Efficient lattice-based zero-knowledge proofs and applications

Rafaël Del Pino

THÈSE DE DOCTORAT

de l'Université de recherche Paris Sciences et Lettres
PSL Research University

Préparée à l'École normale supérieure

Efficient Lattice-Based Zero-Knowledge Proofs and Applications

Ecole doctorale n°386
Sciences Mathématiques de Paris Centre

Spécialité Informatique

COMPOSITION DU JURY :

Mme . AGRAWAL Shweta
IIT Madras
Examinatrice

M. FOUQUE Pierre-Alain
Université Rennes 1
Rapporteur, examinateur

M. LYUBASHEVSKY Vadim
IBM Research Zurich
Directeur de thèse

M. POINTCHEVAL David
CNRS, École normale supérieure
Directeur de thèse

M. STEHLÉ Damien
École normale supérieure de Lyon
Président du jury

M. WEE Hoeteck
CNRS, École normale supérieure
Examinateur

M. BRAKERSKI Zvika
Weizmann Institute of Science
Rapporteur (absent du jury)

Soutenue par
Rafaël del PINO
le 1er juin 2018

Dirigée par
Vadim LYUBASHEVSKY
et David POINTCHEVAL

ENS ÉCOLE NORMALE SUPÉRIEURE

PSL RESEARCH UNIVERSITY PARIS

Efficient Lattice-Based Zero-Knowledge Proofs and Applications

Rafaël del Pino

Thèse de doctorat dirigée par
Vadim Lyubashevsky et David Pointcheval

Résumé

Le chiffrement à base de réseaux euclidiens a connu un grand essor durant les vingt dernières années. Autant grâce à l'apparition de nouvelles primitives telles que le chiffrement complètement homomorphe, que grâce à l'amélioration des primitives existantes, comme le chiffrement á clef publique ou les signatures digitales, qui commencent désormais à rivaliser avec leurs homologues fondés sur la théorie des nombres. Cela dit les preuves à divulgation nulle de connaissance, bien qu'elles représentent un des piliers des protocols de confidentialité, n'ont pas autant progressé, que ce soit au niveau de leur expressivité que de leur efficacité.

Cette thèse s'attelle dans un premier temps à améliorer l'état de l'art en matière de preuves à divulgation nulle de connaissance. Nous construisons une preuve d'appartenance à un sous ensemble dont la taille est indépendante de l'ensemble en question. Nous construisons de même une preuve de connaissance amortie qui est plus efficace et plus simple que toutes les constructions qui la précèdent.

Notre second propos est d'utiliser ces preuves à divulgation nulle de connaissance pour construire de nouvelles primitives cryptographiques. Nous concevons une signature de groupe dont la taille est indépendante du groupe en question, ainsi qu'un schéma de vote électronique hautement efficace, y compris pour des élections à grand échelle.

Abstract

Lattice based cryptography has developed greatly in the last two decades, both with new and stimulating results such as fully-homomorphic encryption, and with great progress in the efficiency of existing cryptographic primitives like encryption and signatures which are becoming competitive with their number theoretic counterparts. On the other hand, even though they are a crucial part of many privacy-based protocols, zero-knowledge proofs of knowledge are still lagging behind in expressiveness and efficiency.

The first goal of this thesis is to improve the quality of lattice-based proofs of knowledge. We construct new zero-knowledge proofs of knowledge such as a subset membership proof with size independent of the subset. We also work towards making zero-knowledge proofs more practical, by introducing a new amortized proof of knowledge that subsumes all previous results.

Our second objective will be to use the proofs of knowledge we designed to construct novel and efficient cryptographic primitives. We build a group signature whose size does not depend on the size of the group, as well as a practical and highly scalable lattice-based e-voting scheme.

Acknowledgments

> You won't ever be able to tell if everything is going to be totally haywire, or maybe if truly everything is perfectly fine
>
> BMO, Adventure Time

Ce sont plus de trois années de thèse qui prennent fin par la rédaction de ces remerciements. Malgré – ou peut être grâce à – ses hauts et ses bas, cette thèse a été une expérience unique que je ne regrette pas un seul instant. C'est grâce à tous ceux qui m'ont aidé que je peux être fier du travail que j'ai fourni pendant ces trois années. Qu'ils eurent été des mentors, des collègues ou simplement des proches, je tiens à tous les remercier, et je m'excuse d'avance auprès de ceux que je ne manquerai pas d'oublier... à cet effet cette thèse fait aussi office de coupon pour une bière gratuite.

My first thanks go to my advisor Vadim Lyubashevsky. I am grateful for his keen insights and our fruitful discussions, which have led to multiple papers I am really proud of. He has given me the opportunity to work in what I think are some of the best research environments in the world, whether it be at ENS or IBM Zurich. Working with him for more than three years was truly a pleasure, and I am sure we will get to collaborate again in the future.

Je voudrais remercier le directeur du laboratoire de Crypto de l'École normale supérieure, David Pointcheval. J'aurais aimé avoir eu plus d'occasions de travailler avec lui, notre unique collaboration fut un plaisir de part sa rigueur scientifique et sa pédagogie.

Je remercie mes rapporteurs Zvika Brakerski et Pierre-Alain Fouque qui ont accepté de relire ma thèse dans des délais presque prohibitifs. J'aimerais aussi remercier Shweta Agrawal, Damien Stehlé et Hoeteck Wee d'avoir accepté de faire partie de mon jury.

Je tiens à remercier tous mes collègues de l'ENS: Adrian, Aisling, Alain, Anca, Angelo, Antonia, Aurélien, Aurore, Balthazar, Céline, Chloé, Dahmun, Damien, David, Édouard, Fabrice, Geoffroy, Georg, Hieu, Hoeteck, Houda, Itai, Jérémy, Julia, Léo, Louiza, Mario, Mélissa, Michel, Michele Minelli, Michele Orrù, Nicky, Pierre-Alain, Phong, Pierrick, Pooya, Raphaël, Razvan, Rémi, Romain, Simon, Sonia, Sylvain, Tancrède, Thierry, Thomas Peters, et Thomas Prest. Je suis particulièrement redevable a Michele Minelli dont j'ai fait bien malgré lui mon émissaire à l'ENS quand je n'étais plus en France. Je sais qu'il gardera la flamme des lattices vivante au sein de l'ENS. Je remercie aussi Florian avec qui j'ai travaillé sur des problèmes passionants, et parfois même fait de la cryptographie. If you need some help staying awake during my elating defense here is an enigma from Florian which has kept me awake for many nights:
The evil Dr. No has captured 10 mathematicians and wants to play a little game with them. He explains the rules beforehand and lets them agree on a strategy: "I have 10 copies of the

same room containing an infinite countable number of boxes, indexed from 0 upwards. Each one of those boxes contains a real number of my choosing – the same in each room. Each of you will go into one copy of this room and will be free to open any number of boxes (even a countable number). However, you have to leave at least one box closed (they can chose which one, and it can be different for each of them), and guess which number is inside this box. If all but one of you guess correctly, I will let you free. Otherwise, you will be tortured to death." How can the mathematicians make sure they survive this challenge?

Je remercie l'administration de l'ENS et de l'INRIA: Joëlle, Lise-Marie, Nathalie, Stéphane, et Valérie. C'est grâce à leur efficacité que j'ai pu me concentrer exclusivement sur la recherche.

I would like to thank Michael Osborne for welcoming me in the Security & Privacy team of IBM Zurich, as well as everyone in the team: Anja, Anna, Björn, Cecilia, Christian, David, Eduarda, Gregor, Gregory, Jan, Maria, Manu, Patrick, Thijs, Tommaso, and Vadim. Working in IBM was an enriching experience and I have learned a lot by interacting with my peers.

Enfin je remercie ma famille. Mes parents Catherine et Basilio qui m'ont soutenu à 100% quand je vivais encore avec eux et à 200% après que j'ai quitté le cocon familial, et ma sœur Alicia qui a sacrifié ses congés pour visiter des contrées exotiques avec son petit frère.

Contents

Chapter 1

Introduction

From its genesis to less than a century ago, cryptography has been a tool of war. Encrypting messages so they would not fall into enemy hands was done using the now famous Caesar cipher in ancient Rome, and using the state of the art Enigma machines during World War II. A common point between these two cryptosystems, and all the ones that came in-between, is their use of symmetric key cryptography: they allow secure communication between two parties who share a common secret information.

In fact, different types of cryptography were not even conceived before 1970 when James H. Ellis surmised the possibility of "non-secret" encryption, which would allow two parties to communicate securely without prior agreement on a common secret. The first concrete example of public-key cryptography comes from a 1974 work of Ralph C.Merkle who devised a protocol in which two parties can exchange a secret key through an insecure channel without needing to know any common secret information.

Since then cryptography has grown at an impressive rate, with a plethora of new concepts such as identification schemes, which allow an individual to prove his identity via the knowledge of a secret without revealing it, fine-grained public key infrastructures, in which one gets to choose who should be able to decrypt which part of a message, and many more... This boom is due to a change of paradigm in regard to the purpose of cryptography: the emergence of information technology and in particular the internet has created a need for efficient and novel cryptographic tools.

Zero-knowledge proofs of knowledge (ZKPoK) are an important primitive in privacy preserving schemes. A ZKPoK is a two-party protocol in which a prover, Alice, wants to convince a verifier, Bob, that she knows a secret information "x" without revealing any other information about "x". Such proofs are ubiquitous in privacy protocols, they are used to construct digital signatures, authentication schemes, and they gained renewed interest with the rapid expansion of blockchain and cryptocurrencies.

1.1 Lattice-Based Cryptography

"When will large scale quantum computers exist?" Ask ten people and you will receive ten different answers. However, the fact remains that tremendous work is being put towards the construction of quantum computers and it is becoming apparent that the existence of large scale quantum computers is really a matter of "when" and not "if". Once such computers are built number theoretic cryptography, which is by and large the only cryptography that

is currently being used in practice, will collapse. In the last few years a lot of progress has been made towards constructing post-quantum cryptography, i.e. cryptography that relies on computational assumptions which are believed to resist quantum computers, as is evidenced by the NIST call for "Post-Quantum Cryptography Standardization", and lattice-based cryptography is one of the most promising candidates.

Intuitively a lattice can be seen as a periodic grid in $\mathbb{R}^n$, it is formally defined as the set of all integer-valued linear combinations of some basis vectors $(\mathbf{b}_1, \ldots, \mathbf{b}_k)$.

Cryptography on lattices started with the seminal work of Ajtai[Ajt96] in which he introduces the Short Integer Solution (SIS) problem and proves that this problem is on average as hard as some worst-case well-known problems on lattices. In parallel Jeffrey Hoffstein, Jill Pipher, and Joseph Silverman developed the NTRU encryption scheme [HPS98] based on polynomial rings. While this scheme did not enjoy a theoretical security proof, it has still resisted 20 years of cryptanalysis. In 2005, Regev [Reg05] introduced the Learning With Errors (LWE) problem and proved it to be as hard as standard lattice problems. The LWE problem was a breakthrough in lattice-based cryptography allowing for more efficient schemes and new cryptographic primitives such as fully-homomorphic encryption [1].

Schemes based on LWE and SIS suffer from one major caveat when compared to schemes based on number theoretic assumptions: they have substantially larger key and ciphertext sizes. To remedy this issue ring variants of these problems R-LWE and R-SIS were introduced respectively by Lyubashevsky, Peikert and Regev [LPR10], and Micciancio [Mic02]. These new problems use polynomials instead of matrices, effectively reducing parameters by up to a square root factor. This increased efficiency comes at a price in security since these problem can only be reduced to standard problems on ideal lattices which are more structured lattices. It is however worth noting that, except for marginal cases, attacks on ideal lattices do not perform better than attacks on standard lattices.

1.1.1 Lattice-Based Zero-Knowledge

The security of lattice-based cryptographic primitives is based on the hardness of recovering a short vector $\mathbf{s}$ when given a matrix $\mathbf{A}$ and $\mathbf{t} = \mathbf{As}$ as inputs, this problem is known as the Inhomogeneous Short Integer Solution (ISIS) problem. The operations are performed over some ring R, which most commonly is either $\mathbb{Z}_a$ or a polynomial ring $\mathbb{Z}[X]/X^d + 1$.[2] Depending on the primitive, $\mathbf{s}$ can represent the secret key, the randomness used during encryption, the signature of a message, or anything else that one should not be able to obtain just by knowing $\mathbf{A}$ and $\mathbf{t}$. In many cryptographic protocols, someone announcing the value $\mathbf{t}$ would also need to be able to prove the knowledge of a short pre-image $\bar{\mathbf{s}}$ (the pre-image may not be unique) of $\mathbf{t}$ satisfying

$$\mathbf{A} \cdot \bar{\mathbf{s}} = \mathbf{t} \tag{1.1}$$

The first approach developed for constructing such a zero-knowledge proof is using Stern-type proofs for codes [Ste94] adapted to the lattice setting [KTX08; LNSW13]. The main downside of this technique is that it has rather long proofs. Each run of the protocol has soundness error 2/3, thus requiring over 200 repetitions for 128 bits of security and over 400 repetitions if one would like to have 128 bits of *quantum* security in the resulting NIZK proof

[1] Though the original FHE construction of Gentry [Gen09] did not use the LWE problem, all the more recent constructions do.

[2] We will use this polynomial ring throughout our paper as it is the one that is almost exclusively used in practice. Instantiations with other monic polynomials are also possible.

constructed via the Fiat-Shamir technique. Even the most basic application, such as proving the knowledge of a solution to a hash function (e.g. the one in [LMPR08]), would require around 1KB for every round, thus making the total proof size approximately 400 KB. More complicated applications, as well as those in which the inputs are not taken from $\{0,1\}^k$, quickly push such proofs to the order of Megabytes.

Another technique for creating zero-knowledge proofs is the "Fiat-Shamir with Aborts" approach which allows to create a proof of knowledge of a vector $\bar{\mathbf{s}}$ with small coefficients (though larger than those in $\mathbf{s}$) and a ring element $\bar{c}$ with very small coefficients satisfying $\mathbf{A}\bar{\mathbf{s}} = \bar{c}\mathbf{t}$ [Lyu09; Lyu12]. As long as the ring $\mathcal{R}$ has many elements with small coefficients, such proofs are very efficient, producing soundness of $1 - 2^{-128}$ with just one iteration. While these proofs are good enough for constructing practical digital signatures (e.g. [GLP12; DDLL13; BG14]), commitment schemes with proofs of knowledge [BKLP15; BDOP16], and certain variants of verifiable encryption schemes [LN17], they prove less than what the honest prover knows. In many applications where zero-knowledge proofs are used, in particular those that need to take advantage of additive homomorphisms, the presence of the element $\bar{c}$ makes these kinds of "approximate" proofs too weak to be useful. As of today, we do not have any truly practical zero-knowledge proof systems that give a proof of Equation (1.1).

The situation is considerably more promising when one considers *amortized* proofs, and the increased efficiency of such proofs has already been exploited in practical multi-party computation [DPSZ12]. A series of works [DPSZ12; BDLN16; CDXY17] have considered how to obtain efficient exact proofs with overwhelming soundness when proving many equations at the same time, resulting in [PL17] in proofs with constant overhead and small slack. These proofs can however only be use when amortized over thousands of equations like (1.1), making them inapplicable in many scenarios.

1.2 Personal Contributions

1.2.1 Contributions in this Thesis

[PLNS17] In this paper we propose a lattice-based electronic voting scheme, EVOLVE (Electronic Voting from Lattices with Verification), which is conjectured to resist attacks by quantum computers. Our protocol involves a number of voting authorities so that vote privacy is maintained as long as at least one of the authorities is honest, while the integrity of the result is guaranteed even when all authorities collude. Furthermore, the result of the vote can be independently computed by any observer. At the core of the protocol is the utilization of a homomorphic commitment scheme with strategically orchestrated zero-knowledge proofs: voters use approximate but efficient "Fiat-Shamir with Aborts" proofs to show the validity of their vote, while the authorities use amortized exact proofs to show that the commitments are well-formed. We also present a novel efficient zero-knowledge proof that one of two lattice-based statements is true (so-called OR proof) and a new mechanism to control the size of the randomness when applying the homomorphism to commitments.

[PLS18] In this paper we present a new lattice-based zero-knowledge proof system for proving that a committed value belongs to a particular set of small size. The sets for which our proofs are applicable are exactly those that contain elements that remain stable under Galois automorphisms of the underlying cyclotomic number field of our protocol.

An application of our new proofs is that they allow the use of the selectively-secure signature scheme (i.e. a signature scheme in which the adversary declares the forgery message before seeing the public key) of Agrawal et al. [ABB10] for constructing lattice-based privacy protocols. For selectively-secure schemes to be meaningfully converted to standard signature schemes, it is crucial that the size of the message space is small. Using our zero-knowledge proofs, we can strategically pick sets for which we can provide efficient zero-knowledge proofs of membership.

[BBC+18] In this paper we present a surprisingly simple zero-knowledge proof for pre-images of linear relations whose amortized communication complexity depends only logarithmically on the number of relations being proved. This latter protocol is a substantial improvement, both theoretically and in practice, over the previous results in this line of research of Damgård et al. [DPSZ12], Baum et al. [BDLN16], Cramer et al. [CDXY17] and del Pino and Lyubashevsky [PL17].

1.2.2 Other Contributions

[BPMW16] In this paper we study the circuit privacy of fully homomorphic encryption (FHE). We carefully analysis of the noise growth in the FHE scheme of Alperin-Sheriff and Peiker[AP14] and prove that the addition of a small well-chosen noise to the output of computations is sufficient to ensure circuit privacy.

[PLP16] In this paper, we show that by simultaneously considering the secrecy and authenticity requirements of an authenticated key exchange (AKE), we can construct a scheme that is more secure and with smaller communication complexity than a scheme created by a generic combination of a key encapsulation mechanism (KEM) with a signature scheme. We first observe that relaxing the correctness property of the KEM so that it is no longer overwhelming allows for better parameters at virtually no impact in security. Our second improvement is showing that certain hash-and-sign lattice signatures can be used in "message-recovery" mode. In this mode, the signature size is doubled but this longer signature is enough to recover an even longer message – thus the signature is longer but the message does not need to be sent. This is advantageous when signing relatively long messages, such as the public keys and ciphertexts generated by a lattice-based KEM.

[PL17] In this paper we present an amortized zero-knowledge proof of knowledge for lattice-based one-way functions. This work is an improvement of a paper by Cramer et al. [CDXY17] and reduces the number of equations needed for amortization by a parameter $\sim \log^2 \alpha$ at a cost in running time of α. For example, increasing the running time by a factor of 8 allows us to decrease the required number of samples from 69000 to 4500 (a factor of 15).

1.3 Organization of this Thesis

In Chapter 2 we define the notations that will be used throughout this thesis, we recall basic notions on lattices and Gaussians, and we give formal definitions as well as lattice-based instantiations of the cryptographic building blocks we will use.

In chapter 3 we define Σ'-protocols, and give constructions of proofs of knowledge for: one-way function preimages, commitment openings, disjunction of NP-languages, and subset membership. We also give an amortized proof of knowledge for one-way function preimages.
In chapter 4 we construct a group signature using the proof of subset membership of Chapter 3. We give concrete parameters as well as the resulting signature size.
In chapter 5 we describe a security model for e-voting and use the proof for disjunctions as well as the amortized proof of Chapter 3 to build an e-voting scheme. We fix parameters and present the size and timings obtained from our proof-of-concept implementation.
Finally, we conclude in Chapter 6 and present some open questions.

Chapter 2

Preliminaries

In this chapter we present the notations that will be used throughout this thesis. Then we we give some properties on the invertibility of small polynomials in the ring $\mathbb{Z}_q[X]/X^d+1$, which will be useful for the proofs of knowledge of Chapter 3. We give some background on lattices and Gaussian sampling, and we give a succinct overview on the cryptanalysis of lattice problems, which we will use to set the parameters of the schemes presented in this thesis. Finally we give formal definitions, security models and lattice-based constructions for commitments and public key cryptography which we will use in the constructions of Chapters 4 and 5.

Contents

2.1 Notations

Sets, Integers. We denote by $\mathbb{Z}$ the set of integers, by $\mathbb{N}$ the set of non-negative integers, and by $\mathbb{R}$ the set of real numbers. For two integers $a, b \in \mathbb{Z}$ such that $a \leq b$, we denote by $\{a, \ldots, b\}$ the set of all integers $c \in \mathbb{Z}$ such that $a \leq c \leq b$. For $a \in \mathbb{N}$ such that $a > 0$, the notation $[a]$ is equivalent to $\{1, \ldots, a\}$. We denote the size of a finite set $\mathcal{S}$ by $|\mathcal{S}|$. For a positive integer q we denote by $\mathbb{Z}_q$ the ring of integers modulo q for which we will consider the set of representatives $\left\{-\left\lfloor \frac{q-1}{2} \right\rfloor, \ldots, \left\lceil \frac{q-1}{2} \right\rceil\right\}$.
Rings, Vectors, Matrices. We will consider vectors and matrices over a ring $\mathcal{R}$ which will be either $\mathcal{R} = \mathbb{Z}$ or $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for some polynomial f. Elements in $\mathcal{R}$ as well as $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ will be written in lower case (e.g. $c \in \mathcal{R}$). Vectors will be in bold lower case (e.g. $\mathbf{y} \in \mathcal{R}^m$) and will be column vectors. Matrices will be in bold upper case (e.g. $\mathbf{A} \in \mathcal{R}^{n \times m}$).
Vector norms. Polynomials will be identified as the vector of their coefficients for the purposes of norms, i.e. for $g = \sum_0^{d-1} g_i X^i$ and a positive integer p, the ℓ_p norm is defined as

$$\|g\|_p = \left(\sum_1^{d-1} |g_i|^p \right)^{\frac{1}{p}}.$$

Norms are extended to vectors and matrices over $\mathcal{R}$ in the natural way, i.e. for $\mathbf{v} = (v_1, \ldots, v_m) \in \mathcal{R}^m$, $\|\mathbf{v}\|_p = \left(\sum_1^{m-1} \|v_i\|_p^p\right)^{\frac{1}{p}}$ and for $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_m) \in \mathcal{R}^{n \times m}$, $\|\mathbf{V}\|_p = \left(\sum_1^{m-1} \|\mathbf{v}_i\|_p^p\right)^{\frac{1}{p}}$. Since we will mostly consider the euclidean norm of vectors we will abbreviate the notation $\|\cdot\|_2$ as $\|\cdot\|$.
The operator norm of a matrix $\mathbf{A} \in \mathcal{R}^{n \times m}$ is defined as

$$s_1(\mathbf{A}) = \max_{\mathbf{x} \in \mathcal{R}^m \setminus \{\mathbf{0}\}} \left(\frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \right),$$

we also define a matrix norm that will be useful for the amortized zero-knowledge proofs of Chapter 3 which we call "max norm": for $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_m) \in \mathcal{R}^{n \times m}$,

$$\|\mathbf{A}\|_{\max} = \max_{j \in [m]} (\|\mathbf{a}_j\|).$$

2.2 Polynomial Rings

For the rest of this thesis we will consider $\mathcal{R}$ to be the ring $\mathbb{Z}[X]/X^d + 1$ for d a power of two, in the case $d = 1$ we will simply identify $\mathcal{R}$ with $\mathbb{Z}$. For many of our protocols we will require that "small" polynomials are invertible. The following lemmata guarantee such a properties for well chosen power-of-two cyclotomics.

Lemma 2.2.1 (Inverse of differences of monomials[BCK+14])**.** *Let d be a power of 2, let $a, b \in \{\pm X^i \ : \ i \geq 0\} \cup \{0\}$. Then $2(a-b)^{-1} \mod X^d + 1$ only has coefficients in $\{-1, 0, 1\}$.*

Lemma 2.2.2 (Invertibility of small polynomials [LS17])**.** *Let $d \geq k > 1$ be powers of 2 and q a prime such that $X^d + 1$ splits into k different irreducible polynomials modulo q. Then any c in $\mathbb{Z}_q[X]/(X^d + 1)$ such that $0 < \|c\| < q^{1/k}$ has an inverse in the ring.*

For this lemma to be meaningful we need to know how $X^d + 1$ splits modulo q, this is specified in [LS17, Corollary 1.2].

Lemma 2.2.3 (Factors of $X^d + 1$)**.** *Let $d \geq k > 1$ be powers of 2 and $q = 2k + 1 \mod 4k$ be a prime. Then the polynomial $X^d + 1$ factors as*

$$X^d + 1 = \prod_{j=1}^{k} (X^{n/k} - r_j) \mod q$$

For distinct $r_j \in \mathbb{Z}_q^*$ where $X^{n/k} - r_j$ are irreducible in the ring $\mathbb{Z}_q[X]$.

Polynomial and matrix products. Products of polynomials in $\mathcal{R}$ (and $\mathcal{R}_q$) can be represented by matrix/vector products in $\mathbb{Z}$ (and $\mathbb{Z}_q$). For a polynomial $a = \sum a_i X^i \in \mathcal{R}$ we will define the rotation matrix of a as

$$\mathrm{Rot}(a) := \left[a \mid aX \mid aX^2 \mid \ldots \mid aX^{d-1} \right] = \begin{bmatrix} a_0 & -a_{d-1} & \ldots & -a_1 \\ a_1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -a_{d-1} \\ a_{d-1} & \ldots & a_1 & a_0 \end{bmatrix}$$

It is apparent that if we write as $\mathbf{b} \in \mathbb{Z}^d$ the coefficients of $b \in \mathcal{R}$ then $\mathrm{Rot}(a) \cdot \mathbf{b} \in \mathbb{Z}^d$ is the vector that contains the coefficients of $a \cdot b \in \mathcal{R}$. Similarly for $\mathbf{A} \in \mathcal{R}^{n \times m}$, we will write $\mathrm{Rot}(\mathbf{A}) \in \mathbb{Z}^{dn \times dm}$ as the matrix in which we have applied Rot component-wise.

2.3 Lattices

2.3.1 Definitions and Properties

An n-dimensional lattice $\mathcal{L}$ is a discrete subgroup of $\mathbb{R}^n$. Any lattice can be expressed as the set of linear combinations of some linearly independent vectors $\mathbf{B} = (\mathbf{b}_1, \ldots, \mathbf{b}_k)$, i.e.

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) := \left\{ \sum_{i=1}^{k} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z} \right\} = \mathbf{B}\mathbb{Z}^k$$

$\mathbf{B}$ is a called a basis of $\mathcal{L}$ and its number of columns k is the rank of the lattice. Since $\mathcal{L}$ is a subgroup of $\mathbb{R}^n$, we can consider the quotient group $\mathbb{R}^n/\mathcal{L}$ of cosets. For $\mathbf{c} \in \mathbb{R}^n$ the coset of $\mathbf{c}$ is

$$\mathbf{c} + \mathcal{L} = \{\mathbf{c} + \mathbf{x} \mid \mathbf{x} \in \mathcal{L}\}$$

Definition 2.3.1 (Successive minima)**.** *Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice of rank k. For $i \in [k]$ the i-th successive minimum of $\mathcal{L}$ is defined as*

$$\lambda_i(\mathcal{L}) := \inf \left\{ r \mid \dim \left(\mathrm{Span}(\mathcal{L} \cap \bar{B}(\mathbf{0}, r) \right) \geq i \right\}$$

Definition 2.3.2 (Determinant of a lattice)**.** *Let $\mathcal{L} = \mathcal{L}(\mathbf{B})$. The determinant, or volume, of $\mathcal{L}$ is defined as*

$$\mathrm{Vol}(\mathcal{L}) := \det(\mathcal{L}) := \sqrt{\det(\mathbf{B}^T \mathbf{B})}$$

The notion of volume will be useful to estimate the first minimum of a lattice. The Gaussian heuristic surmises that the number of lattice point in a convex body $\mathcal{S}$(we will usually consider a ball) is equal to the volume of $\mathcal{S}$ divided by the volume of the lattice.

$$|\mathcal{L} \cap \mathcal{S}| \approx \mathrm{Vol}(\mathcal{S})/\mathrm{Vol}(\mathcal{L}).$$

Using this heuristic with a sphere of radius $\lambda_1(\mathcal{L})$ gives the following approximation:

$$\lambda_1(\mathcal{L}) \approx \sqrt{\frac{n}{2\pi e}}\mathrm{Vol}(\mathcal{L})^{1/n}$$

Definition 2.3.3 (Dual Lattice)**.** *Let $\mathcal{L} \subset \mathbb{R}^n$ be a lattice. The dual of $\mathcal{L}$ is the lattice defined as*

$$\mathcal{L}^* := \{\mathbf{y} \in \mathrm{Span}(\mathcal{L}) \mid \forall \mathbf{x} \in \mathcal{L}, \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\}$$

Gram Schmidt orthogonalization. The Gram-Schmidt orthogonalization (GSO) is an algorithm that takes as input a basis $\mathbf{B}$ of a vector space and outputs an orthogonal basis $\tilde{\mathbf{B}}$ of the same vector space. The GSO is useful both in cryptography as it is for example used in the sampling of Gaussians over lattice in the GPV sampler of Section 2.3.5, and in cryptanalysis where it is an important component of lattice reduction algorithms.

Definition 2.3.4 (Gram-Schmidt orthogonalization)**.** *Let $\mathbf{B} := \left[\mathbf{b}_1 \mid \ldots \mid \mathbf{b}_n\right] \in \mathbb{R}^{n \times n}$ be a basis of $\mathbb{R}^n$. The Gram-Schmidt orthogonalization of $\mathbf{B}$ is the matrix $\tilde{\mathbf{B}}$ defined as follows:*

$$\tilde{\mathbf{b}}_1 := \mathbf{b}_1$$
$$\tilde{\mathbf{b}}_i := \mathrm{Proj}^{\perp}_{\mathbf{b}_1,\ldots,\mathbf{b}_{i-1}}(\mathbf{b}_i)$$

Q-ary lattices. In most lattice-based crypto applications we will consider lattices of a very specific form that comes from the SIS problems. For a matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, we define the q-ary lattice associated with $\mathbf{A}$ as:

$$\mathcal{L}(\mathbf{A})_q^{\perp} := \{\mathbf{x} \in \mathcal{R}^m \ : \ \mathbf{A}\mathbf{x} = \mathbf{0} \mod q\}.$$

It is easy to see that $\mathcal{L}(\mathbf{A})_q^{\perp}$ is indeed a lattice, and that it is of dimension dm over $\mathbb{R}$ (this lattice can be interpreted over $\mathbb{R}$ as $\mathcal{L}(\mathrm{Rot}(\mathbf{A}))_q^{\perp}$). It is also of rank dm, a simple proof of this being that the vectors of the form $(0, \ldots, 0, qX^i, 0, \ldots, 0) \in \mathcal{R}^m \simeq (0, \ldots, 0, q, 0, \ldots, 0) \in \mathbb{Z}^{dm}$ are in $\mathcal{L}(\mathbf{A})_q^{\perp}$ and there are dm independent such vectors. For $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$, with $n \leq m$, the lattice $\mathcal{L}(\mathbf{A})_q^{\perp}$ will have with high probability volume

$$\mathrm{Vol}(\mathcal{L}(\mathbf{A})_q^{\perp}) = q^{dn}.$$

To show this first write $\left[\mathbf{A}_1 \mid \mathbf{A}_2\right] := \mathrm{Rot}(\mathbf{A})$,where $\mathbf{A}_1 \in \mathbb{Z}_q^{dn \times dn}$. With high probability $\mathbf{A}_1$ will have an inverse and it is easy to notice that

$$\begin{bmatrix} q\mathbf{I}_{dn} & \mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{0} & \mathbf{I}_{d(m-n)} \end{bmatrix}$$

is a basis of $\mathcal{L}(\mathrm{Rot}(\mathbf{A}))_q^{\perp}$ and has determinant q^{nd}.

2.3.2 Gaussians.

Continuous and discrete Gaussians.
For any integer $n \geq 1$ and real $\sigma > 0$, the n-dimensional spherical Gaussian function $\rho_\sigma : \mathbb{R}^n \to (0, 1]$ is defined as[1]:

$$\rho_\sigma(\mathbf{x}) := \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right)$$

We define the discrete Gaussian over the coset $\mathbf{c} + \mathcal{L}$ of $\mathcal{L} \subset \mathbb{R}^n$, with standard deviation σ via the following probability density function:

$$\forall x \in \mathbf{c} + \mathcal{L};\ D_{\mathbf{c}+\mathcal{L},\sigma}(\mathbf{x}) = \frac{\rho_\sigma(\mathbf{x})}{\rho_\sigma(\mathbf{c} + \mathcal{L})}$$

where $\rho_\sigma(\mathbf{c} + \mathcal{L}) = \sum_{\mathbf{u}\in\mathcal{L}} \rho_\sigma(\mathbf{c} + \mathbf{u})$. When $\mathcal{L} = \mathbb{Z}$ and $\mathbf{c} = 0$ we will ignore the corresponding subscript, we will also write $\mathbf{x} \leftarrow D_\sigma^n$ as an alternative to $\mathbf{x} \leftarrow D_{\mathbb{Z}^n,\sigma}$ as sampling the n coefficients of $\mathbf{x}$ independently results in the same distribution. For a polynomial ring $\mathcal{R}$ we will write $x \leftarrow D_{\mathcal{R},\sigma}$ to denote that all the coefficients of $x \in \mathcal{R}$ are taken according to D_σ.
Norm Bounds. Using the tail bounds for the 0-centered discrete Gaussian distribution, we can show that for any $\sigma > 0$, $x \leftarrow D_\sigma$ is likely to be close to σ.

