] \ldots b_\tau[\ell],\ \sigma_{3,3,2},\ b_{\tau,\mathsf{U}}[1] \ldots b_{\tau,\mathsf{U}}[\ell_I]), \end{cases}$$

let Γ_ϕ be the permutation that, when applying to vector $\mathbf{s}$ of the form $(\mathbf{w}^T_{\mathsf{tree}} \| \mathbf{w}^T_{\mathsf{BP}} \| \mathbf{w}^T_3)^T \in \mathbb{Z}^D$, where $\mathbf{w}_{\mathsf{tree}}, \mathbf{w}_{\mathsf{BP}}, \mathbf{w}_3$ have the form (9.25), (9.28), and (9.32), respectively, transforms the block-vectors as follows:

- For each $(\theta, i) \in [L] \times [\delta_\kappa]$: $\widetilde{\mathbf{g}}_{\theta,i} \mapsto \sigma_{g,\theta,i}(\widetilde{\mathbf{g}}_{\theta,i})$ and
$$\widehat{\mathbf{g}}_{\theta,i} \mapsto T_{\mathsf{exp}}[b_{d,\theta,i}, \sigma_{g,\theta,i}](\widehat{\mathbf{g}}_{\theta,i}); \qquad \widehat{\mathbf{t}}_{\theta,i} \mapsto T_{\mathsf{exp}}[b_{d,\theta,i}, \sigma_{t,\theta,i}](\widehat{\mathbf{t}}_{\theta,i}).$$
- For each $\theta \in [L]$: $\mathbf{y}_\theta \mapsto T_2[b_{y,\theta}](\mathbf{y}_\theta)$; $\widehat{\mathbf{r}} \mapsto \sigma_r(\widehat{\mathbf{r}})$.
- For each $\theta = 1, \ldots, L-1$: $\mathbf{s}_\theta \mapsto T_5[b_{\eta,\theta}](\mathbf{s}_\theta)$.
- For each $\theta = 2, \ldots, L$: $\mathbf{e}_\theta \mapsto T_5[b_{\eta,\theta-1}](\mathbf{e}_\theta)$.
- For each $(\theta, i) \in [2, L] \times [0, 4]$: $\mathbf{c}_{\theta,i} \mapsto T_2[b_{c,\theta,i}](\mathsf{c}_{\theta,i})$.
- For each $(\theta, j, i) \in [2, L] \times \{0,1\} \times [0,4]$: $\mathbf{z}_{\theta,j,i} \mapsto T_{5\times 2}[b_{\pi,\theta,j,i}, b_{c,\theta,i}](\mathbf{z}_{\theta,j,i})$.
- For each $(\theta, j) \in [2, L] \times \{0,1\}$: $\mathbf{f}_{\theta,j} \mapsto T_5[b_{f,\theta,j}](\mathbf{f}_{\theta,j})$.
- $\mathbf{h}_{1,0} \mapsto T_{5\times 2}[b_{\pi,1,0,0}, b_{y,1}](\mathbf{h}_{1,0})$; $\mathbf{h}_{1,1} \mapsto T_{5\times 2}[b_{\pi,1,1,0}, b_{y,1}](\mathbf{h}_{1,1})$.

- For each $(\theta, j) \in [2, L] \times \{0, 1\}$: $\mathbf{h}_{\theta,j} \mapsto T_{5\times 2}[b_{f,\theta,j}, b_{y,\theta}](\mathbf{h}_{\theta,j})$.
- For each $(\theta, i) \in [L] \times [\delta_\kappa]$: $\mathbf{d}_{\theta,i} \mapsto T_2[b_{d,\theta,i}](\mathbf{d}_{\theta,i})$.
- For each $(\theta, j, i) \in [L] \times \{0, 1\} \times [0, 4]$: $\Pi_{\theta,j,i} \mapsto T_5[b_{\pi,\theta,j,i}](\Pi_{\theta,j,i})$.
- For each $i \in [1, 2]$: $\mathbf{w}_{3,i} \mapsto \sigma_{3,i}(\mathbf{w}_{3,i})$.
- $\mathsf{s}_0 \mapsto \sigma_{3,3,1}(\mathsf{s}_0)$. For each $j \in [\ell]$: $\mathbf{s}_j \mapsto T_{\mathsf{exp}}[b_\tau[j], \sigma_{3,3,1}](\mathbf{s}_j)$.
- $\mathsf{s}_{\mathsf{U},0} \mapsto \sigma_{3,3,2}(\mathsf{s}_{\mathsf{U},0})$. For each $j \in [\ell_I]$: $\mathsf{s}_{\mathsf{U},j} \mapsto T_{\mathsf{exp}}[b_{\tau,\mathsf{U}}[j], \sigma_{3,3,2}](\mathsf{s}_{\mathsf{U},j})$.

Based on the equivalences observed in Table 9.1, it can be checked that the conditions of (4.3) hold, namely:

$$\begin{cases} \mathbf{w} \in \mathsf{VALID} \iff \Gamma_\phi(\mathbf{w}) \in \mathsf{VALID}, \\ \text{If } \mathbf{w} \in \mathsf{VALID} \text{ and } \phi \text{ is uniform in } \mathcal{S}, \text{ then } \Gamma_\phi(\mathbf{w}) \text{ is uniform in } \mathsf{VALID}. \end{cases}$$

Our desired argument system then works as follows. At the beginning of the interaction, the prover computes commitments $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1} \in \mathbb{Z}_q^n$ and send them once to the verifier. Both parties construct matrix $\mathbf{M}$ and vector $\mathbf{v}$ based on the public input as well as $\mathsf{com}_0, \ldots, \mathsf{com}_{\kappa-1}$, while the prover prepares vector $\mathbf{w}$, as described. Finally, they run the protocol of Section 4.3.2, which has communication cost $\mathcal{O}(D \log q) = \mathcal{O}(L \cdot \log \kappa + \kappa)$.

9.6 Reducing the Communication Complexity in the Random Oracle Model

One limitation of our basic adaptive OT protocol is that it requires the sender to repeat the zero-knowledge proofs of the initialization phase for each user. In total, the communication cost of the initialization phase thus amounts to $\Omega(\lambda NU)$, which is even more expensive than the $O(\lambda(N + U))$ complexities of [CNs07, GH07, CDN09, JL09]. As pointed out by Green and Hohenberger [GH11], decreasing the cost of the initialization phase to be independent of the number of users is highly desirable: ideally, one would certainly prefer a non-interactive initialization phase where the Sender can publicize a $O(\lambda N)$-size commitment to the database, which can subsequently be used by arbitrarily many receivers.

In the random oracle model, we show that our protocols can both be modified to obtain this optimized communication complexity. This can be achieve by the simple expedient of making the sender's zero-knowledge proofs non-interactive via the Fiat-Shamir heuristic. By removing interaction from *all* the sender's proof (i.e., even those of the transfer phase), we also minimize the number of communication rounds since we only need the verifier's arguments to be witness indistinguishable and we can thus safely repeat them in parallel. Relying on the random oracle model thus allows the sender to publicize the entire database and proofs on a public repository so as to avoid repeating these proofs for each receiver.

9.6.1 Description

The description hereunder relies on the same parameters as in sections 9.3 and 9.4. Namely, we use $m = 2n\lceil \log q \rceil$, a modulus q for which the noise distribution χ is αq-bounded, for

some $0 < \alpha < 1$, and also define an integer B as a randomization parameter such that $(m+1)\alpha q/B$ is negligible and choosing α such that $(m+1)\alpha \leq 1/5$ ensures decryption correctness.

We assume two random oracles $H_F : \{0,1\}^* \to \mathbb{Z}_q^{n\times m}$ and $H_{\mathsf{FS}} : \{0,1\}^* \to \{1,2,3\}^{\varsigma}$, for some $\varsigma \in \omega(\log n)$. The former will be used to derive the sender's public matrix $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$ while the latter will provide the verifier's challenges when we apply the Fiat-Shamir heuristic.

Initialization$(\mathsf{S}_{\mathsf{I}}(1^\lambda, \mathbf{DB}), \mathsf{R}_{\mathsf{I}}(1^\lambda))$**:** In this protocol, the sender S_{I} has a database $\mathsf{DB} = (M_1, \ldots, M_N)$ of N messages, where $M_i \in \{0,1\}^t$ for each $i \in [N]$, for some $t \in \mathsf{poly}(\lambda)$. It interacts with the receiver R_{I} as follows.

1. Generate a key pair for the signature scheme of Section 9.2.1 in order to sign $Q = N$ messages of length $m_d = (n+t)\cdot\lceil \log q \rceil$ each. This key pair consists of $SK_{sig} = \mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m\times m}$ and

$$PK_{sig} := (\mathbf{A}, \{\mathbf{A}_j\}_{j=0}^{\ell}, \mathbf{D}, \mathbf{u}),$$

   where $\ell = \log N$ and $\mathbf{A}, \mathbf{A}_0, \ldots, \mathbf{A}_\ell \in U(\mathbb{Z}_q^{n\times m})$, $\mathbf{D} \in U(\mathbb{Z}_q^{n\times m_d})$ with $m = 2n\lceil \log q \rceil$, $m_d = (n+t)\lceil \log q \rceil$. The counter is initialized to $\tau = 0$.

2. Choose a matrix $\mathbf{S} \hookleftarrow \chi^{n\times t}$ that will serve as a secret key for an LWE-based encryption scheme. Then, define the matrix $\mathbf{F} = H_F(\varepsilon) \in \mathbb{Z}_q^{n\times m}$ and sample a matrix $\mathbf{E} \hookleftarrow \chi^{m\times t}$ to compute

$$\mathbf{P} = [\mathbf{p}_1 \mid \ldots \mid \mathbf{p}_t] = \mathbf{F}^T \cdot \mathbf{S} + \mathbf{E} \in \mathbb{Z}_q^{m\times t} \tag{9.35}$$

   so that $(\mathbf{F}, \mathbf{P}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{m\times t}$ forms a public key for a t-bit variant of Regev's encryption scheme [Reg05] (or, equivalently, a set of m encryptions of the all-zeroes t-bit string).

3. Sample vectors $\mathbf{a}_1, \ldots, \mathbf{a}_N \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{x}_1, \ldots, \mathbf{x}_N \hookleftarrow \chi^t$ to compute

$$(\mathbf{a}_i, \mathbf{b}_i) = (\mathbf{a}_i,\ \mathbf{S}^T \cdot \mathbf{a}_i + \mathbf{x}_i + M_i \cdot \lfloor q/2 \rfloor) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t \qquad \forall i \in [N] \tag{9.36}$$

4. For each $i = 1$ to N, generate a signature $(\tau_i, \mathbf{v}_i) \leftarrow \mathsf{Sign}(SK_{sig}, \tau, \mathfrak{m}_i)$ on the message $\mathfrak{m}_i = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_i|\mathbf{b}_i) \in \{0,1\}^{m_d}$ obtained by decomposing $(\mathbf{a}_i^T|\mathbf{b}_i^T)^T \in \mathbb{Z}_q^{n+t}$.