Lemma 2.3.5 (Gaussian tails)**.** *For any $k > 0$*

$$\Pr_{x\leftarrow D_\sigma}[|x| > k\sigma] \leq 2e^{-k^2/2}, \tag{2.1}$$

and when $\mathbf{x}$ is drawn from D_σ^n, we have

$$\Pr_{\mathbf{x}\leftarrow D_\sigma^n}[\|\mathbf{x}\| > \sqrt{2n} \cdot \sigma] < 2^{-n/4}. \tag{2.2}$$

Rejection Sampling. We give an important lemma on rejection sampling which will guarantee that the responses used in our zero-knowledge protocols do not leak information.

Algorithm 1 $\mathsf{Rej}(\mathbf{Z}, \mathbf{B}, \sigma)$

$u \leftarrow [0, 1)$
if $u > \frac{1}{3}\exp\left(\frac{-2\langle\mathbf{Z},\mathbf{B}\rangle+\|\mathbf{B}\|^2}{2\sigma^2}\right)$ **then return** 0
else return 1

Lemma 2.3.6 ([Lyu12])**.** *Let V be a subset of $\mathcal{R}^{n\times m}$ with elements of norm less than T, let h be a distribution over V, and $\mathbf{B} \leftarrow h$. Consider a procedure that samples $\mathbf{Y} \leftarrow D_{\mathcal{R},\sigma}^{n\times m}$ and then returns the output of $\mathsf{Rej}(\mathbf{Z} := \mathbf{Y} + \mathbf{B}, \mathbf{B}, \sigma)$ where $\sigma \geq 11\|\mathbf{B}\|$. The probability that this procedure outputs 1 is within 2^{-100} of $1/3$. The distribution of $\mathbf{Z}$, conditioned on the output being 1, is within statistical distance 2^{-100} of $D_{\mathcal{R},\sigma}^{n\times m}$.*

[1] Multiple works define the Gaussian function as $\rho_\sigma(\mathbf{x}) := \exp\left(-\frac{\pi\|\mathbf{x}\|^2}{\sigma^2}\right)$ which is a dilation of the function we use. We choose our definition because it entails that the discrete Gaussian we obtain from it will have standard deviation σ which is conceptually simpler.

2.3.3 Hard Problems on Lattices.

We define some of the computational problems defined on lattice. We will only consider the the problems which will be useful for the security analysis of Section 2.3.4.

Definition 2.3.7 (SVP - Shortest vector problem)**.** *Given an n-dimensional lattice $\mathcal{L}$, find a lattice vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$.*

Definition 2.3.8 (SVP_γ - Approximate shortest vector problem)**.** *Given an n-dimensional lattice $\mathcal{L}$ and $\gamma(n) \geq 1$, find a non-zero lattice vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \gamma(n)\lambda_1(\mathcal{L})$.*

Definition 2.3.9 (BDD_γ - Bounded distance decoding problem)**.** *Given an n-dimensional lattice $\mathcal{L}$, $\gamma(n) \geq 1$, and a target point $\mathbf{t} \in \mathbb{R}^n$ with the guarantee that $dist(\mathcal{L}, \mathbf{t}) < d = \lambda_1(\mathcal{L})/2\gamma(n)$, find the unique lattice vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v} - \mathbf{t}\| \leq d$.*

Definition 2.3.10 (uSVP_γ - Unique shortest vector problem)**.** *Given an n-dimensional lattice $\mathcal{L}$ and $\gamma(n) \geq 1$ with the guarantee that $\lambda_2(\mathcal{L}) \geq \gamma(n)\lambda_1(\mathcal{L})$, find a non-zero lattice vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \gamma(n)\lambda_1(\mathcal{L})$.*

Cryptographic Problems. We now consider two problems on which the security of most lattice-based schemes rely: SIS and LWE. The SIS problem was introduced by Ajtai [Ajt96] and the LWE problem by Regev [Reg05]. These problems were extended to consider polynomial rings respectively by by Micciancio [Mic02] (and then revisited and proven secure in [LM06; PR06]) and Lyubashevsky et al [LPR10]. In a recent work Langlois and Stelhé [LS15] defined a module variant of the SIS and LWE problems which encompasses the previous definitions. Remark that these problems can be defined for modules over any polynomial ring $\mathcal{R}$ but using rings other than $\mathbb{Z}[X]/X^d + 1$ makes the definitions rather cumbersome and will not be necessary here.

Definition 2.3.11 ($\text{M-SIS}_{q,n,m,\beta}$ - Module Short Integer Solution [LS15])**.** *For integers $q, n, m > 0$, real $\beta > 0$, given $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$, find $\mathbf{z} \in \mathcal{R}^m$ such that $\mathbf{Az} = 0 \mod q$ and $0 < \|\mathbf{z}\| \leq \beta$.*

Definition 2.3.12 ($\text{M-LWE}_{q,n,\sigma}$ - Module Learning With Errors adapted from [LS15])**.** *For integers $q, n > 0$, real $\sigma > 0$, and $\mathbf{s} \leftarrow D^n_{\mathcal{R},\sigma}$, let $\mathcal{A}_{q,\mathbf{s},\sigma}$ be the distribution obtained by sampling $\mathbf{a} \xleftarrow{\$} \mathcal{R}_q^n$, $e \leftarrow D_{\mathcal{R},\sigma}$ and outputting $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e) \in \mathcal{R}_q^n \times \mathcal{R}_q$. Given $\mathbf{s} \leftarrow D^n_{\mathcal{R},\sigma}$, distinguish between $\mathcal{A}_{q,\mathbf{s},\sigma}$ and $\left(\mathcal{U}(\mathcal{R}_q^n), \mathcal{U}(\mathcal{R}_q)\right)$.*

Both of these problems are shown to be as hard as some standard module-lattice problems in [LS15][2]. It is worth noting that these definition of M-SIS and M-LWE collapse to the aforementioned problems SIS and LWE when $d = 1$, and to R-SIS and R-LWE when $n = 1$.

Definition 2.3.13 ($\text{NTRU}_{q,\sigma}$)**.** *For an integer $q > 0$, a real $\sigma > 0$. Let $\mathcal{N}_{q,\sigma}$ be the distribution obtained by sampling $f, g \leftarrow D_{\mathcal{R},\sigma}$ conditioned on g being invertible in $\mathcal{R}_q$ and outputting $h = f/g \in \mathcal{R}_q$. Distinguish between $\mathcal{N}_{q,\sigma}$ and $\mathcal{U}(\mathcal{R}_q)$.*

The NTRU problem was only shown to be hard for $\sigma = \Omega(d\sqrt{q})$ with a reduction to R-LWE by Stelhé and Steinfield [SS11]. However slightly smaller parameters such as $\sigma = \Omega(\sqrt{q})$ are not known to have more efficient cryptanalysis that R-LWE instances of comparable standard deviation.

[2] In fact M-LWE is only shown to be hard in its decision variant for slightly non-spherical Gaussians but all practical applications consider spherical nonetheless.

2.3.4 Cryptanalysis

In this section we succinctly introduce the cryptanalytic tools from which we can evaluate the hardness of M-LWE and M-SIS.

Lattice reduction algorithms. First introduced by Lenstra, Lenstra, and Lovász [LLL82] lattice reduction algorithms aim at outputting a "short" basis of a lattice. Many improvements to the original LLL algorithm have been designed with various tradeoffs between running times and quality of output. We will be interested in the BKZ-β algorithm [Sch87] which is in practice the best available lattice reduction algorithm. The BKZ-β algorithm depends on a parameter called the block size β and uses as a subroutine an algorithm that solves the SVP problem exactly in dimension β, it has running time $poly(|\mathbf{B}|) \cdot T$ where $\mathbf{B}$ is the input basis and T is the running time of the SVP solving subroutine. Since solving SVP exactly takes exponential time, we usually equate the running time of BKZ-β with the one of the exact solver, different parameters β offer trade offs between the running time of the algorithm and the quality of the output. The BKZ-β algorithm, when run on a lattice $\mathcal{L}$ of dimension n, guarantees that the first basis vector it outputs will be $\mathbf{b}_1$ such that:

$$\|\mathbf{b}_1\| \leq \delta_0^n \cdot \mathrm{Vol}(\mathcal{L})^{1/n}$$

with δ_0 the root Hermite factor:

$$\delta_0 = \left(\frac{(\pi\beta)^{1/\beta}\beta}{2\pi e}\right)^{\frac{1}{2(\beta-1)}}.$$

The BKZ-β algorithm can therefore be used to find "somewhat" short vectors by setting δ_0 to be sufficiently small and using the corresponding block size. However this approach will not work when the lattice $\mathcal{L}$ has abnormally short vectors (which will typically be true when attacking LWE). Luckily Alkim et al. [ADPS16] have observed that in the presence of a very small vector the BKZ-β algorithm will not behave as expected and as a result the short vector will be detected. The basis $\mathbf{B}$ output by BKZ-β is usually (e.g. for random lattices) expected to follow the geometric series assumption:

$$\left\|\tilde{\mathbf{b}}_i\right\| = \alpha^{i-1} \left\|b_1\right\|, \text{ for } \alpha = \delta_0^{-2n/(n-1)}.$$

On the other hand an invariant of the BKZ-β algorithm guarantees that the vector $\mathbf{b}_{n-\beta+1}$ will be such that $\left\|\tilde{\mathbf{b}}_{n-\beta+1}\right\|$ is the norm of the shortest vector in the lattice obtained by projecting $\mathcal{L}$ orthogonally to $\mathbf{b}_1, \ldots, \mathbf{b}_{n-\beta}$. Suppose $\mathcal{L}$ contains a very short vector $\mathbf{v}$ and that $\mathbf{v}$ is taken uniformly on the sphere of radius $\|\mathbf{v}\|$, then the expected norm of the projection of $\mathbf{v}$ against $\mathbf{b}_1, \ldots, \mathbf{b}_{n-\beta}$ will be $\sqrt{\frac{\beta}{n}}\|\mathbf{v}\|$, while the geometric series assumption gives

$$\left\|\tilde{\mathbf{b}}_{n-\beta+1}\right\| = \delta_0^{\frac{n}{n-1}(2\beta-n-1)}\mathrm{Vol}(\mathcal{L})^{1/n} \approx \delta_0^{2\beta-n}\mathrm{Vol}(\mathcal{L})^{1/n}.$$

Which entails that the geometric series assumption will be broken when

$$\sqrt{\frac{\beta}{n}}\|\mathbf{v}\| < \delta_0^{2\beta-n}\mathrm{Vol}(\mathcal{L})^{1/n}.$$

It has been observed in practice that the BKZ-β detects such cases and that the full vector $\mathbf{v}$ can be recovered.

Solving M-SIS and M-LWE. To solve M-SIS$_{q,n,m,B}$, given $\mathbf{A} \leftarrow \mathcal{R}_q^{n\times m}$ we want to find a vector $\mathbf{z} \in \mathcal{R}^m$ of norm B such that $\mathbf{Az} = 0 \mod q$, i.e. we want to find a short vector in the lattice $\mathcal{L}(\mathbf{A})_q^\perp$ which is a dm-dimensional lattice of volume q^{dn}. This problem becomes harder when it gets closer to an instance of SVP (without approximation), i.e. when B is the size of the expected smallest vector in $\mathcal{L}(\mathbf{A})_q^\perp$, i.e. $B \approx \sqrt{\frac{dm}{2\pi e}} q^{n/m}$. If B is larger, meaning we have an approximate-SVP instance, we will simply compute the root Hermite factor necessary for solving M-SIS$_{q,n,m,B}$ as

$$\delta_0 = \left(Bq^{-n/m}\right)^{1/dm},$$

find the corresponding block size β and apply BKZ-β to find a short vector $\mathbf{z}$[3].
When B is smaller than the expected value of the shortest vector, meaning we have a unique-SVP instance, we will rely on the other approach we have described. i.e. we will solve

$$\sqrt{\frac{\beta}{dm}} B = \delta_0^{2\beta - dm} q^{n/m}$$

for β (by using the equation between δ_0 and β) and then use BKZ-β.
To solve M-LWE$_{q,n,\sigma}$, notice that for a matrix $\mathbf{A} \in \mathcal{R}_q^{m\times n}$ and a sample $\mathbf{b} := \mathbf{As} + \mathbf{e}$ with $\mathbf{s} \leftarrow D_{\mathcal{R},\sigma}^n$ and $\mathbf{e} \leftarrow D_{\mathcal{R},\sigma}^m$, we have:

$$\mathbf{Mz} = 0 \mod q, \text{ where } \mathbf{M} = \begin{bmatrix} \mathbf{A} & | & \mathbf{I}_{dm} & | & -\mathbf{b} \end{bmatrix} \in \mathbb{Z}_q^{dm \times d(m+n)+1} \text{ and } \mathbf{z} = \begin{bmatrix} \mathbf{s} \\ \mathbf{e} \\ 1 \end{bmatrix} \in \mathbb{Z}^{d(m+n)+1}$$

We are thus looking for a vector of norm $\sqrt{d(n+m)\sigma^2 + 1} \approx \sqrt{d(n+m)}\sigma$ in the lattice $\mathcal{L}(\mathbf{M})_q^\perp$ of volume $q^{m/(n+m)}$. In all "interesting" instances of M-LWE this vector will be much shorter than the expected shortest vector of $\mathcal{L}(\mathbf{M})_q^\perp$ and we will thus search for it by setting δ_0 such that:

$$\sqrt{\beta}\sigma = \delta_0^{2\beta - d(n+m)} q^{m/(n+m)}$$

Solving SVP. Since the BKZ-β algorithm relies on a subroutine that solves the shortest vector problem exactly in dimension β, it will be crucial to optimize this solver. There are two main approaches to solving SVP exactly. The first one is enumeration, which is in essence an exhaustive search algorithm. Enumeration algorithms run in constant space but they have worst case complexity $2^{O(\beta^2)}$. Due to multiple heuristic optimizations (such as pruning [GN08]) enumeration tends to be the most efficient algorithms for "feasible" blocksizes such as $\beta = 40$ or 50, however the super exponential behaviour of these algorithms quickly make them unusable for larger block sizes. We thus use sieving algorithms to obtain estimate the complexity of BKZ-β for large block sizes. Sieving algorithms proceed by first sampling an exponential number of lattice vectors, and then obtain increasingly shorter vectors by computing differences of vectors that are close to one another. Since these algorithms start by sampling exponentially many vectors they all require exponential space, on the other hand they also achieve (simply) exponential complexity. The best known sieving

[3] This approach can be slightly improved by noticing that we can ignore columns of $\mathbf{A}$ by setting the corresponding coefficient of $\mathbf{z}$ to 0 and reduce the problem to solving a M-SIS$_{q,n,m',B}$ instance for $m' \leq m$, we can then optimize over the value of m'. We find that $m' = \sqrt{\frac{n \log q}{\log \delta_0}}$ gives the best result.

algorithms [BDGL16; Laa15] have a heuristic time complexity of $2^{0.292\beta}$ which can be reduced to a quantum complexity of $2^{0.265\beta}$ [ADPS16] by using Grover's quantum search algorithm. Both algorithms have a space complexity of $2^{0.2075\beta}$.

2.3.5 Trapdoors.

Informally a trapdoored one-way function is a function that is hard invert except when given access to a secret "trapdoor". In the context of lattices the one-way functions we consider are of the form

$$\begin{array}{rccc} f_{\mathbf{A}}: & \{\mathbf{x} \in \mathcal{R}^m \mid \|\mathbf{x}\| \leq \beta\} & \to & \mathcal{R}^n \\ & \mathbf{x} & \mapsto & \mathbf{A}\mathbf{x} \end{array}$$

one can observe that the set of elements $\mathbf{x}$ s.t. $\mathbf{A}\mathbf{x} = \mathbf{0}$ forms a lattice and that solutions to $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{t}$ are small elements in a coset of this lattice. As such the purpose of a lattice trapdoor will be enable one to sample small vectors in a lattice. Gentry et al. [GPV08] present an algorithm which takes as input a basis $\mathbf{B}$ of a lattice $\mathcal{L}$ and outputs a sample from $D_{\mathbf{c}+\mathcal{L},\sigma}$ for any coset $\mathbf{c} + \mathcal{L}$ and for a standard deviation σ which depends on the size of $\mathbf{B}$.

Theorem 2.3.14 (GPV sampler [GPV08])**.** *Let $\varepsilon > 0$. There is a PPT algorithm* **Sample** *such that, for a basis $\mathbf{B}$ of a lattice $\mathcal{L} \subset \mathbb{Z}^n$, a vector $\mathbf{c} \in \mathbb{R}^n$, and a standard deviation $\sigma \geq \frac{1}{\pi}\sqrt{\frac{1}{2}\log\left(2n(1+\frac{1}{\varepsilon})\right)}\|\tilde{\mathbf{B}}\|_{\max}$. The output of* $\mathbf{Sample}(\mathbf{B}, \sigma, \mathbf{c})$ *is statistically close to $D_{\mathbf{c}+\mathcal{L},\sigma}$, i.e.*

$$\Delta\left(\mathbf{Sample}(\mathbf{B}, \sigma, \mathbf{c}), D_{\mathbf{c}+\mathcal{L},\sigma}\right) \leq \varepsilon$$

GPV Trapdoor. To use this theorem one needs to know a short basis of $\mathcal{L}$. While this is a hard problem for a random lattice, a series of work [Ajt99; Pei10; MP12] have tackled the issue of generating a trapdoored-lattice (i.e. with a known small basis) that is induistinguishable from a random lattice for anyone who does not know the trapdoor. We succintly present the result of [MP12] which uses "gadget"-based trapdoors for q-ary lattices.

Definition 2.3.15 (Gadget matrix [GPV08])**.** *For a modulus q and a base B, let $\ell := \lceil \log_B(q) \rceil$. The gadget vector $\mathbf{g} \in \mathcal{R}^\ell$ is defined as*

$$\mathbf{g} := \left(1, 2, 4, \ldots, 2^{\ell-1}\right)$$

The gadget matrix of dimension n is defined as

$$\mathbf{G} := \mathbf{I}_n \otimes \mathbf{g} = \begin{bmatrix} \mathbf{g} & & \\ & \ddots & \\ & & \mathbf{g} \end{bmatrix} \in \mathcal{R}^{n \times n\ell}$$

Theorem 2.3.16 ([MP12] trapdoor)**.** *Let $\sigma > 0$, $\bar{\mathbf{A}} \xleftarrow{\$} \mathcal{R}_q^{n \times 2n}$, let $\mathbf{R} \leftarrow D_{\mathcal{R},\sigma}^{2n \times n\ell}$, and $\mathbf{H} \in \mathcal{R}_q^{n \times n}$ with $\mathbf{H}$ invertible in $\mathcal{R}_q^{n \times n}$. Let $m = n(\ell + 2)$, and*

$$\mathbf{A} := \left[\bar{\mathbf{A}} \mid \bar{\mathbf{A}}\mathbf{R} - \mathbf{H}\mathbf{G}\right]$$

The matrix $\mathbf{A}$ is indistinguishable from uniform under the M-$LWE_{q,n,\sigma}$ assumption, and the lattice $\mathcal{L}_q^\perp(\mathbf{A})$ has an efficiently computable basis $\mathbf{B}$ such that $\|\tilde{\mathbf{B}}\|_{\max} \leq (s_1(\mathbf{R}) + 1)\sqrt{B^2 + 1}$

From theorems 2.3.14 and 2.3.16, one can sample a matrix $\mathbf{A} \in \mathcal{R}_q^{n \times n(\ell+2)}$ indistinguishable from uniform, and a trapdoor $\mathbf{B}$ for the lattice $\mathcal{L}_q^{\perp}(\mathbf{A})$. For any $\mathbf{t} \in \mathcal{R}_q^n$, and $\sigma \geq \frac{1}{\pi}\sqrt{\frac{1}{2}\log\left(2n(1+\frac{1}{\varepsilon})\right)}(s_1(\mathbf{R})+1)\sqrt{B^2+1}$, one can then sample $\mathbf{v} \leftarrow D_{\mathcal{R},\sigma}^{n(\ell+2)}$ such that

$$\mathbf{Av} = \mathbf{t} \mod q$$

by using $\mathbf{Sample}(\mathbf{B}, \sigma, \mathbf{c})$ for any $\mathbf{c} \in \mathcal{R}_q^{n(\ell+2)}$ such that $\mathbf{Ac} = \mathbf{t} \mod q$.

NTRU Trapdoors. A simple construction for trapdoored lattices is through the NTRU assumption. Consider the matrix $\mathbf{A} = \begin{bmatrix} 1 & h \end{bmatrix}$, for $h = f/g \leftarrow \mathcal{N}_{q,\sigma}$ as per definition 2.3.13. It is shown in [DLP14] that by choosing a standard deviation $\sigma = 1.17\sqrt{\frac{q}{2d}}$ for the coefficients of f and g, one can efficiently compute a basis $\mathbf{B}$ of $\mathcal{L}_q^{\perp}(\mathbf{A})$ such that $\|\tilde{\mathbf{B}}\|_{\max} \leq 1.17\sqrt{q}$. While such a standard deviation is not sufficient to argue that the NTRU assumption is as hard as the R-LWE assumption, there is no known distinguishing attack that exploits the NTRU structure of $\mathbf{A}$ for these parameters.

2.4 Basic Cryptographic Primitives

2.4.1 Commitments

Definition 2.4.1 (Commitment scheme)**.** *A commitment scheme is a tuple of three probabilistic polynomial time algorithms* $(\mathbf{CSetup}, \mathbf{Com}, \mathbf{Open})$ *such that:*

- $\mathbf{CSetup}(1^\lambda) \mapsto CK$ *generates the public commitment key.*
- $\mathbf{Com}_{CK}(m) \mapsto (c, d)$ *generates a commitment* c *for the message* m *under the key* CK*, and an opening* d *to this commitment.*
- $\mathbf{Open}_{CK}(c, d) \mapsto \tilde{m}$ *opens the commitment* c *using the opening* d*. If the commitment is not valid then* $\tilde{m} = \perp$*.*

For completeness opening any well formed commitment/opening tuple should recover the original message with overwhelming probability, i.e. for any $CK \leftarrow \mathbf{CSetup}(1^\lambda)$ and any message m: $\mathbf{Open}_{CK}(\mathbf{Com}_{CK}(m)) = m$ with overwhelming probability over the random coins of $\mathbf{Com}_{CK}$. For security the commitment should be binding, meaning that a commitment c should be "openable" to at most one message, and hiding, meaning that one should not be able to distinguish commitments to two different messages. Both the binding and the hiding property can be either statistical or computational, we will only consider commitment schemes that are computationally binding and hiding as we aim to obtain efficient constructions.

Definition 2.4.2 (Security)**.** *A valid commitment scheme should achieve the following properties:*

- (Computational) Hiding. *It is hard for any PPT adversary* $\mathcal{A}$ *to generate to messages* $m_0 \neq m_1$ *such that* $\mathcal{A}$ *can distinguish between* $\mathbf{Com}(m_1)$ *and* $\mathbf{Com}(m_2)$*. Formally:*

$$\Pr\left[b = b' \;\middle|\; \begin{array}{c} CK \leftarrow \mathbf{CSetup}(1^\lambda), (m_0, m_1, St) \leftarrow \mathcal{A}(CK), b \xleftarrow{\$} \{0,1\} \\ (c, d) \leftarrow \mathbf{Com}_{CK}(m_b), b' \leftarrow \mathcal{A}(c, St) \end{array} \right] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

- (Computational) Binding. *It is hard for any PPT adversary $\mathcal{A}$ to come up with a tuple (c, d, d') such that* $\mathbf{Open}(c, d) = m \neq \perp$, $\mathbf{Open}(c, d') = m' \neq \perp$ *and* $m \neq m'$. *Formally:*

$$\Pr\left[m \neq m' \wedge m, m' \neq \perp \;\middle|\; \begin{array}{c} CK \leftarrow \mathbf{CSetup}(1^\lambda), (c, d, d') \leftarrow \mathcal{A}(CK) \\ m \leftarrow \mathbf{Open}_{CK}(c, d), m' \leftarrow \mathbf{Open}_{CK}(c, d') \end{array} \right] \leq \mathrm{negl}(\lambda)$$

In most applications the commitment key CK will be clear from the context and we will thus omit it when referring to the algorithms **Com** and **Open**.

Lattice-Based Commitments. We consider a commitment scheme with message space $\mathcal{R}_q^l$.

$\mathbf{CSetup}(1^\lambda)$:

- Generate public parameters $pp := (q, n, m, l, \sigma, B_{Com}, \bar{\mathcal{C}})$
- Sample $\mathbf{A}'_1 \xleftarrow{\$} \mathcal{R}_q^{n \times (n-m)}$, set $\mathbf{A}_1 = \begin{bmatrix} \mathbf{I}_n & \mathbf{A}'_1 \end{bmatrix} \in \mathcal{R}_q^{n \times m}$
- Sample $\mathbf{A}'_2 \xleftarrow{\$} \mathcal{R}_q^{l \times (m-n-l)}$, set $\mathbf{A}_2 = \begin{bmatrix} \mathbf{0}^{l \times n} & \mathbf{I}_l & \mathbf{A}'_2 \end{bmatrix} \in \mathcal{R}_q^{l \times m}$
- Set $\mathbf{A} := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$
- Output $(pp, \mathbf{A})$

$\mathbf{Com}(\mathbf{m} \in \mathcal{R}_q^l)$

- Sample $\mathbf{r} \leftarrow D^m_{\mathcal{R},\sigma}$
- Output $\left(\mathbf{Ar} + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}, \mathbf{r} \right)$

$\mathbf{Open}(\mathbf{t} \in \mathcal{R}_q^{n+l}, \mathbf{r} \in \mathcal{R}^m, \bar{c} \in \bar{\mathcal{C}})$

- Parse $\mathbf{t}$ as $\begin{bmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{bmatrix}$ with $\mathbf{t}_1 \in \mathcal{R}_q^n$ and $\mathbf{t}_2 \in \mathcal{R}_q^l$
- If $\bar{c}\mathbf{t}_1 - \mathbf{A}_1\mathbf{r} \neq 0$, or $\|\mathbf{r}\| > B_{Com}$, or $\bar{c} \notin \bar{\mathcal{C}}$, output $\perp$
- Else, output $\mathbf{t}_2 - \bar{c}^{-1}\mathbf{A}_2\mathbf{r}$

We could consider an opening algorithm that does not include the $\bar{c}$ term. However defining **Open** like this will be useful for Chapter 3, where the knowledge extractor of our proofs of knowledge will only be able to exctract $\bar{c}\mathbf{t}$ for some small polynomial $\bar{c}$. In practice the set $\bar{C}$ will be the set of difference of challenges of our proofs (except 0) and we will always ensure that all the elements in $\bar{\mathcal{C}}$ are small and invertible. For ease of presentation we will sometimes write $\mathbf{t} := \mathbf{Com}(\mathbf{m}; \mathbf{r})$ to denote the commitment of $\mathbf{m}$ with the randomness $\mathbf{r}$.

Lemma 2.4.3 (Correctness)**.** *Let $B_{Com} > \sqrt{2dm}\sigma$ then* $(\mathbf{CSetup}, \mathbf{Com}, \mathbf{Open})$ *is correct with overwhelming probability.*

Proof. For correctness we will need that well formed commitments can be opened with overwhelming probability. Let $(\mathbf{t}, \mathbf{r}) := \mathbf{Com}(\mathbf{m})$, if $\|\mathbf{r}\| \leq B_{Com}$ then it is clear that $\mathbf{Open}(\mathbf{t}, \mathbf{r}, 1) = \mathbf{m}$. We know by Lemma 2.3.5 that $\|\mathbf{r}\| \leq \sqrt{2dm}\sigma \leq B_{Com}$ with overwhelming probability. □

Lemma 2.4.4 (Hiding)**.** *For any $\mathbf{m}, \mathbf{m}' \in \mathcal{R}_q^l$, if there is an adversary $\mathcal{A}$ who can distinguish between $\mathbf{Com}(\mathbf{m})$ and $\mathbf{Com}(\mathbf{m}')$ with advantage ε, then there exists an algorithm $\mathcal{A}'$ who runs in the same time and breaks $M\text{-}LWE_{q,m-n-l,\sigma}$ with probability $\varepsilon/2$.*

Proof. Given an instance $(\mathbf{B}, \mathbf{y}) \in \mathcal{R}_q^{(n+l)\times(m-n-l)} \times \mathcal{R}_q^{n+l}$ of $M\text{-}LWE_{q,m-n-l,\sigma}$, $\mathcal{A}'$ samples $\mathbf{R} \xleftarrow{\$} \mathcal{R}_q^{n\times l}$ and sets:

$$\mathbf{A} := \begin{bmatrix} \mathbf{I}_n & \mathbf{R} \\ \mathbf{0}^{l\times n} & \mathbf{I}_l \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I}_{n+l} & \mathbf{B} \end{bmatrix}$$

$\mathcal{A}'$ sends $\mathbf{A}$ to the adversary $\mathcal{A}$ and receives messages $\mathbf{m}_0, \mathbf{m}_1 \in \mathcal{R}_q^l$ such that $\mathbf{m}_0 \neq \mathbf{m}_1$. $\mathcal{A}'$ samples $b \xleftarrow{\$} \{0, 1\}$, computes:

$$\mathbf{t} = \begin{bmatrix} \mathbf{I}_n & \mathbf{R} \\ \mathbf{0}^{l\times n} & \mathbf{I}_l \end{bmatrix} \mathbf{y} + \begin{bmatrix} \mathbf{0} \\ \mathbf{m}_b \end{bmatrix}$$

and sends $\mathbf{t}$ to $\mathcal{A}$. When $\mathcal{A}$ returns b', $\mathcal{A}'$ returns 1 if $b' = b$ and 0 otherwise.
We first show that the public commitment matrix $\mathbf{A}$ is taken according to the correct distribution. Rewrite $\mathbf{B}$ as $\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$ with $\mathbf{B}_1 \in \mathcal{R}_q^{n\times(m-n-l)}$, and $\mathbf{B}_2 \in \mathcal{R}_q^{l\times(m-n-l)}$. Then we have:

$$\mathbf{A} = \begin{bmatrix} \mathbf{I}_n & \mathbf{R} & \mathbf{B}_1 + \mathbf{R}\mathbf{B}_2 \\ \mathbf{0}^{l\times n} & \mathbf{I}_l & \mathbf{B}_2 \end{bmatrix}$$

Since $\mathbf{R}, \mathbf{B}_1$, and $\mathbf{B}_2$ are uniform and independent, the distribution of $\mathbf{A}$ is identical to the one output by **CSetup**.
If $\mathbf{y}$ is uniform in $\mathcal{R}_q^{n+l}$ then $\mathbf{t}$ is uniform in $\mathcal{R}_q^{n+l}$ and $b' = b$ with probability exactly $1/2$. However if $\mathbf{y} = \begin{bmatrix} \mathbf{I}_{n+l} & \mathbf{B} \end{bmatrix} \mathbf{r}$, then :

$$\mathbf{t} = \mathbf{A}\mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \mathbf{m}_b \end{bmatrix}$$

and $\mathcal{A}$ will output $b' = b$ with probability $1/2 + \varepsilon$. $\mathcal{A}'$ therefore has advantage $\varepsilon/2$ in the $M\text{-}LWE_{q,m-n-l,\psi}$ problem. □

Lemma 2.4.5 (Binding)**.** *Let $B_{\mathcal{C}} \geq \max_{c\in\mathcal{C}}(\|c\|_1)$. If there is an adversary $\mathcal{A}$ who can output a commitment $\mathbf{t}$ with two valid openings $(\mathbf{m}, \mathbf{r}, c)$ and $(\mathbf{m}', \mathbf{r}', c')$ such that $\mathbf{m} \neq \mathbf{m}'$ with probability ε, then there is an algorithm $\mathcal{A}'$ who can break $M\text{-}SIS_{q,m,4B_{\mathcal{C}}B_{Com}}$ in the same time and with advantage ε.*

Proof. Given an instance $\mathbf{A}_1 = \begin{bmatrix} \mathbf{I}_n & \mathbf{A}_1' \end{bmatrix} \mathcal{R}_q^{n\times m}$ of $M\text{-}SIS_{q,m,2B_{\mathcal{C}}B_{Com}}$. $\mathcal{A}'$ samples $\mathbf{A}_2 := \begin{bmatrix} 0^{l\times n} & \mathbf{I}_l & \mathbf{A}_2' \end{bmatrix}$ as per **CSetup** and outputs $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$. When $\mathcal{A}$ comes up with a commitment $\mathbf{t}$ such that $\mathbf{Open}(\mathbf{t}, \mathbf{r}, c) = \mathbf{m} \neq \perp$, $\mathbf{Open}(\mathbf{t}, \mathbf{r}', c') = \mathbf{m}' \neq \perp$, and $\mathbf{m} \neq \mathbf{m}'$. We have

$\mathbf{m} = \mathbf{t}_2 - c^{-1}\mathbf{A}_2\mathbf{r}$ and $\mathbf{m}' = \mathbf{t}_2 - c'^{-1}\mathbf{A}_2\mathbf{r}'$, which implies $cc'(\mathbf{m} - \mathbf{m}') = \mathbf{A}_2(c\mathbf{r}' - c'\mathbf{r})$, and since $\mathbf{m} \neq \mathbf{m}'$ we get:

$$c\mathbf{r}' - c'\mathbf{r} \neq 0$$

Additionally the verification equations of the opening entail that $\mathbf{A}_1\mathbf{r} = c\mathbf{t}_1$ and $\mathbf{A}_1\mathbf{r}' = c'\mathbf{t}_1$, and thus:

$$\mathbf{A}_1(c'\mathbf{r} - c\mathbf{r}') = \mathbf{0}$$

We also know that $\|\mathbf{r}\|, \|\mathbf{r}'\| \leq B_{Com}$, and $\|c\|_1, \|c'\|_1 \leq 2B_{\mathcal{C}}$, from which we obtain $\|c\mathbf{r}' - c'\mathbf{r}\| \leq 4B_{Com}B_{\mathcal{C}}$. □

Commitments with different moduli. We present a commitment scheme identical to the previous one except for the fact that the n first rows of the commitment will be taken modulo q_1, and the l last rows will be taken modulo q_2. In most applications using different moduli does not give better parameters or security, but we will see in Chapter 4 that it can be of use in our group signature scheme.