5. S_{I} sends R_{I} the initialization data

$$R_0 = (PK_{sig},\ (\mathbf{F}, \mathbf{P}),\ \{(\mathbf{a}_i, \mathbf{b}_i), (\tau_i, \mathbf{v}_i)\}_{i=1}^N,\ \pi_K), \tag{9.37}$$

   which includes a NIZK argument of knowledge π_K of small-norm matrices $\mathbf{S} \in \mathbb{Z}^{n\times t}$ and $\mathbf{E} \in \chi^{m\times t}$ and t-bit messages $\{M_i\}_{i=1}^N$ that are consistent with (9.35)-(9.36). The argument π_K is built by taking the following steps:

   a. Define $\mathbf{A}_{\mathsf{DB}} = [\mathbf{a}_1|\ldots|\mathbf{a}_N] \in \mathbb{Z}_q^{n\times N}$, $\mathbf{B}_{\mathsf{DB}} = [\mathbf{b}_1|\ldots|\mathbf{b}_N] \in \mathbb{Z}_q^{t\times N}$, $\mathbf{M} = [M_1|\ldots|M_N] \in \{0,1\}^{t\times N}$, $\mathbf{X} = [\mathbf{x}_1|\ldots|\mathbf{x}_N] \in \chi^{t\times N}$ and parse $\mathbf{S}$ and $\mathbf{E}$ as $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t] \in \chi^{n\times t}$, $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m\times t}$.

b. For each $j \in [t]$, define $\bar{M}_j \in \{0,1\}^N$ to be the j-th column of $\mathbf{M}^T = [\bar{M}_1|\ldots|\bar{M}_t]$. Likewise, let $\bar{\mathbf{b}}_j \in \mathbb{Z}_q^N$ (resp. $\bar{\mathbf{x}}_j \in \chi^N$) be the j-th column of $\mathbf{B}_{\mathsf{DB}}^T = [\bar{\mathbf{b}}_1|\ldots|\bar{\mathbf{b}}_t] \in \mathbb{Z}_q^{N\times t}$ (resp. $\mathbf{X}^T = [\bar{\mathbf{x}}_1|\ldots|\bar{\mathbf{x}}_t]$) and generate a signature of knowledge of $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$, $\bar{\mathbf{x}}_j \in \chi^N$, $\bar{M}_j \in \{0,1\}^N$, for $j \in [t]$, such that

$$\left[\begin{array}{c|c|c|c} \mathbf{F}^T & \mathbf{I}_m & & \\ \hline \mathbf{A}_{\mathsf{DB}}^T & & \mathbf{I}_N & \lfloor q/2\rfloor \cdot \mathbf{I}_N \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{s}_j \\ \hline \mathbf{e}_j \\ \hline \bar{\mathbf{x}}_j \\ \hline \bar{M}_j \end{array}\right] = \left[\begin{array}{c} \mathbf{p}_j \\ \hline \bar{\mathbf{b}}_j \end{array}\right] \tag{9.38}$$

Let the NIZK proof be $\pi_K = (\{\mathsf{Comm}_{K,j}\}_{j=1}^{\varsigma}, \mathsf{Chall}_K, \{\mathsf{Resp}_{K,j}\}_{j=1}^{\varsigma})$, where $\mathsf{Chall}_K = H_{\mathsf{FS}}((\mathbf{F},\mathbf{P},\mathbf{A}_{\mathsf{DB}},\mathbf{B}_{\mathsf{DB}}),\{\mathsf{Comm}_{K,j}\}_{j=1}^{\varsigma}) \in \{1,2,3\}^{\varsigma}$.

c. If the proof of knowledge π_K does not verify or if there exists $i \in [N]$ such that $(\tau_i, \mathbf{v}_i)$ is an invalid signature on $\mathsf{vdec}_{n+t,q-1}\big((\mathbf{a}_i^T|\mathbf{b}_i^T)^T\big)^T$, then R_{I} aborts.

6. Finally S_{I} defines $S_0 = \big((\mathbf{S},\mathbf{E}),(\mathbf{F},\mathbf{P}),PK_{sig}\big)$, which it keeps to itself.

Transfer$\big(\mathsf{S}_{\mathsf{T}}(S_{i-1}),\mathsf{R}_{\mathsf{T}}(R_{i-1},\rho_i)\big)$**:** At the i-th transfer, the receiver R_{T} has state R_{i-1} and an index $\rho_i \in [1,N]$. It interacts as follows with the sender S_{T} that has state S_{i-1} in order to obtain M_{ρ_i} from DB.

1. R_{T} samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and a random $\nu \hookleftarrow U([-B,B]^t)$ to compute

$$(\mathbf{c}_0,\mathbf{c}_1) = \big(\mathbf{a}_{\rho_i} + \mathbf{F}\cdot\mathbf{e},\ \mathbf{b}_{\rho_i} + \mathbf{P}^T\cdot\mathbf{e} + \mu\cdot\lfloor q/2\rfloor + \nu\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t, \tag{9.39}$$

which is a re-randomization of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i} + \mu\cdot\lfloor q/2\rfloor)$. The resulting ciphertext $(\mathbf{c}_0,\mathbf{c}_1)$ is sent to S_{T}. In addition, R_{T} provides an interactive WI argument that $(\mathbf{c}_0,\mathbf{c}_1)$ is indeed a re-randomization of $(\mathbf{a}_{\rho_i},\mathbf{b}_{\rho_i})$ for some index $\rho_i \in [N]$. To this end, R_{T} argues knowledge of short vectors $\mathfrak{m} = \mathsf{vdec}_{n+1,q-1}(\mathbf{a}_i|\mathbf{b}_i) \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $\mathbf{v} = (\mathbf{v}_1^T|\mathbf{v}_2^T)^T \in \mathbb{Z}^{2m}$ such that

$$\left[\begin{array}{c|c|c|c} \mathbf{H}_{n,q-1} & \mathbf{F} & & \\ \hline \mathbf{H}_{t,q-1} & \mathbf{P}^T & \mathbf{I}_t\cdot\lfloor q/2\rfloor & \mathbf{I}_t \end{array}\right] \cdot \left[\begin{array}{c} \mathfrak{m} \\ \hline \mathbf{e} \\ \hline \mu \\ \hline \nu \end{array}\right] = \left[\begin{array}{c} \mathbf{c}_0 \\ \hline \mathbf{c}_1 \end{array}\right] \tag{9.40}$$

and

$$\left[\begin{array}{c|c|c|c} \mathbf{A} & \mathbf{A}_0 & \cdots & \mathbf{A}_\ell \end{array}\right] \cdot \left[\begin{array}{c} \mathbf{v}_1 \\ \hline \mathbf{v}_2 \\ \hline \tau[1]\cdot\mathbf{v}_2 \\ \hline \vdots \\ \hline \tau[\ell]\cdot\mathbf{v}_2 \end{array}\right] = \mathbf{u} + \mathbf{D}\cdot\mathfrak{m} \mod q \tag{9.41}$$

2. If the WI argument of step 1 verifies, S_T uses $\mathbf{S} \in \chi^{n\times t}$ to decrypt $(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ and obtain
$$M' = \lfloor (\mathbf{c}_1 - \mathbf{S}^T \cdot \mathbf{c}_0)/(q/2) \rceil \in \{0,1\}^t,$$
which is sent back to R_T. In addition, S_T provides a NIZK argument π_T of knowledge of $\mathbf{y} = \mathbf{c}_1 - \mathbf{S}^T \cdot \mathbf{c}_0 - M' \cdot \lfloor q/2 \rfloor \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ and $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m\times t}$ satisfying (modulo q)
$$\mathbf{P} \;=\; \mathbf{F}^T \cdot \mathbf{S} + \mathbf{E}\,, \qquad \mathbf{c}_0^T \cdot \mathbf{S} + \mathbf{y}^T = \mathbf{c}_1^T - M'^T \cdot \lfloor q/2 \rfloor. \quad (9.42)$$
Given $\mathbf{y} = (\mathbf{y}[1], \ldots, \mathbf{y}[t])^T \in \mathbb{Z}^t$ and $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t]$, this amounts to proving, for each $j \in [t]$, knowledge of $\mathbf{s}_j \in \chi^n$, $\mathbf{y}[j] \in \mathbb{Z}$ such that $|\mathbf{y}[j]| < q/4$ and $\mathbf{e}_j \in \chi^m$, such that
$$\left[\begin{array}{c|c|c} \mathbf{F}^T & \mathbf{I}_m & \\ \hline \mathbf{c}_0^T & & 1 \end{array}\right] \cdot \left(\begin{array}{c} \mathbf{s}_j \\ \hline \mathbf{e}_j \\ \hline \mathbf{y}[j] \end{array}\right) = \left(\begin{array}{c} \mathbf{p}_j \\ \hline \mathbf{c}_1[j] - M'[j] \cdot \lfloor q/2 \rfloor \end{array}\right) \quad \forall j \in [t], \quad (9.43)$$
where $\mathbf{c}_1 = (\mathbf{c}_1[1], \ldots, \mathbf{c}_1[t])^T$ and $M' = (M'[1], \ldots, M'[t])^T$. Let the NIZK argument be $\pi_T = (\{\mathsf{Comm}_{T,j}\}_{j=1}^\varsigma, \mathsf{Chall}_T, \{\mathsf{Resp}_{T,j}\}_{j=1}^\varsigma)$, where $\mathsf{Chall}_T = H_\mathsf{FS}((\mathbf{F}, \mathbf{P}, \mathbf{c}_0, \mathbf{c}_1), \{\mathsf{Comm}_{T,j}\}_{j=1}^\varsigma) \in \{1,2,3\}^\varsigma$.
3. If th argument π_T produced by S_T does not properly verify, R_T halts and outputs $\perp$. Otherwise, R_T recalls the random string $\mu \in \{0,1\}^t$ that was chosen at step 1 and computes $M_{\rho_i} = M' \oplus \mu$. The transfer ends with S_T and R_T outputting $S_i = S_{i-1}$ and $R_i = R_{i-1}$, respectively.

9.6.2 Security

For simplicity, our proofs are given in the single-receiver setting but they readily carry over to the multi-receiver setting, as defined in [GH11, Appendix B].

Theorem 9.7. *The above $\mathcal{OT}_{k\times 1}^N$ protocol provides receiver security under the* SIS *assumption in the random oracle model.*

Proof. We show how to map any real-world cheating sender $\hat{\mathsf{S}}$ to an ideal-world cheating sender $\hat{\mathsf{S}}'$ such that, under the SIS assumption, the distributions $\mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}$ and $\mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}$ under common input $(N, k, M_1, \ldots, M_N, \rho_1, \ldots, \rho_k)$ are computationally indistinguishable.

We consider a sequence of hybrid experiments with binary outputs. In each experiment Exp_i, a distinguisher $\mathcal{D}$ inputs the states (S_k, R_k) produced by $\hat{\mathsf{S}}$ and R' at the end of Exp_i and outputs a bit. We define W_i to be the event that the output of Exp_i is 1. The first experiment outputs whatever the distinguisher $\mathcal{D}$ outputs and corresponds to the real interaction between the cheating sender $\hat{\mathsf{S}}$ and the receiver R.