CSetup(1^λ) :

- Generate public parameters $pp := (q_1, q_2, n, m, l, \sigma, B_{Com}, \mathcal{C})$
- Sample $\mathbf{A}'_1 \xleftarrow{\$} \mathcal{R}_{q_1}^{n\times(m-n)}$, set $\mathbf{A}_1 = \begin{bmatrix}\mathbf{I}_n & \mathbf{A}'_1\end{bmatrix} \in \mathcal{R}_{q_1}^{n\times m}$
- Sample $\mathbf{A}'_2 \xleftarrow{\$} \mathcal{R}_{q_2}^{l\times(m-n-l)}$, set $\mathbf{A}_2 = \begin{bmatrix}\mathbf{0}^{l\times n} & \mathbf{I}_l & \mathbf{A}'_2\end{bmatrix} \in \mathcal{R}_{q_2}^{l\times m}$
- Set $\mathbf{A} := \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{A}_2\end{bmatrix}$
- Output $(pp, \mathbf{A})$

For the hiding property to hold we will need to artificially modify the distribution of the randomness in the commitment. Let $\sigma' := \sqrt{\frac{q_1}{q_2}\sigma + 1 + 2d(m-n-l)\sigma^2}$.

Com$(\mathbf{m} \in \mathcal{R}_{q_2}^l)$

- Sample $\mathbf{r} \leftarrow D_{\mathcal{R},\sigma'}^n \times D_{\mathcal{R},\sigma}^{m-n}$
- Output $\left(\mathbf{A}\mathbf{r} + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix}, \mathbf{r}\right)$

The fact that the randomness is skewed is an artefact of the modulus switching we use in the proof of hiding. In practice we will use spherical gaussians as it is very unlikely that different moduli would help in attacking the scheme. **Open**$(\mathbf{t} \in \mathcal{R}_{q_1}^n \times \mathcal{R}_{q_2}^l, \mathbf{r} \in \mathcal{R}^m, \bar{c} \in \bar{\mathcal{C}})$

- Parse $\mathbf{t}$ as $\begin{bmatrix}\mathbf{t}_1 \\ \mathbf{t}_2\end{bmatrix}$ with $\mathbf{t}_1 \in \mathcal{R}_{q_1}^n$ and $\mathbf{t}_2 \in \mathcal{R}_{q_2}^l$
- If $\bar{c}\mathbf{t}_1 - \mathbf{A}_1\mathbf{r} \neq 0$, or $\|\mathbf{r}\| > B_{Com}$, or $\bar{c} \notin \bar{\mathcal{C}}$, output $\perp$
- Else, output $\mathbf{t}_2 - \bar{c}^{-1}\mathbf{A}_2\mathbf{r}$

Lemma 2.4.6 (Correctness)**.** *Let $B_{Com} > \sqrt{2dm}\sigma$ then* (**CSetup**, **Com**, **Open**) *is correct with overwhelming probability.*

Proof. For correctness we will need that well formed commitments can be opened with overwhelming probability. Let $(\mathbf{t}, \mathbf{r}) := \mathbf{Com}(\mathbf{m})$, if $\|\mathbf{r}\| \leq B_{Com}$ then it is clear that $\mathbf{Open}(\mathbf{t}, \mathbf{r}, 1) = \mathbf{m}$. We know by Lemma 2.3.5 that $\|\mathbf{r}\| \leq \sqrt{2dm}\sigma \leq B_{Com}$ with overwhelming probability. □

Lemma 2.4.7 (Binding)**.** *Let $B_{\mathcal{C}} \geq \max_{c \in \mathcal{C}} (\|c\|_1)$. If there is an adversary $\mathcal{A}$ who can output a commitment $\mathbf{t}$ with two valid openings $(\mathbf{m}, \mathbf{r}, c)$ and $(\mathbf{m}', \mathbf{r}', c')$ such that $\mathbf{m} \neq \mathbf{m}'$ with probability ε, then there is an algorithm $\mathcal{A}'$ who can break M-SIS$_{q_1,m,4B_{\mathcal{C}}B_{Com}}$ in the same time and with advantage ε.*

Proof. The proof is identical to the one with a single modulus. □

Lemma 2.4.8 (Hiding)**.** *For any $\mathbf{m}, \mathbf{m}' \in \mathcal{R}_{q_2}^l$, if there is an adversary $\mathcal{A}$ who can distinguish between $\mathbf{Com}(\mathbf{m})$ and $\mathbf{Com}(\mathbf{m}')$ with advantage ε, then there exists an algorithm $\mathcal{A}'$ who runs in the same time and breaks M-LWE$_{q_2,m-n-l,\sigma}$ with probability $\varepsilon/2$.*

Proof. Given an instance $(\mathbf{B}, \mathbf{y}) \in \mathcal{R}_{q_2}^{(n+l)\times(m-n-l)} \times \mathcal{R}_{q_2}^{n+l}$ of M-LWE$_{q_2,m-n-l,\sigma}$, parse $\mathbf{B}$ and $\mathbf{y}$ as $\begin{bmatrix}\mathbf{B}_1 \\ \mathbf{B}_2\end{bmatrix}$ and $\begin{bmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{bmatrix}$. Let $\rho : \mathbb{R} \to \mathbb{Z}$ be a randomized rounding function which maps $x \in \mathbb{R}$ to $\rho(x) \leftarrow \lfloor x \rfloor + B_{x-\lfloor x \rfloor}$, where $B_{x-\lfloor x \rfloor}$ is a Bernouilli variable which outputs 1 with probability $x - \lfloor x \rfloor$. Remark that for $q_1 \leq q_2$, $\rho\left(\mathcal{U}(\frac{q_1}{q_2}\mathbb{Z}_{q_2})\right) = \mathcal{U}(\mathbb{Z}_{q_1})$. Let $\mathbf{B}_1' := \rho\left(\frac{q_1}{q_2}\mathbf{B}_1\right)$ and $\mathbf{y}_1' := \rho(\frac{q_1}{q_2}\mathbf{y}_1)$. $\mathcal{A}'$ samples $\mathbf{R} \xleftarrow{\$} \mathcal{R}_{q_1}^{n\times l}$ and sets:

$$\mathbf{A} := \begin{bmatrix}\mathbf{I}_n & \mathbf{R} \\ \mathbf{0}^{l\times n} & \mathbf{I}_l\end{bmatrix} \cdot \begin{bmatrix}\mathbf{I}_n & \mathbf{0}^{n\times l} & \mathbf{B}_1' \\ \mathbf{0}^{l\times n} & \mathbf{I}_l & \mathbf{B}_2\end{bmatrix}$$

where the products are done over the integers and then taken modulo q_1 for the top part and modulo q_2 for the bottom part. $\mathcal{A}'$ sends $\mathbf{A}$ to the adversary $\mathcal{A}$ and receives messages $\mathbf{m}_0, \mathbf{m}_1 \in \mathcal{R}_{q_2}^l$ such that $\mathbf{m}_0 \neq \mathbf{m}_1$. $\mathcal{A}'$ samples $b \xleftarrow{\$} \{0, 1\}$, computes:

$$\mathbf{t} = \begin{bmatrix}\mathbf{I}_n & \mathbf{R} \\ \mathbf{0}^{l\times n} & \mathbf{I}_l\end{bmatrix} \begin{bmatrix}\mathbf{y}_1' \\ \mathbf{y}_2\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}_b\end{bmatrix}$$

where the products are done over the integers and then taken modulo q_1 for the top part and modulo q_2 for the bottom part, and sends $\mathbf{t}$ to $\mathcal{A}$. When $\mathcal{A}$ returns b', $\mathcal{A}'$ returns 1 if $b' = b$ and 0 otherwise.
We first show that the public commitment matrix $\mathbf{A}$ is taken according to the correct distribution. We have:

$$\mathbf{A} = \begin{bmatrix}\mathbf{I}_n & \mathbf{R} & \mathbf{B}_1' + \mathbf{R}\mathbf{B}_2 \\ \mathbf{0}^{l\times n} & \mathbf{I}_l & \mathbf{B}_2\end{bmatrix}$$

Since $\mathbf{B}_1'$ is uniform modulo q_1 and independent from $\mathbf{R}$ and $\mathbf{B}_2$, $\mathbf{B}_1' + \mathbf{R}\mathbf{B}_2 \mod q_1$ is uniform modulo q_1. Since $\mathbf{R}$ and $\mathbf{B}_2$ are also uniform, the distribution of $\mathbf{A}$ is identical to the one output by **CSetup**.
If $\mathbf{y}$ is uniform in $\mathcal{R}_{q_2}^{n+l}$ then $\mathbf{y}_1'$ is uniform in $\mathcal{R}_{q_1}^n$ and $\mathbf{t}$ is uniform in $\mathcal{R}_{q_1}^{n+l}$ and $b' = b$ with probability exactly 1/2. However if $\mathbf{y} = \begin{bmatrix}\mathbf{I}_{n+l} & \mathbf{B}\end{bmatrix}\mathbf{r}$, write $\mathbf{r}$ as $\begin{bmatrix}\mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3\end{bmatrix}$, with $\mathbf{r}_1 \in \mathcal{R}^n$, $\mathbf{r}_2 \in \mathcal{R}^l$,

and $\mathbf{r}_3 \in \mathcal{R}^{m-n-l}$. Applying ρ component-wise to $\frac{q_1}{q_2}(\mathbf{B}_1, \mathbf{B}_1\mathbf{r}_3 + \mathbf{r}_1)$ we get:

$$\begin{aligned}\left(\frac{q_1}{q_2}\mathbf{B}_1 + \Delta, \frac{q_1}{q_2}B_1\mathbf{r}_3 + \frac{q_1}{q_2}\mathbf{r}_1 + \delta\right) &= \left(\frac{q_1}{q_2}\mathbf{B}_1 + \Delta, \left(\frac{q_1}{q_2}B_1 + \Delta\right)\mathbf{r}_3 + \frac{q_1}{q_2}\mathbf{r}_1 + \delta - \Delta\mathbf{r}_3\right) \\ &= \left(\mathbf{B}_1', \mathbf{B}_1'\mathbf{r}_3 + \frac{q_1}{q_2}\mathbf{r}_1 + \delta - \Delta\mathbf{r}_3\right) \\ &= (\mathbf{B}_1', \mathbf{B}_1'\mathbf{r}_3 + \mathbf{r}_1')\end{aligned}$$

where $\mathbf{r}_1'$ is subgaussian with parameter $\sqrt{\frac{q_1}{q_2}\alpha + 1 + \|\mathbf{r}_3\|^2} \leq \sigma'$. Setting $\mathbf{r}' = \begin{bmatrix}\mathbf{r}_1' \\ \mathbf{r}_2 \\ \mathbf{r}_3\end{bmatrix}$ we have that

$$\mathbf{t} = \mathbf{A}\mathbf{r}' + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}_b\end{bmatrix}$$

is distributed according to $\mathbf{Com}(\mathbf{m}_b)$, and $\mathcal{A}$ will output $b' = b$ with probability $1/2 + \varepsilon$. $\mathcal{A}'$ therefore has advantage $\varepsilon/2$ in the M-LWE$_{q_2,m-n-l,\sigma}$ problem. □

2.4.2 Public Key Encryption

Definition 2.4.9 (Public Key Encryption)**.** *A public key encryption scheme is a tuple of three probabilistic polynomial time algorithms* $(\mathbf{PKESetup}, \mathbf{Enc}, \mathbf{Dec})$ *such that:*

- $\mathbf{PKESetup}(1^\lambda) \mapsto (sk, pk)$ *Generates the secret key and the public key.*
- $\mathbf{Enc}(pk, m) \mapsto ct$ *outputs a ciphertext ct which encrypts the message m under the public key pk.*
- $\mathbf{Dec}(sk, ct) \mapsto \tilde{m}$ *Decrypts the ciphertext ct to a message m using the secret key sk. Outputs the error symbol* $\perp$ *if decryption fails.*

For completeness decrypting any well formed ciphertext should recover the original message with overwhelming probability, i.e. for any $(pk, sk) \leftarrow \mathbf{PKESetup}(1^\lambda)$ and any message m: $\mathbf{Dec}(sk, \mathbf{Enc}(pk, m)) = m$ with overwhelming probability over the random coins of **Enc**.

$\mathsf{Exp}^{ind-cpa-b}(\mathcal{A}, \lambda)$:
$(sk, pk) \leftarrow \mathbf{PKESetup}(1^\lambda)$
$(m_0, m_1, st) \leftarrow \mathcal{A}(pk)$
$ct^* \leftarrow \mathbf{Enc}(pk, m_b)$
$b' \leftarrow \mathcal{A}(ct^*, st)$
Return b'

Figure 2.1: Experiment for IND-CPA security

Definition 2.4.10 (Indistinguishability under chosen-plaintext attack)**.** *For a PKE scheme* $(\mathbf{PKESetup}, \mathbf{Enc}, \mathbf{Dec})$*, we define security against chosen-plaintext attacks* (IND-CPA) *with adversary* $\mathcal{A}$ *via the experiment of Figure 2.1. The advantage of* $\mathcal{A}$ *is:*

$$\mathsf{Adv}^{ind-cpa}(\mathcal{A}, \lambda) := \left| \Pr_{b \xleftarrow{\$} \{0,1\}} \left[\mathsf{Exp}^{ind-cpa-b}(\mathcal{A}, \lambda) = b\right] - \frac{1}{2} \right|$$

The scheme is secure if this advantage is negligible in λ for any PPT adversary $\mathcal{A}$.

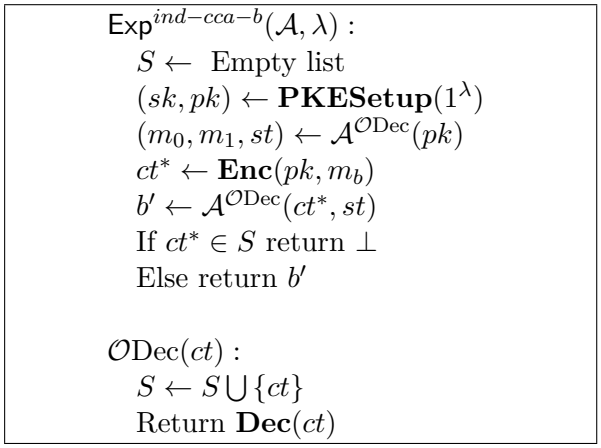

Figure 2.2: Experiment for IND-CPA security

Definition 2.4.11 (Indistinguishability under chosen-ciphertext attack)**.** *For a PKE scheme* $(\mathbf{PKESetup}, \mathbf{Enc}, \mathbf{Dec})$*, we define security against chosen-ciphertext attacks* (IND-CCA-2) *with adversary $\mathcal{A}$ via the experiment of Figure 2.2. The advantage of $\mathcal{A}$ is:*

$$\mathsf{Adv}^{ind-cca}(\mathcal{A},\lambda) := \left| \Pr_{b \xleftarrow{\$} \{0,1\}} \left[\mathsf{Exp}^{ind-cca-b}(\mathcal{A},\lambda) = b \right] - \frac{1}{2} \right|$$

The scheme is secure if this advantage is negligible in λ for any PPT adversary $\mathcal{A}$.

Lattice-Based PKE. Lyubashevsky introduced in [LPR13] a ring variant of the seminal CPA-secure LWE encryption scheme of Regev [Reg05]. We present an adaptation of this scheme to module lattices, we also modify the scheme to accommodate for larger message spaces.

$\mathbf{PKESetup}(1^\lambda)$:

- Generate public parameters $pp := (q, p, n, l, \sigma)$
- Sample $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times n}$
- Sample $\mathbf{S}, \mathbf{E} \leftarrow D_{\mathcal{R},\sigma}^{n \times l}$
- Set $\mathbf{B} := \mathbf{AS} + \mathbf{E}$
- Output $(\mathbf{s}, (pp, \mathbf{A}, \mathbf{B}))$

$\mathbf{Enc}((pp, \mathbf{A} \in \mathcal{R}_q^{n \times n}, \mathbf{B} \in \mathcal{R}_q^{n \times l}), \mathbf{m} \in \mathcal{R}_q^l)$:

- Sample $\mathbf{r}, \mathbf{e}_1 \leftarrow D_{\mathcal{R},\sigma}^n$
- Sample $\mathbf{e}_2 \leftarrow D_{\mathcal{R},\sigma}^l$
- Set $\mathbf{u} := p(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1)$

- Set $\mathbf{v} := p(\mathbf{B}^T\mathbf{r} + \mathbf{e}_2) + \mathbf{m}$
- Output $(\mathbf{u}, \mathbf{v})$

$\mathbf{Dec}(\mathbf{S} \in \mathcal{R}_q^{n\times l}, (\mathbf{u} \in \mathcal{R}_q^n, \mathbf{v} \in \mathcal{R}_q^l))$:

- Output $\mathbf{v} - \mathbf{S}^T\mathbf{u} \mod p$

Lemma 2.4.12 (Correctness)**.** *Let the modulus q be such that:*

$$q > 64pnd\sqrt{2}\sigma^2$$

then the scheme $(\mathbf{PKESetup}, \mathbf{Enc}, \mathbf{Dec})$ *has overwhelming correctness.*

Proof. Decryption will be correct if $\mathbf{v} - \mathbf{S}^T\mathbf{u} = \mathbf{v} - \mathbf{S}^T\mathbf{u} \mod q$, indeed in that case we have:

$$\mathbf{v} - \mathbf{S}^T\mathbf{u} = p(\mathbf{E}^T\mathbf{r} + \mathbf{e}_2 - \mathbf{S}^T\mathbf{e}_1) + \mathbf{m}$$

and $\mathbf{v} - \mathbf{S}^T\mathbf{u} = \mathbf{m} \mod p$. Decryption will thus be correct if

$$\left\|p(\mathbf{E}^T\mathbf{r} + \mathbf{e}_2 - \mathbf{S}^T\mathbf{e}_1) + \mathbf{m}\right\|_\infty \leq q/2p.$$

Let $\mathbf{e}, \mathbf{s} \leftarrow D^n_{\mathcal{R},\sigma}$ be the first column of $\mathbf{E}$ and $\mathbf{S}$, by applying a union bound it is apparent that proving $\left\|p(\mathbf{e}^T\mathbf{r} + \mathbf{e}_2 - \mathbf{s}^T\mathbf{e}_1) + \mathbf{m}\right\|_\infty \leq q/2p$ with overwhelming probability is sufficient. Let $x, y \leftarrow D_{\mathcal{R},\sigma}$, we have $\|xy\|_\infty \leq \|x\|_1 \|y\|_\infty \leq \sqrt{d}\,\|x\|\,\|y\|_\infty$.
By union bound

$$\begin{aligned}\Pr\left[\|xy\|_\infty > 14d\sqrt{2}\sigma^2\right] &\leq \Pr\left[\|x\| > \sqrt{2d}\sigma\right] + \Pr\left[\|y\|_\infty > 14\sigma\right] \\ &\leq 2^{-d/4} + d2e^{-98}\end{aligned}$$

Since $\mathbf{e} \leftarrow D^n_{\mathcal{R},\sigma}$, we have $\left\|\mathbf{e}^T\mathbf{s}\right\|_\infty \leq \sum \|e_i s_i\|_\infty \leq 14nd\sqrt{2}\sigma^2$ with overwhelming probability. The same argument can be applied to $\left\|\mathbf{s}^T\mathbf{e}_1\right\|$. Since $\mathbf{m} \in \mathbb{Z}_p^n$ and $\mathbf{e}_2 \leftarrow D^k_{\mathcal{R},\sigma}$, we have:

$$\left\|p(\mathbf{e}^T\mathbf{r} + \mathbf{e}_2 - \mathbf{s}^T\mathbf{e}_1) + \mathbf{m}\right\|_\infty \leq 64pnd\sqrt{2}\sigma^2$$

□

Lemma 2.4.13 (IND-CPA Security)**.** *For any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ for the* M-LWE$_{q,n,\sigma}$ *such that:*

$$\mathsf{Adv}^{ind-cca}(\mathcal{A}, \lambda) \leq (k+1)\mathsf{Adv}^{\text{M-LWE}_{q,n,\sigma}}(\mathcal{B})$$

Proof. Let $\mathcal{A}$ be an adversary for the IND-CPA security of our scheme. We use a succession of games.
Game $\mathbf{G}_0$: In this game we sample $b \xleftarrow{\$} \{0,1\}$ run experiment $\mathsf{Exp}^{ind-cpa-b}(\mathcal{A}, \lambda)$.
Game $\mathbf{G}_{1,0<i\leq k}$: This game is identical to the previous one except for the fact the the i^th column of $\mathbf{B}$ is now taken uniformly in $\mathcal{R}_q^n$. Note that we had $\mathbf{b}_i := \mathbf{A}\mathbf{s}_i + \mathbf{e}_i$ in the previous game. The advantage of $\mathcal{A}$ in distinguishing this game from the previous is thus:

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{1,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{1,i-1}}\right| \leq \mathsf{Adv}^{\text{M-LWE}_{q,n,\sigma}}(\mathcal{A})$$

Game $\mathbf{G}_2$: In this game we replace $\mathbf{u}, \mathbf{v}$ by uniformly random vector. Note that since $\mathbf{B} \xleftarrow{\$} \mathcal{R}_q^{n\times k}$ in **Game** $\mathbf{G}_{1,k-1}$, we had $\left[\mathbf{A} \mid \mathbf{B}\right] \xleftarrow{\$} \mathcal{R}_q^{n\times(n+k)}$ and

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v}-\mathbf{m} \end{bmatrix} := \begin{bmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}$$

was a M-LWE$_{q,n,\sigma}$ sample. The advantage of distinguishing this game from the previous one is thus:

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_2} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{1,k-1}}\right| \leq \mathsf{Adv}^{\text{M-LWE}_{q,n,\sigma}}(\mathcal{A})$$

Since $\mathbf{v}$ is uniform in $\mathcal{R}_q^k$ in **Game** $\mathbf{G}_2$, $\mathcal{A}$ has no advantage in guessing b:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_2} = 0$$

By summing the successive inequalities we obtain the desired result. □

Chapter 3

Lattice-Based Zero-Knowledge

In this chapter we define zero-knowledge proofs of knowledge over lattices through the notion of Σ'-protocol introduced in [BCK+14]. We then present the basic "Fiat-Shamir with aborts" from [Lyu09] and give two variants: an "exact" protocol and an "approximate" protocol. We then show how to use this Σ'-protocol to prove knowledge of openings of commitments and linear relations on openings, we also construct a proof for the disjunction of statements. We present a proof that the opening of a message belongs to a subset of $\mathcal{R}$ that is fixed by a set of automorphisms.
Finally we present an exact proof with constant overhead for lattice-based one-way functions, this proof is amortized and requires $O(\lambda)$ equations to be efficient.

Contents

3.1 Definitions

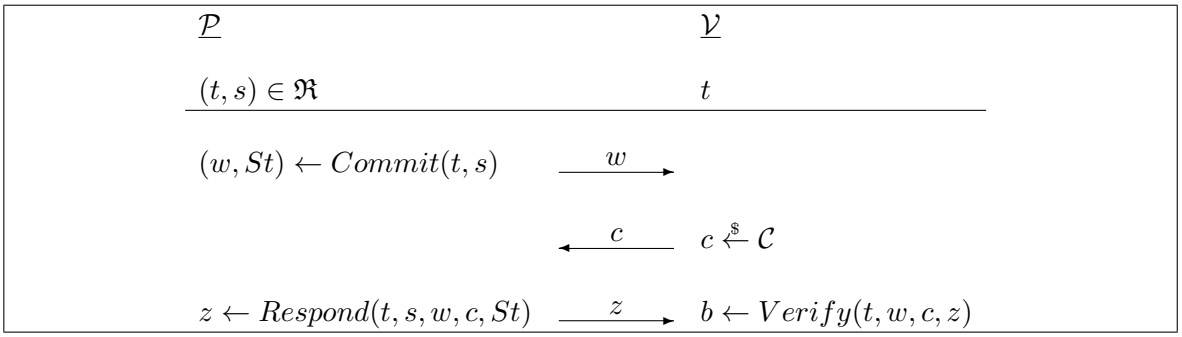

Figure 3.1: Three round form of a Σ-protocol.

A zero-knowledge proof of knowledge, abbreviated as ZKPoK, is an interactive protocol in which a prover $\mathcal{P}$ must convince a verifier $\mathcal{V}$ that a statement is true without revealing any other information about the statement. Consider a binary relation $\mathfrak{R}$ associated to an NP language $\mathfrak{L}$, i.e. for any statement $t \in \mathfrak{L}$ there exists a witness s such that $(t,s) \in \mathfrak{R}$. A ZKPoK for $\mathfrak{R}$ should achieve the following properties.

Completeness: A prover with witness s for $t \in \mathfrak{L}$ can convince the verifier.

Soundness: A prover cannot convince the verifier when $t \notin \mathfrak{L}$.

Zero-knowledge: The interaction should not reveal anything to the verifier except that $t \in \mathfrak{L}$. In particular, it should not reveal the prover's witness s.

Sigma' Protocols. Σ-protocols (introduced by Cramer in [Cra97]) are three round interactive proofs of knowledge between a prover $\mathcal{P}$ who knows $(t,s) \in \mathfrak{R}$ and a verifier $\mathcal{V}$ who knows t, c.f. Figure 3.1. In this thesis we will consider Σ'-protocols, introduced in [BCK+14], which are an adaptation of Σ-protocols to the context of lattice-based cryptography.

Definition 3.1.1 (Σ'-Protocol)**.** *Let $\mathfrak{R} \subset \mathfrak{R}'$ be two binary relations, let $(\mathcal{P}, \mathcal{V})$ be a two-party protocol where $\mathcal{V}$ is PPT. $(\mathcal{P}, \mathcal{V})$ is a Σ'-protocol for $\mathfrak{R}, \mathfrak{R}'$ with challenge set $\mathcal{C}$, public input t and private input s, if:*

- *Three-move form: $(\mathcal{P}, \mathcal{V})$ is of the form described in Figure 3.1*
- *Completeness: There is a constant $\alpha > 0$ such that whenever $(t,s) \in \mathfrak{R}$, the verifier accepts with probability at least α.*
- *Special soundness: There exists a PPT algorithm $\mathcal{E}$, called knowledge extractor, who on input two accepting transcripts, (w,c,z) and (w,c',z') such that $c \neq c'$, outputs $\bar{s}$ such that $(t,\bar{s}) \in \mathfrak{R}'$.*
- *Special honest-verifier zero-knowledge: There exists a PPT algorithm $\mathcal{S}$, called simulator, who on input $y \in \mathfrak{L}(\mathfrak{R})$ and $c \in \mathcal{C}$, outputs (w,s) such that (w,c,s) is indistinguishable from an accepting transcript generated by $(\mathcal{P}, \mathcal{V})$.*

This definition differs from that of Σ-protocols in two ways. The completeness property has been relaxed from overwhelming to constant, that is to accommodate the use of rejection sampling, in which the prover will only send the reply z with some probability that depends on (s, t, w, c) and abort the protocol otherwise. This step is necessary to ensure that the reply z is independent of the witness s and can be simulated for zero-knowledge, we will ensure that the probability that the protocol does not abort is overwhelmingly close to a constant (which we will take to be 1/3 for simplicity). The second difference is that the soundness extractor does not recover a witness from $\mathfrak{R}$ but a slightly larger relation $\mathfrak{R}'$, we will need this relaxation to accommodate the soundness slack that comes with lattice-based zero-knowledge proofs (i.e. the fact that extracted witness $\bar{s}$ will be "larger" than s) and because some of the Σ'-protocols we present will be "approximate" (c.f. Section 3.2).

The special soundness property achieved by Σ-protocol is a stronger variant of the knowledge soundness introduced by Bellare and Goldreich [BG93]. We recall this definition as some of our protocols do not achieve special soundness (we will still call such protocols Σ'-protocols for ease of presentation).

Definition 3.1.2 (Knowledge soundness)**.** *Let $\kappa : \{0,1\}^* \to [0,1]$, let $\mathcal{E}$ be a PPT extractor who gets an input $t \in \mathfrak{R}$ and black box access to a prover $\mathcal{P}^*$. $(\mathcal{P}, \mathcal{V})$ is κ-knowledge sound (or has soundness error κ) if for any prover $\mathcal{P}^*$ who has probability $\varepsilon(t)$ of convincing $\mathcal{V}$ on input t, there exists a constant c such that if $\varepsilon(t) > \kappa(t)$ then $\mathcal{E}$ can output a witness $\bar{s}$ such that $(t, \bar{s}) \in \mathfrak{R}'$ in expected time at most*

$$\frac{|t|^c}{\varepsilon(t) - \kappa(t)}$$

where access to P^* counts as a single step.

Lemma 3.1.3 ([Dam10] Theorem 1)**.** *Let $(\mathcal{P}, \mathcal{V})$ be a Σ'-protocol with challenge space $\mathcal{C}$. If $(\mathcal{P}, \mathcal{V})$ achieves special soundness then it has soundness error $1/|\mathcal{C}|$.*

Non-interactive proofs. The Fiat-Shamir transform was originally introduced in [FS87] to transform a three move proof of knowledge into a digital signature scheme. Using the Fiat-Shamir transform we can obtain a non-interactive proof of knowledge from a Σ'-protocol. We will define the non-interactive variant of $(\mathcal{P}, \mathcal{V})$ as the pair of algorithms $\Pi(t; s)$, $\mathbf{Verify}(t, c, z)$ given in Algorithm 2 and Algorithm 3, where H is a collision resistant random oracle.

Algorithm 2 $\Pi(t, s)$

Require: Public information: t. Private information: s such that $(t, s) \in \mathfrak{R}$

1: $(w, St) \leftarrow Commit(t, s)$
2: $c \leftarrow H(t, w, c)$
3: $z \leftarrow Respond(t, s, w, c, St)$
4: **return** (w, c, z)

Algorithm 3 Verify(t, c, z)

Require: Public information: $t \in \mathfrak{L}(\mathfrak{R})$, $(w, c, z) \leftarrow \Pi(t; s)$.

1: **if** $c = H(t, w, c)$ **then**
2: **return** $Verify(t, w, c, z)$
3: **elsereturn** 0

3.2 Exact and Approximate ZKPoKs

3.2.1 Σ'-Protocol for one way functions

Our first ZKPoK will consist in proving knowledge of a small preimage $\mathbf{s} \in \mathcal{R}^m$ for an image $\mathbf{t} \in \mathcal{R}_q^n$ of the one-way function defined by a random matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$. We present in Figure 3.1 a basic Σ'-protocol for the following relations:

$$\mathfrak{R} = \left\{(\mathbf{t}, \mathbf{s}, 1) \in \mathcal{R}_q^n \times \mathcal{R}^m \times \mathcal{R} \mid \mathbf{As} = \mathbf{t},\ \|\mathbf{s}\| \leq B\right\}$$
$$\mathfrak{R}' = \left\{(\mathbf{t}, \mathbf{s}, \bar{c}) \in \mathcal{R}_q^n \times \mathcal{R}^m \times \mathcal{R} \mid \mathbf{As} = \mathbf{t}\bar{c},\ \|\mathbf{s}\| \leq 2B', \bar{c} \in \bar{\mathcal{C}}\right\}$$

where $\bar{\mathcal{C}}$ is the set of differences of elements of $\mathcal{C}$ except for 0. If $\mathcal{R}$ is taken to be $\mathbb{Z}$ then the challenge set we bill fixed to $\mathcal{C} = \{0, 1\}$ (and thus $\bar{\mathcal{C}} = \{1\}$) resulting in a protocol with soundness $1/2$. If $\mathcal{R}$ is a polynomial ring then one can chose $\mathcal{C}$ to be a larger set (e.g. polynomials of small norm) to obtain soundness $1/|\mathcal{C}|$, this comes at a cost since the set $\bar{\mathcal{C}}$ will also be large.

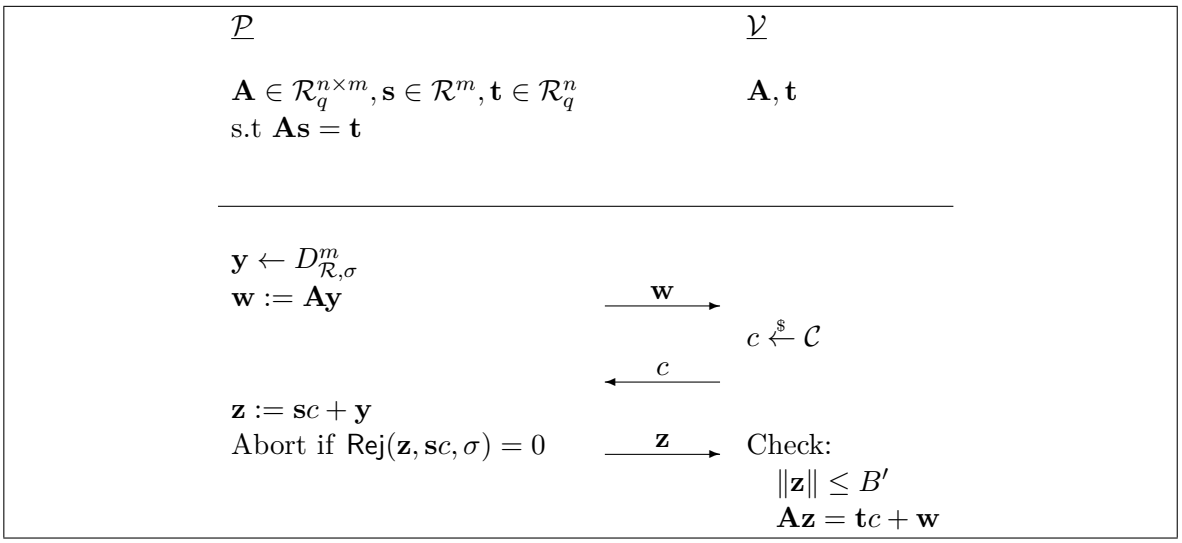

Figure 3.2: Generic ZKPoK for lattice-based one-way functions.

Theorem 3.2.1. *Let* $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$ *and* $\mathbf{s} \in \mathcal{R}^m$ *such that* $\mathbf{t} := \mathbf{As}$ *and* $\|\mathbf{s}\| \leq B$. *Let* $B_{\mathcal{C}} \geq \max_{c \in \mathcal{C}} (\|c\|_1)$, $\sigma \geq 11 B_{\mathcal{C}} B$, *and* $B' \geq \sqrt{2md}\sigma$. *The protocol* $\Pi_{OWF}(\mathbf{A}, \mathbf{t}; \mathbf{s})$ *of Figure 3.1 is a* Σ'*-protocol for relations* $\mathfrak{R}$ *and* $\mathfrak{R}'$ *with success probability* $1/3$ *and soundness error* $1/|\mathcal{C}|$.

Proof.

Correctness: Using lemma 2.3.6 we know that since $\sigma \geq 11B_{\mathcal{C}}B \geq 11\|\mathbf{s}c\|$ the rejection step will accept with probability overwhelmingly close to $1/3$ and the vector $\mathbf{z}$ output will be statistically close to $D^m_{\mathcal{R},\sigma}$. By lemma 2.3.5 we have $\|\mathbf{z}\| \leq \sqrt{2md}\sigma \leq B'$ with overwhelming probability. It is clear that in the honest protocol the response is computed in such a way that the verification equation $\mathbf{Az} = \mathbf{t}c + \mathbf{w}$ stands.

Special Soundness: Let $\mathbf{w}, c, \mathbf{z}$ and $\mathbf{w}, c', \mathbf{z}'$ be two accepting transcripts with $c \neq c'$. Let $\bar{\mathbf{z}} = \mathbf{z} - \mathbf{z}'$ and $\bar{c} = c - c'$, by correctness we have $\mathbf{A}\bar{\mathbf{z}} = \mathbf{t}\bar{c}$, with $\bar{c} \in \bar{\mathcal{C}}$ and $\|\bar{\mathbf{z}}\| \leq 2B'$.

Honest Verifier Zero-Knowledge: We only prove that accepting transcripts do not leak information. The reason for this is that all concrete instantiations of the proofs of knowledge described in this section will be made non-interactive via the Fiat-Shamir transform, in which case only accepting transcripts will matter since aborting runs of the protocol will not output anything. Moreover the protocol of Figure 3.2 can be made zero-knowledge even for non-accepting transcripts by sending $B(\mathbf{w})$, where B is a commitment, instead of $\mathbf{w}$ and then sending the opening of that commitment in the response flow.
Let $\mathcal{S}(\mathbf{A}, \mathbf{t})$ be the following PPT algorithm:

- Sample $c \xleftarrow{\$} \mathcal{C}$
- Sample $\mathbf{z} \leftarrow D^m_{\mathcal{R},\sigma}$
- Set $\mathbf{w} = \mathbf{Az} - \mathbf{t}c$
- Output $(\mathbf{w}, c, \mathbf{z})$

It is clear that $\mathbf{z}$ verifies with overwhelming probability. We already showed in the proof of correctness that in the real protocol when no abort occurs the distribution of $\mathbf{z}$ is within statistical distance 2^{-100} of $D^m_{\mathcal{R},\sigma}$. Since $\mathbf{w}$ is completely determined by $\mathbf{A}, \mathbf{t}, \mathbf{z}$ and c, the distribution of $(\mathbf{w}, c, \mathbf{z})$ output by $\mathcal{S}$ is within distance 2^{-100} of the distribution of these variables in the actual protocol.

□

Multiple criteria come into play when considering the efficiency of a ZKPoK. The main concern will be the communication overhead (or proof size for the non-interactive variant) which we will consider to be $|\mathbf{z}| / |\mathbf{s}|$[1]. The soundness error heavily influences the communication overhead of the proof, as a protocol with soundness error C will need to be repeated $\lambda / \log(1/C)$ times to achieve overwhelming soundness and will thus multiply the overhead by the same amount. Another important attribute of lattice-based zero-knowledge will be the slack of the protocol, defined as $\|\mathbf{z}\| / \|\mathbf{s}\|$. The slack of the proof of knowledge will affect how we set parameters, for example when proving knowledge of a preimage for the one way function defined by $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ we will of course want this function to be hard to invert for $\mathbf{s}$ but also for $\mathbf{z}$, otherwise the solution extracted by the knowledge extractor would be vacuous. Hence a large slack will imply larger parameters and in turn larger overhead. Finally the computation complexity of the protocol can be of importance (though it is often a smaller concern), most of the computational complexity comes from one-way function evaluations (which are matrix/vector products) and discrete Gaussian sampling. As discrete Gaussians

[1] We do not consider $|\mathbf{w}|$ and $|c|$ because $\mathbf{w}$ will not be output in non-interactive proofs (as it can be recovered from $\mathbf{A}, \mathbf{z}, \mathbf{t}, c$) and $|c|$ is usually negligible.

can be computationally intensive to sample some applications will forego Gaussians in favor of uniform or other simpler distributions, at a cost in parameters.

ZKPoKs over the integers. Here we consider the ring $\mathcal{R} = \mathbb{Z}$. In this case the challenge space will be $\mathcal{C} = \{0, 1\}$, we could consider larger challenge space for better soundness but the slack of the protocol depends on the bound $B_{\mathcal{C}}$ on the norm of the challenges which would grow linearly with the size of the challenge space. These parameters result in an "exact" proof of knowledge (since $\bar{\mathcal{C}} = \{1\}$) with soundness $1/2$ and slack $11\sqrt{2dm}$. The main issue here is the constant soundness, meaning that the ZKPoK will need to be repeated λ times to achieve overwhelming soundness, resulting in impractically large proofs.

ZKPoKs over polynomial rings. If we consider a ring of dimension larger than one we can obtain new tradeoffs. Let $\mathcal{R} = \mathbb{Z}[X]/X^d + 1$, for d a power of two. We consider the challenge set $\mathcal{C} = \{c \in \mathcal{R} \mid \|c\|_1 = \kappa, \|c\|_\infty = 1\}$. Using such a challenge set results in an "approximate" proof of knowledge with soundness error $1/|\mathcal{C}| = \binom{d}{\kappa}^{-1} 2^{-\kappa}$ and slack $11\kappa\sqrt{2md}$, the bound κ will be chosen so that the protocol achieves overwhelming soundness. Using this challenge set we obtain a proof of knowledge with roughly equivalent slack and without any need for repetition. The downside being that it is no longer clear exactly what this "approximate" proof really proves as the knowledge extractor can only extract $\bar{\mathbf{z}}$ such that $\mathbf{A}\bar{\mathbf{z}} = \mathbf{t}\bar{c}$ for some $\bar{c} \in \bar{\mathcal{C}}$. Lyubashevsky shows in [Lyu08] that such a ZKPoK can be used for digital signatures, we will also use "approximate" proofs in Chapters 4 and 5 to construct a group signature and an e-voting scheme.

ZKPoKs with monomial challenges. In this paragraph we will also consider the ring $\mathbb{Z}[X]/X^d + 1$ but with the challenge space $\mathcal{C} = \{0\} \bigcup \{\pm X^j\}_{j<d}$. While this challenge set is only of size $2d + 1$ it has the property that elements $\bar{c} \in \bar{\mathcal{C}}$ have inverses in $\mathcal{R}_q$ such that $\|2\bar{c}^{-1}\|_1 \leq d$ (cf. lemma 2.2.1). Meaning that we can consider ZKPoKs which extract witnesses from the relation

$$\mathfrak{R}'' = \left\{(\mathbf{t}, \mathbf{s}, 2) \in \mathcal{R}_q^n \times \mathcal{R}^m \times \mathcal{R} \mid \mathbf{A}\mathbf{s} = 2\mathbf{t},\ \|\mathbf{s}\| \leq 2dB'\right\}$$

which is such that $\mathfrak{R}' \subset \mathfrak{R}''$ as one can map $(\mathbf{t}, \mathbf{s}, \bar{c}) \in \mathfrak{R}'$ to $(\mathbf{t}, 2\mathbf{s}\bar{c}^{-1}, 2) \in \mathfrak{R}''$. Using this challenge set we obtain a proof with soundness error $1/(2d + 1)$ and slack $11d\sqrt{2dm}$, resulting in proofs that are $\log(2d + 1)$ times smaller but with d times more slack than for $\mathcal{C} = \{0, 1\}$. Once again the proof is "approximate" since extraction recovers a preimage of $2\mathbf{t}$ rather than $\mathbf{t}$ but this "fixed" approximation can be better than the solution of the previous paragraph. For example extracted values can be summed, i.e. if $\mathbf{A}\mathbf{s}_1 = 2\mathbf{t}_1$ and $\mathbf{A}\mathbf{s}_2 = 2\mathbf{t}_2$ then $\mathbf{A}(\mathbf{s}_1 + \mathbf{s}_2) = 2(\mathbf{t}_1 + \mathbf{t}_2)$, such a property does not hold for elements in $\mathfrak{R}'$, this will come in handy when we construct our e-voting scheme in Chapter 5.

3.2.2 Applications to Commitments

Proof of Opening. We consider a commitment $\mathbf{t}$ defined as in Section 2.4.1, i.e. of the form

$$\mathbf{t} = \begin{bmatrix}\mathbf{t}_1 \\ \mathbf{t}_2\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{A}_2\end{bmatrix}\mathbf{r} + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix}$$

One can prove knowledge of an opening simply by using the protocol of Figure 3.1 to prove knowledge of a preimage of $\mathbf{t}_1$. Consider the corresponding binary relations:

$$\mathfrak{R} = \left\{(\mathbf{t}_1, \mathbf{r}, 1) \in \mathcal{R}_q^n \times \mathcal{R}^m \times \mathcal{R} \mid \mathbf{A}_1\mathbf{s} = \mathbf{t},\ \|\mathbf{r}\| \leq B\right\}$$
$$\mathfrak{R}' = \left\{(\mathbf{t}_1, \mathbf{r}, \bar{c}) \in \mathcal{R}_q^n \times \mathcal{R}^m \times \mathcal{R} \mid \mathbf{A}_1\mathbf{r} = \mathbf{t}_1\bar{c},\ \|\mathbf{r}\| \leq 2B', \bar{c} \in \bar{\mathcal{C}}\right\}$$

The soundness extractor of the protocol $\Pi_{OWF}(\mathbf{A}_1, \mathbf{t}_1; \mathbf{r})$ will extract a tuple $(\mathbf{t}_1, \bar{\mathbf{r}}, \bar{c}) \in \mathfrak{R}'$. By definition of the opening of a commitment it is clear that $\bar{\mathbf{r}}, \bar{c}$ is a valid opening for the message $\mathbf{t}_2\bar{c} - \mathbf{A}_2\bar{\mathbf{r}}$ as long as $\|\bar{\mathbf{r}}\| \leq 2B' \leq B_{Com}$. Similarly one can prove knowledge of an opening to a specific message $\mathbf{m}$ by proving knowledge of a preimage of $\mathbf{t} - \begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix}$ for the matrix $\mathbf{A}$, i.e. from the proof $\Pi_{OWF}\left(\mathbf{A}, \mathbf{t} - \begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix}; \mathbf{r}\right)$ one can extract $\bar{\mathbf{r}}, \bar{c}$ such that

$$\mathbf{t}\bar{c} = \mathbf{A}\bar{\mathbf{r}} + \bar{c}\begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix}$$

Proofs of Linear Relations. We now consider how to prove that the messages of two commitments verify a certain linear relation. Formally for some public $u, v \in \mathcal{R}$, $\mathbf{t} = \mathbf{Com}(\mathbf{m}; \mathbf{r})$ and $\mathbf{t}' = \mathbf{Com}(\mathbf{m}'; \mathbf{r}')$ we want a proof that $u\mathbf{m} + v\mathbf{m}' = \mathbf{0}$. We thus consider the following relations:

$$\mathfrak{R} = \left\{ \begin{array}{c} (\mathbf{t}, \mathbf{t}', \mathbf{r}, \mathbf{r}', \mathbf{m}, \mathbf{m}', 1) \in \mathcal{R}_q^{n+l} \times \mathcal{R}_q^{n+l} \times \mathcal{R}^m \times \mathcal{R}^m \times \mathcal{R}^l \times \mathcal{R}^l \times \mathcal{R} \\ \text{s.t. } \mathbf{t} = \mathbf{Com}(\mathbf{m}; \mathbf{r}), \mathbf{t}' = \mathbf{Com}(\mathbf{m}', \mathbf{r}'), \|\mathbf{r}\| \leq B, \|\mathbf{r}'\| \leq B, u\mathbf{m} + v\mathbf{m}' = \mathbf{0} \end{array} \right\}$$
$$\mathfrak{R}' = \left\{ \begin{array}{c} (\mathbf{t}, \mathbf{t}', \mathbf{r}, \mathbf{r}', \mathbf{m}, \mathbf{m}', \bar{c}) \in \mathcal{R}_q^{n+l} \times \mathcal{R}_q^{n+l} \times \mathcal{R}^m \times \mathcal{R}^m \times \mathcal{R}^l \times \mathcal{R}^l \times \mathcal{R} \\ \text{s.t. } \bar{c}\mathbf{t} = \mathbf{Com}(\bar{c}\mathbf{m}; \mathbf{r}), \bar{c}\mathbf{t}' = \mathbf{Com}(\bar{c}\mathbf{m}', \mathbf{r}'), \|\mathbf{r}\| \leq 2B', \|\mathbf{r}'\| \leq 2B', u\mathbf{m} + v\mathbf{m}' = \mathbf{0} \end{array} \right\}$$

Such a proof can be obtained by applying the proof for one-way functions to a well chosen matrix. Let $\mathbf{B}$ be the following matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_1 \\ u\mathbf{A}_2 & v\mathbf{A}_2 \end{bmatrix}$$

observe that if our commitment scheme is binding for randomnesses of norm less than B, then

$$(\mathbf{t}, \mathbf{t}', \mathbf{r}, \mathbf{r}', \mathbf{m}, \mathbf{m}', 1) \in \mathfrak{R} \Leftrightarrow \mathbf{B}\begin{bmatrix}\mathbf{r}\\ \mathbf{r}'\end{bmatrix} = \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_1'\\ u\mathbf{t}_2 + v\mathbf{t}_2'\end{bmatrix} \wedge \|\mathbf{r}\| \leq B \wedge \|\mathbf{r}'\| \leq B$$

The protocol $\Pi_{OWF}\left(\mathbf{B}, \begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_1'\\ u\mathbf{t}_2 + v\mathbf{t}_2'\end{bmatrix}; \begin{bmatrix}\mathbf{r}\\ \mathbf{r}'\end{bmatrix}\right)$ is therefore a valid ZKPoK with knowledge extractor in $\mathfrak{R}'$.

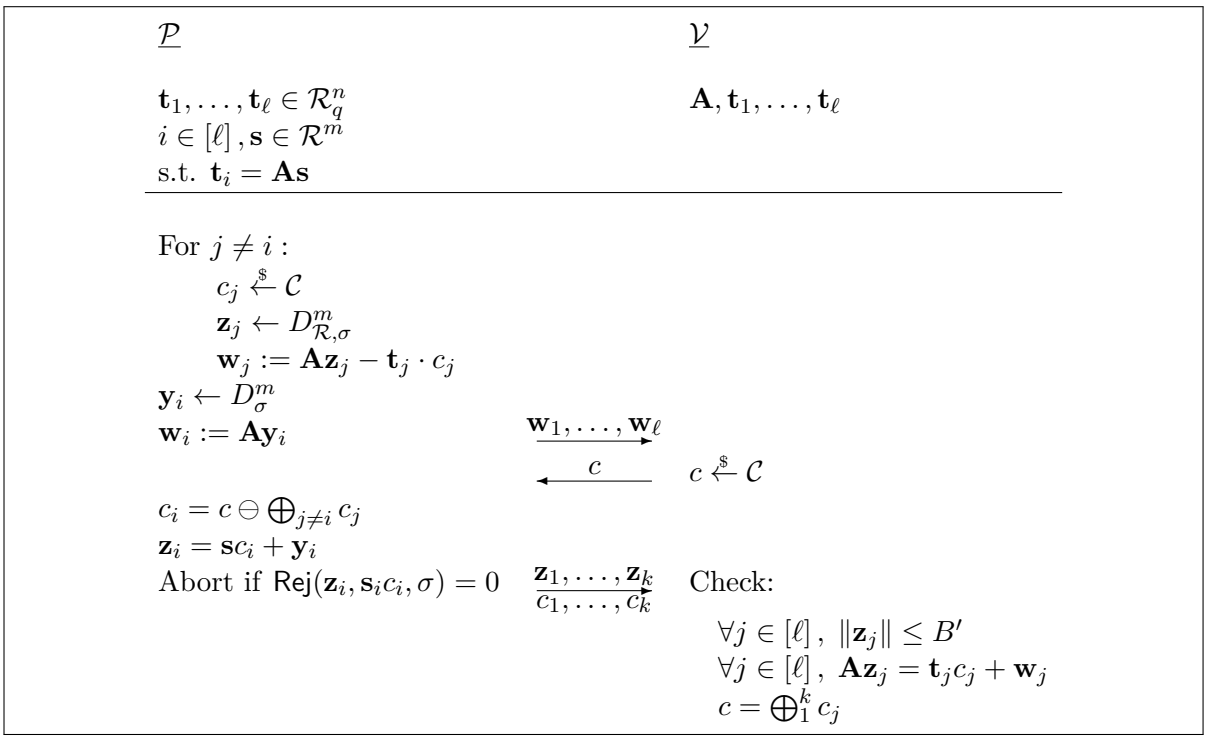

Figure 3.3: Proof of knowledge of a preimage $\mathbf{s}$ of one of ℓ vectors $\mathbf{t}_j$.

3.3 OR-Proofs

Given Σ-protocols for two relations $\mathfrak{R}_0$ and $\mathfrak{R}_1$, one can prove that he knows a witness for either $t_0 \in \mathfrak{L}(\mathfrak{R}_0)$ or $t_1 \in \mathfrak{L}(\mathfrak{R}_1)$ without revealing which is the case. Let $b \in \{0,1\}$ be such that $\mathcal{P}$ knows s_b with $(t_b, s_b) \in \mathfrak{R}_b$, the prover first computes a transcript using the simulator of the relation for which he does not have a witness, and then uses $c - c_{1-b}$ as a challenge for the relation for the relation for which he knows the witness, where c_{1-b} is the challenge used in the simulator and c a challenge sent by the verifier. This protocol (given in e.g. [Dam10]) assumes that the challenges form a group as the verifier should not be able to distinguish whether the prover computed $c_0 = c - c_1$ or $c_1 = c - c_0$. The challenge space we used in lattice based ZKPoK typically do not have this property, to remedy this we will endow $\mathcal{C}$ with an ad-hoc group law. For any $\ell \geq 2$ we describe in Figure 3.3 an or-proof for the relations

$$\mathfrak{R}_{OR} = \left\{ (\mathbf{t}_1, \ldots, \mathbf{t}_\ell, \mathbf{s}, 1) \in \left(\mathcal{R}_q^n\right)^\ell \times \mathcal{R}^m \times \mathcal{R} \mid \exists i, \mathbf{A}\mathbf{s} = \mathbf{t}_i,\ \|\mathbf{s}\| \leq B \right\}$$

$$\mathfrak{R}'_{OR} = \left\{ (\mathbf{t}_1, \ldots, \mathbf{t}_\ell, \mathbf{s}, \bar{c}) \in \left(\mathcal{R}_q^n\right)^\ell \times \mathcal{R}^m \times \mathcal{R} \mid \exists i, \mathbf{A}\mathbf{s} = \mathbf{t}_i \bar{c},\ \|\mathbf{s}\| \leq 2B', \bar{c} \in \bar{\mathcal{C}} \right\}$$

We will consider $\oplus$ to be an efficiently computable group law over the challenge space, we will show how to obtain such a group law.

Lemma 3.3.1. *Let $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $\mathbf{t}_1, \ldots, \mathbf{t}_\ell \in \mathcal{R}_q^n$, $\mathbf{s} \in \mathcal{R}_q^m$, and $i \in [\ell]$ such that $\mathbf{t}_i := \mathbf{A}\mathbf{s}$ and $\|\mathbf{s}\| \leq B$. Let $B_{\mathcal{C}} \geq \max_{c \in \mathcal{C}} (\|c\|_1)$, $\sigma \geq 11 B_{\mathcal{C}} B$, and $B' \geq \sqrt{2md}\sigma$. The protocol $\Pi_{OR}(\mathbf{A}, \mathbf{t}, \ldots, \mathbf{t}_\ell; \mathbf{s})$ of Figure 3.3 is a Σ'-protocol for relations $\mathfrak{R}_{OR}$ and $\mathfrak{R}'_{OR}$ with success probability $1/3$ and soundness $1/|\mathcal{C}|$.*

Proof.

Correctness: Using lemma 2.3.6 we know that since $\sigma \geq 11 B_{\mathcal{C}} B \geq \|\mathbf{s}c_i\|$ the rejection step will accept with probability overwhelmingly close to $1/3$ and the vector $\mathbf{z}_i$ output will be statistically close to D_σ^m, for $i \neq j$ the vector $\mathbf{z}_j$ comes exactly from D_σ^m. By lemma 2.3.5 we have for all $j \in [\ell]$, $\|\mathbf{z}_j\| \leq \sqrt{2md}\sigma \leq B'$ with overwhelming probability. It is clear that in the honest protocol the response is computed in such a way that the verification equations stand.

Special Soundness: Consider two accepting transcripts $(\mathbf{w}_1, \ldots, \mathbf{w}_\ell, c, \mathbf{z}_1, \ldots, \mathbf{z}_\ell, c_1, \ldots, c_\ell)$ and $(\mathbf{w}_1, \ldots, \mathbf{w}_\ell, c, \mathbf{z}'_1, \ldots, \mathbf{z}'_\ell, c'_1, \ldots, c'_\ell)$ with $c \neq c'$. Since $c \neq c'$, there exists $i \in [\ell]$ such that $c_i \neq c'_i$. Let $\bar{\mathbf{z}}_i = \mathbf{z}_i - \mathbf{z}'_i$ and $\bar{c}_i = c_i - c'_i$, by correctness we have $\mathbf{A}\bar{\mathbf{z}}_i = \mathbf{t}_i \bar{c}_i$, with $\bar{c}_i \in \bar{\mathcal{C}}$ and $\|\bar{\mathbf{z}}_i\| \leq 2B'$.

Honest Verifier Zero-Knowledge: We only prove that accepting transcripts do not leak information. Let $\mathcal{S}(\mathbf{A}, \mathbf{t}_1, \ldots, \mathbf{t}_\ell)$ be the following PPT algorithm:

- Sample $c \xleftarrow{\$} \mathcal{C}$
- Sample $c_j \xleftarrow{\$} \mathcal{C}$ for $j < \ell$, and set $c_\ell = c \ominus \bigoplus_{j \neq \ell} c_j$
- Set $\mathbf{w}_j = \mathbf{A}\mathbf{z}_j - \mathbf{t}_j c_j$ for $j \leq \ell$
- Output $(\mathbf{w}_1, \ldots, \mathbf{w}_\ell, c, \mathbf{z}_1, \ldots, \mathbf{z}_\ell, c_1, \ldots, c_\ell)$

It is clear that the transcript output by the simulator verifies with overwhelming probability. We already showed in the proof of correctness that in the real protocol when no abort occurs the distribution of $\mathbf{z}_j$ is within statistical distance 2^{-100} of D_σ^m for all $j \in [\ell]$. Since $\mathcal{C}$ is a group for $\oplus$ the vector $(c_1, \ldots, c_l)$ is sampled uniformly over $\mathcal{C}^\ell$ conditioned on $\bigoplus c_j = c$ in both the simulator and the real protocol. Since $(\mathbf{w}_1, \ldots, \mathbf{w}_\ell)$ is completely determined by $\mathbf{A}, \mathbf{t}, \ldots, \mathbf{t}_l, \mathbf{z}_1, \ldots, \mathbf{z}_l, c_1, \ldots, c_l$ and c, the distribution of the transcript output by $\mathcal{S}$ is within distance 2^{-100} of the distribution of the one in the actual protocol.

□

Endowing $\mathcal{C}$ with a group structure. If $\mathcal{C} = \{0, 1\}$ or $\mathcal{C} = \{0\} \bigcup \{\pm X^j\}_{j<d}$ then adding a group law to the challenge space is straightforward, we will thus focus on the case where $\mathcal{C} = \{c \in \mathcal{R} \mid \|c\|_1 = \kappa, \|c\|_\infty = 1\}$. One can obtain a group law $\oplus$ by defining a bijection ϕ from $\mathcal{C}$ to $\mathbb{Z}_{|\mathcal{C}|}$ and setting $x \oplus y := \phi^{-1}(\phi(x) + \phi(y))$, to do so we will need both ϕ and ϕ^{-1} to be efficiently computable. Recall that we have fixed $\mathcal{C} = \{c \in \mathcal{R} \mid \|c\|_1 = \kappa, \|c\|_\infty = 1\}$ which is of size $|C| = 2^\kappa \binom{d}{\kappa}$ (one can think of choosing $c \in \mathcal{C}$ as first choosing the κ non-zero coefficients and then choosing their sign). If we define $\mathcal{C}' = \left\{c \in \{0,1\}^d,\ \|c\|_1 = \kappa\right\}$ then any element $c \in \mathcal{C}$ can be directly decomposed as a pair $(a, b) \in \mathcal{C}' \times \mathbb{Z}_2^\kappa$ where a is the "absolute value" of c and $b_i = 1$ iff the i^{th} non-zero coefficient of c is 1. It is clear that finding an efficient bijection $\phi' : \mathcal{C}' \to \mathbb{Z}_{\binom{d}{\kappa}}$ is sufficient.

For $c \in \mathcal{C}'$ let $ind(c) \in \mathbb{Z}_d^\kappa$ be such that $ind(c)_i$ for $0 \leq i < \kappa$ is the index of the i^{th} non-zero coefficient of c, then we define ϕ' as follows:

$$\phi'(c) = \sum_0^{\kappa-1} \binom{ind(c)_i}{i+1}$$

where $\binom{a}{b} = 0$ if $b > a$. This bijection in fact corresponds to ordering $\mathcal{C}'$ in lexicographic order and can be computed in time $O(\kappa d)$. ϕ'^{-1} is somewhat more complicated to compute as there is no convenient closed-form expression but the following algorithm was shown in [Ste88] to compute ϕ'^{-1} in quadratic time:

Algorithm 4 $\phi'^{-1}\left(x \in \mathbb{Z}_{\binom{d}{\kappa}}\right)$

$i := d$
$j := x$
$k := \binom{i}{\kappa}$
while $i > 0$ **do**
  $k' := k \cdot \frac{i-\kappa}{i}$
  if $j \leq k'$ **then**
    output 0
    $k := k'$
  else
    output 1
    $j := j - k'$
    $k := k \cdot \frac{\kappa}{i}$
  $i := i - 1$

3.4 Subset Membership Proofs

The goal of this section will be to prove that the message $\mathbf{m}$ of a commitment $\mathbf{t} := \mathbf{Com}(\mathbf{m})$ is in a certain fixed subset $S \subset \mathcal{R}_q^l$. A simple approach that works for any set S is to use the OR-proof of Section 3.3 on each element of S, of course such a technique quickly becomes unpractical since the size of the proof would grow linearly with the size of S.
A different approach would be to prove that the message belongs to a subring $\mathcal{S}_q$ of $\mathcal{R}_q$ by using the Chinese remainder theorem, for example suppose that X^d+1 split into $X^d+1 = fg \mod q$ then we could take $\mathcal{S}_q = f\mathcal{R}_q = \{fx \ : \ x \in \mathcal{R}_q\}$ and prove that $\mathbf{t}$ commits to $m \in \mathcal{S}_q$ by proving that $mg = 0 \mod q$. This can be done with a proof size independent of $|\mathcal{S}_q|$ by using the linearity proof of Section 3.2.2. A caveat of this solution is that by definition of $\mathcal{S}_q$ its element will not have an inverse in $\mathcal{R}_q$, which is an often important property, e.g. for the group signature of Section 4.2.
We will thus prove that the message $m \in \mathcal{R}_q$ is in a subring $\mathcal{S}_q$ of $\mathcal{R}_q$ with invertible elements, to do so we will use the automorphisms of $\mathcal{R}$.

Galois Group Structure of Cyclotomic Rings. We will only state the results that are useful in constructing our proof of subset membership, for a more in-depth approach to the Galois group structure of cyclotomic rings we refer the reader to [PLS18].
Consider the following functions for $0 \leq i < 2d$:

$$\begin{array}{rccc} \xi_i : & \mathcal{R} & \to & \mathcal{R} \\ & u(X) & \mapsto & u(X^i) \end{array}$$

the set $\Gamma := \{\xi_i \ : \ i \in \mathbb{Z}_{2d}^*\}$ is the so-called Galois group of $\mathcal{R}$[2]. Γ is a group for composition

[2]The definition of the Galois group is in fact more intricate as it is only defined for field extensions. If we

Degree k	Galois group H	Generator α
1	$\langle \xi_{-1}, \xi_5 \rangle$	1
2	$\langle \xi_{-1}, \xi_5^2 \rangle$	$X^{3072} - X^{1024}$
4	$\langle \xi_{-1}, \xi_5^4 \rangle$	$X^{3584} - X^{512}$
8	$\langle \xi_{-1}, \xi_5^8 \rangle$	$X^{3849} - X^{256}$

Table 3.1: Subrings of $\mathcal{R}_q$, for $d = 4096$, of degree at most 8 with generators for the corresponding Galois group H, and generators of the subring.

and the only subset of $\mathcal{R}$ that is fixed by Γ is $\mathbb{Z}$ (i.e. the polynomials of degree 0)

$$\mathcal{R}/\Gamma \simeq \mathbb{Z}.$$

As a group Γ is isomorph to $\mathbb{Z}_{2d}^* \simeq \mathbb{Z}_2 \times \mathbb{Z}_{d/2}$ and is therefore generated by two elements, these elements are $\xi_{-1} : X \mapsto X^{2d-1} = X^{-1}$ and $\xi_5 : X \mapsto X^5$. From this we know that the only elements of $\mathcal{R}$ fixed by both ξ_{-1} and ξ_5 are the constant polynomials. To obtain a larger subring we can consider the elements that are fixed by a subgroup of Γ, for any power of two $k < d$ we have:

$$\mathcal{R}/\left\langle \xi_{-1}, \xi_5^k \right\rangle \simeq \mathbb{Z}^k.$$

We can thus obtain subrings of $\mathcal{R}$ of dimension k for any power of two less than d. However while this is true for $\mathcal{R}$ it is not trivial for $\mathcal{R}_q$, the set Γ will still be a group of automorphism of $\mathcal{R}_q$ but it might be possible that, for example, there exists $u \in \mathcal{R}_q$ that is fixed by both ξ_{-1}, and ξ_5 but is not a constant. Fortunately we prove in [PLS18] that for a well chosen modulus q there are no such problems, i.e.:

$$\mathcal{R}_q/\left\langle \xi_{-1}, \xi_5^k \right\rangle \simeq \mathbb{Z}_q^k.$$

We can now prove that a message is in a subring of dimension k by proving that it is fixed by ξ_{-1} and ξ_5. Moreover such subrings only contain invertible elements, and have a $\mathbb{Z}_q$-basis that consists of the power of an efficiently computable polynomial $\alpha \in \mathcal{R}_q$, i.e.

$$\mathcal{R}/\left\langle \xi_{-1}, \xi_5^k \right\rangle = \left\{ c_0 + c_1\alpha + \ldots, +c_{k-1}\alpha^{k-1} \ : \ c_i \in \mathbb{Z}_q \right\}$$

Theorem 3.4.1 ([PLS18] Theorem 3.2)**.** *Let $d > k \geq 1$ be powers of 2, let q be a prime congruent to 3 or 5 modulo 8. Let $H = \langle \xi_{-1}, \xi_5^k \rangle$. The subring*

$$\mathcal{R}_q/\left\langle \xi_{-1}, \xi_5^k \right\rangle$$

is of dimension k (and thus size q^k), is generated by $\alpha = X^{d-\frac{d}{2k}} - X^{\frac{d}{2k}}$, and all of its elements are invertible in $\mathcal{R}_q$.

We give in Table 3.1 a few example of subrings of $\mathcal{R}_q$ and their generators.