Exp_0: This experiment involves a real execution of $\hat{\mathsf{S}}$ in interaction with a honest receiver R which queries the index $\rho_i \in [N]$ at the i-th transfer for each $i \in [k]$. The output of Exp_0 is exactly the output of the distinguisher $\mathcal{D}$ on input of $X = (S_k, R_k) \leftarrow \mathbf{Real}_{\mathsf{S},\hat{\mathsf{R}}}$, so that we have
$$\Pr[W_0] = \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}].$$

$\mathbf{Exp}_1$: This experiment is like Exp_0 except that R' programs the random oracle $H_F : \{0,1\}^* \to \mathbb{Z}_q^{n\times m}$ in the following way. It runs the trapdoor generation algorithm $(\mathbf{F}, \mathbf{T}_\mathbf{F}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$ of [AP09] so as to obtain a statistically uniform matrix $\mathbf{F} \in \mathbb{Z}_q^{n\times m}$ and a small-norm $\mathbf{T}_\mathbf{F} \in \mathbb{Z}^{m\times m}$ basis of $\Lambda_q^\perp(\mathbf{F})$. It then programs the random oracle so as to have $H_F(\varepsilon) = \mathbf{F} \in \mathbb{Z}_q^{n\times m}$. Clearly, this change leaves the adversary's view statistically unchanged: we have $|\Pr[W_1] - \Pr[W_0]| \in \mathsf{negl}(\lambda)$.

$\mathbf{Exp}_2$: is as Exp_1 but, at step 5 of the initialization phase, R' uses the short basis $\mathbf{T}_\mathbf{F} \in \mathbb{Z}^{m\times m}$ of $\Lambda_q^\perp(\mathbf{F})$ (which satisfies $\mathbf{F}\cdot\mathbf{T}_\mathbf{F} = \mathbf{0}^n \bmod q$) to extract witnesses $\mathbf{s}_j \in \chi^n, \mathbf{e}_j \in \chi^m$ from the columns $\mathbf{p}_j = \mathbf{F}^T\cdot\mathbf{s}_j + \mathbf{e}_j \in \mathbb{Z}^m$ of the matrix $\mathbf{P} = [\mathbf{p}_1 \mid \ldots \mid \mathbf{p}_t] \in \mathbb{Z}_q^{m\times t}$ for each $j \in [t]$. Note that this can be done by inverting the LWE function (see, e.g., [GKV10, Section 2.3]). At this point, R' aborts the interaction in the event that one of the following conditions holds:

- E.1: The LWE-inversion algorithm fails to compute small-norm vectors $\mathbf{s}_j \in \chi^n$, $\mathbf{e}_j \in \chi^m$ such that $\mathbf{p}_j = \mathbf{F}^T\cdot\mathbf{s}_j + \mathbf{e}_j \in \mathbb{Z}_q^m$ for some $j \in [t]$.
- E.2: The columns of $\mathbf{S} = [\mathbf{s}_1 \mid \ldots \mid \mathbf{s}_t] \in \chi^{n\times t}$ are successfully extracted but there exists $i \in [N]$ such that one of the coordinates of $\mathbf{b}_i - \mathbf{S}^T\cdot\mathbf{a}_i \bmod q$ is neither close to 0 nor $\lfloor q/2\rfloor$ (i.e., the inequalities $|\mathbf{b}_i - \mathbf{S}^T\cdot\mathbf{a}_i \bmod q| > \alpha q$ and $|(\mathbf{b}_i - \mathbf{S}^T\cdot\mathbf{a}_i \bmod q) - \lfloor q/2\rfloor| > \alpha q$ are both satisfied).

In either of the above situations, R' infers that $\hat{\mathsf{S}}$ managed to create a convincing argument for a false statement and aborts the interaction. In such a situation, however, R' can be turned into an algorithm that breaks the binding property of the commitment scheme used in the ZK argument (which contradicts the SIS assumption if the statistically hiding commitment of [KTX08] is used) by replaying the adversary with the same random tape but a different random oracle H_{FS}. According to the General Forking Lemma of [BPVY00], replaying $\hat{\mathsf{S}}$ up to $32\cdot Q_H/(\varepsilon - 3^{-t})$ times (where Q_H is the number of queries to H_{FS} is sufficient to extract a breach in the binding property of the commitment). Otherwise (i.e., if R' does not fail), the matrix $\mathbf{S} \in \chi^{n\times t}$ allows R' to decode the messages $M_1, \ldots, M_N \in \{0,1\}^t$ from the encrypted database $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$. Under the SIS assumption, it follows that Exp_1 returns 1 with about the same probability as Exp_0. In the random oracle model, the SIS assumption thus implies that $|\Pr[W_2] - \Pr[W_1]| \in \mathsf{negl}(\lambda)$.

$\mathbf{Exp}_3$: is identical to Exp_2 except that the receiver R' makes use of the matrix $\mathbf{S} \in \chi^{n\times t}$, which was extracted at step 5 of the initialization phase. At step 2 of each transfer, R' uses $\mathbf{S}$ to determine if the NIZK argument π_T really proves a true statement or if $\hat{\mathsf{S}}$ managed to break its soundness. Namely, upon receiving $\hat{\mathsf{S}}$'s response $M' \in \{0,1\}^t$ at step 2, R' uses the previously extracted $\mathbf{S} \in \chi^{n\times t}$ to determine if there exists $\mathbf{y} \in \mathbb{Z}^t$ of norm $\|\mathbf{y}\|_\infty \leq q/5$ such that

$$\mathbf{c}_0^T\cdot\mathbf{S} + \mathbf{y}^T = \mathbf{c}_1^T - M'^T\cdot\lfloor q/2\rfloor. \tag{9.44}$$

If such vector $\mathbf{y}$ turns out not to exist, R' deduces R' that $\hat{\mathsf{S}}$ was able to fake a convincing argument for a false statement and aborts the interaction. However, R' can then be turned into a PPT adversary against the binding property of the commitment scheme used in the ZK argument (and thus the SIS assumption if the

commitment of [KTX08] is used) by replaying the adversary according to the General Forking technique [BPVY00]. The result of [BPVY00] tells us that replaying $\hat{\mathsf{S}}$ up to $32 \cdot Q_H/(\varepsilon - 3^{-t})$ times (where Q_H is the number of queries to H_{FS}) suffices to break the binding property of the commitment. Under the SIS assumption, we have $|\Pr[W_3] - \Pr[W_2]| \in \mathbf{negl}(\lambda)$.

Exp$_4$: This experiment is like Exp$_3$ with the difference that, at each transfer, the receiver R' chooses the index $\rho_i = 1$ and thus always requests the first message of the encrypted database. In more details, at each transfer, R' samples vectors $\mathbf{e} \hookleftarrow U(\{-1,0,1\}^m)$, $\mu \hookleftarrow U(\{0,1\}^t)$ and $\nu \hookleftarrow U([-B,B]^t)$ to compute and send

$$(\mathbf{c}_0, \mathbf{c}_1) = \big(\mathbf{a}_1 + \mathbf{F} \cdot \mathbf{e},\ \mathbf{b}_1 + \mathbf{P}^T \cdot \mathbf{e} + \mu \cdot \lfloor q/2 \rfloor + \nu\big) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t,$$

which is a re-randomization of $(\mathbf{a}_1, \mathbf{b}_1 + \mu \cdot \lfloor q/2 \rfloor)$. Moreover, R'_{T} uses the witness $\rho_i = 1$ to faithfully generate an interactive WI argument that $(\mathbf{c}_0, \mathbf{c}_1)$ is a re-randomization of $(\mathbf{a}_{\rho_i}, \mathbf{b}_{\rho_i})$. It thus generates a WI argument of knowledge of vectors $\mathfrak{m} = \mathsf{vdec}_{n+t,q-1}(\mathbf{a}_1|\mathbf{b}_1) \in \{0,1\}^{m_d}$, $\mathbf{e} \in \{-1,0,1\}^t$, $\mu \in \{0,1\}^t$, $\nu \in [-B,B]^t$, $\tau \in \{0,1\}^\ell$ and $(\mathbf{v}_1^T|\mathbf{v}_2^T)^T \in \mathbb{Z}^{2m}$ satisfying relations (9.22). By the statistically WI of the interactive argument system, this modification has no noticeable impact on the output distribution of a cheating sender $\hat{\mathsf{S}}$ whatsoever. We have $|\Pr[W_4] - \Pr[W_3]| \in \mathsf{negl}(\lambda)$.

In Exp$_4$, we define the ideal-world cheating sender $\hat{\mathsf{S}}'$ in the following way. It programs the random oracle $H_F : \{0,1\}^* \to \mathbb{Z}_q^{n \times m}$ in such a way that $H_F(\varepsilon) = \mathbf{F} \in \mathbb{Z}_q^{n \times m}$ for some statistically random matrix produced as $(\mathbf{F}, \mathbf{T}_{\mathbf{F}}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. At the initialization phase, $\hat{\mathsf{S}}'$ uses the small-norm basis $\mathbf{T}_{\mathbf{F}}$ of $\Lambda_q^{\perp}(\mathbf{F})$ to extract the small-norm matrices $\mathbf{S} = [\mathbf{s}_1|\ldots|\mathbf{s}_t] \in \chi^{n \times t}$ and $\mathbf{E} = [\mathbf{e}_1|\ldots|\mathbf{e}_t] \in \chi^{m \times t}$ satisfying (9.35) and decrypt $\{(\mathbf{a}_i, \mathbf{b}_i)\}_{i=1}^N$ into messages $M_1, \ldots, M_N \in \{0,1\}^N$. If the extraction fails because one of the events E.1 and E.2 (as defined in Exp$_2$) comes about, $\hat{\mathsf{S}}'$ aborts. Otherwise, it then submits $M_1, \ldots, M_N \in \{0,1\}^N$ to the trusted party TT. As in Exp$_2$, during each transfer phase, $\hat{\mathsf{S}}'$ computes $(\mathbf{c}_0, \mathbf{c}_1)$ as a re-randomization of $(\mathbf{a}_1, \mathbf{b}_1) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t$ and faithfully generates the receiver's argument of knowledge using the witness $\rho_i = 1$ at step 1. At step 2 of each transfer, $\hat{\mathsf{S}}'$ aborts if it realizes that $\hat{\mathsf{S}}$ created a convincing NIZK argument π_T for a false statement. If π_T correctly verifies and indeed relates to a true statement (which $\hat{\mathsf{S}}'$ can detect by applying the test (9.44) using the matrix $\mathbf{S} \in \chi^{n \times t}$ extracted in the initialization phase), $\hat{\mathsf{S}}'$ sends 1 to the trusted party TT so as to authorize the transfer in the ideal world. Otherwise, $\hat{\mathsf{S}}'$ sends 0 to TT. At the end of the k-th transfer phase, $\hat{\mathsf{S}}'$ outputs whatever $\hat{\mathsf{S}}$ outputs as its final state S_k.
In Exp$_4$, it is easy to see that

$$\Pr[W_4] = \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}].$$

Putting the above altogether, we find that the SIS assumption implies such that

$$|\Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Real}_{\hat{\mathsf{S}},\mathsf{R}}] - \Pr[\mathcal{D}(X) = 1 \mid X \leftarrow \mathbf{Ideal}_{\hat{\mathsf{S}}',\mathsf{R}'}]| \in \mathsf{negl}(\lambda).$$

□

The proof of security against a dishonest receiver is almost identical to the proof of Theorem 9.5. The only difference is that, from experience Exp_3 onwards, the sender's ZK arguments are non-interactive and can be simulated by programming the random oracle $H_{\mathsf{FS}} : \{0,1\}^* \to \{1,2,3\}^\varsigma$ in the standard way. The detailed proof of the following theorem is thus omitted.

Theorem 9.8. *The above $\mathcal{OT}^N_{k\times 1}$ protocol provides sender security under the* SIS *assumption in the random oracle model.*

As mentioned in [GH11], extending the simulation-based definitions to the multi-receiver setting is rather straightforward (see [GH11, Appendix B] for details).
Analogously to the Green-Hohenberger protocol [GH11, Section 4], our proof of sender security goes through in the multi-receiver setting as long as the receivers interact with the sender in a sequential manner. This restriction is important since the simulator has to rewind the receiver's zero-knowledge arguments at step 1 of each transfer, which would not be possible in concurrent sessions.

9.7 Comparison of Oblivious Transfer Schemes

Protocol	Initialization Cost	Transfer Cost	Assumptions	Security
Folklore	·	$\mathcal{O}(\lambda N)$	general	Full Sim
NP [NP99]	·	$\mathcal{O}(\lambda \cdot \log(N))$	DDH + OT^2_1	Half Sim
KPN [KPN10]	$\mathcal{O}(\lambda(N \cdot U))$	$\mathcal{O}(\lambda)$	DDH	Full Sim
CNS [CNs07]	$\mathcal{O}(\lambda(N + U))$	$\mathcal{O}(\lambda)$	q-type	Full Sim
GH08 [GH08]	$\mathcal{O}(\lambda(N + U))$	$\mathcal{O}(\lambda)$	DLIN + q-type	UC
JL [JL09]	$\mathcal{O}(\lambda(N + U))$	$\mathcal{O}(\lambda)$	Comp. Dec. Residuosity + q-type	Full Sim
GH11 [GH11]	$\mathcal{O}(\lambda(N + U))$	$\mathcal{O}(\lambda)$	Decision 3-Party DH	Full Sim
GH11 [GH11]	$\mathcal{O}(\lambda N)$	$\mathcal{O}(\lambda)$	3-Party DDH + DLIN	Full Sim
Ours, §9.1	$\mathcal{O}(\lambda(N \cdot U))$	$\mathcal{O}(\lambda \cdot \log N)$	LWE + SIS	Full Sim
Ours, App 9.6	$\mathcal{O}(\lambda N)$	$\mathcal{O}(\lambda \cdot \log N)$	LWE + SIS	Full Sim (ROM)

Table 9.2 – Overview of the different adaptive OT (without access control) protocols secure in the standard model (except for our scheme in Section 9.6 of this Supplementary Material). In this table, λ denotes the security parameter, N the size of the database and U the number of receivers. The horizontal lines separate the different schemes into categories based of their efficiency. We note that, like those of [KPN11], the KPN [KPN10] scheme is secure in a strictly weaker model than ours. In particular, the sender detects if the same record is obtained twice, as pointed out in [GH11].

In this section, we present, in Tables 9.2 and 9.3, comparisons between existing adaptive oblivious transfer protocols and ours. These results are to be taken carefully, as the existing schemes are mostly designed in the pairing-based cryptography setting. The communication complexities thus take into account the number of underlying mathematical objects exchanged during each interactive protocols, which are group elements in the previous constructions, and vectors in our case.

Another remark is that the other schemes which support access control, shown in Table 9.3, manage access policy in the fashion of Camenisch *et al.* [CDN09]. In their work, they model the *access policy* as access categories bounded to users (like their role, or their permission)

which are delivered by the issuer. A given message in the database is made available for a *conjunction* of access categories: meaning that to access a given file, a user has to be in *all* the categories the message in linked to. To handle disjunctions, the file is duplicated. The number of messages in the database N in these schemes is then dependent of the access policy, and a cost for duplications is to take into account, as the database has to prove that encryption of the same message with different access policy is indeed the encryption of the same message.

By handling access control through branching programs, we avoid the hidden cost of disjunctions, while enabling access control for attribute's language in $\mathsf{NC1}$.

Protocol	Initialization Cost	Transfer Cost	Assumptions	Policies	Private Policies	Security
CDN [CDN09]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \mathsf{poly}(\lambda)$	q-type	Conj.	✗	Full Sim
CDNZ [CDNZ11]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \mathsf{poly}(\lambda)$	q-type + XDDH	Conj.	✓	Full Sim
ACDN [ACDN13]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda) \cdot \mathsf{poly}(\lambda)$	DLIN + SXDH	Conj.	✗	UC
ZAW+ [ZAW+10]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda)$	CP-ABE + q-type	$\mathsf{NC1}$	✗	Full-Sim
CDEN [CDEN12]	$\mathcal{O}(\lambda \cdot N)$	$\mathcal{O}(\lambda \log N) + \mathsf{poly}(\lambda)$	CP-ABE + GGM	CNF^-	✓	Full-Sim
Ours, §9.4	$\mathcal{O}(\lambda \cdot N)$	$\widetilde{\mathcal{O}}(\lambda \log N) + \mathsf{poly}(\lambda)$	LWE + SIS	$\mathsf{NC1}$	✗	Full Sim

Table 9.3 – Overview of the different adaptive OT-AC protocols secure in the standard model. Here N denotes the size of the database. The polynomial $\mathsf{poly}(\lambda)$ in transfer costs captures the expense of access policies. In CDEN, GGM stands for generic group model, and CNF^- means a restricted version of conjunctive normal form formulas, namely a user has to possess *all* attributes in its access credentials, and to do so, it is able to provides a disjunction of its accesses. Finally "Conj." means "Conjunctions", meaning that the user has to possess all the credential for a given message, and disjunctions can be achieved at the expense of duplications of database entries.

Conclusion

In this thesis, we presented new cryptographic schemes that rely on lattice or pairing assumptions. These contributions focus on the design and the analysis of new cryptographic schemes that target privacy-preserving applications.

In pairing-based cryptography, we proposed a practical dynamic group signature scheme, whose security relies on well-understood assumptions in the random oracle. It relies on widely used assumptions with simple and constant-size descriptions which have been studied for more than ten years. This work is also supported by an implementation in `C`.

The results in the lattice setting gave rise to three realizations of fundamental primitives that were missing in the landscape of lattice-based privacy-preserving cryptography. Even if these schemes suffer from a lack of efficiency due to their novelty, we do believe that they take one step towards a quantum-secure privacy-friendly world.

On the road, improvements have been made in the state of the art of zero-knowledge proofs in the lattice setting by providing building blocks that, we believe, are of independent interest. For example, our signature with efficient protocols has already been used to design a privacy-preserving lattice-based e-cash system [LLNW17].

All these works are proven to satisfy strong security models under simple assumptions. This provides a breeding ground for new theoretical constructions.

Open Problems

The path of providing new cryptographic primitives and proving them secure is full of pitfalls. The most obvious question that stems from this work is how to tackle the trade-offs we made in the design of those primitives. In particular, the specific question naturally arise:

Question 1. *Is it possible to build a fully-simulatable adaptive oblivious transfer (even without access control) secure under* LWE *with polynomially large modulus?*

In other words, is it possible to avoid the use of noise flooding to guarantee receiver-security in the adaptive oblivious transfer scheme of Chapter 9. In our current protocol, this issue arises from the use of Regev's encryption scheme, where we need to prevent the noise distribution from leaking the receiver's index. However, while a finer analysis of the noise in

GSW ciphertexts [GSW13] seems promising to achieve this at reasonable cost [BdPMW16], it is not sufficient in our setting because it would leak the norm of the noise vector of ciphertexts. Then, another difficulty is to have zero-knowledge proofs compatible with the access control and the encryption components.

Question 2. *Can we construct provably-secure adaptive oblivious transfer schemes in the universal composability model?*

Our adaptive oblivious transfer scheme relies on zero-knowledge proofs to hedge against malicious adversaries. The security proofs take advantage of the fact that the proofs can be rewound to extract a witness (as described in Definition 4.2). The Peikert-Vaikuntanathan-Waters [PVW08] construction, based on dual-mode encryption, achieves 1-out-of-2 composable oblivious transfer (which can be generalized to 1-out-of-2^t OT), without relying on zero-knowledge proofs, but it does not imply OT with adaptive queries (i.e., where each index ρ_i may depend on messages received in previous transfers). Actually, the use of ZK proofs is not ruled out in this setting, as shown by the pairing-based construction of Green and Hohenberger [GH08]. However, this protocol uses the trapdoor extractability of Groth-Sahai proofs [GS08] to achieve straight-line extraction. It is not known to be possible in the lattice setting.

Question 3. *Can we obtain a more efficient compact e-cash system from lattice assumptions?*

Another privacy-preserving primitive is compact e-cash [Cha82, Cha83, CHL05b]. As explained in the introduction, it is the digital equivalent of real-life money. A body of research followed its introduction [CFN88, OO91, CP92, FY93, Oka95, Tsi97], and the first compact realization was given by Camenisch, Hohenberger and Lysyanskaya [CHL05b] (here, "compact" means that the complexity of coin transfers is at most logarithmic in the value of withdrawn wallets). Before the work of Libert, Ling, Nguyen and Wang [LLNW17], all compact constructions were based on traditional number-theoretic techniques. This construction still suffers from efficiency issues akin to the problem we met in this thesis. It is thus interesting to improve the efficiency of this scheme and obtain viable constructions of anonymous e-cash from post-quantum assumptions.

Zero-Knowledge Proofs

Question 4. *Can we provide NIZK proofs in the standard model for all NP languages while relying on the standard* LWE *assumption only?*

Extending the work of Groth, Ostrovsky and Sahai [GOS06] to the lattice setting would be a breakthrough result for lattice-based cryptography in general. This question remains open for more than 10 years [PV08]. A recent line of work makes steps forward in this direction [KW18, RSS18], but they rely on primitives that do not exist yet [RSS18] (NIZK proofs for a variant of the bounded decoding distance problem) or assume pre-processing [KW18].

The Stern-like proof systems we studied in this thesis, despite being flexible enough to prove a large variety of statements, suffer from the stiffness of being combinatorial. The choice of permutations used to ensure the zero-knowledge property (and thus witness-indistinguishability) is quite strict, and forces the challenge space to be ternary. This turns out to be a real bottleneck in the efficiency of such proof systems.

Question 5. *Can we get negligible soundness error in one shot for expressive statements in the post-quantum setting?*

This question can be restated as "can we combine the expressiveness of Stern-like proofs with the efficiency of Schnorr-like proof with rejection sampling?". For Stern-like protocols, decreasing the soundness error from $2/3$ to $1/2$ would already be an interesting improvements with a direct impact on the efficiency of all lattice-based schemes presented in this thesis. Recall that the *soundness error* is the probability that a cheating prover convinces an honest verifier of a false statement. As long as it is noticeably different from 1, it is possible to make the soundness error negligible by repeating the protocol a sufficient number of times. Likewise, isogeny-based proof systems [JDF11, GPS17] suffer from similar issues as the challenge space is small (binary). The $2/3$ soundness error is also present in [IKOS07], which is a technique to obtain zero-knowledge proofs relying on secure multi-party computation. With this technique, however, the size of the proof is proportional to the size of the circuit describing the relation we want to prove (which is not the case with Stern-like protocols). Thus, the question of having efficient post-quantum zero-knowledge proofs for expressive statements is a difficult question and remains open as of today.