Proving Stability Under Automorphisms. We present in this section a proof of knowledge that a commitment opens to a message $\mathbf{m} \in \mathcal{R}_q^l$ that is invariant under a certain set of

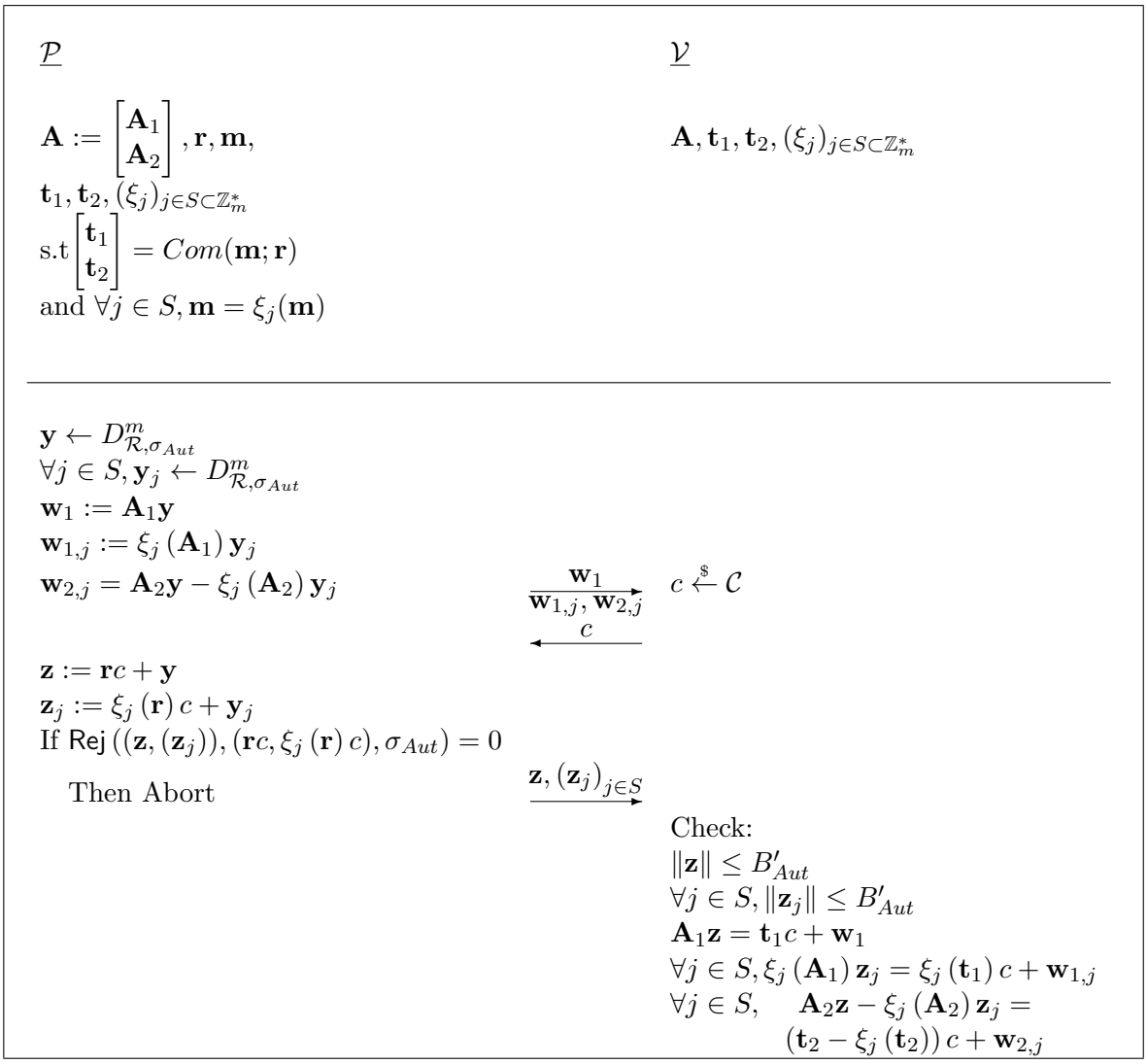

Figure 3.4: Proof that the opening of a commitment is invariant under a set of automorphisms $(\xi_j)_{j\in S}$

automorphisms $(\xi_j)_{j\in S\subset\mathbb{Z}_m^*}$, as a special case we can show that $\mathbf{m} \in \mathbb{Z}_q^l$ by proving that it is invariant under two automorphisms (specifically ξ_{-1} and ξ_5).

$$\mathfrak{R}_{Auto} = \left\{ \begin{array}{c} (\mathbf{t},\mathbf{r},\mathbf{m},1) \in \mathcal{R}_q^{n+l} \times \mathcal{R}^m \times \mathcal{R}^l \times \mathcal{R} \\ \text{s.t. } \mathbf{t} = \mathbf{Com}(\mathbf{m};\mathbf{r}), \|\mathbf{r}\| \le B_{Aut},\ \forall j \in S,\ \xi_j(\mathbf{m}) = \mathbf{m} \end{array} \right\}$$

$$\mathfrak{R}'_{Auto} = \left\{ \begin{array}{c} (\mathbf{t},\mathbf{r},\mathbf{m},\bar{c}) \in \mathcal{R}_q^{n+l} \times \mathcal{R}^m \times \mathcal{R}^l \times \mathcal{R} \\ \text{s.t. } \bar{c}\mathbf{t} = \mathbf{Com}(\bar{c}\mathbf{m};\mathbf{r}), \|\mathbf{r}\| \le 2B'_{Aut},\ \forall j \in S,\ \xi_j(\mathbf{m}) = \mathbf{m} \end{array} \right\}$$

Theorem 3.4.2. *Let $\|\mathbf{r}\| \le B_{Aut}$. Let S be a set of automorphisms of size $|S|$. Let $\sigma \ge 11B_{\mathcal{C}}B_{Aut}\sqrt{|S|+1}$ and $B'_{Aut} \ge \sqrt{2md}\sigma_{Aut}$. If $B_{com} \ge 2B'_{Aut}$, then the protocol*

consider $\mathbb{K} := \mathbb{Q}(\zeta)$, where ζ is a complex root of $X^d + 1$ then the Galois group $Gal(\mathbb{K}/\mathbb{Q})$ is the set of all automorphisms of $\mathbb{K}$ that fix $\mathbb{Q}$. This set is exactly $\left\{\xi_i : \zeta \to \zeta^i \ :\ i \in \mathbb{Z}_{2d}^*\right\}$.

$\Pi_{Auto}(\mathbf{A}, \mathbf{t}, S; \mathbf{s}, \mathbf{m})$ *of Figure 3.4 is a* Σ'*-protocol for relations* $\mathfrak{R}_{Auto}$ *and* $\mathfrak{R}'_{Auto}$ *with success probability* $1/3$ *and soundness* $1/|\mathcal{C}|$.

Proof. The arguments for correctness and zero-knowledge are identical to the ones of Theorem 3.2.1, we will focus on special soundness.
Let $(\mathbf{w}_1, \mathbf{w}_{1,j}, \mathbf{w}_{2,j}, c, \mathbf{z}, \mathbf{z}_j)_{j\in S}$ and $\left(\mathbf{w}_1, \mathbf{w}_{1,j}, \mathbf{w}_{2,j}, c', \mathbf{z}', \mathbf{z}'_j\right)_{j\in S}$ be two accepting transcripts. We will prove that there exists a message $\bar{\mathbf{m}}$ such that $(\mathbf{z}-\mathbf{z}', \bar{\mathbf{m}}, c-c')$ is a valid opening of $\begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}$ and $\forall j \in S, \xi_j(\bar{\mathbf{m}}) = \bar{\mathbf{m}}$. For a fixed $j \in S$, let $\bar{\mathbf{z}} = \mathbf{z}-\mathbf{z}'$, $\bar{\mathbf{z}}_j = \mathbf{z}_j - \mathbf{z}'_j$, and $\bar{c} = c - c'$. By taking the difference of the verification equations for both transcripts we obtain:

$$\mathbf{A}_1\bar{\mathbf{z}} = \mathbf{t}_1\bar{c} \tag{3.1}$$
$$\xi_j(\mathbf{A}_1)\,\bar{\mathbf{z}}_j = \xi_j(\mathbf{t}_1)\,\bar{c} \tag{3.2}$$
$$\mathbf{A}_2\bar{\mathbf{z}} - \xi_j(\mathbf{A}_2)\,\bar{\mathbf{z}}_j = (\mathbf{t}_2 - \xi_j(\mathbf{t}_2))\bar{c} \tag{3.3}$$

If we apply the automorphism ξ_j^{-1} to equation 3.2 we have $\mathbf{A}_1\xi_j^{-1}(\bar{\mathbf{z}}_j) = \mathbf{t}_1\xi_j^{-1}(\bar{c})$, we can then multiply this equation by $\bar{c}$ and equation 3.1 by $\xi_j^{-1}(\bar{c})$ and take the difference to get:

$$\mathbf{A}_1\left(\xi_j^{-1}(\bar{\mathbf{z}}_j)\bar{c} - \bar{\mathbf{z}}\xi_j^{-1}(\bar{c})\right) = 0$$

Using verification we know that $\|\bar{\mathbf{z}}\| \le 2B'_{Aut}$ and $\|\bar{\mathbf{z}}_j\| \le 2B'_{Aut}$, which implies

$$\left\|\xi_j^{-1}(\bar{\mathbf{z}}_j)\bar{c} - \bar{\mathbf{z}}\xi_j^{-1}(\bar{c})\right\| \le 4B_{\mathcal{C}}B'_{Aut}.$$

By the binding property of our commitment scheme we obtain that $\xi_j^{-1}(\bar{\mathbf{z}}_j)\bar{c} = \mathbf{z}\xi_j^{-1}(\bar{c})$, or equivalently $\bar{\mathbf{z}}_j\xi_j(\bar{c}) = \xi_j(\bar{\mathbf{z}})\bar{c}$. We multiply equation 3.3 by $\xi_j(\bar{c})$ and use the previous equality to obtain:

$$A_2\bar{\mathbf{z}}\xi_j(\bar{c}) - \xi_j(A_2)\,\xi_j(\bar{\mathbf{z}})\bar{c} = (\mathbf{t}_2 - \xi_j(\mathbf{t}_2))\bar{c}\xi_j(\bar{c}) \tag{3.4}$$

Since $\bar{c}$ has an inverse in $\mathcal{R}_q$ we can define $\bar{\mathbf{m}} \in \mathcal{R}_q^l$ such that $\bar{c}\mathbf{t}_2 = \mathbf{A}_2\bar{\mathbf{z}} + \bar{c}\bar{\mathbf{m}}$. By replacing $\bar{c}\mathbf{t}_2$ in equation 3.4 we have:

$$\bar{\mathbf{m}} = \xi_j(\bar{\mathbf{m}})$$

In conclusion we have extracted $\bar{\mathbf{z}}$, $\bar{\mathbf{m}}$, and $\bar{c}$ such that:

$$\begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix}\bar{\mathbf{z}} + \begin{bmatrix}0\\ \bar{c}\bar{\mathbf{m}}\end{bmatrix} = \bar{c}\begin{bmatrix}\mathbf{t}_1\\ \mathbf{t}_2\end{bmatrix}$$

with $\|\bar{\mathbf{z}}\| \le 2B'_{Aut}$ and $\bar{\mathbf{m}} = \xi_j(\bar{\mathbf{m}})$. The same argument can be applied for the other $j \in S$. □

3.5 Amortized Zero-Knowledge

In section 3.2 we present a simple protocol to prove knowledge of a preimage for lattice-based one-way functions. i.e. Given a matrix $\mathbf{A}$ and a target vector $\mathbf{t} = \mathbf{A}\mathbf{s}$ for a small secret $\mathbf{s}$ we describe a proof of knowledge of a vector $\bar{\mathbf{s}}$ with small coefficients (though larger than those in $\mathbf{s}$) and a ring element $\bar{c}$ with very small coefficients satisfying $\mathbf{A}\bar{\mathbf{s}} = \bar{c}\mathbf{t}$. As long as the ring $\mathcal{R}$ has many elements with small coefficients, such proofs are very efficient, producing soundness

of $1 - 2^{-128}$ with just one iteration. While these proofs are good enough for constructing practical digital signatures (e.g. [GLP12; DDLL13; BG14]), commitment schemes with proofs of knowledge [BKLP15; BDOP16], and certain variants of verifiable encryption schemes [LN17], they prove less than what the honest prover knows. In many applications where zero-knowledge proofs are used, in particular those that need to take advantage of additive homomorphisms, the presence of the element $\bar{c}$ makes these kinds of "approximate" proofs too weak to be useful. As of today, we do not have any truly practical zero-knowledge proof systems that give a proof of knowledge for a single preimage.

Previous approaches. The situation is considerably more promising when one considers amortized proofs, in which one wants to simultaneously prove knowledge of $\mathbf{s}_1, \ldots, \mathbf{s}_\ell$ such that $\mathbf{t}_j = \mathbf{A}\mathbf{s}_j, \forall j \in [\ell]$. The amortized proof given in [DPSZ12] using the protocol from [CD09] showed that a prover only needs to generate $\lambda + \ell$ vectors to prove the knowledge of ℓ equations, which gives an overhead of $1 + \lambda/\ell$. The main downside of this protocol is that the slack is exponential in ℓ.

The works of [BDLN16] and [CDXY17] gave a novel protocol, still using the Fiat-Shamir with Aborts with 0/1 challenges idea, in which the overhead was constant (down to 2), while the slack could be bounded by a small polynomial in the security parameter (To be precise, only by a super-polynomial factor in the first work). The main downside of these protocols is that they require the number of equations to be fairly large before amortization kicks in. In particular one needs to have more than $k = 4\lambda^2$ equations. Thus scenarios where one does not have too many (around λ as in [DPSZ12]) equations to prove would not benefit from the protocol.

The work of [PL17] showed that one could decrease the number of equations in the above protocol by a factor of $\log^2 \alpha$ by increasing the running time of the proof by a factor of α. It also gave a protocol requiring even fewer equations when the functions are over polynomial rings of dimension d by using the monomial challenge space $\mathcal{C} = \{0\} \bigcup \{\pm X^j\}_{j<d}$ presented in Section 3.2.1. This further reduces the required number of equations by a factor approximately $\log d$. Nevertheless, one still needs at least a few thousand of them in order to be able to use amortization in a practical manner.

Our construction. We present a surprisingly simple zero-knowledge proof for proving the knowledge of ℓ preimages. Formally we consider a Σ'-protocol for the following relations

$$\mathfrak{R}_{Amo} = \left\{(\mathbf{T}, \mathbf{S}) \in \mathcal{R}_q^{n\times\ell} \times \mathcal{R}^{m\times\ell} \mid \mathbf{AS} = \mathbf{T},\ s_1(\mathbf{S}) \leq B\right\}$$
$$\mathfrak{R}'_{Amo} = \left\{(\mathbf{T}, \mathbf{S}) \in \mathcal{R}_q^{n\times\ell} \times \mathcal{R}^{m\times\ell} \mid \mathbf{AS} = \mathbf{T},\ \|\mathbf{S}\|_{\max} \leq 2B'\right\}$$

We first give a useful lemma for knowledge extraction. In essence this lemma will be used to show that a prover who can output a verifying output for a challenge $\mathbf{c}_1, \ldots, \mathbf{c}_\ell$ has a high probability of also being able to answer a challenge $\mathbf{c}'_1, \mathbf{c}_2, \ldots, \mathbf{c}_\ell$ in which only $\mathbf{c}'_1 \neq \mathbf{c}_1$.

Lemma 3.5.1 ([Dam10])**.** *Let $H \in \{0,1\}^{l\times k}$ for some $k, l > 1$, such that a fraction ε of the inputs of H are 1. We say that a row of H is "heavy" if it contains a fraction at least $\varepsilon/2$ of ones. Then more than half of the ones in H are located in heavy rows.*

We describe our proof in Figure 3.5. Our first instantiation will be with the challenge set $\mathcal{C}^{\ell\times k}$ for $\mathcal{C} = \{0, 1\}$, this solution allows the extractor of the protocol to obtain exact

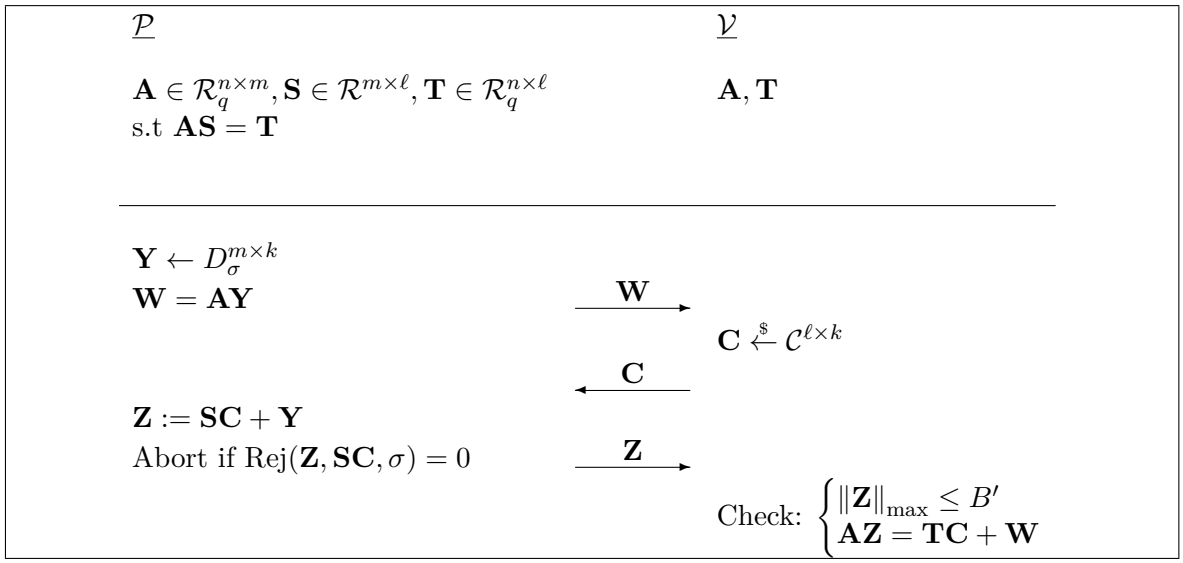

Figure 3.5: Amortized proof for ℓ equations. The ring $\mathcal{R}$ can be either $\mathbb{Z}$ or $\mathbb{Z}[X]/X^d + 1$, the challenge set will be respectively $\{0, 1\}$ or $\{0\} \bigcup \{\pm X^j\}_{j<d}$

preimages of the $\mathbf{t}_i$ and requires $k \geq \lambda + 2$. This ensures that communication only grows linearly in λ regardless of the size of ℓ (since $\mathbf{Z} \in \mathbb{Z}_q^{m\times\lambda}$).

Theorem 3.5.2. *Let $\mathcal{C} = \{0, 1\}$, and $k \geq \lambda + 2$. Let $s_1(\mathbf{S}) \leq B$, $\sigma \geq 11B\sqrt{\ell k}$, and $B' \geq \sqrt{2md}\sigma$. The protocol $\Pi_{Amo1}(\mathbf{A}, \mathbf{T}; \mathbf{S})$ of Figure 3.5 is a Σ'-protocol for relations $\mathfrak{R}_{Amo}$ and $\mathfrak{R}'_{Amo}$ with success probability $1/3$ and soundness error less than $2^{-\lambda}$.*

Proof.

Correctness: Using lemma 2.3.6 we know that since $\sigma \geq 11B\sqrt{\ell k} \geq \|\mathbf{SC}\|$ the rejection step will accept with probability overwhelmingly close to $1/3$ and the vector $\mathbf{Z}$ output will be statistically close to $D_{\mathcal{R},\sigma}^{m\times\lambda}$. By lemma 2.3.5 we have $\|\mathbf{Z}\|_{\max} \leq \sqrt{2md}\sigma \leq B'$ with overwhelming probability. It is clear that in the honest protocol the response is computed in such a way that the verification equation $\mathbf{AZ} = \mathbf{TC} + \mathbf{W}$ stands.

Honest Verifier Zero-Knowledge: We only prove that accepting transcripts do not leak information. Let $\mathcal{S}(\mathbf{A}, \mathbf{T})$ be the following PPT algorithm:

- Sample $C \xleftarrow{\$} \mathcal{C}^{\ell\times k}$
- Sample $\mathbf{Z} \leftarrow D_\sigma^{m\times\lambda}$
- Set $\mathbf{W} = \mathbf{AZ} - \mathbf{TC}$
- Output $(\mathbf{W}, \mathbf{C}, \mathbf{Z})$

It is clear that $\mathbf{Z}$ verifies with overwhelming probability. We already showed in the proof of correctness that in the real protocol when no abort occurs the distribution of $\mathbf{Z}$ is within statistical distance 2^{-100} of $D_\sigma^{m\times\lambda}$. Since $\mathbf{W}$ is completely determined by $\mathbf{A}, \mathbf{T}, \mathbf{Z}$ and $\mathbf{C}$, the distribution of $(\mathbf{W}, \mathbf{C}, \mathbf{Z})$ output by $\mathcal{S}$ is within distance 2^{-100} of the distribution of these variables in the actual protocol.

Soundness: We defer the proof of soundness to lemma 3.5.3, the proof is somewhat more complicated than for the protocols of section 3.2 as we cannot argue special soundness.

□

Lemma 3.5.3 (Soundness)**.** *For any prover $\mathcal{P}^*$ who succeeds with probability $\varepsilon > 2^{-\lambda}$ (i.e. $\geq 2^{-k+2}$) over his random tape $\chi \in \{0,1\}^x$ and the challenge choice $\mathbf{C} \xleftarrow{\$} \mathcal{C}^{\ell \times k}$, there exists a knowledge extractor $\mathcal{E}$ running in expected time $poly(\lambda)/\varepsilon$ who can extract a witness $\mathbf{S}' \in \mathcal{R}^{m \times \ell}$, such that $\mathbf{AS}' = \mathbf{T}$, and $\|\mathbf{S}'\|_{\max} \leq 2B'$.*

Proof. For $i \in [\ell]$, let $\mathbf{t}_i \in \mathcal{R}^n$ be the ith column of $\mathbf{T}$, and $\mathbf{c}_i^T \in \mathcal{R}^{1 \times k}$ be the ith row of $\mathbf{C}$ (note that $\mathbf{c}_i^T$ are not the transpose of the columns of $\mathbf{C}$ but really its rows). Note that $\mathbf{t}_i \mathbf{c}_i^T \in \mathcal{R}^{n \times k}$ and $\mathbf{TC} = \sum_{i=1}^{\ell} \mathbf{t}_i \mathbf{c}_i^T$. For any fixed i, we describe an extractor $\mathcal{E}_i$ who can extract a preimage of $\mathbf{t}_i$ of norm less than $2B'$ in expected time $O(1/\varepsilon)$, and the full result follows by running each extractor (of which there are $\ell = poly(\lambda)$).
Consider a matrix $H_i \in \{0,1\}^{2^{k(\ell-1)+x} \times 2^k}$ whose rows are indexed by the value of

$$(\chi, \mathbf{c}_1^T, \ldots, \mathbf{c}_{i-1}^T, \mathbf{c}_{i+1}^T, \ldots, \mathbf{c}_\ell^T)$$

and whose columns are indexed by the value of $\mathbf{c}_i^T$. An entry of H_i will be 1 if $\mathcal{P}^*$ succeeds for the corresponding randomness and challenge (i.e. produces an accepting $\mathbf{Z}$). We will say that a row of H_i is "heavy" if it contains a fraction of at least $\varepsilon/2$ ones, i.e. if it contains more than $2^k * \varepsilon/2 > 2$ ones. The extractor $\mathcal{E}_i$ will proceed as follow:

1. Run $\mathcal{P}^*$ on random $(\chi', \mathbf{C}')$ until it succeeds, and obtains $\mathbf{Z}'$ that verifies. This takes expected time $1/\varepsilon$.

2. Run $\mathcal{P}^*$ on random $(\chi'', \mathbf{C}'')$ where $\chi'' = \chi'$, $\forall j \neq i, \mathbf{c}_j''^T = \mathbf{c}_j'^T$, and $\mathbf{c}_i''^T$ is freshly sampled. If after λ/ε attempts $\mathcal{P}^*$ has not output a valid response, $\mathbf{Z}''$ abort.

The extractor $\mathcal{E}_i$ runs in expected time $poly(\lambda)/\varepsilon$, and aborts with probability less than $1/2 + 2^{-\lambda}$. The running time is clear from the definition of $\mathcal{E}_i$. To compute the abort probability note that in step 2 all the challenges $(\chi'', \mathbf{C}'')$ considered are in the same row of H_i as $(\chi', \mathbf{C}')$, if we call *Abort* the event where $\mathcal{E}_i$ aborts and *Heavy* the event that $(\chi', \mathbf{C}')$ is in a heavy row of H_i, we have:

$$\Pr[\mathit{Abort}] = \Pr\left[\ \mathit{Abort} \mid \mathit{Heavy}\ \right] \Pr[\mathit{Heavy}] + \Pr\left[\ \mathit{Abort} \mid \neg \mathit{Heavy}\ \right] \Pr[\neg \mathit{Heavy}]$$

According to Lemma 3.5.1, $\Pr[\neg \mathit{Heavy}] < 1/2$. On the other hand if the row is heavy then for a random sample, different from $(\chi', \mathbf{C}')$, in this row $\mathcal{P}^*$ has probability at least $\varepsilon/2 - 2^{-k} > \varepsilon/4$ of outputting a valid answer (the probability is $\varepsilon/2 - 2^{-k}$ and not $\varepsilon/2$ because we want a reply for a challenge different from $(\chi', \mathbf{C}')$). Which entails that the probability that $\mathcal{P}^*$ does not succeed on any of the λ/ε challenges $\mathbf{C}''$ is:

$$\Pr\left[\ \mathit{Abort} \mid \mathit{Heavy}\ \right] < (1 - \varepsilon/4)^{\lambda\varepsilon} < e^{-4\lambda} < 2^{-\lambda}$$

From this we get:

$$\Pr[\mathit{Abort}] < 1/2 + 2^{-\lambda}$$

By running $\mathcal{E}_i$ $O(\lambda)$ times we obtain an extractor that runs in expected time $poly(\lambda)/\varepsilon$ and outputs two valid pairs $\mathbf{C}', \mathbf{Z}'$ and $\mathbf{C}'', \mathbf{Z}''$ such that $\forall j \neq i, \mathbf{c}_j'^T = \mathbf{c}_j''^T$, and $\mathbf{c}_i'^T \neq \mathbf{c}_i''^T$.
Since both transcripts verify we know:

$$\mathbf{AZ}' = \mathbf{TC}' + \mathbf{W} = \sum_{j=1}^{n} \mathbf{t}_j \mathbf{c}_j'^T + \mathbf{W}$$

$$\mathbf{AZ}'' = \mathbf{TC}'' + \mathbf{W} = \sum_{j=1}^{n} \mathbf{t}_j \mathbf{c}_j''^T + \mathbf{W}$$

Which implies:

$$\mathbf{A}(\mathbf{Z}' - \mathbf{Z}'') = \sum_{j=1}^{\ell} \mathbf{t}_j (\mathbf{c}_j'^T - \mathbf{c}_j''^T) = \mathbf{t}_i (\mathbf{c}_i'^T - \mathbf{c}_i''^T)$$

If we consider an index $l \in [\ell]$ such that $\mathbf{c}_i'^T[l] \neq \mathbf{c}_i''^T[l]$, and assume w.l.o.g that $\mathbf{c}_i'^T[l] - \mathbf{c}_i''^T[l] = 1$, then by only considering the l^{th} column of the previous equation we obtain:

$$\mathbf{A}(\mathbf{z}_l' - \mathbf{z}_l'') = \mathbf{t}_i$$

where $\|\mathbf{z}_l' - \mathbf{z}_l''\| \leq 2B'$.

□

Our second instantiation uses $\mathcal{R} = \mathbb{Z}[X]/X^d + 1$ and $\mathcal{C} = \{0\} \bigcup \{\pm X^j\}_{j<d}$, in this protocol the extractor will only obtain preimages of $2\mathbf{t}_i$ but the number of columns in the response matrix $\mathbf{Z}$ can be reduced by a factor of $\log(2d+1)$ as the soundness now only requires that $k \log(2d+1) \geq \lambda + 2$. It is worth noting that in this protocol the values of n and m would typically be chosen to be around d times smaller than in the instantiation with $\mathcal{R} = \mathbb{Z}$, that is because $\mathbf{A}$ will be a matrix of polynomials of degree d. In this context the language $\mathfrak{R}_{Amo}$ is unchanged but the language $\mathfrak{R}'_{Amo}$ is somewhat larger.

$$\mathfrak{R}'_{Amo} = \left\{ (\mathbf{T}, \mathbf{S}) \in \mathcal{R}_q^{n \times \ell} \times \mathcal{R}^{m \times \ell} \mid \mathbf{AS} = 2\mathbf{T},\ \|\mathbf{S}\|_{\max} \leq 2dB' \right\}$$

Theorem 3.5.4. *Let $\mathcal{R} = \mathbb{Z}[X]/X^d + 1$, $\mathcal{C} = \{0\} \bigcup \{\pm X^j\}_{j<d}$, and $k \geq (\lambda+2)/\log(2d+1)$. Let $s_1(\mathbf{S}) \leq B$, $\sigma \geq 11B\sqrt{\ell k}$, and $B' \geq \sqrt{2md}\sigma$. The protocol $\Pi_{Amo2}(\mathbf{A}, \mathbf{T}; \mathbf{S})$ of Figure 3.5 is a Σ'-protocol for relations $\mathfrak{R}_{Amo}$ and $\mathfrak{R}'_{Amo}$ with success probability $1/3$ and soundness error less than $2^{-\lambda}$.*

Proof. The proofs for correctness and zero-knowledge are nearly identical to the ones of Theorem 3.5.2. We will prove soundness in Lemma 3.5.5. □

Lemma 3.5.5 (Soundness)**.** *For any prover $\mathcal{P}^*$ who succeeds with probability $\varepsilon > 2^{-\lambda} (\geq 2^{-k\log(2d+1)+2})$ over his random tape $\chi \in \{0,1\}^x$ and the challenge choice $\mathbf{C} \leftarrow \mathcal{C}^{\ell \times k}$ there exists a knowledge extractor $\mathcal{E}$ who can extract a witness $\mathbf{S}' \in \mathcal{R}^{m \times \ell}$, such that $\mathbf{AS}' = 2\mathbf{T}$, and $\mathbf{S}'_{\max} \leq 2dB$, in expected time $poly(\lambda)/\varepsilon$.*

Proof. The first part of the proof (obtaining $\mathbf{C}', \mathbf{Z}'$ and $\mathbf{C}'', \mathbf{Z}''$) is identical to the one of Lemma 3.5.3 except for the fact that the matrix H_i has different dimensions. Let $\delta = \log(2d+1)$. Since for each $j \in [\ell]$, $\mathbf{c}_j^T$ is sampled from a set of size $2^{k\delta}$, we have $H_i \in \{0,1\}^{2^{k\delta(\ell-1)+x} \times 2^{k\delta}}$. The heavy rows of H_i will contain $2^{k\delta}\varepsilon/2 > 2$ ones, and the

extractor can proceed as in the proof of Lemma 3.5.3.
Assume that $\mathcal{E}_i$ has extracted $\mathbf{C}', \mathbf{Z}$" and $\mathbf{C}'', \mathbf{Z}''$ such that $\forall j \neq i, \mathbf{c}_j'^T = \mathbf{c}_j''^T$, and $\mathbf{c}_i'^T \neq \mathbf{c}_i''^T$. As previously we have:

$$\mathbf{A}(\mathbf{Z}' - \mathbf{Z}'') = \sum_{j=1}^{\ell} \mathbf{t}_j(\mathbf{c}_j'^T - \mathbf{c}_j''^T) = \mathbf{t}_i(\mathbf{c}_i'^T - \mathbf{c}_i''^T)$$

If we consider an index $l \in [\ell]$ such that $\mathbf{c}_i'^T[l] \neq \mathbf{c}_i''^T[l]$, since $\mathcal{C} = \{0\} \bigcup \{\pm X^j\}_{0 \leq j \leq d-1}$, we have according to Lemma 2.2.1 that there exists a $g \in \mathcal{R}$ such that $2^{-1}(\mathbf{c}_i'^T[l] - \mathbf{c}_i''^T[l])g = 1$ and $\|g\| 81 \leq d$. Hence:

$$\mathbf{A}(\mathbf{z}_l' - \mathbf{z}_l'')g = 2\mathbf{t}_i \cdot 2^{-1}(\mathbf{c}_i'^T[l] - \mathbf{c}_i''^T[l])g = 2\mathbf{t}_i$$

With $\|(\mathbf{z}_l' - \mathbf{z}_l'')g\| \leq 2dB'$.

□

Chapter 4

Group Signature

In this chapter we will show how to use the proofs of knowledge from Chapter 3, specifically the proof of subset membership, to construct efficient group signatures.
For ease of comprehension we first describe a group signature without the traceability property and then show how to use the verifiable encryption of [LN17] to obtain a group signature with full traceability. We give formal proofs for the anonymity and traceability of our scheme, and derive parameters as well as the corresponding signature size.