Cryptographic Constructions

Question 6. *Can we construct more efficient lattice-based signature schemes compatible with zero-knowledge proofs?*

In the general lattice setting, the most efficient signature schemes require at least as many matrices as the length ℓ of the random tag used in the signature (like the scheme in Section 7.1). This cost has direct impact on the efficiency and public-key size of schemes or protocols that use them: in our group signatures of Chapter 7, for example, ℓ is logarithmic in the maximal number of members the group can accept N_{gs}. In ideal lattices, it is possible to reduce this cost to a vector of size ℓ [DM14]. In the group signature scheme of [LNWX18], which is based on ideal lattice problems, they use this property to allow an exponential number of group members to join the group, and thus propose a "constant-size" group signature scheme. The method used to construct this group signature is essentially the same as in Chapter 7, where matrices are hidden in the ring structure of the ideal lattice [LS14]. In the construction of [LNWX18], the dependency on $\log N_{\mathsf{gs}}$ is actually hidden in the dimension of the ring. As these signatures are a fundamental building block for privacy-preserving cryptography, any improvement on them has a direct impact on the primitives or protocols that use them as a building block.

Question 7. *Can we obtain more efficient lattice-based one-time signatures in general lattices?*

In our group signature and group encryption schemes (in Chapter 7 and Chapter 8 respectively), signature and ciphertext contain a public key for a one-time signature scheme. One efficiency issue is that, in lattice-based one-time signatures [LM08, Moh11], the public-key contains a full matrix, that is part of the signature/ciphertext. Therefore, this matrix significantly increase the size of the signature/ciphertext. As security requirements for one-time signature are weaker than those of full-fledged signatures (namely, the adversary has access to only one signature per public key), we can hope for more efficient constructions

of one-time signatures based on general lattices where, the public-key is smaller that a full-matrix.

As we explained in the introduction, advanced cryptography from lattices often suffers from the use of lattice trapdoors. Thus, a natural question may be:

Question 8. *Does an efficient trapdoor-free (H)IBE exist?*

In the group encryption scheme of Chapter 8, for instance, trapdoors are used for two distinct purposes. They are used to build a secure public-key encryption scheme under adaptive chosen-ciphertext attacks and a signature scheme. These primitives are both induced by identity-based encryption: the Canetti-Halevi-Katz transform generically turns an IBE into a IND-CCA2 PKE [CHK04], and signatures are directly implied by IND-CPA-secure IBE [BF01, BLS01]. Actually, a recent construction due to Brakerski, Lombardi, Segev and Vaikuntanathan [BLSV18] (inspired by [DG17a]) gives a candidate which relies on garbled circuits, and is fairly inefficient compared to IBE schemes with trapdoors. Even the question of a trapdoor-less IND-CCA2 public key encryption still does not have a satisfactory solution. The construction of Peikert and Waters [PW08] is trapdoor-free, but remains very expensive.

Bibliography

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 1st edition, 2009. Citations: § 13

[ABB10] S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Eurocrypt*, volume 6110 of *LNCS*, pages 553–572. Springer, 2010. Citations: § xviii, 4, 26, 86, 90, 101, 102, 124, 125, 130, 131, 132, 134, 137, 138, 142, and 154

[ACDN13] M. Abe, J. Camenisch, M. Dubovitskaya, and R. Nishimaki. Universally composable adaptive oblivious transfer (with access control) from standard assumptions. In *ACM Workshop on Digital Identity Management*, pages 1–12, 2013. Citations: § xxii, 7, 147, and 187

[ACJT00] G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *Crypto*, volume 1880 of *LNCS*, pages 255–270. Springer, Springer, 2000. Citations: § 43

[ADRSD15] D. Aggarwal, D. Dadush, O. Regev, and N. Stephens-Davidowitz. Solving the Shortest Vector Problem in 2^n Time Using Discrete Gaussian Sampling. In *STOC*, pages 733–742. ACM, 2015. Citations: § 22

[ADSD15] D. Aggarwal, D. Dadush, and N. Stephens-Davidowitz. Solving the Closest Vector Problem in 2^n Time — The Discrete Gaussian Strikes Again! In *FOCS*. ACM, 2015. Citations: § 22

[AFG+10] M. Abe, G. Fuchsbauer, J. Groth, K. Haralambiev, and M. Ohkubo. Structure-preserving signatures and commitments to group elements. In *Crypto*, volume 6223 of *LNCS*, pages 209–236. Springer, 2010. Citations: § 96

[AFG14] M. R. Albrecht, R. Fitzpatrick, and F. Göpfert. On the Efficacy of Solving LWE by Reduction to Unique-SVP. In *ICISC 2013*, pages 293–310, Cham, 2014. Springer. Citations: § 22

[AG] D. F. Aranha and C. P. L. Gouvêa. RELIC is an Efficient LIbrary for Cryptography. https://github.com/relic-toolkit/relic. Citations: § 75

[AIR01] W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In *Eurocrypt*, pages 119–135, 2001. Citations: § 146 and 147

[AJLA+12] G. Asharov, A. Jain, A. Lopez-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Eurocrypt*, volume 7237 of *LNCS*, pages 483–501. Springer, 2012. Citations: § 90 and 158

[Ajt96] M. Ajtai. Generating Hard Instances of Lattice Problems. In ACM, editor, *STOC*, pages 99–108, 1996. Citations: § 23

[AMBB+13] C. Aguilar-Melchor, S. Bettaieb, X. Boyen, L. Fousse, and P. Gaborit. Adapting Lyubashevsky's Signature Schemes to the Ring Signature Setting. In *Africacrypt*, volume 7918 of *LNCS*, pages 1–25. Springer, 2013. Citations: § 118

[AP09] J. Alwen and C. Peikert. Generating shorter bases for hard random lattices. In *STACS*, volume 3 of *LNCS*, pages 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009. Citations: § 25 and 184

[Ban93] W. Banaszczyk. New bounds in some transference theorems in the geometry of number. 296:625–635, 1993. Citations: § 23

[Bar86] D. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. In *STOC'86*, pages 1–5, 1986. Citations: § 146

[BB04] D. Boneh and X. Boyen. Efficient selective-ID secure identity-based encryption without random oracles. In *Eurocrypt*, volume 3027 of *LNCS*, pages 223–238. Springer, 2004. Citations: § xix, 4, 15, 26, 52, and 53

[BBDP01] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-Privacy in Public-Key Encryption. In *PKC*, LNCS, pages 566–582. Springer, 2001. Citations: § 131

[BBKS07] M. Bellare, A. Boldyreva, K. Kurosawa, and J. Staddon. Multirecipient Encryption Schemes: How to Save on Bandwidth and Computation Without Sacrificing Security. *IEEE Trans. on Information Theory*, 53(11):3927–3943, 2007. Citations: § 66

[BBS04] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Crypto*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004. Citations: § xviii, 4, 15, 21, 51, 52, 53, and 74

[BCC+09] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable Proofs and Delegatable Anonymous Credentials. In *Crypto*, volume 5677 of *LNCS*, pages 108–125. Springer, 2009. Citations: § xviii and 4

[BCC+16] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, and J. Groth. Foundations of Fully Dynamic Group Signatures. In *ACNS*, LNCS, pages 117–136. Springer, 2016. Citations: § 44

[BCK+14] F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *Asiacrypt*, number 8873 in LNCS, pages 551–572, 2014. Citations: § 118

[BCKL08] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. P-signatures and Noninteractive Anonymous Credentials. In *TCC*, number 4948 in LNCS, pages 356–374. Springer, 2008. Citations: § xviii, 4, and 54

[BCKL09] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya. Compact E-Cash and Simulatable VRFs Revisited. In *Pairing*, volume 5671 of *LNCS*, pages 114–131. Springer, 2009. Citations: § xviii and 4

[BCN+10] P. Bichsel, J. Camenisch, G. Neven, N. P. Smart, and B. Warinschi. Get Shorty via Group Signatures without Encryption. In *SCN*, LNCS, pages 381–398. Springer, 2010. Citations: § 75

[BD18] R. Barbulescu and S. Duquesne. Updating Key Size Estimations for Pairings. *Journal of Cryptology*, pages 1–39, 2018. Citations: § xix, 4, and 21

[BdPMW16] F. Bourse, R. del Pino, M. Minelli, and H. Wee. FHE Circuit Privacy Almost for Free. In *Crypto*, number 9815 in LNCS, pages 62–89, 2016. Citations: § 158 and 190

[BF01] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In J. Kilian, editor, *Crypto*, pages 213–229. Springer, 2001. Citations: § 21 and 192

[BG92] M. Bellare and O. Goldreich. On Defining Proofs of Knowledge. In *Crypto*, volume 740 of *LNCS*, pages 390–420. Springer, 1992. Citations: § 28

[BGdMM05] L. Ballard, M. Green, B. de Medeiros, and F. Monrose. Correlation-resistant storage via keyword-searchable encryption. Cryptology ePrint Archive, Report 2005/417, 2005. https://ia.cr/2005/417. Citations: § 22 and 53

[BHJ+15] F. Böhl, D. Hofheinz, T. Jager, J. Koch, and C. Striecks. Confined guessing: New signatures from standard assumptions. *Journal of Cryptology*, 28(1):176–208, 2015. Citations: § xviii, 3, 5, 26, 77, 79, 82, 90, 131, 152, and 153

[BKLP15] F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS*, volume 9326 of *LNCS*, pages 305–325. Springer, 2015. to appear. Citations: § 118

[BLL+15] S. Bai, A. Langlois, T. Lepoint, D. Stehlé, and R. Steinfeld. Improved security proofs in lattice-based cryptography: Using the Rényi divergence rather than the statistical distance. In *Asiacrypt*, volume 9452 of *LNCS*. Springer, 2015. Citations: § 18, 78, 81, 90, and 91

[BLP+13] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé. On the classical hardness of learning with errors. In *STOC*, pages 575–584. ACM, 2013. Citations: § 25

[BLS01] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Asiacrypt*, LNCS, pages 514–532. Springer, 2001. Citations: § 192

[BLSV18] Z. Brakerski, A. Lombardi, G. Segev, and V. Vaikuntanathan. Anonymous IBE, Leakage Resilience and Circular Security from New Assumptions. In *Eurocrypt*, LNCS, pages 535–564. Springer, 2018. Citations: § 192

[Blu81] M. Blum. Coin Flipping by Telephone. In *Crypto*, pages 11–15. Springer, 1981. Citations: § 29

[BMW03] M. Bellare, D. Micciancio, and B. Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *Eurocrypt*, volume 3376 of *LNCS*, pages 614–629. Springer, 2003. Citations: § 43 and 44

[BN06] P. S. L. M. Barreto and M. Naehrig. Pairing-friendly elliptic curves of prime order. In B. Preneel and S. Tavares, editors, *Selected Areas in Cryptography*, pages 319–331. Springer, 2006. Citations: § xviii, 4, 21, 71, 72, 74, and 75

[Boy10] X. Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In *PKC*, volume 6056 of *LNCS*, pages 499–517. Springer, 2010. Citations: § xviii, xix, 3, 5, 26, 77, 82, 96, and 154

[BPVY00] E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung. Design validations for discrete logarithm based signature schemes. In *PKC*, volume 1751 of *LNCS*, pages 276–292. Springer, 2000. Citations: § 103, 106, 107, 184, and 185

[BR93] M. Bellare and P. Rogaway. Random Oracles Are Practical: A Paradigm for Designing Efficient Protocols. In *CCS*. ACM, 1993. Citations: § 16, 31, and 51

[BSS05] I. F. Blake, G. Seroussi, and N. P. Smart. *Advances in elliptic curve cryptography*, volume 317. Cambridge University Press, 2005. Citations: § 21 and 53

[BSZ05] M. Bellare, H. Shi, and C. Zhang. Foundations of group signatures: The case of dynamic groups. In *CT-RSA*, volume 2656 of *LNCS*, pages 136–153. Springer, 2005. Citations: § 43, 44, 46, 65, 96, and 99

[BV11] Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In *FOCS*, pages 97–106, 2011. Citations: § xviii, xix, and 4

[Can01] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001. Citations: § 11, 19, 147, and 148

[CDEN12] J. Camenisch, M. Dubovitskaya, R. Enderlein, and G. Neven. Oblivious transfer with hidden access control from attribute-based encryption. In *SCN*, volume 7485 of *LNCS*, pages 559–579, 2012. Citations: § 146, 147, and 187

[CDM00] R. Cramer, I. Damgård, and P. MacKenzie. Efficient Zero-Knowledge Proofs of Knowledge Without Intractability Assumptions. In *PKC*, pages 354–372, 2000. Citations: § 67

[CDN09] J. Camenisch, M. Dubovitskaya, and G. Neven. Oblivious transfer with access control. In *ACM-CCS*, pages 131–140, 2009. Citations: § xxii, 7, 27, 146, 147, 148, 150, 152, 180, 186, and 187

[CDNZ11] J. Camenisch, M. Dubovitskaya, G. Neven, and G. Zaverucha. Oblivious transfer with hidden access control policies. In *PKC'11*, volume 6571 of *LNCS*, pages 192–209, 2011. Citations: § xxii, 7, 146, 147, and 187

[CF01] R. Canetti and M. Fischlin. Universally composable commitments. In *Crypto*, pages 19–40. Springer, 2001. Citations: § 20

[CFN88] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Crypto*, volume 403 of *LNCS*, pages 319–327, 1988. Citations: § xv, 1, and 190

[CGH98] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *STOC*, volume 45. ACM, 1998. Citations: § 16

[CGH09] S. Coull, M. Green, and S. Hohenberger. Controlling access to an oblivious database using stateful anonymous credentials. In *PKC*, number 5443 in LNCS, 2009. Citations: § 147

[CGKS95] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *FOCS*, pages 41–50, 1995. Citations: § 146

[Cha82] D. Chaum. Blind signatures for untraceable payments. In *Crypto*, LNCS, pages 199–203, 1982. Citations: § xvi, 2, and 190

[Cha83] D. Chaum. Blind signature system. In *Crypto*, LNCS, page 153, 1983. Citations: § 190

[Cha85] D. Chaum. Security without Identification: Transactions System to Make Big Brother Obsolete. 28(10):1030–1044, 1985. Citations: § xvi, 2, 27, and 51

[Che06] J. H. Cheon. Security analysis of the strong diffie-hellman problem. In *Eurocrypt*, volume 4004 of *LNCS*. Springer, 2006. Citations: § 15 and 22

[CHK04] R. Canetti, S. Halevi, and J. Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In *Eurocrypt*, LNCS, pages 207–222. Springer, 2004. Citations: § 77, 130, and 192

[CHKP10] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Eurocrypt*, volume 6110 of *LNCS*, pages 523–552. Springer, 2010. Citations: § xix, 4, 26, and 80

[CHL05a] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Balancing Accountability and Privacy Using E-Cash. In *SCN*, number 4116 in LNCS, pages 141–155, 2005. Citations: § 53

[CHL05b] J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *Eurocrypt*, number 3494 in LNCS, pages 302–321. Springer, 2005. Citations: § 27 and 190

[CHL+15] J. H. Cheon, K. Han, C. Lee, H. Ryu, and D. Stehlé. Cryptanalysis of the Multilinear Map over the Integers. In *Eurocrypt*, 2015. Citations: § xvii and 2

[CKL06] R. Canetti, E. Kushilevitz, and Y. Lindell. On the limitations of universally composable two-party computation without set-up assumptions. *Journal of Cryptology*, 19(2):135–167, 2006. Citations: § 11

[CKL+15] J. Camenisch, S. Krenn, A. Lehmann, G.-L. Mikkelsen, G. Neven, and M.-. Pedersen. Formal treatment of privacy-enhancing credential systems. In *SAC*, LNCS, pages 3–24. Springer, 2015. Citations: § 87 and 90

[CL01] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Eurocrypt*, number 2045 in LNCS, pages 93–118. Springer, 2001. Citations: § xvi, 2, 27, and 51

[CL02a] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *Security and Cryptography for Networks (SCN'02)*, number 2576 in LNCS, pages 268–289, 2002. Citations: § 53, 65, 78, and 79

[CL02b] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. In *SCN*, number 2576 in LNCS, pages 268–289. Springer, 2002. Citations: § 78 and 118

[CL04a] J. Camenisch and A. Lysyanskaya. A Signature Scheme with Efficient Protocols. In *SCN*, LNCS, pages 268–289. Springer, 2004. Citations: § 51

[CL04b] J. Camenisch and A. Lysyanskaya. Signature Schemes and Anonymous Credentials from Bilinear Maps. In *Crypto*, number 3152 in LNCS, pages 56–72. Springer, 2004. Citations: § 51 and 52

[CLY09] J. Cathalo, B. Libert, and M. Yung. Group Encryption: Non-Interactive Realization in the Standard Model. In *Asiacrypt*, number 5912 in LNCS, pages 179–196. Springer, 2009. Citations: § 119 and 121

[CNR12] J. Camenisch, G. Neven, and M. Rückert. Fully anonymous attribute tokens from lattices. In *SCN*, volume 7485 of *LNCS*, pages 57–75. Springer, 2012. Citations: § 78

[CNs07] J. Camenisch, G. Neven, and a. shelat. Simulatable adaptive oblivious transfer. In *Eurocrypt*, volume 4515 of *LNCS*, pages 573–590, 2007. Citations: § 145, 146, 147, 148, 158, 180, and 186

[Coo71] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158. ACM, 1971. Citations: § 11

[CP92] D. Chaum and T. Pedersen. Transferred Cash Grows in Size. In *Eurocrypt*, volume 658 of *LNCS*, pages 390–407, 1992. Citations: § 190

[Cra96] R. Cramer. *Modular Design of Secure, yet Practical Cryptographic Protocols*. PhD thesis, University of Amsterdam, 1996. Citations: § 29

[CS98] R. Cramer and V. Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In *Crypto*, pages 13–25, 1998. Citations: § 66

[CvH91] D. Chaum and E. van Heyst. Group signatures. In *Eurocrypt*, volume 547 of *LNCS*, pages 257–265. Springer, Springer, 1991. Citations: § 43 and 117

[Dam89] I. Damgård. A Design Principle for Hash Functions. In *Crypto*, pages 416–427. Springer, 1989. Citations: § 30

[Dam00] I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Eurocrypt*, volume 1807 of *LNCS*, pages 418–430. Springer, 2000. Citations: § 28, 89, 95, 138, and 141

[DCOR99] G. Di Crescenzo, R. Ostrovsky, and S. Rajagopalan. Conditional oblivious transfer and timed-release encryption. In *Eurocrypt'99*, number 1592 in LNCS, pages 74–89, 1999. Citations: § 147

[DFKS16] N. Döttling, N. Fleischhacker, J. Krupp, and D. Schröder. Two-message, oblivious evaluation of cryptographic functionalities. In *Crypto*, number 9816 in LNCS, pages 619–648, 2016. Citations: § 147

[DG17a] N. Döttling and S. Garg. From Selective IBE to Full IBE and Selective HIBE. In *TCC*, LNCS, pages 372–408. Springer, 2017. Citations: § xix and 192

[DG17b] N. Döttling and S. Garg. Identity-Based Encryption from the Diffie-Hellman Assumption. In *Crypto*, volume 10401 of *LNCS*, pages 537–569. Springer, 2017. Citations: § 4

[DHKT08] I. Damgård, D. Hofheinz, E. Kiltz, and R. Thorbek. Public-key encryption with non-interactive opening. In *CT-RSA*, volume 4964 of *LNCS*, pages 239–255. Springer, 2008. Citations: § 99

[DM14] L. Ducas and D. Micciancio. Improved Short Lattice Signatures in the Standard Model. In *Crypto*, LNCS, pages 335–352. Springer, 2014. Citations: § 191

[DN03] I. Damgård and J.-B. Nielsen. Universally composable efficient multiparty computation from threshold homomorphic encryption. In *Crypto*, number 2729 in LNCS, pages 247–264, 2003. Citations: § 146

[DP06] C. Delerablée and D. Pointcheval. Dynamic fully anonymous short group signatures. In *VietCrypt*, volume 4341 of *LNCS*, pages 193–210. Springer, 2006. Citations: § 52, 53, 66, 71, and 75

[dPLNS17] R. del Pino, V. Lyubashevsky, G. Neven, and G. Seiler. Practical Quantum-Safe Voting from Lattices. In *CCS*, 2017. Citations: § xviii, 4, and 35

[DS16] L. Ducas and D. Stehlé. Sanitization of FHE ciphertexts. In *Eurocrypt*, LNCS, 2016. Cryptology ePrint Archive: Report 2016/164. Citations: § 158 and 161

[EAJ13] L. El Aimani and M. Joye. Toward Practical Group Encryption. In *ACNS 2013*, volume 7954 of *LNCS*, pages 237–252. Springer, 2013. Citations: § 119

[EGL85] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. 28(6):637–647, 1985. Citations: § xxi, 6, and 145

[ELL+15] M. F. Ezerman, H. T. Lee, S. Ling, K. Nguyen, and H. Wang. A provably secure group signature scheme from code-based assumptions. In *Asiacrypt'15*, volume 9452 of *LNCS*, pages 260–285. Springer, 2015. Citations: § 128

[FIPR05] M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold. Keyword search and oblivious pseudorandom functions. In *TCC*, volume 3378 of *LNCS*, pages 303–324, 2005. Citations: § 147

[Fre10] D. M. Freeman. Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups. In *Eurocrypt*, LNCS, pages 44–61. Springer, 2010. Citations: § 52

[FS86] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Crypto*, pages 186–194. Springer, 1986. Citations: § 16, 27, 31, 66, 89, 98, and 160

[FS90] U. Feige and A. Shamir. Witness Indistinguishable and Witness Hiding Protocols. In ACM, editor, *STOC*, pages 416–426, 1990. Citations: § 28 and 34

[FY93] M. Franklin and M. Yung. Secure and efficient off-line digital money. In *ICALP*, volume 700 of *LNCS*, pages 265–276. Springer, 1993. Citations: § 190

[Gen09] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. ACM, 2009. Citations: § xix, 4, and 22

[GH07] M. Green and S. Hohenberger. Blind identity-based encryption and simulatable oblivious transfer. In *Asiacrypt*, volume 4833 of *LNCS*, pages 265–282, 2007. Citations: § 145, 147, and 180

[GH08] M. Green and S. Hohenberger. Universally Composable Adaptive Oblivious Transfer. In *Asiacrypt*, number 5350 in LNCS, pages 179–197. Springer, 2008. Citations: § 145, 147, 186, and 190

[GH11] M. Green and S. Hohenberger. Practical adaptive oblivious transfer from simple assumptions. In *TCC*, volume 6597 of *LNCS*, pages 347–363, 2011. Citations: § 145, 147, 180, 183, and 186