Contents

4.1 Introduction

4.1.1 Our Contribution

One application of our new proof system presented in section 3.4 is towards constructing more practical lattice-based privacy protocols. In this chapter we will use the specific example of group signatures. A group signature scheme consists of three parties – the group manager, the opener, and group members. The group manager has a public key and generates secret keys for all the group members. Using their secret keys, the group members can sign messages in a way that anyone can verify that a message was signed by a member of the group, but the identity of the signer remains secret (one should not even be able to tell that two messages were signed by the same member) to everyone except for the opener. The opener should be able to recover the identity of any signer.

A common way of constructing group signatures is via the sign-and-encrypt approach. The group manager's public key is the public key to some signature scheme, and the secret key of a user with identity i is a signature of i. To sign a message, the group member produces a non-interactive ZKPoK that he has the manager's signature of some identity i.[1] Furthermore, the group member encrypts his identity i using the opener's public key, and gives another ZKPoK of the fact that the encryption is of the same identity as was used in the proof.

To create a practical scheme using the above approach, one typically needs to have a very efficient *standard model* signature scheme that is used by the group manager to sign user identities.[2] While there exist efficient standard model signature schemes based on classical assumptions (e.g. [CL03]) which can be used for constructions of fairly compact group signatures, the non-existence of such signatures based on lattice assumptions is the main culprit in the fact that the only "efficiency" lattice-based group signatures have is asymptotic.

Lattice-based signature schemes in the standard model are built based on Boyen's framework [Boy10]. There have been efficiency improvements to this scheme (e.g. [DM15; KY16]) that used polynomial lattices, but they still appear to be unsuitable for producing practical (group) signatures. In fact, the only group signature that we're aware of that actually proposes concrete parameters uses different techniques, and the signatures in it are on the order of 50MB [LLNW16].

While lattice-based signatures in the standard model are inefficient, there is a much more efficient *selectively-secure* lattice-based digital signature scheme that is implicit from the works of [ABB10; Boy10]. A selectively-secure signature scheme is one in which the adversary declares the message that he will forge on prior to seeing the public key. A scheme like this can be converted to a regular signature scheme with a reduction loss of $1/|S|$, where S is the message space simply by guessing the message that the Adversary will forge on. Thus for small message spaces, this becomes a signature scheme with a meaningful reduction from hard lattice problems.

There have been several works that utilized the above-mentioned selectively secure scheme for group signatures and related applications [NZZ15; BCN17]. In those papers, proving that the identity i is in a particular set resulted in either a significant increase in the proof

[1] The ZKPoK is a Fiat-Shamir transformation of a Σ-protocol, and so the message that the group member signs is simply added into the random oracle input.

[2] The reason that signature schemes using cryptographic hash functions (and proved secure in the random oracle model) are not suitable is that their lack of algebraic structure makes it very difficult to construct efficient proofs of knowledge that prove something about the identity i when it is an input to the random oracle.

size, restrictions on the challenge space, and/or a very noticeable loss in the tightness of the proof. For example, in [NZZ15], the identity space was all polynomials with only a constant coefficient (i.e. of degree zero). But to prove this fact, the size of the proof of knowledge significantly increased due to the fact that one needed to map integers to a space with small coefficients. In [BCN17], which used a similar high-level idea with polynomial rings, the message space of the selectively-secure signature scheme was a polynomial in a subfield of $\mathcal{R}_q$. The degree of the subfield could not be too small because the soundness error of the ZKPoK inversely depends on the degree of the subfield, and the degree cannot be too large because it would make the security reduction meaningless. Further complicating matters is that the messages have to have small coefficients, yet the proof of knowledge only proves that the coefficients are in a significantly larger space causing an additional loss in the reduction.

In the present work we show how our new proofs for stability under automorphisms gives rise to a fairly straight-forward group signature scheme. In particular, the set of identities will be exactly those elements in $\mathcal{R}_q$ that are preserved under some set of automorphisms. The size of these sets can be small (as small as q), and so we will only lose a factor of the group size in the reduction. The idea for the ZKPoK will then be to do the proof of knowledge with the commitments of i rather than with i (thus not revealing the identity) and prove that our commitments are to elements in the appropriate set of identities. The only other property that we use from our commitment scheme is that if $i \cdot s = u$, then $\mathbf{Com}(i;\mathbf{r}) \cdot s = \begin{bmatrix} 0 \\ u \end{bmatrix} + \mathbf{Com}(0;\mathbf{r}')$. The encryption to the opener can be done using the main idea from the verifiable encryption scheme from [LN17]. A point of note is that the selectively-secure signature scheme requires that the messages come from a set S such that the difference of any two elements from the set is invertible. This is compatible with our definition of sets because they turn out to be subfields of the original ring $\mathcal{R}_q$.

Instantiating our scheme with concrete parameters gives group signatures of around 580 KB, which is almost a 2 order of magnitude reduction from [LLNW16]. Our main technique should also applicable to a variety of other privacy applications that require similar proofs of knowledge. For example, one should be able to apply these techniques in a very similar manner to the constructions of anonymous credentials as in [BCN17].

4.1.2 Overview of our Construction

We will use a particular instantiation of the commitment scheme from Section 2.4.1 where the common reference string public key is

$$\begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} = \begin{bmatrix} 1 & a_1 & a_2 \\ 0 & 1 & a_3 \end{bmatrix} \in \mathcal{R}_q^{2\times 3} \tag{4.1}$$

The master public key of the group manager will be a public key for the selectively secure signature scheme from [ABB10] adapted to polynomial rings:

$$[\ \mathbf{a}^T \mid \mathbf{b}^T\ ], u = \mathbf{a}^T\mathbf{s}_1' + \mathbf{b}^T\mathbf{s}_2' + \mathbf{a}_2^T\mathbf{s}_3' \tag{4.2}$$

where $\mathbf{a}^T = [a\ a']$ for a uniformly-random $a, a' \in \mathcal{R}_q$ and $\mathbf{b} = [b_1\ b_2] = \mathbf{a}^T \begin{bmatrix} r_1 & r_2 \\ e_1 & e_2 \end{bmatrix}$ where r_i, e_i are polynomials in $\mathcal{R}$ with small coefficients such that (a, b_1, b_2) are indistinguishable from random based on the hardness of the R-LWE problem. The group member identities

are polynomials $i \in S \subseteq \mathcal{R}_q$ where the set S is preserved under some set of automorphisms of $\mathcal{R}_q$. The secret key of a user with identity i consists of vectors $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$ that are generated by the group manager using his secret trapdoor key $\mathbf{R} = \begin{bmatrix} r_1 & r_2 \\ e_1 & e_2 \end{bmatrix}$. The group manager first picks a short vector $\mathbf{s}_3$ from a particular distribution, and then "pre-image samples" short vectors $\mathbf{s}_1, \mathbf{s}_2$ such that

$$[\ \mathbf{a}^T \mid \mathbf{b}^T + i \cdot [\ 1\ \sqrt{q}\ ]\ ] \cdot \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} = u + \mathbf{a}_2^T \mathbf{s}_3. \tag{4.3}$$

The matrix $[\ 1\ \sqrt{q}\ ]$[3] is the "gadget matrix" defined in Section 2.3.5 that allows for efficient pre-image sampling of short vectors $\mathbf{s}_1, \mathbf{s}_2$ as in Equation (4.3) for all $i \neq 0$. When $i = 0$, the group manager can output $\mathbf{s}_1', \mathbf{s}_2', -\mathbf{s}_3'$ as the key.[4] The purpose of the $\mathbf{a}_2^T \mathbf{s}_3$ part of the construction is only necessary for the proof – it's unclear if it truly serves any security purpose. In the security proof, because $\mathbf{b}$ is only computationally-indistinguishable from random (rather than statistically), one needs a "double trap-door" for switching between games and $\mathbf{a}_2$ serves that purpose. Normally, this would decrease the efficiency of the scheme because one would need to include the extra term $\mathbf{a}_2 \cdot \mathbf{s}_3$ in the security proof. But in our case, we use the $\mathbf{a}_2$ that appears as the common reference string in our commitment scheme, and this term anyway appears in our proof below. The only downside is that in practice one needs to do the extra sampling of $\mathbf{s}_3$ and multiplication by $\mathbf{a}_2$ in the key generation, and that the size of the solution to the Ring-SIS problem in the security proof is a small (virtually inconsequential) additive factor larger.

Signing. The high level idea for signing is for the user with identity i to prove knowledge of $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$ that satisfies Equation (4.3). If the proof of knowledge is a Σ-protocol, then it can be converted into a non-interactive proof using the Fiat-Shamir heuristic, which turns the Σ-protocol into a signature scheme if one inputs the message into the random oracle. The main difficulty lies in doing this proof without revealing i.

To hide i in our proof, the signer will commit to i and $i\sqrt{q}$ using the commitment scheme from Section 2.4.1 and publish his commitments as part of the signature. The main observation is that

$$\left[\begin{bmatrix} \mathbf{0} \\ \mathbf{a}^T \end{bmatrix} \mid \begin{bmatrix} \mathbf{0} \\ \mathbf{b}^T \end{bmatrix} + [\mathbf{Com}(i; \mathbf{r})\ \mathbf{Com}(i\sqrt{q}; \mathbf{r}')]\right] \cdot \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ u + \mathbf{a}_2^T \mathbf{s}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \cdot \tilde{\mathbf{r}}. \tag{4.4}$$

The signer will give an approximate ZKPoK of the short randomnesses $\mathbf{r}, \mathbf{r}'$ that open the commitments to $i, i\sqrt{q}$ and also that $i \in S$.[5] In other words, he'll prove knowledge of

[3] We write $\sqrt{q}$ instead of $\lceil \sqrt{q} \rfloor$ for readability.

[4] For $i \neq 0$, the group manager is able to output many possible valid $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$ using his trapdoor and the gadget matrix. For $i = 0$, however, the gadget matrix disappears and so the group manager is only able to return one $\mathbf{s}_1', \mathbf{s}_2', -\mathbf{s}_3$ that he "planted" when creating u in Equation (4.2). For the security proof, it will be necessary that the distribution for all i is the same, and so for this reason, we make the pre-image sampling procedure for all i deterministic. In other words, the randomness used in the sampling will be derived by the group manager using a keyed PRF whose input depends on i.

[5] Due to the slack in our zero-knowledge protocols, the proofs will be for larger values of $\mathbf{r}, \mathbf{r}'$ than those used in the commitments. But for simplicity of exposition in the introduction of this paper, we will use the same notation.

$$\begin{bmatrix}\mathbf{a}_1^T\\ \mathbf{a}_2^T\end{bmatrix}\cdot[\mathbf{r}\ \mathbf{r}']+\begin{bmatrix}0 & 0\\ ci & ci\sqrt{q}\end{bmatrix}=c\cdot\begin{bmatrix}t_1^{(1)} & t_1^{(2)}\\ t_2^{(1)} & t_2^{(2)}\end{bmatrix}. \tag{4.5}$$

In parallel, the signer will also prove that

$$\left[\mathbf{a}^T \mid \mathbf{b}^T+[t_2^{(1)}\ t_2^{(2)}]\right]\cdot\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}=cu+\mathbf{a}_2^T\mathbf{r}''.^{6} \tag{4.6}$$

Multiplying Equation (4.6) by c and combining with Equation (4.5) produces the equation

$$\left[\mathbf{a}^T \mid c\mathbf{b}^T+\mathbf{a}_2^T\cdot[\mathbf{r}\ \mathbf{r}']+c\cdot[i\ i\sqrt{q}]\right]\cdot\begin{bmatrix}c\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}=c^2u+c\mathbf{a}_2^T\mathbf{r}''. \tag{4.7}$$

We can then show that if an Adversary can produce polynomial vectors $\mathbf{r},\mathbf{r}',\mathbf{s}_1,\mathbf{s}_2,\mathbf{r}'',c$ with small coefficients that satisfy the above equation, then he is able to solve the Ring-SIS problem. The proof is very similar to the proof of selective security for the signature scheme of [ABB10] . Intuitively, suppose that the Adversary in the impersonation game produces a solution (i.e. the extracted values from the PoK) for Equation (4.7) for $i=0$. Then, using the fact that $\mathbf{b}^T=\mathbf{a}^T\mathbf{R}$ and $u=\mathbf{a}^T\mathbf{s}_1'+\mathbf{b}^T\mathbf{s}_2'+\mathbf{a}_2^T\mathbf{s}_3'$ and writing $\mathbf{R}'=[\mathbf{r}\ \mathbf{r}']$, Equation (4.7) can be rewritten as

$$\mathbf{a}^T\left(c\mathbf{s}_1+c\mathbf{R}\mathbf{s}_2-c^2\mathbf{s}_1'-c^2\mathbf{R}\mathbf{s}_2'\right)+\mathbf{a}_2^T\left(\mathbf{R}'\mathbf{s}_2-c\mathbf{r}''-c^2\mathbf{s}_3'\right)=0, \tag{4.8}$$

which is a solution to the Ring-SIS problem because the coefficients of all the terms in parentheses are small relative to q.

Of course, the Adversary is not guaranteed to impersonate on identity $i=0$, but may choose an arbitrary $i'\in S$. To handle this, we use the standard "puncturing" technique. In the security proof we would not choose $\mathbf{b}^T=\mathbf{a}^T\mathbf{R}$ as part of the public key, but we rather pick a uniformly-random "guess" $i'\in S$, and set $\mathbf{b}^T=\mathbf{a}^T\mathbf{R}-[i'\ i'\sqrt{q}]$ as part of the public key. It's not hard to see that if the Adversary produces a solution for Equation (4.7) with $i=i'$, then one again obtains the same Ring-SIS solution as in Equation (4.8). If the Adversary cannot tell how $\mathbf{b}$ was constructed, even after querying for preimages $\mathbf{s}_1,\mathbf{s}_2,\mathbf{s}_3$, then there is exactly a $1/|S|$ chance that $i=i'$. Therefore there is a $1/|S|$ loss in the tightness of the security reduction. We give a first set of parameters which does not take this loss in tightness into account, there are multiple reason we consider such parameters. First the loss in tightness does not correspond to a practical attack, if we consider how to concretely attack the scheme it is not clear what advantage this reduction from a selective signature would give. The second reason being that the best cryptanalytic algorithms (c.f Section 2.3.4) run in exponential space which would arguably not be affected by a loss in tightness, for our parameters polynomial space algorithms would guarantee much more than 128 bits of security regardless of any complexity leveraging. Finally we include a second set of parameters in Table 4.2 (and the corresponding signature size in Table 4.1) which obtain more than 128 bits of post-quantum security even when accounting for the loss in tightness.

For the purposes of allowing opening, the signer will also create a Ring-LWE encryption of the three polynomials comprising the vector $\mathbf{r}$ used in the commitment of i in Equation (4.5)

[6] Notice that we combined the $\mathbf{a}_2\cdot\mathbf{s}_3$ term with $\mathbf{a}_2\cdot\tilde{\mathbf{r}}$ term to obtain $\mathbf{a}_2\cdot\mathbf{r}''$. This was the reason that we used exactly $\mathbf{a}_2$ from the commitment scheme in the key generation in Equation (4.3).

using the one-shot verifiable encryption / proof of plaintext knowledge from [LN17] combined with the proofs of knowledge for Equation (4.5). The reason that we encrypt $\mathbf{r}$ rather than i is that the coefficients in $\mathbf{r}$ are small, whereas i comes from a set that is stable under some automorphism, and such sets contain elements with large coefficients. Once the opener decrypted the $\mathbf{r}$, he knows from Equation (4.5) that $\mathbf{a}_1^T\mathbf{r} = c \cdot t_1^{(1)}$, and so he can recover c. Then using this c, he can recover i from the equality $\mathbf{a}_2^T\mathbf{r} + ci = c \cdot t_2^{(1)}$.

Reducing the Commitment Size. To reduce the size of the signature, we can slightly modify the commitment so that it works over two different moduli, one for the top and another for the bottom part (call them q_1 and q_2 respectively). In our group signature scheme, the value of q_2 needs to be large due to the fact that the Ring-SIS solution in Equation (4.8) is fairly large itself. The value of q_1, on the other hand, only needs to be set so that the commitments to i and $i\sqrt{q}$ in Equation (4.5) are binding and hiding. Since a smaller q_1 will result in smaller sizes of $t_1^{(1)}$, $t_1^{(2)}$ in the commitment, it is sensible to set it as small as possible. We show that our proofs of automorphism stability still work if the two moduli are different.

4.2 Our Group Signature

4.2.1 Commitment Scheme

We will use the commitment scheme of Section 2.4.1 for messages in $\mathcal{R}_{q_2}$. For the purpose of this scheme it will be advantageous to use two different moduli for the "top" and "bottom" part of the commitment. The security of our group signature will rely on the hardness of finding collisions on the bottom part of the commitment matrix (while the binding property of the commitment scheme only needs the top part to be collision resistant), this is why taking a larger modulo q_2 for the bottom part makes sense. While we prove that the hiding property of the scheme when taking two moduli can be reduced to M-LWE if the randomness is taken from a skewed distribution, here we will take $\mathbf{r} \leftarrow S_1^k$ where $S_1 = \mathcal{U}(\{x \in \mathcal{R} \ : \ \|x\|_\infty = 1\})$. As argued in Section 2.4.1 the fact that the distribution of the randomness is skewed in the commitment with different moduli is mostly an artefact of the reduction. Moreover we chose to take a small uniform distribution rather than Gaussian for simplicity, for a bounded number of samples the cryptanalysis of Section 2.3.4 still represents the best known attacks for such a distribution.

- **CSetup**(1^λ) outputs a commitment matrix $\mathbf{A} := \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix}$ with $\mathbf{a}_1 \in \mathcal{R}_{q_1}^3$ and $\mathbf{a}_2 \in \mathcal{R}_{q_2}^3$.
- **Com**$(m \in \mathcal{R}_{q_2})$ outputs the commitment $\mathbf{t} := \mathbf{A}\mathbf{r} + \begin{bmatrix} 0 \\ m \end{bmatrix}$, and randomness $\mathbf{r} \leftarrow S_1^3$.
- **Open**$(\mathbf{t}, \mathbf{r}, c)$ checks that the commitment is valid and outputs $\tilde{m} := t_2 - c^{-1}\mathbf{a}_2^T\mathbf{r} \in \mathcal{R}_{q_2}$.

We have fixed the dimensions of $\mathbf{A}$ to 2×3 this entails that the scheme relies on the R-LWE problem for hiding and R-SIS problem for binding.

ZKPoKs with different moduli. The proofs of knowledge presented in Chapter 3 only consider one modulus q, they can however be seamlessly adapted to two moduli. We will use the proof for linear relations of Section 3.2.2 to prove that two commitments $\mathbf{t}_1$ and

	Full signature	Commitment	Ciphertext	Proof	Secret key
Without Complexity Leveraging	581 KB	113 KB	123 KB	345 KB	146 KB
With Complexity Leveraging	1173 KB	204 KB	254 KB	715 KB	292 KB

Table 4.1: Size of a signature

$\mathbf{t}_2$ open to messages $m_1, m_2 \in \mathcal{R}_{q_2}$ such that $m_2 = \sqrt{q}m_1$, and the proof of automorphism stability to prove that a commitment $\mathbf{t}$ opens to a message $m \in \mathbb{Z}_{q_2}$. We do not recall these ZKPoKs as they are identical to the ones of Section 3.2, we however formally describe the proof output by the signature of a group member in Section 4.2.5. This proof will contain the two aforementioned proofs as well as a proof of verifiable encryption and will be used as a signature using the Fiat-Shamir transform.

4.2.2 Definitions

We first recall the definitions and security model of group signatures. A group signature scheme consists of a tuple of four algorithms (**GSetup**, **Sign**, **Verify**, **Open**):

- **GSetup**$(1^\lambda, 1^N)$:Takes as input the security parameter λ as well as the maximum number of identities N. Outputs the group public key gpk, the group manager secret key $gmsk$, and the secret keys of each identity $sk_1, \ldots, sk_N$.
- **Sign**(sk_i, M) : Takes as input a user secret key sk_i and a message $M \in \{0,1\}^*$. Outputs a signature z of M.
- **Verify**(gpk, M, z) : Takes as input the group public key gpk, a message M, and a signature z. Outputs 1 if z is a valid signature of M and 0 otherwise.
- **Open**$(gmsk, M, z)$: Takes as input the group manager secret key $gmsk$, a message M, and a valid signature z of M. Outputs an identity $id \in [N]$ or $\perp$.

For correctness, we want that for any $(gpk, gmsk, sk_1, \ldots, sk_n) \leftarrow$ **GSetup**(1^λ), any $j \in [N]$, any $M \in \{0,1\}^*$, and any $z \leftarrow$ **Sign**(gpk, sk_j, M), with overwhelming probability:

$$\textbf{Verify}(gpk, M, z) = 1, \text{ and } \textbf{Open}(gpk, gmsk, M, z) = j$$

The security of the group signature is captured by two notions: anonymity and traceability [BMW03]. For anonymity we consider a PPT adversary $\mathcal{A}$ who has access to all the signing keys $sk_1, \ldots, sk_N$ but not the manager key $gmsk$. $\mathcal{A}$ chooses a message M and two identities i_0 and i_1, his goal is to distinguish between signatures of M under these identities. There are multiple flavors of anonymity depending on whether $\mathcal{A}$ can access an opening oracle (full

anonymity) or not (weak anonymity), intuitively full anonymity will be achieved when the PKE used in the opening is CCA-secure while weak anonymity corresponds to a CPA-secure encryption scheme. In this paper we present a weakly anonymous group signature. The verifiable encryption scheme we use [LN17] can achieve CCA security but this comes at a cost in efficiency, moreover it is likely that in real implementations access to the tracing functionality of the scheme will be restricted, in which case weak anonymity seems to be an appropriate notion.

$\mathsf{Exp}^{anon-b}(\mathcal{A}, \lambda, N)$:
$(gpk, gmsk, sk_1, \ldots, sk_N) \leftarrow \mathbf{GSetup}(1^\lambda, 1^N)$
$(i_0, i_1, M, St) \leftarrow \mathcal{A}(gpk, sk_1, \ldots, sk_N)$
$z^* \leftarrow \mathbf{Sign}(gpk, sk_{i_b}, M^*)$
$b' \leftarrow \mathcal{A}(z^*, St)$ Return b'

Figure 4.1: Experiment for weak anonymity

Definition 4.2.1 (Weak anonymity)**.** *For a group signature* (**GSetup**, **Sign**, **Verify**, **Open**)*, we define weak anonymity with adversary* $\mathcal{A}$ *via the experiment of Figure 4.1. The advantage of* $\mathcal{A}$ *is:*

$$\mathsf{Adv}^{anon}(\mathcal{A}, \lambda, N) := |\Pr\left[\mathsf{Exp}^{anon-1}(\mathcal{A}, \lambda, N) = 1\right] - \Pr\left[\mathsf{Exp}^{anon-0}(\mathcal{A}, \lambda, N) = 1\right]|$$

In full-traceability the adversary $\mathcal{A}$ has access to the signing keys $(sk_i)_{i \in S}$ for any arbitrary set $S \subset [N]$ (possible $S = [N]$) as well as the manager secret key $gmsk$, his goal is to produce a valid signature z of some message M (i.e. which passes verification) such that either $\mathbf{Open}(gpk, gmsk, M, z) = j \notin S$ or $\mathbf{Open}(gpk, gmsk, M, z) = \perp$. Full-traceability captures the notion that all signatures, even when computed by a collusion of users and the group manager, should trace to a member of the forging coalition.

Definition 4.2.2 (Weak anonymity)**.** *For a group signature* (**GSetup**, **Sign**, **Verify**, **Open**)*, we define full traceability with adversary* $\mathcal{A}$ *via the experiment of Figure 4.2. The advantage of* $\mathcal{A}$ *is:*

$$\mathsf{Adv}^{trace}(\mathcal{A}, \lambda, N) := \Pr\left[\mathsf{Exp}^{trace}(\mathcal{A}, \lambda, N) = 1\right]$$

4.2.3 The Scheme

The group signature we present in this section will be for fixed parameters as per Table 4.2, for which the signatures will be of size 581 KB, as described in appendix 4.3. In particular we consider the power-of-two cyclotomic ring $\mathcal{R} = \mathbb{Z}[X]/X^{4096} + 1$, and identity set $[N] = \mathbb{Z}_{q_2}$. This entails that user identities are exactly the elements $x \in \mathcal{R}_{q_2}$ that are left invariant under the automorphisms $\xi_{-1} : X \to X^{-1} = -X^{d-1}$ and $\xi_5 : X \to X^5$. We also use commitments that rely on R-LWE and R-SIS (which can be seen as specific instances of the module variant of the corresponding problems for modules of dimension 1). Using other cyclotomic rings can result in smaller signatures (especially since for some of them only one automorphism is needed to prove that elements belong to $\mathbb{Z}_{q_2}$), and using higher dimension commitments that rely on the module variants of LWE and SIS would allow for more fine-tuned parameters.

$\mathsf{Exp}^{trace-b}(\mathcal{A}, \lambda, N)$:
  $S \leftarrow$ Empty List
  $(gpk, gmsk, sk_1, \ldots, sk_N) \leftarrow \mathbf{GSetup}(1^\lambda, 1^N)$
  $(M^*, z^*) \leftarrow \mathcal{A}^{\mathcal{O}\mathbf{Sign}, \mathcal{O}\mathbf{Corrupt}}(gpk, gmsk)$
  If $\mathbf{Verify}(gpk, M^*, z^*) = 0$ return 0
  If $\mathbf{Open}(gmsk, M^*, z^*) = \perp$ return 1
  If $\mathbf{Open}(gmsk, M^*, z^*) = j^* \in [N]$
    and $j^* \notin C$
    and $(j^*, M^*) \notin S$
  Then return 1
  Else return 0

$\mathcal{O}\mathbf{Sign}(i, M)$:
  $S \leftarrow S \cup \{(i, M)\}$
  Return $\mathbf{Sign}(sk_i, M)$

$\mathcal{O}\mathbf{Corrupt}(i)$:
  $C \leftarrow C \cup i$
  Return sk_i

Figure 4.2: Experiment for full traceability

We have chosen the parameters in this section as such for easier presentation and because they allow for simpler implementations.
We will first present in this section a group signature scheme without opening, and show in section 4.2.4 how to add an opening.
Let $\delta = \lceil \sqrt{q_2} \rceil$, and $\mathbf{g}^T$ be the gadget matrix $\begin{bmatrix} 1 & \delta \end{bmatrix} \in \mathcal{R}_{q_2}^{1\times 2}$, we will consider the set of identities $\mathcal{I}d = \mathbb{Z}_{q_2}$.
GSetup(1^λ):

- Sample $\mathbf{A} := \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \end{bmatrix} \leftarrow \mathbf{CSetup}(1^\lambda)$, with $\mathbf{a}_1 \in \mathcal{R}_{q_1}^3$, and $\mathbf{a}_2 \in \mathcal{R}_{q_2}^3$.
- Sample $\mathbf{a} \xleftarrow{\$} \mathcal{R}_{q_2}^2$.
- Sample $\mathbf{R} \xleftarrow{\$} S_1^{2\times 2}$ and set $\mathbf{b}^T = \mathbf{a}^T\mathbf{R} \in \mathcal{R}_{q_2}^{1\times 2}$.
- Sample $(\mathbf{s}_{0_1}, \mathbf{s}_{0_2}, \mathbf{s}_{0_3}) \leftarrow D_{R,s} \times D_{R,s} \times D_{R,r}$
- Set $u := \begin{bmatrix} \mathbf{a}^T & | & \mathbf{b}^T & | & \mathbf{a}_2^T \end{bmatrix} \begin{bmatrix} \mathbf{s}_{0_1} \\ \mathbf{s}_{0_2} \\ \mathbf{s}_{0_3} \end{bmatrix}$
- Set $gpk := (\mathbf{A}, \mathbf{a}, \mathbf{b}, u)$
- For $i \in \mathbb{Z}_{q_2}^*$, sample $\mathbf{s}_{i_3} \leftarrow D_{\mathcal{R},r}^3$

Parameter	Notation	Value	Value with complexity leveraging
Ring dimension	d	4096	8192
Commitment modulus ("Top")	q_1	$\sim 2^{30}$	$\sim 2^{20}$
Commitment modulus ("Bottom")	q_2	$\sim 2^{80}$	$\sim 2^{80}$
Commitment row dimension ("Top")	n	1	1
Commitment message dimension	l	1	1
Verifiable encryption plaintext module	p	$\sim 2^{27}$	$\sim 2^{27}$
Verifiable encryption ciphertext module	Q	$\sim 2^{60}$	$\sim 2^{62}$
Bound on the challenge space	$B_{\mathcal{C}} = \kappa$	26	24
Standard deviation of the GPV trapdoor	s	$\sim 2^{49}$	$\sim 2^{50}$
Standard deviation of the NTRU trapdoor	r	$\sim 2^{42}$	$\sim 2^{42}$

Table 4.2: Concrete parameters for our Group signature

- For $i \in \mathbb{Z}_{q_2}^*$, sample $(\mathbf{s}_{i_1}, \mathbf{s}_{i_2}) \in \mathcal{R}^4$ s.t $\begin{bmatrix}\mathbf{a}^T & | & \mathbf{b}^T + i\mathbf{g}^T\end{bmatrix} \begin{bmatrix}\mathbf{s}_{i_1} \\ \mathbf{s}_{i_2}\end{bmatrix} = u - \mathbf{a}_2^T \mathbf{s}_{i_3}$, and $(\mathbf{s}_{i_1}, \mathbf{s}_{i_2}) \leftarrow D_{\mathcal{R},s}^4$
- For $i \in \mathbb{Z}_{q_2}$, set $sk_i := \mathbf{s}_i := (\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \mathbf{s}_{i_3})$

Intuitively user i could sign a message $M \in \{0,1\}^*$ by doing a non-interactive proof that he knows a small $\mathbf{s} \in \mathcal{R}^6$ such that $\begin{bmatrix}\mathbf{a}^T & | & \mathbf{b}^T + i\mathbf{g}^T & | & \mathbf{a}_2^T\end{bmatrix} \mathbf{s} = u$ in which the message is part of the hash that generates the challenge. However doing so would reveal his identity, indeed a verifier would need to know the matrix $\begin{bmatrix}\mathbf{a}^T & | & \mathbf{b}^T + i\mathbf{g}^T & | & \mathbf{a}_2^T\end{bmatrix}$ to verify the signature and since $\mathbf{a}$, $\mathbf{a}_2$, $\mathbf{b}$, and $\mathbf{g}$ are public he could recover the identity of the signer. As explained in section 4.1.2 we circumvent this issue by committing to the part of the matrix that depends on i (that is $i\mathbf{g}^T$) and proving that the previous equation still stands but for a commitment of u rather than u itself.