[Gil77] J. Gill. Computational Complexity of Probabilistic Turing Machines. *SIAM J. on Computing*, 6(4):675–695, 1977. Citations: § 13

[GKV10] S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Asiacrypt*, volume 2647 of *LNCS*, pages 395–412. Springer, 2010. Citations: § 43, 78, 118, and 184

[GLOW12] M. Gerbush, A. Lewko, A. O'Neill, and B. Waters. Dual Form Signatures: An Approach for Proving Security from Static Assumptions. In *Asiacrypt*, LNCS, pages 25–42. Springer, 2012. Citations: § 52

[GM82] S. Goldwasser and S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *STOC*, pages 365–377. ACM, 1982. Citations: § xvi, 2, 17, and 19

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC*, pages 291–304. ACM, ACM, 1985. Citations: § 27, 28, and 118

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In *STOC*, pages 218–229, 1987. Citations: § 145

[GMY03] J. Garay, P. MacKenzie, and K. Yang. Strengthening Zero-Knowledge Protocols Using Signatures. In *Eurocrypt*, LNCS, pages 177–194. Springer, 2003. Citations: § 28

[Gol04] O. Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004. Citations: § 19

[GOS06] J. Groth, R. Ostrovsky, and A. Sahai. Perfect Non-interactive Zero Knowledge for NP. In *Eurocrypt*, 2006. Citations: § xvii, xviii, 3, 4, 32, and 190

[GPS17] S. D. Galbraith, C. Petit, and J. Silva. Identification Protocols and Signature Schemes Based on Supersingular Isogeny Problems. In *Asiacrypt*, LNCS, pages 3–33. Springer, 2017. Citations: § 191

[GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. ACM, 2008. Citations: § xviii, xix, 4, 22, 24, 25, 77, 87, 99, 108, 131, and 132

[Gro07] J. Groth. Fully anonymous group signatures without random oracles. In *Asiacrypt*, volume 4833 of *LNCS*, pages 164–180. Springer, 2007. Citations: § xviii and 4

[GS08] J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008. Citations: § xvii, xviii, 3, 4, 32, 118, and 190

[GSW13] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Crypto*, number 8042 in LNCS, pages 75–92, 2013. Citations: § xviii, xix, 4, 22, 158, and 190

[HAO15] R. Hiromasa, M. Abe, and T. Okamoto. Packing messages and optimizing bootstrapping in GSW-FHE. In *PKC*, number 9020 in LNCS, pages 699–715, 2015. Citations: § 158

[Her11] J. Herranz. Restricted adaptive oblivious transfer. *Theoretical Computer Science*, 412(46):6498–6506, 2011. Citations: § 147

[HK17] G. Herold and E. Kirshanova. Improved algorithms for the approximate k-list problem in Euclidean norm. In *PKC'17*, pages 16–40. Springer, 2017. Citations: § 22

[HW09] S. Hohenberger and B. Waters. Short and stateless signatures from the RSA assumption. In *Crypto*, volume 5677 of *LNCS*, pages 654–670. Springer, 2009. Citations: § 82 and 154

[IKOS07] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge from Secure Multiparty Computation. In *STOC*, pages 21–30. ACM, 2007. Citations: § 191

[IPV10] M. Izabachène, D. Pointcheval, and D. Vergnaud. Mediated traceable anonymous encryption. In *LATINCRYPT 2010*, volume 6212 of *LNCS*, pages 40–60. Springer, 2010. Citations: § 119

[JDF11] D. Jao and L. De Feo. Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies. In *PQCrypto*, LNCS, pages 19–34. Springer, 2011. Citations: § 191

[JKPT12] A. Jain, S. Krenn, K. Pietrzak, and A. Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *Asiacrypt*, volume 7658 of *LNCS*, pages 663–680. Springer, 2012. Citations: § 118

[JL09] S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In *TCC*, volume 5444 of *LNCS*, pages 577–594, 2009. Citations: § 147, 180, and 186

[Jou00] A. Joux. A one round protocol for tripartite diffie–hellman. In W. Bosma, editor, *Algorithmic Number Theory*, pages 385–393. Springer, 2000. Citations: § xviii and 4

[JR13] C. Jutla and A. Roy. Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In *Asiacrypt*, LNCS, pages 1–20. Springer, 2013. Citations: § 32, 52, and 54

[JR14] C. Jutla and A. Roy. Switching Lemma for Bilinear Tests and Constant-Size NIZK Proofs for Linear Subspaces. In *Crypto*, volume 8617 of *LNCS*, pages 295–312. Springer, 2014. Citations: § 52 and 53

[KB16] T. Kim and R. Barbulescu. Extended tower number field sieve: A new complexity for the medium prime case. In M. Robshaw and J. Katz, editors, *Crypto*, pages 543–571. Springer, 2016. Citations: § xix, 4, and 21

[KL07] J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007. Citations: § 15 and 16

[KPN10] K. Kurosawa, L. Phong, and R. Nojima. Efficiency-improved fully simulatable adaptive OT under the DDH assumption. In *SCN*, volume 6280 of *LNCS*, pages 172–181, 2010. Citations: § 147 and 186

[KPN11] K. Kurosawa, L. Phong, and R. Nojima. Generic fully simulatable adaptive oblivious transfer. In *ACNS*, volume 6715 of *LNCS*, pages 274–291, 2011. Citations: § 147 and 186

[KSS08] E. J. Kachisa, E. F. Schaefer, and M. Scott. Constructing brezing-weng pairing-friendly elliptic curves using elements in the cyclotomic field. In S. D. Galbraith and K. G. Paterson, editors, *Pairing-Based Cryptography – Pairing*, pages 126–135. Springer, 2008. Citations: § 21

[KTX08] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Asiacrypt*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008. Citations: § xvii, xviii, 3, 30, 31, 35, 36, 38, 78, 79, 81, 87, 118, 132, 144, 160, 162, 174, 184, and 185

[KTY04] A. Kiayias, Y. Tsiounis, and M. Yung. Traceable signatures. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 571–589. Springer, 2004. Citations: § 119

[KTY07] A. Kiayias, Y. Tsiounis, and M. Yung. Group encryption. In *Asiacrypt*, number 4833 in LNCS, pages 181–199. Springer, 2007. Citations: § xx, xxi, 6, 117, 118, 119, 120, 121, 123, 131, and 132

[KW03] J. Katz and N. Wang. Efficiency improvements for signature schemes with tight security reductions. In *CCS*, pages 155–164. ACM, 2003. Citations: § 70

[KW15] E. Kiltz and H. Wee. Quasi-Adaptive NIZK for Linear Subspaces Revisited. In *Eurocrypt*, 2015. Citations: § 32, 52, 53, 54, 55, 58, and 69

[KW18] S. Kim and D. J. Wu. Multi-Theorem Preprocessing NIZKs from Lattices. In *Crypto*, LNCS, page To appear. Springen, 2018. Citations: § 190

[KY05] A. Kiayias and M. Yung. Group signatures with efficient concurrent join. In *Eurocrypt*, number 3494 in LNCS, pages 198–214. Springer, 2005. Citations: § 96 and 119

[KY06] A. Kiayias and M. Yung. Secure scalable group signature with dynamic joins and separable authorities. 1(1):24–45, 2006. Citations: § 43, 44, 46, 65, 95, 96, and 99

[Lin08] A. Y. Lindell. Efficient fully-simulatable oblivious transfer. In *CT-RSA*, LNCS, 2008. Citations: § 146

[LLLS13] F. Laguillaumie, A. Langlois, B. Libert, and D. Stehlé. Lattice-based group signatures with logarithmic signature size. In *Asiacrypt*, volume 8270 of *LNCS*, pages 41–61. Springer, 2013. Citations: § 77, 78, and 118

[LLM+16a] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *Asiacrypt*, 2016. Citations: § xx, 5, 6, 16, 35, 38, 118, 128, 131, 132, and 169

[LLM+16b] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Zero-knowledge arguments for matrix-vector relations and lattice-based group encryption. In *Asiacrypt*, 2016. Citations: § xx, xxi, 5, 6, 35, 169, and 174

[LLM+17] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Adaptive oblivious transfer with access control from lattice assumptions. In T. Takagi and T. Peyrin, editors, *Asiacrypt*, pages 533–563. Springer, 2017. Citations: § xx, xxii, 5, 6, 7, and 168

[LLNW14] A. Langlois, S. Ling, K. Nguyen, and H. Wang. Lattice-Based Group Signature Scheme with Verifier-Local Revocation. In *PKC*, volume 8383 of *LNCS*, pages 345–361. Springer, 2014. Citations: § 44 and 118

[LLNW16] B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-size Ring Signatures and Group Signatures Without Trapdoors. In *Eurocrypt*, volume 9666 of *LNCS*, pages 1–31. Springer, 2016. Citations: § xix, 5, 77, 78, 80, 118, 169, and 174

[LLNW17] B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-Knowledge Arguments for Lattice-Based PRFs and Applications to E-Cash. In *Asiacrypt*, LNCS, pages 304–335. Springer, 2017. Citations: § 35, 189, and 190

[LM08] V. Lyubashevsky and D. Micciancio. Asymptotically Efficient Lattice-Based Digital Signatures. In *TCC*, LNCS, pages 37–54. Springer, 2008. Citations: § 191

[LMPY16] B. Libert, F. Mouhartem, T. Peters, and M. Yung. Practical "signatures with efficient protocols" from simple assumptions. In *AsiaCCS*, pages 511–522. ACM, 2016. Citations: § xx, 5, and 16

[LNSW13] S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications. In *PKC*, volume 7778, pages 107–124. Springer, 2013. Citations: § 35, 36, 38, 78, 79, 118, 125, 128, 130, 168, and 169

[LNW15] S. Ling, K. Nguyen, and H. Wang. Group Signatures from Lattices: Simpler, Tighter, Shorter, Ring-Based. In *PKC*, volume 9020 of *LNCS*, pages 427–449. Springer, 2015. Citations: § 43, 77, 78, 79, 98, 107, 118, 128, 131, and 169

[LNWX17] S. Ling, K. Nguyen, H. Wang, and Y. Xu. Lattice-Based Group Signatures: Achieving Full Dynamicity (and Deniability) with Ease. In *ACNS*, LNCS. Springer, 2017. Citations: § 44

[LNWX18] S. Ling, K. Nguyen, H. Wang, and Y. Xu. Constant-Size Group Signatures from Lattices. In *PKC*, LNCS, pages 58–88. Springer, 2018. Citations: § 191

[LP07] Y. Lindell and B. Pinkas. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In M. Naor, editor, *Eurocrypt*, pages 52–78. Springer, 2007. Citations: § 11

[LPJY13] B. Libert, T. Peters, M. Joye, and M. Yung. Linearly Homomorphic Structure-Preserving Signatures and Their Applications. In *Crypto*, LNCS, pages 289–307. Springer, 2013. Citations: § 57

[LPJY14] B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from Malleability: Simulation-Sound Quasi-Adaptive NIZK Proofs and CCA2-Secure Encryption from Homomorphic Signatures. In *Eurocrypt*, LNCS, pages 514–532. Springer, 2014. Citations: § 52, 53, and 54

[LPJY15] B. Libert, T. Peters, M. Joye, and M. Yung. Compactly Hiding Linear Spans: Tightly Secure Constant-Size Simulation-Sound QA-NIZK Proofs and Applications. In *Asiacrypt*, pages 681–707, 2015. Citations: § 32