Sign$(M, \mathbf{s}_i)$:

- Set $\mathbf{t} := \begin{bmatrix}t_1 \\ t_2\end{bmatrix} = \mathbf{Com}(i, \mathbf{r}) \in \mathcal{R}_{q_1} \times \mathcal{R}_{q_2}$, where $\mathbf{r} \leftarrow S_1^3$.
- Set $\mathbf{t}' := \begin{bmatrix}t'_1 \\ t'_2\end{bmatrix} = \mathbf{Com}(i\delta, \mathbf{r}')$, where $\mathbf{r}' \leftarrow S_1^3$.
- Set $\mathbf{v}^T := \begin{bmatrix}\mathbf{a}^T & | & \mathbf{b}^T + \begin{bmatrix}t_2 & t'_2\end{bmatrix} & | & \mathbf{a}_2^T\end{bmatrix} \in \mathcal{R}_{q_2}^{1\times 7}$, and $\mathbf{s}' = \begin{bmatrix}\mathbf{s}_{i_1} \\ \mathbf{s}_{i_2} \\ \mathbf{s}_{i_3} - \begin{bmatrix}\mathbf{r} & \mathbf{r}'\end{bmatrix} \mathbf{s}_{i_2}\end{bmatrix} \mathcal{R}^7$, observe that $\mathbf{v}^T \mathbf{s}' = u$
- Compute $\Pi_1 = \Pi_{lin}(\mathbf{t}, \mathbf{t}', \delta; \mathbf{r}, \mathbf{r}', i, i\delta)$
- Compute $\Pi_2 = \Pi_{auto}(\mathbf{t}, (\xi_{-1}, \xi_5); \mathbf{r}, i)$

- Compute $\Pi_3 = \Pi_{OWF}(\mathbf{v}; \mathbf{s}')$
- Output the signature $z = (\mathbf{t}, \mathbf{t}', \Pi_1, \Pi_2, \Pi_3)$

The proofs $\Pi_{1,2,3}$ will use the Fiat-Shamir heuristic to transform interactive proofs into non-interactive proofs in the random oracle model, we will also include the message M in the random oracle call to obtain a signature. For unforgeability we will need all of these proofs to be executed with the same challenge, i.e. the signer will run all three proofs in parallel and compute a common challenge for all three as a hash of all relevant information. We describe the full non-interactive proof, including the opening, in more details in Section 4.2.5.
To verify a signature one simply verifies the proofs $\Pi_{1,2,3}$.
Verify$(\mathbf{t}, \mathbf{t}', \Pi_1, \Pi_2, \Pi_3)$:

- Let $\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} := \mathbf{t}$
- Let $\begin{bmatrix} t'_1 \\ t'_2 \end{bmatrix} := \mathbf{t}'$
- Let $\mathbf{v}^T = \begin{bmatrix} \mathbf{a}^T & | & \mathbf{b}^T + \begin{bmatrix} t_2 & t'_2 \end{bmatrix} & | & \mathbf{a}_2^T \end{bmatrix}$
- Verify Π_1 using $\mathbf{t}, \mathbf{t}', \delta$
- Verify Π_2 using $\mathbf{t}, \xi_{-1}, \xi_5$
- Verify Π_3 using $\mathbf{v}$

4.2.4 Adding the Opening

To be able to open the group signature scheme of Section 4.2.3 we will add a verifiable encryption to the signature. In essence we want the signer to encrypt his identity, using a public key associated to a decryption key that the group manager possesses, and prove that this encryption is indeed of his identity. To do so we will encrypt the randomness $\mathbf{r}$ of $\mathbf{t} = \mathbf{Com}(id; \mathbf{r})$ and prove that $\mathbf{a}_1^T \mathbf{r} = t_1$, note that encrypting id directly would result in a smaller ciphertext but a very large proof since id itself is not small. We use the verifiable encryption of [LN17] which consists in a *R-LWE* encryption and a proof of knowledge. We let p be the modulus of the plaintext space of our encryption scheme (which we only need large enough to accommodate the decryption slack, see [LN17]) and Q the modulus of the ciphertext.
PKESetup(1^λ):

- Sample $a \xleftarrow{\$} \mathcal{R}_Q$
- Sample $\mathbf{s}, \mathbf{e} \leftarrow S_1^3$
- Set $\mathbf{b} := a\mathbf{s} + \mathbf{e} \in \mathcal{R}_Q^3$
- Output $(\mathbf{s}, (a, \mathbf{b}))$

Encryption will consist in creating a standard *R-LWE* encryption and a proof that the message $\mathbf{r}$ encrypted is the randomness in $\mathbf{t} = \mathbf{Com}(id; \mathbf{r})$.
Enc$((a, \mathbf{b}), \mathbf{r}, t_1)$:

- Sample $r, e_1 \leftarrow S_1$
- Sample $\mathbf{e}_2 \leftarrow S_1^3$
- Set $u := p(ar + e_1)$
- Set $\mathbf{v} := p(\mathbf{b}r + \mathbf{e}_2) + \mathbf{m}$
- Set $\mathbf{M}_1 := \begin{bmatrix} pa & p & 0 & 0 & 0 & 0 & 0 & 0 \\ pb_1 & 0 & p & 0 & 0 & 1 & 0 & 0 \\ pb_2 & 0 & 0 & p & 0 & 0 & 1 & 0 \\ pb_3 & 0 & 0 & 0 & p & 0 & 0 & 1 \end{bmatrix} \in \mathcal{R}_Q^{4\times 8}$
- Set $\mathbf{M}_2 := \begin{bmatrix} \mathbf{0}^{1\times 5} & \mathbf{a}_1^T \end{bmatrix} \in \mathcal{R}_{q_1}^{1\times 8}$
- Set $\mathbf{M} := \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{bmatrix}$
- Set $\mathbf{x} := \begin{bmatrix} r \\ e_1 \\ \mathbf{e}_2 \\ \mathbf{r} \end{bmatrix} \in R^8$
- Set $\mathbf{y} := \begin{bmatrix} u \\ \mathbf{v} \\ t_1 \end{bmatrix} \in \mathcal{R}_Q^4 \times \mathcal{R}_{q_1}$
- Set $\Pi := \Pi_{OWF}(\mathbf{M}, \mathbf{y}; \mathbf{x})$
- Output $(u, \mathbf{v}, \Pi)$

To verify an encryption one simply verifies the proof Π.
Verify$((u, \mathbf{v}, \Pi), t_1)$:

- Set $\mathbf{M}_1 := \begin{bmatrix} pa & p & 0 & 0 & 0 & 0 & 0 & 0 \\ pb_1 & 0 & p & 0 & 0 & 1 & 0 & 0 \\ pb_2 & 0 & 0 & p & 0 & 0 & 1 & 0 \\ pb_3 & 0 & 0 & 0 & p & 0 & 0 & 1 \end{bmatrix} \in \mathcal{R}_Q^{4\times 8}$
- Set $\mathbf{M}_2 := \begin{bmatrix} \mathbf{0}^{1\times 5} & \mathbf{a}_1^T \end{bmatrix} \in \mathcal{R}_{q_1}^{1\times 8}$
- Set $\mathbf{M} := \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{bmatrix}$
- Set $\mathbf{y} := \begin{bmatrix} u \\ \mathbf{v} \\ t_1 \end{bmatrix} \in \mathcal{R}_Q^4 \times \mathcal{R}_{q_1}$
- Output $\mathbf{Verify}_{OWF}(\Pi, \mathbf{M}, \mathbf{y})$

Decryption is not as simple as standard R-LWE decryption. By completeness we know that honestly generated ciphertexts can be decrypted but soundness should guarantee that as long as the proof verifies one should be able to decrypt. This is not clear since the proof Π does not imply that $(u, \mathbf{v})$ is a valid ciphertext but that there exists some $\bar{c} \in \bar{\mathcal{C}}$ such that $(\bar{c}u, \bar{c}\mathbf{v})$ is a valid ciphertext and we do not know which one. In [LN17] the authors show that in fact trying random $\bar{c}$ is a valid approach and will take as many attempts as the number of oracle calls that were needed to generate the proof (in particular only one attempt is necessary if the prover is honest). This will be sufficient for our scheme.
Dec$((u, \mathbf{v}, \Pi), \mathbf{s})$:

- If **Verify**$((u, \mathbf{v}, \Pi), t_1) = 1$, Let c be the challenge used in Π
- Loop:
- $c' \leftarrow \mathcal{C}$
- $\bar{c} := c - c'$
- $\bar{\mathbf{r}} := (\mathbf{v} - u\mathbf{s})\bar{c} \mod Q$
- If $\|\bar{\mathbf{r}}\|_\infty \leq Q/8B_{\mathcal{C}}$ then:
- $\bar{\mathbf{r}} := \bar{\mathbf{r}} \mod q$
- return $(\bar{\mathbf{r}}, \bar{c})$

The following lemma shows that if decryption succeeds then the decrypted value $(\bar{\mathbf{r}}, \bar{c})$ will essentially be a preimage for the zero-knowledge proof.

Lemma 4.2.3 ([LN17] Lemma 3.1)**.** *Let $sk = \mathbf{s}$, and $\mathbf{e}$ be the error in $\mathbf{b} = a\mathbf{s} + \mathbf{e}$. If for given*

$$(u, \mathbf{v}, t_1) \in \mathcal{R}_Q^4 \times \mathcal{R}_{q_1}$$

there exists $\bar{\mathbf{r}}_M := (\bar{r}, \bar{e}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{r}} \in \mathcal{R}^8$, and $\bar{c} \in \mathcal{R}$ such that :

$$\mathbf{Mr}_B = \begin{bmatrix} u & \mod Q \\ \mathbf{v} & \mod Q \\ t_1 & \mod q_1 \end{bmatrix}$$

and

$$\|p(\bar{u}\mathbf{e} + \bar{\mathbf{e}}_2 - \bar{e}\mathbf{s}) + \bar{\mathbf{r}}\|_\infty \leq Q/4B_{\mathcal{C}} \tag{4.9}$$

Then for $(\bar{\mathbf{r}}', \bar{c}') = \mathbf{Dec}(u, \mathbf{v}, \Pi, t_1)$, we have:

$$\frac{\bar{\mathbf{r}}}{\bar{c}} \mod p = \frac{\bar{\mathbf{r}}'}{\bar{c}'} \mod p$$

Once we have verifiable encryption adding traceability to our group signature is straightforward. During key generation we will create $(pk, sk) \leftarrow$ **PKESetup**(1^λ), add pk to the group public key and set $gmsk = sk$. When signing a user will compute an encryption v of his randomness $\mathbf{r}$, which is such that $\mathbf{a}_1^T\mathbf{r} = t_1 \mod q_1$, and add v to the signature. For verification one only needs to check the extra proof Π. We consider how to open a signature, this is not completely straightforward because soundness only guarantees that a verifying signature will open to $\bar{c}\mathbf{r}$ for some $\bar{c} \in \bar{\mathcal{C}}$.
Open(msk, z) :

- Parse z as $(\mathbf{t}, \mathbf{t}', \Pi_1, \Pi_2, \Pi_3, v)$
- Let $(\bar{\mathbf{r}}, \bar{c})$ =Dec(msk, t_1, z)
- Set $id := \bar{c}^{-1}(t_2 - \mathbf{a}_2^T \bar{\mathbf{r}}) \in \mathcal{R}_{q_2}$
- If $id \in \mathbb{Z}_{q_2}$ then output id, otherwise output $\perp$

Note that if decryption succeeds then the proof Π verifies, which entails that there exists $\bar{\mathbf{r}}', \bar{c}'$ such that $\mathbf{a}_1^T \bar{\mathbf{r}}' = \bar{c}' t_1 \mod q_1$ and by lemma 4.2.3 we know that:

$$\frac{\bar{\mathbf{r}}}{\bar{c}} \mod p = \frac{\bar{\mathbf{r}}'}{\bar{c}'} \mod p$$

if we multiply this equation by $\bar{c}$ and $\bar{c}'$ we have that $\bar{\mathbf{r}}' \bar{c} = \bar{\mathbf{r}} \bar{c}' \mod p$, and since both sides are smaller than p this equation will be true over the integer. From which we get:

$$\mathbf{a}_1^T \bar{\mathbf{r}} = \bar{c} t_1 \mod q_1$$

which entails that if $\mathbf{t} = (t_1, t_2)$ is a well formed commitment the identity returned by the Open algorithm will be its message.

4.2.5 The Full Non-Interactive Proof

We give the full non-interactive zero-knowledge proof that the signer will output. We only consider the parameter choice made in section 4.3. The user $i \in \mathbb{Z}_{q_2}$ will use the following elements for his proof:

$$\mathbf{t} := \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \mathbf{A}\mathbf{r} + \begin{bmatrix} 0 \\ i \end{bmatrix} \in \mathcal{R}_{q_1} \times \mathcal{R}_{q_2}$$

$$\mathbf{t}' := \begin{bmatrix} t_1' \\ t_2' \end{bmatrix} = \mathbf{A}\mathbf{r}' + \begin{bmatrix} 0 \\ i\delta \end{bmatrix} \in \mathcal{R}_{q_1} \times \mathcal{R}_{q_2}$$

$$\mathbf{v}^T = \begin{bmatrix} \mathbf{a}^T & | & \mathbf{b}^T + \begin{bmatrix} t_2 & t_2' \end{bmatrix} & | & \mathbf{a}_2^T \end{bmatrix} \in \mathcal{R}_{q_2}^{1 \times 7}$$

$$\mathbf{s}' = \begin{bmatrix} \mathbf{s}_{i_1} \\ \mathbf{s}_{i_2} \\ \mathbf{s}_{i_3} - \begin{bmatrix} \mathbf{r} & \mathbf{r}' \end{bmatrix} \mathbf{s}_{i_2} \end{bmatrix} \in \mathcal{R}^7$$

We first note that since $\mathbf{a}_2^T = \begin{bmatrix} 0 & 1 & a_2' \end{bmatrix}$, we can ignore the 5^{th} coefficient of $\mathbf{v}^T$ (corresponding to 0) in $\mathbf{v}^T \mathbf{s}' = u$ and thus consider $\mathbf{v}^T \in \mathcal{R}_{q_2}^{1 \times 6}$ and $\mathbf{s}' \in \mathcal{R}^6$ such that $\mathbf{v}^T \mathbf{s}' = u$. The gain in proof size obtained by discarding one element of this equation may seems negligible at first but it is in fact rather important because the last three coefficients of $\mathbf{s}'$ will be much larger than the other four. We also recall the matrices needed for the proof of verifiable encryption:

$$\mathbf{M}_1 = \begin{bmatrix} pa & p & 0 & 0 & 0 & 0 & 0 & 0 \\ pb_1 & 0 & p & 0 & 0 & 1 & 0 & 0 \\ pb_2 & 0 & 0 & p & 0 & 0 & 1 & 0 \\ pb_3 & 0 & 0 & 0 & p & 0 & 0 & 1 \end{bmatrix} \in \mathcal{R}_Q^{4 \times 8}$$

$$\mathbf{M}_2 = \begin{bmatrix} \mathbf{0}^{1 \times 5} & \mathbf{a}_1^T \end{bmatrix} \in \mathcal{R}_{q_1}^{1 \times 8}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 \\ \mathbf{M}_2 \end{bmatrix}$$

Which are such that :

$$\mathbf{M}\mathbf{r}_M = \begin{bmatrix} u & \bmod Q \\ v_1 & \bmod Q \\ v_2 & \bmod Q \\ v_3 & \bmod Q \\ t_1 & \bmod q_1 \end{bmatrix}, \text{ for } \mathbf{r}_M = \begin{bmatrix} r \\ e_1 \\ \mathbf{e}_2 \\ \mathbf{r} \end{bmatrix}$$

Rather than just computing all the proofs simultaneously one can optimize the proof. Remark that since the first reply in both Π_{lin} and Πaut is the same $\mathbf{z} = \mathbf{r}c + \mathbf{y}$, we will only to need to send it once. An important point for proof size will be rejection sampling. After doing rejection sampling $\mathrm{Rej}(\mathbf{z}, \mathbf{a}, \sigma)$ on a vector $\mathbf{z}$ we know by lemma 2.3.6 that all of its coefficients will be statistically close to D_σ with $\sigma \geq 11 \|\mathbf{a}\|$, meaning that for very imbalanced vectors it would be worthwhile to do rejection sampling multiple times. For example if $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2)$ with $\|\mathbf{a}_2\| >> \|\mathbf{a}_1\|$ then by doing two rejection samplings $\mathrm{Rej}(\mathbf{z}_1, \mathbf{a}_1, \sigma_1)$ and $\mathrm{Rej}(\mathbf{z}_2, \mathbf{a}_2, \sigma_2)$ one obtains a smaller vector $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$ at the cost of having acceptation probability $1/27$, since the proof is non-interactive aborts have a minimal impact and this approach can help reduce the proof size significantly. We will use two rejection samplings for $\mathbf{s}' \in \mathcal{R}^6$ in which the last two coefficients will be much larger than the other four (because they correspond to a product of $\mathbf{s}_i$ and $\mathbf{r}, \mathbf{r}'$). We will thus write $\mathbf{s}'$ as $\mathbf{s}' = (\mathbf{s}'_1, \mathbf{s}'_2) \in \mathcal{R}^4 \times \mathcal{R}^2$. We can now write the full zero-knowledge proof of the verifier.

Lemma 4.2.4. *Let* $\mathbf{r}, \mathbf{r}' \leftarrow S_1^3$, *let* $\mathbf{s}_{i_1}, \mathbf{s}_{i_2} \leftarrow D^2_{\mathcal{R},s}$, *let* $\mathbf{s}_{i_3} \leftarrow D^3_{\mathcal{R},r}$, *let* $u, v_1, v_2, v_3 \leftarrow S_1$. *Let* $\mathbf{t}, \mathbf{t}', \mathbf{v}, \mathbf{s}', \mathbf{M}, \mathbf{r}_M$ *be defined as previously. Let* $\sigma \geq 11B_{\mathcal{C}}\sqrt{20d}$, $\sigma_1 \geq 11B_{\mathcal{C}}\sqrt{8d}s$, $\sigma_2 \geq 11B_{\mathcal{C}}(d\sqrt{24}s + \sqrt{2d}r)$ *and* $B \geq 2\sqrt{10d}\sigma$, $B_1 \geq 2\sqrt{2d}\sigma_1$, $B_2 \geq 2\sqrt{d}\sigma_2$. *If* $B_{com} \geq 2B$, *then the algorithm* Π_{sign} *achieves the following properties:*

- Correctness: *The prover restarts with probability at most* $1/27 + 2^{-100}$, *and if he does not abort the verifier accepts with overwhelming probability.*
- Honest-Verifier Zero-Knowledge: *Signatures can be simulated with statistically indistinguishable distribution.*
- Special Soundness: *Given two accepting transcripts one can extract* $\bar{\mathbf{z}} \in \mathcal{R}^3$, $\bar{id} \in \mathbb{Z}_{q_2}$, $\bar{\mathbf{z}}' \in \mathcal{R}^3$, $\bar{\mathbf{z}}_s \in \mathcal{R}^7$, $\bar{\mathbf{z}}_B \in \mathcal{R}^8$, $\bar{c} \in \bar{\mathcal{C}}$ *such that:*

$$\begin{aligned} &\bar{c}\mathbf{t} = Com(\bar{c}\bar{id}; \bar{\mathbf{z}}) \\ &\bar{c}\mathbf{t}' = Com(\bar{c}\bar{id}\delta; \bar{\mathbf{z}}') \\ &\bar{c} \begin{bmatrix} u \\ v_1 \\ v_2 \\ v_3 \\ t_1 \end{bmatrix} = \mathbf{M}\bar{\mathbf{z}}_M \\ &\bar{c}u = \mathbf{v}^T\bar{\mathbf{z}}_s \end{aligned}$$

  such that $\|(\bar{\mathbf{z}}, \bar{\mathbf{z}}', \bar{\mathbf{z}}_B)\| \leq 2B \wedge \|\bar{\mathbf{z}}_{s_1}\| \leq 2B_1 \wedge \|\bar{\mathbf{z}}_{s_2}\| \leq 2B_2$.

Proof. The proof is simply a combination of the proofs for Theorems 3.2.1 and 3.4.2. □

Algorithm 5 Π_{Sign}

Require: Message $M \in \{0,1\}^*$. Public information: $\mathbf{t}, \mathbf{t}', \mathbf{v}^T, \mathbf{B}, \delta = \lfloor \sqrt{q} \rceil, \xi_{-1}, \xi_5$. Private information: $\mathbf{r}, \mathbf{r}', i, \mathbf{s}', \mathbf{r}_B$

$\mathbf{y}, \mathbf{y}', \mathbf{y}_{-1}, \mathbf{y}_5 \leftarrow D^3_{\mathcal{R},\sigma}$
$\mathbf{y}_M \leftarrow D^8_{\mathcal{R},\sigma}$
$\mathbf{y}_{s_1} \leftarrow D^4_{\mathcal{R},\sigma_1}$
$\mathbf{y}_{s_2} \leftarrow D^2_{\mathcal{R},\sigma_2}$
$\mathbf{y}_s = (\mathbf{y}_{s_1}, \mathbf{y}_{s_2})$
$\mathbf{w}_1 := \mathbf{a}_1^T \mathbf{y}$
$\mathbf{w}'_1 := \mathbf{a}_1^T \mathbf{y}'$
$\mathbf{w}_{1,-1} := \mathbf{a}_1^T \mathbf{y}_{-1}$
$\mathbf{w}_{1,5} := \mathbf{a}_1^T \mathbf{y}_5$
$\mathbf{w}_2 := \delta \mathbf{a}_2^T \mathbf{y} - \mathbf{a}_2^T \mathbf{y}'$
$\mathbf{w}_{2,-1} := \mathbf{a}_2^T \mathbf{y} - \xi_{-1}(\mathbf{a}_2)\mathbf{y}_{-1}$
$\mathbf{w}_{2,5} := \mathbf{a}_2^T \mathbf{y} - \xi_5(\mathbf{a}_2)\mathbf{y}_5$
$\mathbf{w}_s := \mathbf{v}^T \mathbf{y}_s$
$\mathbf{w}_B := \mathbf{B}\mathbf{y}_B$
$c := H(\mathbf{t}, \mathbf{t}', \mathbf{v}, \mathbf{B}, \delta, \xi_{-1}, \xi_5, \mathbf{w}_1, \mathbf{w}'_1, \mathbf{w}_{1,-1}, \mathbf{w}_{1,5}, \mathbf{w}_2, \mathbf{w}_{2,-1}, \mathbf{w}_{2,5}, \mathbf{w}_s, \mathbf{w}_B, M)$
$\mathbf{z} := \mathbf{r}c + \mathbf{y}$
$\mathbf{z}' := \mathbf{r}'c + \mathbf{y}'$
$\mathbf{z}_{-1} := \xi_{-1}(\mathbf{r})c + \mathbf{y}_{-1}$
$\mathbf{z}_5 := \xi_5(\mathbf{r})c + \mathbf{y}_5$
$\mathbf{z}_{s_1} := \mathbf{s}'_1 c + \mathbf{y}_{s_1}$
$\mathbf{z}_{s_2} := \mathbf{s}'_2 c + \mathbf{y}_{s_2}$
$\mathbf{z}_M := \mathbf{r}_M c + \mathbf{y}_M$
if $\text{Rej}((\mathbf{z}, \mathbf{z}', \mathbf{z}_{-1}, \mathbf{z}_5, \mathbf{z}_M), (\mathbf{r}c, \mathbf{r}'c, \xi_{-1}(\mathbf{r})c, \xi_5(\mathbf{r})c, \mathbf{r}_M c), \sigma) = 1 \wedge \text{Rej}(\mathbf{z}_{s_1}, \mathbf{s}'_1 c, \sigma_1) = 1 \wedge \text{Rej}(\mathbf{z}_{s_2}, \mathbf{s}'_2 c, \sigma_2) = 1$ **then**
    Output $z = (\mathbf{z}, \mathbf{z}', \mathbf{z}_{-1}, \mathbf{z}_5, \mathbf{z}_{s_1}, \mathbf{z}_{s_2}, \mathbf{z}_M, c)$
else
    Restart

Algorithm 6 Verify

Require: Message $M \in \{0,1\}^*$. Signature $\Pi = (\mathbf{z}, \mathbf{z}', \mathbf{z}_{-1}, \mathbf{z}_5, \mathbf{z}_{s_1}, \mathbf{z}_{s_2}, \mathbf{z}_M)$.Public information: $\mathbf{t}, \mathbf{t}', \mathbf{v}^T, \mathbf{M}, \delta = \lfloor \sqrt{q} \rceil, \xi_{-1}, \xi_5$.

$\mathbf{w}_1 := \mathbf{a}_1^T\mathbf{z} - t_1 c$
$\mathbf{w}'_1 := \mathbf{a}_1^T\mathbf{z}' - t'_1 c$
$\mathbf{w}_{1,-1} := \xi_{-1}(\mathbf{a}_1^T)\mathbf{z}_{-1} - \xi_{-1}(t_1)c$
$\mathbf{w}_{1,5} := \xi_5(\mathbf{a}_1^T)\mathbf{z}_5 - \xi_5(t_1)c$
$\mathbf{w}_2 := \delta\mathbf{a}_2^T\mathbf{z} - \mathbf{a}_2^T\mathbf{z}' - (\delta t_2 - t'_2)\, c$
$\mathbf{w}_{2,-1} := \mathbf{a}_2^T\mathbf{z} - \xi_{-1}(\mathbf{a}_2^T)\mathbf{z}_{-1} - (t_2 - \xi_{-1}(t_2))\, c$
$\mathbf{w}_{2,5} := \mathbf{a}_2^T\mathbf{z} - \xi_5(\mathbf{a}_2^T)\mathbf{z}_5 - (t_2 - \xi_5(t_2))\, c$
$\mathbf{w}_s := \mathbf{v}^T\mathbf{z}_s - uc$
$\mathbf{w}_M := \mathbf{B}\mathbf{z}_B - (v, v_1, v_2, v_3, t_1)c \in \mathcal{R}_Q^4 \times \mathcal{R}_{q_1}$
if $\|(\mathbf{z}, \mathbf{z}', \mathbf{z}_{-1}, \mathbf{z}_5, \mathbf{z}_B)\| \leq B \wedge \|\mathbf{z}_{s_1}\| \leq B_1 \wedge \|\mathbf{z}_{s_2}\| \leq B_2$
and $c = H(\mathbf{t}, \mathbf{t}', \mathbf{v}, \mathbf{M}, \delta, \xi_{-1}, \xi_5, \mathbf{w}_1, \mathbf{w}'_1, \mathbf{w}_{1,-1}, \mathbf{w}_{1,5}, \mathbf{w}_2, \mathbf{w}_{2,-1}, \mathbf{w}_{2,5}, \mathbf{w}_s, \mathbf{w}_M)$ **then**
    Output 1
else
    Output 0

4.3 Fixing the Parameters

We will set the parameters as per Table 4.2. In this section we discuss the bounds that must be verified by these parameters and the resulting security guarantees. We will consider the security of our scheme in terms of "root-hermite factor" δ_0.

First we fix the dimension to $d = 4096$, we use this dimension as anything smaller does not allow the existence of parameters that make our scheme secure. For this dimension the challenge set $\{c \in \mathcal{R} \ : \ \|c\|_1 = \kappa, \|c\|_\infty = 1\}$ will be of size greater than 2^{256} if we fix $B_{\mathcal{C}} = \kappa = 26$.

The standard deviations s and r are fixed by the quality of our trapdoors to $s = 6\sqrt{dq_2}$ and $r = 1.17\sqrt{q_2}$.

We will need for the hiding property of our commitment that R-LWE is hard for dimension d, errors sampled in S_1 and both modulus q_1 and q_2. We do not use the reduction of Section 2.4.1 for commitments with multiple moduli as in practice the best attack will be to either solve R-LWE modulo q_1 or modulo q_2. Since we will have $q_2 > q_1$ and the hardness of R-LWE decreases as the modulus increases, we will only consider q_2 as being relevant here. The analysis of Section 5.6 shows that for the parameters in table 4.2 we have $\delta_0 = 1.0036$ for R-LWE with modulo q_2.

To fix q_1 we consider the requirements on the binding property of our commitment, from section 4.2 we have that the M-SIS$_{q_1,1,3,4B_{\mathcal{C}}B}$ problem has to be for vectors of norm $4B_{\mathcal{C}}B = 88 \cdot \kappa^2 \cdot \sqrt{200d}$, for $q_1 \approx 2^{30}$, as in table 4.2, we obtain $\delta_0 = 1.0036$.

To fix q_2 we will need the M-SIS$_{q_2,1,4,B_S}$ for B_S as per lemma 4.4.3 to be hard. We can compute that $B_S = \|\tilde{\mathbf{z}} - \mathbf{s}^*\| \approx 180224 \cdot \sqrt{2} \cdot d^2 \cdot \sqrt{q_2}$ for $q_2 \approx 2^{80}$ as in table 4.2 the root-hermite factor of this problem will be $\delta_0 = 1.0036$.

The only constraint we have on p the plaintext modulus of our verifiable encryption scheme is that if $\bar{\mathbf{r}}\tilde{c} = \tilde{\mathbf{r}}\bar{c} \mod p$ for some $\bar{\mathbf{r}}, \bar{c}, \tilde{\mathbf{r}}, \tilde{c}$ extracted in Π_{Sign} then this equation should hold over the integers, i.e. $\|\bar{\mathbf{z}}\tilde{c} - \tilde{\mathbf{z}}\bar{c}\|_\infty \leq p/2$. Since the vector $\mathbf{z}$ output in Π_{Sign} will have coefficients distributed according to D_σ, we will have with overwhelming probability that

$\|\mathbf{z}\|_\infty \leq 12\sigma$ (we can add this as an explicit check in the verification algorithm), in which case we will require $p \geq 4 \cdot \kappa \cdot 12\sigma \geq 2^{26.5}$.
The ciphertext modulus Q will be fixed by equation 4.9 which gives:

$$Q \geq 264\sqrt{34}\kappa p d^{3/2} \geq 2^{59.5}$$

All the parameters were set to achieve the same root Hermite factor of 1.0036 which corresponds to a block size of $\beta = 450$. Resulting in a security of 93 bits in space, 131 bits in time, and 119 bits in time for post-quantum security.
We consider the proof size that results from this parameter choice. The secret key will consist in 4 polynomials of standard deviation s and two polynomials of standard deviation r resulting in a size of $4d\log(12s) + 2d\log(12r) = 154KB$. The signature itself will consist of two commitments, one ciphertext and one zero-knowledge proof, which are respectively of size:

$$\begin{aligned} 2d\log q_1 + 2d\log q_2 &= 113KB \\ 4d\log Q &= 123KB \\ 13d\log(12\sigma) + 4d\log(12\sigma_1) + 2d\log(12\sigma_2) &= 345KB \end{aligned}$$

4.3.1 Accounting For Complexity Leveraging

The proof for full-traceability of Section 4.4 reduces the security of our group signature to that of a selectively secure signature by guessing the identity of the forgery, we thus lose a factor of q_2 in the reduction. If we want to account for this security loss when setting the parameters we will need to target a security of $128 + \log q_2$ bits for the M-SIS$_{q_2,1,3,B_S}$ problem. The dimension $d = 4096$ is no longer enough to reach such a security and we will thus be forced to set $d = 8192$. For this dimension the M-SIS$_{q_2,1,3,B_S}$ has a root hermite factor of $\delta_0 = 1.002$, corresponding to a block size $\beta = 1000$ and thus a a security of 207 bits in space, 292 bits in time, and 262 bits in time for post-quantum security. This is enough to tolerate a loss of $\log q_2 = 80$ bits of security. The R-LWE problem in dimension $d = 8192$ has a block size of more than $\beta = 1200$ resulting in more than 300 bits of security, similarly the M-SIS$_{q_1,1,3,4B_C B}$ becomes harder with a higher dimension and q_1 can be reduced accordingly. The rest of the parameters will be changed to the values given in Table 4.2, resulting in a signature size of (c.f. Table 4.1).

4.4 Security of the Scheme

Lemma 4.4.1 (Anonymity)**.** *Let $\mathcal{A}$ be a PPT adversary. Let $\mathsf{Adv}_{\mathcal{A}}^{Hid}(\lambda)$ be the advantage of $\mathcal{A}$ over the Hiding property of the commitment scheme. Let $\mathsf{Adv}_{\mathcal{A}}^{ind-cpa}(\lambda)$ be the advantage of $\mathcal{A}$ over the IND-CPA property of the encryption scheme. The advantage of $\mathcal{A}$ against the CPA-anonymity of our group signature is at most:*

$$\mathsf{Adv}_{\mathcal{A}}^{anon}(\lambda) \leq 2\mathsf{Adv}_{\mathcal{A}}^{Hid}(\lambda) + \mathsf{Adv}_{\mathcal{A}}^{ind-cpa}(\lambda) + 2^{-\lambda}$$

Proof. We use a succession of games.
Game $\mathbf{G}_0$: In this game the challenger runs **GSetup** honestly and gives $(gpk, sk_1, \ldots, sk_N)$ to $\mathcal{A}$. $\mathcal{A}$ outputs a message M^* and two identities $i_0, i_1 \in [N]$. The challenger chooses a bit

$b \xleftarrow{\$} \{0,1\}$ and computes $z^* := (\mathbf{t}, \mathbf{t}', e, \pi) \leftarrow \mathbf{Sign}(M^*, sk_{i_b})$
Game $\mathbf{G}_1$: In this the challenger uses the simulator of the proof Π_{Sign} when queried for $\mathbf{Sign}(M^*, sk_{i_b})$. This game is statistically indistinguishable from the previous by the zero-knowledge of Π_{Sign}.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_1} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_0}\right| \leq 2^{-\lambda}$$

Game $\mathbf{G}_2$: In this the challenger replaces the commitment $\mathbf{t}$ by a commitment of 0 when answering the query $\mathbf{Sign}(M^*, sk_{i_b})$. The proof Π_{Sign} can still be used since it uses the simulator, and this game is indistinguishable from the previous one by the hiding property of the commitment.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_2} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_1}\right| \leq \mathsf{Adv}_{\mathcal{A}}^{Hid}(\lambda)$$

Game $\mathbf{G}_3$: In this the challenger replaces the commitment $\mathbf{t}'$ by a commitment of 0 when answering the query $\mathbf{Sign}(M^*, sk_{i_b})$. This game is indistinguishable from the previous one by the hiding property of the commitment.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_3} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_2}\right| \leq \mathsf{Adv}_{\mathcal{A}}^{Hid}(\lambda)$$

Game $\mathbf{G}_4$: In this the challenger replaces the commitment ciphertext e with an encryption of 0. Since the proof Π_{Sign} uses the simulator it is independent of the decryption of e. This game is indistinguishable from the previous one by the IND-CPA property of the encryption scheme.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_4} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_3}\right| \leq \mathsf{Adv}_{\mathcal{A}}^{ind-cpa}(\lambda)$$

The signature $(\mathbf{t}, \mathbf{t}', e, \pi)$ output in **Game $\mathbf{G}_3$** is independent of i_b and the adversary has thus probability $1/2$ of outputting $b' = b$. We obtain the desired result by summing the advantages. □

We will prove traceability in two steps. We will first prove that an adversary $\mathcal{A}$ cannot distinguish between the regular traceability game and the traceability game in which the setup algorithm has been replaced by $\mathbf{GSetup}^*$ which we define below. We will then prove that a challenger $\mathcal{B}$ can extract an M-SIS solution from an adversary who succeeds in producing a forgery in the traceability game with $\mathbf{GSetup}^*$.
$\mathbf{GSetup}^*(1^\lambda)$:

- Sample $i^* \xleftarrow{\$} \mathbb{Z}_{q_2}$
- Sample $\mathbf{A} := \begin{bmatrix}\mathbf{a}_1^T \\ \mathbf{a}_2^T\end{bmatrix} \leftarrow \mathbf{CSetup}(1^\lambda)$, with $\mathbf{a}_1 \in \mathcal{R}_{q_1}^3$, and $\mathbf{a}_2 \in \mathcal{R}_{q_2}^3$.
- Sample $\mathbf{a} \xleftarrow{\$} \mathcal{R}_{q_2}^2$.
- Sample $\mathbf{R} \xleftarrow{\$} S_1^{2\times 2}$ and set $\mathbf{b}^T = \mathbf{a}^T\mathbf{R} \in \mathcal{R}_{q_2}^{1\times 2}$.
- Sample $(\mathbf{s}_{i_1^*}, \mathbf{s}_{i_2^*}, \mathbf{s}_{i_3^*}) \leftarrow D_{R,s} \times D_{R,s} \times D_{R,r}$
- Set $u := \begin{bmatrix}\mathbf{a}^T & | & \mathbf{b}^T & | & \mathbf{a}_2^T\end{bmatrix} \begin{bmatrix}\mathbf{s}_{i_1^*} \\ \mathbf{s}_{i_2^*} \\ \mathbf{s}_{i_3^*}\end{bmatrix}$

- Set $gpk := \left(\mathbf{A}, \mathbf{a}, \mathbf{b} - i^*\mathbf{g}^T, u\right)$
- For $i \in \mathbb{Z}_{q_2} \setminus \{i^*\}$, sample $\mathbf{s}_{i_3} \leftarrow D^3_{\mathcal{R},r}$
- For $i \in \mathbb{Z}_{q_2} \setminus \{i^*\}$, sample $(\mathbf{s}_{i_1}, \mathbf{s}_{i_2}) \in \mathcal{R}^4$ s.t $\begin{bmatrix}\mathbf{a}^T & | & \mathbf{b}^T + (i-i^*)\mathbf{g}^T\end{bmatrix}\begin{bmatrix}\mathbf{s}_{i_1}\\ \mathbf{s}_{i_2}\end{bmatrix} = u - \mathbf{a}_2^T\mathbf{s}_{i_3}$, and $(\mathbf{s}_{i_1}, \mathbf{s}_{i_2}) \leftarrow D^4_{\mathcal{R},s}$
- For $i \in \mathbb{Z}_{q_2}$, set $sk_i := \mathbf{s}_i := (\mathbf{s}_{i_1}, \mathbf{s}_{i_2}, \mathbf{s}_{i_3})$

We consider the following advantages for an adversary $\mathcal{A}$

- $\mathsf{Adv}^{trace}_{\mathcal{A}}(\lambda)$ the advantage of $\mathcal{A}$ in the traceability game.
- $\mathsf{Adv}^{trace*}_{\mathcal{A}}(\lambda)$ the advantage of $\mathcal{A}$ in the traceability game where **GSetup** is replaced with **GSetup***.
- $\mathsf{Adv}^{NTRU}_{\mathcal{A}}(\lambda)$ the advantage of $\mathcal{A}$ in solving the NTRU$_{q,r}$ problem.
- $\mathsf{Adv}^{MLWE}_{\mathcal{A}}(\lambda)$ the advantage of $\mathcal{A}$ in solving the M-LWE$_{q,1,s}$ problem.