[LPQ17] B. Libert, T. Peters, and C. Qian. Structure-Preserving Chosen-Ciphertext Security with Shorter Verifiable Ciphertexts. In *PKC*, volume 10174 of *LNCS*, pages 247–276. Springer, 2017. Citations: § xviii and 4

[LPY15] B. Libert, T. Peters, and M. Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In *Crypto*, volume 9216 of *LNCS*, pages 296–316. Springer, 2015. Citations: § xviii, 3, 17, 43, 51, 54, 56, and 69

[LRSW99] A. Lysyanskaya, R. L. Rivest, A. Sahai, and S. Wolf. Pseudonym Systems. In *SAC*, pages 184–199, 1999. Citations: § 51 and 75

[LS14] A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 2014. Citations: § 191

[LSS14] A. Langlois, D. Stehlé, and R. Steinfeld. GGHLite: More efficient multilinear maps from ideal lattices. In *Eurocrypt*, volume 8441 of *LNCS*, pages 239–256. Springer, 2014. Citations: § 90 and 91

[LYJP14] B. Libert, M. Yung, M. Joye, and T. Peters. Traceable group encryption. In *PKC 2014*, volume 8383 of *LNCS*, pages 592–610. Springer, 2014. Citations: § xviii, xx, 4, and 119

[Lyu08] V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *PKC*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008. Citations: § xvii, 3, 16, 34, 78, and 118

[Lyu09] V. Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In *Asiacrypt*, pages 598–616. Springer, 2009. Citations: § xvii, 3, and 34

[Lyu12] V. Lyubashevsky. Lattice signatures without trapdoors. In *Eurocrypt*, volume 7237 of *LNCS*. Springer, 2012. Citations: § xix and 5

[Mer79] R. C. Merkle. *Secrecy, Authentication, and Public Key Systems*. PhD thesis, Stanford University, June 1979. http://www.merkle.com/papers/Thesis1979.pdf. Citations: § 30

[Mer89] R. C. Merkle. A Certified Digital Signature. In *Crypto*, pages 218–238. Springer, 1989. Citations: § 30

[Moh11] P. Mohassel. One-Time Signatures and Chameleon Hash Functions. In *SAC*, LCNS, pages 302–319. Springer, 2011. Citations: § 191

[MP12] D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012. Citations: § xix, 4, 22, 25, 82, 87, 131, 132, 154, 157, and 158

[MSS17] A. Menezes, P. Sarkar, and S. Singh. Challenges with Assessing the Impact of NFS Advances on the Security of Pairing-Based Cryptography. In R. C.-W. Phan and M. Yung, editors, *Paradigms in Cryptology – Mycrypt. Malicious and Exploratory Cryptology*, pages 83–108. Springer, 2017. Citations: § xix, 4, and 21

[MV03] D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Crypto*, volume 2729 of *LNCS*, pages 282–298. Springer, 2003. Citations: § 118

[Nao03] M. Naor. On cryptographic assumptions and challenges. In Springer, editor, *Crypto*, pages 96–109, 2003. Citations: § 15

[NFHF09] T. Nakanishi, H. Fujii, Y. Hira, and N. Funabiki. Revocable Group Signature Schemes with Constant Costs for Signing and Verifying. In *PKC*, LNCS, pages 463–480. Springer, 2009. Citations: § 27

[NIS17] NIST. NIST post-quantum competition. Round 1., 2017. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions. Citations: § xv, 1, and 22

[NP99] M. Naor and B. Pinkas. Oblivious transfer with adaptive queries. In *Crypto*, volume 1666 of *LNCS*, pages 573–590, 1999. Citations: § xxi, 7, 19, 145, 146, and 186

[NP01] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *SODA*, pages 448–457, 2001. Citations: § 146

[NP05] M. Naor and B. Pinkas. Computationally secure oblivious transfer. 18(1):1–35, 2005. Citations: § 146

[NR97] M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *FOCS*, pages 458–467. IEEE Press, 1997. Citations: § 53

[NYO08] T. Nishide, K. Yoneyama, and K. Ohta. Attribute-based encryption with partially hidden encryptor-specified access structures. In *ACNS'08*, number 5037 in LNCS, pages 111–129, 2008. Citations: § 147

[NZZ15] P. Q. Nguyen, J. Zhang, and Z. Zhang. Simpler efficient group signatures from lattices. In *PKC*, volume 9020 of *LNCS*, pages 401–426. Springer, 2015. Citations: § 77, 78, and 118

[Oka95] T. Okamoto. An efficient divisible electronic cash scheme. In *Crypto*, volume 963 of *LNCS*, pages 438–451. Springer, 1995. Citations: § 190

[Oka06] T. Okamoto. Efficient Blind and Partially Blind Signatures Without Random Oracles. In *TCC*, LNCS, pages 80–99. Springer, 2006. Citations: § 51

[OO91] K. Ohta and T. Okamoto. Universal electronic cash. In *Crypto*, volume 576 of *LNCS*, pages 324–337. Springer, 1991. Citations: § 190

[Pai99] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *EUROCRYPT 1999*, number 1592 in LNCS, pages 223–238. Springer, 1999. Citations: § 118

[Ped91] T. P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *Crypto*, pages 129–140. Springer, 1991. Citations: § 34

[Pei09] C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, pages 333–342. ACM, 2009. Citations: § 25

[Pre17] T. Prest. Sharper Bounds in Lattice-Based Cryptography Using the Rényi Divergence. In *Asiacrypt*, LNCS, pages 347–374o. Springer, 2017. Citations: § 18 and 90

[PS96] D. Pointcheval and J. Stern. Security Proofs for Signature Schemes. In *Eurocrypt*, LNCS, pages 387–398. Springer, 1996. Citations: § 31

[PS00] D. Pointcheval and J. Stern. Security Arguments for Digital Signatures and Blind Signatures. *Journal of Cryptology*, 13(3):361–396, June 2000. Citations: § 71

[PS16] D. Poincheval and O. Sanders. Short Randomizable Signatures. In *CT-RSA*, LNCS, pages 111–126. Springer, 2016. Citations: § 75

[PS18] D. Pointcheval and O. Sanders. Reassessing Security of Randomizable Signatures. In *CT-RSA*, LNCS, pages 319–338. Springer, 2018. Citations: § 51

[PSTY13] C. Papamanthou, E. Shi, R. Tamassia, and K. Yi. Streaming authenticated data structures. In *Eurocrypt*, volume 7881 of *LNCS*, pages 353–370. Springer, 2013. Citations: § 80

[PV08] C. Peikert and V. Vaikuntanathan. Non-interactive statistical zero-knowledge proofs for lattice problems. In *Crypto*, volume 5157 of *LNCS*, pages 536–553. Springer, 2008. Citations: § 118, 131, and 190

[PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Crypto*, volume 5157 of *LNCS*, pages 554–571, 2008. Citations: § 146, 158, and 190

[PW08] C. Peikert and B. Waters. Lossy Trapdoor Functions and Their Applications. In *STOC*, pages 187–196. ACM, 2008. Citations: § 192

[Rab60] M. O. Rabin. Degree of difficulty of computing a function and a partial ordering of recursive sets. Technical Report 2, Hebrew University of Jerusalem, 1960. Citations: § 12

[Rab81] M. Rabin. How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981. Citations: § xvi, xxi, 2, 6, and 145

[Reg05] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93. ACM, 2005. Citations: § xviii, 4, 22, 25, 158, 159, and 181

[RSS18] R. D. Rothblum, A. Sealfon, and K. Sotiraki. Towards Non-Interactive Zero-Knowledge for NP from LWE. iacr ePrint Report, 2018. https://eprint.iacr.org/2018/240. Citations: § 190

[Rü10] M. Rückert. Lattice-Based Blind Signatures. In *Asiacrypt*, volume 6477 of *LNCS*, pages 413–430, 2010. Citations: § 118

[Sch96] C. P. Schnorr. Security of 2^t-Root Identification and Signatures. In *Crypto*, LNCS, pages 143–156. Springer, 1996. Citations: § xvii, 3, and 33

[Sco02] M. Scott. Authenticated ID-based Key Exchange and remote log-in with simple token and PIN number. Cryptology ePrint Archive, 2002. https://ia.cr/2002/164. Citations: § 22

[SG98] V. Shoup and R. Gennaro. Securing Threshold Cryptosystems against Chosen Ciphertext Attack. In *Eurocrypt*, pages 1–16, 1998. Citations: § 66

[Sho99] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999. Citations: § xv and 1

[Sho06] V. Shoup. Sequences of Games: A Tool for Taming Complexity in Security Proofs. Tutorial. http://www.shoup.net/papers/games.pdf, January 2006. Citations: § 17

[SOK00] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems Based on Pairings. In *Symposium on Cryptography and Information Security*, pages 26–28, 2000. Citations: § xviii, 4, and 21

[SSE+12] Y. Sakai, J. Schuldt, K. Emura, G. Hanaoka, and K. Ohta. On the security of dynamic group signatures: Preventing signature hijacking. In *PKC*, volume 7293 of *LNCS*, pages 715–732, 2012. Citations: § 99

[Ste96] J. Stern. A new paradigm for public key identification. 42(6):1757–1768, 1996. Citations: § xvii, 3, 16, 35, 78, 79, 87, and 88

[SW05] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Eurocrypt*, number 3494 in LNCS, pages 457–473, 2005. Citations: § 147

[TK05] Y. Tauman-Kalai. Smooth projective hashing and two-message oblivious transfer. In *Eurocrypt'05*, number 3494 in LNCS, pages 78–95. Springer, 2005. Citations: § 146

[Tsi97] Y. Tsiounis. *Efficient Electronic Cash: New Notions and Techniques*. PhD thesis, 1997. Citations: § 190

[TW05] M. Trolin and D. Wikström. Hierarchical Group Signatures. In *ICALP 2005*, volume 3580 of *LNCS*, pages 446–458. Springer, 2005. Citations: § 118

[VP17] M. Vanhoef and F. Piessens. Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2. In *CCS*, pages 1313–1328. ACM, 2017. Citations: § xvii and 2

[Wat05] B. Waters. Efficient identity-based encryption without random oracles. In R. Cramer, editor, *Eurocrypt*, pages 114–127. Springer, 2005. Citations: § 21

[XXW13] X. Xie, R. Xue, and M. Wang. Zero knowledge proofs from Ring-LWE. In *CANS*, volume 8257 of *LNCS*, pages 57–73. Springer, 2013. Citations: § 118

[Yao86] A. C.-C. Yao. How to generate and exchange secrets. In *FOCS*, 1986. Citations: § 11

[YCZY14] T. H. Yuen, S. S. Chow, C. Zhang, and S. M. Yiu. Exponent-inversion signatures and ibe under static assumptions. Cryptology ePrint Archive, Report 2014/311, 2014. https:/ia.cr/2014/311. Citations: § 52

[ZAW+10] Y. Zhang, M.-H. Au, D. Wong, Q. Huang, N. Mamoulis, D. Cheung, and S.-M. Yiu. Oblivious transfer with access control: Realizing disjunction without duplication. In *Pairing*, number 6847 in LNCS, pages 96–115, 2010. Citations: § xxii, 7, 146, 147, and 187

List of Figures

List of Tables