Lemma 4.4.2. *The advantage of any PPT adversary $\mathcal{A}$ against the traceability game of the group signature is at most:*

$$\mathsf{Adv}^{trace}_{\mathcal{A}}(\lambda) \leq 2(\mathsf{Adv}^{NTRU}_{\mathcal{A}}(\lambda) + \mathsf{Adv}^{MLWE}_{\mathcal{A}}(\lambda)) + \mathsf{Adv}^{trace*}_{\mathcal{A}}(\lambda)$$

Proof. We use a succession of games.
Game $\mathbf{G}_0$: The challenger $\mathcal{B}$ runs the Group signature protocol honestly. He gives $(sk_i)_{i\in S}$ as well as $gmsk$ to $\mathcal{A}$ who has advantage ε in the traceability game.

$$\mathsf{Adv}^{\mathbf{G}_0}_{\mathcal{A}} = \mathsf{Adv}^{trace}_{\mathcal{A}}$$

Game $\mathbf{G}_1$: $\mathcal{B}$ samples $\mathbf{a}_2^T$ as $[0 \mid 1 \mid f/g]$ where $f, g \in\leftarrow D_{\mathcal{R},r}$ are taken as in Section 2.3.5. $\mathbf{G}_2$ is indistinguishable from $\mathbf{G}_1$ under the NTRU$_{q,r}$ assumption.

$$\left|\mathsf{Adv}^{\mathbf{G}_1}_{\mathcal{A}} - \mathsf{Adv}^{\mathbf{G}_0}_{\mathcal{A}}\right| \leq \mathsf{Adv}^{NTRU}_{\mathcal{A}}$$

Game $\mathbf{G}_2$: $\mathcal{B}$ sets $\mathbf{b}^T \xleftarrow{\$} \mathcal{R}^{1\times 2}_{q_2}$. Note that if $\mathbf{b}^T \neq \mathbf{a}^T\mathbf{R}$, $\mathcal{B}$ can no longer use the GPV trapdoor of $\begin{bmatrix}\mathbf{a}^T & | & \mathbf{b}^T + i\mathbf{g}^T\end{bmatrix}$ to sample secret keys for user i. To generate keys for i he will instead sample $\mathbf{s}_{i_1}, \mathbf{s}_{i_2} \leftarrow D^2_s$ and use his NTRU trapdoor on $\mathbf{a}_2$ to sample $\mathbf{s}_{i_3}$. This game will be indistinguishable from the previous one by the hardness of M-LWE$_{q,1,s}$ (since $\mathbf{a}^T\mathbf{R}$ is two M-LWE$_{q,1,s}$ samples).

$$\left|\mathsf{Adv}^{\mathbf{G}_2}_{\mathcal{A}} - \mathsf{Adv}^{\mathbf{G}_1}_{\mathcal{A}}\right| \leq \mathsf{Adv}^{MLWE}_{\mathcal{A}}$$

Game $\mathbf{G}_3$: $\mathcal{B}$ replaces $\mathbf{b}^T$ with $\mathbf{b}^{*T} := \mathbf{b}^T - i^*\mathbf{g}^T$. Since $\mathbf{b}^T$ is uniform this game is identical to the previous one.

$$\mathsf{Adv}^{\mathbf{G}_3}_{\mathcal{A}} = \mathsf{Adv}^{\mathbf{G}_2}_{\mathcal{A}}$$

Game $\mathbf{G}_4$: $\mathcal{B}$ sets $\mathbf{b}^{*T} := \mathbf{a}^T\mathbf{R} - i^*\mathbf{g}^T$. This game is indistinguishable from the previous one under M-LWE$_{q,1,s}$.

$$\left|\mathsf{Adv}^{\mathbf{G}_4}_{\mathcal{A}} - \mathsf{Adv}^{\mathbf{G}_3}_{\mathcal{A}}\right| \leq \mathsf{Adv}^{MLWE}_{\mathcal{A}}$$

Game $\mathbf{G}_5$: $\mathcal{B}$ sets $\mathbf{a}_2^T$ as $[0 \mid 1 \mid a_2]$, with $a_2 \xleftarrow{\$} \mathcal{R}_{q_2}$ and uses the GPV trapdoor of $\left[\mathbf{a}^T \mid \mathbf{b}^T + (i - i^*)\mathbf{g}^T\right]$ to sample secret keys for user i. This game is indistinguishable from the previous one under the $\text{NTRU}_{q,r}$ assumption.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_5} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_4}\right| \leq \mathsf{Adv}_{\mathcal{A}}^{NTRU}$$

Note that **Game $\mathbf{G}_5$** is exactly the traceability game that uses **GSetup*** (simply by renaming $\mathbf{s}_0$ to $\mathbf{s}_{i^*}$), the result follows. □

Lemma 4.4.3. *Let $\mathcal{A}$ be a PPT algorithm with advantage ε in the traceability game with* **GSetup****. Let h be a bound on the number of hash queries made by $\mathcal{A}$. Let $B_S \geq 4B_{\mathcal{C}}B_1 + 12B_{\mathcal{C}}\sqrt{d}B_1 + 2B_{\mathcal{C}}B_2 + 6\sqrt{d}\sigma B_1 + B_{\mathcal{C}}^2(1+3\sqrt{d})2\sqrt{d}s + B_{\mathcal{C}}^2\sqrt{6d}s$. There exists $\mathcal{B}$ a challenger for the* $\text{M-SIS}_{q,1,4,B_S}$ *such that:*

$$\mathsf{Adv}_{\mathcal{B}}^{MSIS}(\lambda) \geq \frac{\varepsilon}{q_2}\left(\frac{\varepsilon}{h} - 2^{-\lambda}\right)$$

Proof. Formally $\mathcal{B}$ is given a matrix $\mathbf{x}^T := \left[x_1, x_2, x_3, x_4\right] \in \mathcal{R}_{q_2}^4$ and must output $\mathbf{y}$ s.t $\mathbf{x}^T\mathbf{y} = 0 \mod q_2$ and $\|\mathbf{y}\| \leq B_S$, w.l.o.g we consider $\mathbf{x} = \left[x_1, x_2, x_3, 1\right]$ instead since with high probability one of the x_i will have an inverse.
$\mathcal{B}$ will set $\mathbf{a} := (x_1, x_2)$ and $\mathbf{a}_2^T := (0, 1, x_3)$ during setup, since x_1, x_2, x_3 are uniform in $\mathcal{R}_{q_2}$ this does not change the distribution of **GSetup***. When asked signing queries, $\mathcal{B}$ runs the signing algorithm honestly, when asked corrupt queries $\mathcal{B}$ outputs the corresponding secret key. Suppose the adversary $\mathcal{A}$ outputs a forgery $z := (\mathbf{t}, \mathbf{t}', \Pi, (u, \mathbf{v}))$ by programming the random oracle with two different challenges $\mathcal{B}$ will be able to extract $\bar{\mathbf{z}} \in \mathcal{R}^3$, $\bar{id} \in \mathbb{Z}_{q_2}$, $\bar{\mathbf{z}}' \in \mathcal{R}^3$, $\bar{\mathbf{z}}_s \in \mathcal{R}^7$, $\bar{\mathbf{z}}_B \in \mathcal{R}^8$, $\bar{c} \in \bar{\mathcal{C}}$ such that:

$$\begin{aligned}
&\bar{c}\mathbf{t} = Com(\bar{c}\bar{id}; \bar{\mathbf{z}}) \\
&\bar{c}\mathbf{t}' = Com(\bar{c}\bar{id}\delta; \bar{\mathbf{z}}') \\
&\bar{c}\begin{bmatrix} u \\ \mathbf{v} \\ t_1 \end{bmatrix} = \mathbf{B}\bar{\mathbf{z}}_B \\
&\bar{c}u = \mathbf{v}^T\bar{\mathbf{z}}_s
\end{aligned}$$

such that $\|(\bar{\mathbf{z}}, \bar{\mathbf{z}}', \bar{\mathbf{z}}_B)\| \leq 2B \wedge \|\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2\| \leq 2B_1 \wedge \|\bar{\mathbf{z}}_3\| \leq 2B_2$, with $(\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, \bar{\mathbf{z}}_3) := \bar{\mathbf{z}}$. Using the forking lemma of [BN06], $\mathcal{B}$ will be able to do this with probability at least $\varepsilon\left(\frac{\varepsilon}{h} - 2^{-\lambda}\right)$.
Let $(\tilde{\mathbf{r}}, \tilde{c}) :=\text{Dec}(u, v)$, the parameters set in section 4.3 are such that, by soundness of the verifiable encryption scheme, with overwhelming probability $\tilde{\mathbf{r}}\bar{c} = \bar{\mathbf{z}}\tilde{c}$ over the integers, which implies that $\mathbf{Open}(z) \in \mathbb{Z}_{q_2}$ i.e. the forgery opens to an identity in $\mathbb{Z}_{q_2}$ and not $\perp$.
Since i^* is taken uniformly at random in **GSetup***, z will open to this identity with probability $1/q_2$. Suppose that z opens to i^*. Then

$$\begin{aligned}
&\bar{c}t_2 = \mathbf{a}_2^T\bar{\mathbf{z}} + \bar{c}i^* \\
&\bar{c}t_2' = \mathbf{a}_2^T\bar{\mathbf{z}}' + \bar{c}i^*\delta \\
&\left[\mathbf{a}^T \mid \mathbf{b}^T + [t_2 \mid t_2'] - i^*\mathbf{g}^T \mid \mathbf{a}_2^T\right]\bar{\mathbf{z}}_s = \bar{c}u
\end{aligned}$$

If we multiply the third equation by $\bar{c}$ and replace $\bar{c}[t_2 \mid t'_2]$ we get:

$$\left[\bar{c}\mathbf{a}^T \quad | \quad \bar{c}\mathbf{b}^T + [\mathbf{a}_2^T\bar{\mathbf{z}} \mid \mathbf{a}_2^T\bar{\mathbf{z}}'] \quad | \quad \bar{c}\bar{\mathbf{a}}_2^T\right]\bar{\mathbf{z}}_s = \bar{c}^2 u$$

Let

$$\tilde{\mathbf{z}} = \begin{bmatrix} \bar{c}\bar{\mathbf{z}}_1 + \mathbf{R}\bar{c}\bar{\mathbf{z}}_2 \\ \bar{c}\bar{\mathbf{z}}_3 - \left[\bar{\mathbf{z}} \quad \bar{\mathbf{z}}'\right]\bar{\mathbf{z}}_2 \end{bmatrix}$$

Then

$$\left[\mathbf{a}^T \quad | \quad \bar{\mathbf{a}}_2^T\right]\tilde{\mathbf{z}} = \bar{c}^2 u$$

Since $\mathcal{A}$ has to output a valid forgery this means that he has never obtained the key sk_{i^*}, we can thus consider that $\mathbf{s}_{i^*}$ was sampled after receiving the forgery, conditioned on $\left[\mathbf{a}^T \mid \mathbf{b} \mid \mathbf{a}_2^T\right]\mathbf{s}_{i^*} = u$. Let $\mathbf{s}^* := \left[\mathbf{s}_{i_1^*} + \mathbf{R}\mathbf{s}_{i_2^*} \mid \mathbf{s}_{i_3^*}\right]$, the probability that $\bar{c}\mathbf{s}^* = \tilde{\mathbf{z}}$ is negligible. Finally we have a solution $\tilde{\mathbf{z}} - \bar{c}\mathbf{s}^*$ to the M-SIS problem defined by $\left[\mathbf{a}^T \quad | \quad \mathbf{a}_2^T\right]$. Using the bounds on the extracted values and the distribution of $\mathbf{s}^*$ we have the following bound on the norm of the solution:

$$\begin{aligned}
\|\tilde{\mathbf{z}} - \bar{c}\mathbf{s}^*\| &\leq \|\tilde{\mathbf{z}}\| + 2B_{\mathcal{C}}\|\mathbf{s}^*\| \\
&\leq 2B_{\mathcal{C}}\|\mathbf{z}_1\| + 6B_{\mathcal{C}}\sqrt{d}\,\|\bar{\mathbf{z}}_2\| + B_{\mathcal{C}}\|\bar{\mathbf{z}}_3\| + 3\sqrt{d}\sigma\,\|\bar{\mathbf{z}}_2\| \\
&\quad + B_{\mathcal{C}}^2(1+3\sqrt{d})(2\sqrt{d}s) + B_{\mathcal{C}}^2\sqrt{6d}s \\
&\leq 4B_{\mathcal{C}}B_1 + 12B_{\mathcal{C}}\sqrt{d}B_1 + 2B_{\mathcal{C}}B_2 + 6\sqrt{d}\sigma B_1 + B_{\mathcal{C}}^2(1+3\sqrt{d})2\sqrt{d}s + B_{\mathcal{C}}^2\sqrt{6d}s
\end{aligned}$$

The largest terms in this solution are by far $2B_{\mathcal{C}}B_2$ and $6\sqrt{d}\sigma B_1$ which we will consider when setting the parameters in section 4.3. □

Chapter 5

E-Voting

In this chapter we present an e-voting scheme for multiple candidates with multiple tallying authorities.
We define a security model, adapted from the one of [BCG+15], in which privacy is guaranteed as long as one of the tallying authorities is honest and unforgeability is guaranteed even if they are all corrupt.
After recalling the building blocks we use from Chapters 2 and 3 we present a first variant of our voting scheme in which the parameters grow with the number of voters.
We then show how to modify this scheme so that the parameters remain independent of the number of voters at virtually no cost.
Finally we give concrete parameters as well as ballot sizes and a formal proof of security.

Contents

5.1 Introduction

Given how information technology has penetrated almost every aspect of our world, one could be surprised at the primitive state of technology used in the process that most influences society: elections. Electronic voting machines, which required voters to physically turn up at polling stations to cast their ballots on dedicated machines, saw a brief rise in popularity until the early 2000s, but many countries have since then gone back to paper-and-pencil voting amidst worries about security and reliability.

These are definitely genuine concerns. Researchers discovered serious security flaws in several models of voting machines that were used in elections in the US, the Netherlands, and Germany [FHF07; GH07]. Recent news reports on massive security breaches and suspicions of foreign meddling in national elections have only aggravated those concerns.

Nevertheless, many countries are warming up to the idea of online voting, in which voters cast their ballots using their personal devices from the comfort of their couch. A handful of countries, including Estonia, Switzerland, and Australia, are already using online voting for local and national elections, and it is quite a common tool among private organizations to elect officers and board members.

There are of course many aspects to securing an online voting system, but the underlying cryptographic protocol is obviously an important ingredient. All of the currently deployed electronic voting systems (e.g., Helios [Adi08], the Swiss voting system [RD17], and the Estonian one) are based on cryptographic primitives that rely on the hardness of factoring or discrete logarithms for their security. Both of these assumptions are well-known to succumb to attacks by quantum computers, meaning that, as soon as sufficiently powerful quantum computers become available, an adversary could use them to break the vote secrecy of a past election, or to tamper with the result of an ongoing one. The threat of foreign meddling in elections gives additional reason for concern: powerful nation states may very well be the first to build quantum computers, and they may not be particularly vocal about their achievement.

Fortunately, we do have some cryptographic problems that resist attacks by quantum computers. Lattices are the most prominent one, offering a good trade-off between efficiency and security for basic primitives such as signatures and encryption. For more advanced protocols, such as those required for electronic voting, lattices tend to, however, suffer from extremely high bandwidth requirements.

The main difficulty in constructing practical lattice-based privacy schemes is the lack of efficient zero-knowledge proofs, which is an important tool in electronic voting schemes to let voters prove that they cast a valid ballot. Most lattice-based zero-knowledge proofs are either Fiat-Shamir proofs with single-bit challenges or Stern-type proofs [Ste94] with soundness error 2/3, which have to be repeated many times to reduce the soundness error. Amortization techniques [BDLN16; CDXY17; PL17] exist when performing thousands of proofs in parallel, but these are not very useful when each voter must prove correctness of his own vote. Lyubashevsky's "Fiat-Shamir with Aborts" technique [Lyu12] yields much more efficient proofs with large challenges, but only allows to prove correctness of the statement up to a small multiple of the witnesses, which could be quite detrimental in the context of voting, as it may allow an attacker to inflate the weight of this vote.

The only quantum-safe voting protocol that we are aware of [CGGI16] therefore shuns zero-knowledge proofs completely and uses fully-homomorphic encryption [Gen09] instead. The paper doesn't give any implementation details or concrete parameter choices, so it's

	Voter	Auth/Voter	Total Size / Voter
Time	10ms	3ms	
Size	28KB	2KB	30KB

Table 5.1: Time and space complexity of the voting scheme with 4 authorities. Using the parameters of Section 5.5, each voter ouputs one OR-Proof and four commitments (one per authority), while each authority outputs one proof per voter.

hard to make statements about efficiency, but due to the "heavy machinery" being utilized, chances are that the protocol is not efficient enough for medium to large-scale elections.

5.1.1 Our Contributions

We present a new lattice-based electronic voting scheme that *does* use zero-knowledge proofs, but overcomes their inefficiencies by re-organizing the proofs so that the voting authorities assist the voters by performing amortized proofs. Our protocol provably guarantees vote privacy as long as one of a number of voting authorities is honest, and guarantees consistency (i.e., that honest votes are correctly counted) even if all voting authorities are corrupt, all under standard lattice-based assumptions in the random-oracle model. We suggest concrete choices for the security parameters and implement a prototype of our protocol. Our experimental results (Table 5.1) show that voters need less than 10ms to cast a vote and a complete bulletin (including all commitments and votes) is of size $30KB$, which we think is well within practical bounds for a large-scale election. To better understand the technical hurdles to obtain this result, we briefly sketch a voting protocol by Cramer et al. [CFSY96] on which our protocol is based. Let's say there are N_V voters and N_A voting authorities that assist in a binary election, i.e., where each voter votes zero or one and the result is the sum of the votes. Let's also say that there is a public bulletin board where voters can post their ballots. The authorities jointly compute the tally and post the result of the election, together with a proof of correctness. The goal is to obtain vote privacy, meaning that as long as one authority is honest, the adversary does not learn anything more about the votes of honest voters than what is already implied by the result, as well as consistency and universal verifiability, meaning that anyone can check that all honest votes were counted correctly, even if all authorities collude to rig the election.

The protocol of Cramer et al. [CFSY96] begins by letting each voter secret-share his vote among the N_A authorities and commit to each of the shares. The voter sends the share and the opening information to each authority, and performs an OR-proof [CDS94] to show that he secret-shared a zero-or-one vote by exploiting a homomorphism in the commitment scheme. When the voting phase closes, all servers check the openings of the commitments they received. Each authority then publishes the sum of all the shares it received together with valid opening information, again using the homomorphism in the commitments. The result of the election is the sum of all these partial sums.

There are a number of hurdles to overcome when translating this approach into lattice-based primitives. The first is that, as discussed above, lattice-based zero-knowledge proofs are either inefficient or approximate, while amortization doesn't help for proofs by individual voters. The second is that commitments typically use short vectors as randomness (i.e., opening information), but applying homomorphisms accumulates the size of this randomness,

which must be compensated for by choosing larger parameters, which comes at a big cost in efficiency. The third problem is that the typical OR-proof technique [CDS94] of XOR-ing challenge values doesn't work for lattices, because challenges are polynomials with small coefficients in a ring, but do not form a group among them.

We address the first problem by strategically splitting up the burden of the proofs between voters and authorities. Namely, we let voters prove that they secret-shared a zero-or-one vote using approximate proofs, but we let the authorities prove that the commitment they received is well-formed, i.e., has short opening information. The authorities do so for all voters simultaneously, so they can use the more efficient amortized proofs which we designed in Section 3.5.

The second problem we address by letting authorities re-commit to the sum of batches of votes, and by letting them prove in zero knowledge that the new commitment indeed contains the sum of all votes in the batch. By repetitively applying this technique, each authority can keep the randomness growth within bounds, so that it eventually ends up with a commitment to the sum of all received shares with short randomness.

Finally, we solve the problem with the OR proofs by interpreting the challenge space of our zero-knowledge proofs as a group, using the techniques described in Section 3.3.

The proofs outlined above are constructed using quantum-secure building blocks via the Fiat-Shamir transform. While there is a known *classical* reduction from hard lattice problems to schemes constructed in this manner, there is no quantum reduction known. The underlying reason as to why a general proof is unlikely to come is due to the fact that classically-secure *computationally binding* commitments are not known to be binding for a quantum committer (c.f. [DFS04]). Nevertheless, there are known quantum security proofs in the QROM for Fiat-Shamir schemes of the same form as ours, but in which the parameters are set differently [Unr17]. Furthermore, there are currently no known natural counter-examples of Fiat-Shamir zero-knowledge proofs (nor of commitment schemes) which are based on quantum-hard problems via classical reductions, but are broken by quantum adversaries. It therefore seems reasonable to assume that such Fiat-Shamir schemes are secure. If one would like to have a reduction that is in the QROM, one could instantiate the schemes as in [AFLT12] and then use the reduction in [Unr17]. This would, however, lead to a noticeable increase in the size of the proofs and public keys.

5.1.2 Overview of the Construction

We will make the convention that the voters (and information pertaining to the voters) are numbered 1 through N_V using a subscript, whereas the information pertaining to the authorities is numbered 1 through N_A and is labeled using a parenthesized superscript. In particular, for elements $x_i^{(j)}$, we will define $x_i = \sum_{j=1}^{N_A} x_i^{(j)}$, $x^{(j)} = \sum_{i=1}^{N_V} x_i^{(j)}$, and $x = \sum_{i=1}^{N_V}\sum_{j=1}^{N_A} x_i^{(j)} = \sum_{j=1}^{N_A}\sum_{i=1}^{N_V} x_i^{(j)}$.

We consider an election for N_C candidates in which votes will be vectors in $\{0,1\}^{N_C}$ and the tally will be the sum of all votes. Since voters will commit to values in $\mathcal{R}_q$, we assume there exists a ring $\mathcal{R}'$ and an isomorphism $\phi : \mathcal{R}_q \to \mathcal{R}'^{N_C}$, and we define the set of valid votes as $\left\{v \in \mathcal{R}_q \ : \ \phi(v) \in \{0,1\}^{N_C}\right\}$.

A voter i who wishes to cast a vote v_i (which is such that $\phi(v_i) \in \{0,1\}^{N_C}$), first splits v_i

into N_A parts $v_i^{(j)}$ where the first $N_A - 1$ of them are chosen uniformly in $\mathcal{R}_q$ and the last share $v_i^{(N_A)}$ is chosen such that $\sum_{j=1}^{N_A} v_i^{(j)} = v_i \mod q$. The voter i then uses the commitment scheme to commit to each share $v_i^{(j)}$ as

$$\mathbf{t}_i^{(j)} := \mathbf{Com}(v_i^{(j)}; \mathbf{r}_i^{(j)}). \tag{5.1}$$

All the commitments are published to the bulletin board. Note that because the commitment scheme is additively homomorphic, we have

$$\sum_{j=1}^{N_A} \mathbf{t}_i^{(j)} = \mathbf{t}_i = \mathbf{Com}(v_i; \mathbf{r}_i),$$

which is a valid commitment to v_i (but with slightly larger randomness $\mathbf{r}_i$). Voter i now creates a zero-knowledge OR-proof that he has knowledge of a vector $\bar{\mathbf{r}}_i$ with small coefficients and a ring element $\bar{c}_i$ with very small coefficients such that

$$\bar{c}_i \mathbf{t}_i = \mathbf{Com}(v_i \bar{c}_i, \bar{\mathbf{r}}_i), \text{ and } \phi(v_i) \in \{0,1\}^{N_C}. \tag{5.2}$$

This proof π_i^V also gets posted to the bulletin board.

Each voter now sends to authority j the encryption (under authority j's public key) of the share $v_i^{(j)}$ and the randomness under which this share was committed $\mathbf{r}_i^{(j)}$ from Equation (5.1) (one can alternatively think that the voters simply post this encryption to the bulletin board). Upon receiving all such encryptions from every voter, authority j needs to create a proof of knowledge that the $\mathbf{r}_i^{(j)}$ all have small coefficients. He uses the Amortized Exact Zero-Knowledge proof to create proofs $\pi_{i,j}^A$ that prove the knowledge of $\hat{\mathbf{r}}_i^{(j)}$ and $\hat{v}_i^{(j)}$ that satisfy

$$\mathbf{t}_i^{(j)} = \mathbf{Com}(\hat{v}_i^{(j)}; \hat{\mathbf{r}}_i^{(j)}). \tag{5.3}$$

If all N_A authorities provide proofs of the above statement, then using the additive homomorphism of the commitment scheme, we obtain a proof of knowledge of an $\hat{\mathbf{r}}_i$ such that

$$\mathbf{t}_i = \sum_{j=1}^{N_A} \mathbf{t}_i^{(j)} = \sum_{j=1}^{N_A} \mathbf{Com}(\hat{v}_i^{(j)}, \hat{\mathbf{r}}_i^{(j)}) = \mathbf{Com}(\hat{v}_i, \hat{\mathbf{r}}_i). \tag{5.4}$$

Combining this with Equation (5.2) gives two valid openings of the commitment $\mathbf{t}_i$. Based on the binding property of **Com**, this implies that $\bar{\mathbf{r}}_i = \bar{c}_i \hat{\mathbf{r}}_i$. One can then rewrite Equation (5.2) as

$$\bar{c}_i \mathbf{t}_i = \mathbf{Com}(\bar{c}_i v_i, \bar{c}_i \hat{\mathbf{r}}_i),$$

and since we choose the challenge set such that $\bar{c}_i$ is invertible in $\mathcal{R}_q$, we can divide by $\bar{c}_i$ to finally obtain

$$\mathbf{t}_i = \mathbf{Com}(v_i, \hat{\mathbf{r}}_i), , \text{ and } \phi(v_i) \in \{0,1\}^{N_C}. \tag{5.5}$$

Because there is no longer the factor $\bar{c}_i$ which could be distinct for every voter, the commitment in Equation (5.5) is additively homomorphic. In particular, if we compute

$$\sum_{i=1}^{N_V} \mathbf{t}_i = \mathbf{Com}(\sum_{i=1}^{N_V} v_i; \hat{\mathbf{r}}), \text{ and } \phi(v_i) \in \{0,1\}^{N_C}, \tag{5.6}$$

and $\hat{\mathbf{r}}$ is a vector with small coefficients, then the quantity

$$\sum_{i=1}^{N_V} \mathbf{t}_i$$

is a commitment to the sum of the votes that have been cast. If there are many voters, then $\hat{\mathbf{r}} = \sum_i \bar{\mathbf{r}}_i$ is not small, but we show how to handle this issue later.

For universal verifiability, we therefore would like the value of $\hat{\mathbf{r}}$ to be publicly computable. For this to happen, each authority simply computes $\sum_{i=1}^{N_V} \mathbf{r}_i^{(j)} = \mathbf{r}^{(j)}$ and reveals it by publishing it to the bulletin board. Any verifier can simply check that $\mathbf{r}^{(j)}$ is a valid opening of $\mathbf{t}^{(j)}$, i.e. that there exists some message $m^{(j)}$ such that

$$\mathbf{t}^{(j)} = \mathbf{Com}(m^{(j)}, \mathbf{r}^{(j)}). \tag{5.7}$$

We now claim that it must be that

$$\mathbf{r} = \sum_{j=1}^{N_A} \mathbf{r}^{(j)} = \hat{\mathbf{r}}.$$

From Equation (5.3), we know that

$$\mathbf{t}^{(j)} = \sum_{i=1}^{N_V} \mathbf{Com}(\hat{v}_i^{(j)}; \hat{\mathbf{r}}_i^{(j)}) = \mathbf{Com}(\hat{v}^{(j)}; \hat{\mathbf{r}}^{(j)})$$

Combining this with Equation (5.7) gives two openings of $\mathbf{t}^{(j)}$. Which implies that $\hat{\mathbf{r}}^{(j)} = \mathbf{r}^{(j)}$, and then we have

$$\hat{\mathbf{r}} = \sum_{j=1}^{N_A} \hat{\mathbf{r}}^{(j)} = \sum_{j=1}^{N_A} \mathbf{r}^{(j)} = \mathbf{r}.$$

Plugging the above into Equation (5.6) implies that

$$\mathbf{t} = \sum_{i=1}^{N_V} \mathbf{t}_i = \mathbf{Com}(\sum_{i=1}^{N_V} v_i; \mathbf{r}), \text{ and } \phi(v_i) \in \{0,1\}^{N_C}. \tag{5.8}$$

If $\mathbf{r}$ is small enough, then the above implies that $\mathbf{t}$ is a commitment to the full vote tally $\sum_{i=1}^{N_V} v_i$, and one can obtain this tally by computing $\mathbf{Open}(\mathbf{t}, \mathbf{r}, 1)$. As long as there are fewer than q voters, we can exactly recover $\phi(\sum_{i=1}^{N_V} v_i)$ over the integers.

5.1.2.1 Reducing the Randomness

An issue that we still need to deal with is how to make sure that the randomness, when summed over all the voters, does not grow too much. This is crucial in order for the final commitment in Equation (5.6) to be meaningful. A trivial way to accomplish this is to simply set the parameters large enough so that a large set of voters can be accommodated. This is an extremely impractical solution that we would like to avoid.

The way that we can overcome this issue is by making the Authorities create votes of partial sums of the vote shares they have, using randomness that is close to the randomnesses used by the individual voters. For example, if voters $1, \ldots, l$ whose commitments to Authority j are $\mathbf{t}_1^{(j)}, \ldots, \mathbf{t}_l^{(j)}$ of values $v_1^{(j)}, \ldots, v_l^{(j)}$, under randomnesses $\mathbf{r}_1^{(j)}, \ldots, \mathbf{r}_l^{(j)}$, then the Authority can create a commitment $\mathbf{t}$ of $\sum_{i=1}^{l} v_i^{(j)}$ using a fresh randomness $\mathbf{r}$. The Authority would then publish the "vote" $\mathbf{t}$, proves that there exists some $\hat{\mathbf{r}}', m'$ such that

$$\mathbf{t} = \mathbf{Com}(m', \mathbf{r}'), \tag{5.9}$$

and also prove that there exists a slightly larger $\hat{\mathbf{r}}$ such that

$$\mathbf{t} + \sum_i \mathbf{t}_i^{(j)} = \mathbf{Com}(0; \hat{\mathbf{r}}) \tag{5.10}$$

The proof in Equation (5.9) can be "amortized-in" with the proofs $\pi_{i,j}^A$ because the size of the randomness is the same. The proof of Equation (5.10), however, contains larger randomness, and so such proofs should be amortized only among themselves.[1]

Notice that because the randomness of commitment $\mathbf{t} = \mathbf{Com}(\sum_{i=1}^{l} v_i^{(j)}, \mathbf{r})$ is as small as in one voter commitment, we have effectively reduced the size of the sum of the randomness in l voter commitments to that of one commitment. If we do this for every block of l voters, then we effectively reduced the sum of the randomness by a factor of l.

It is easy to see that this procedure is repeatable. Once every l blocks of votes have small randomness, we can consider repeating this procedure by summing over l such blocks. This will effectively reduce the total sum of the randomnesses by another factor of l. If we continue this procedure, then we will be effectively adding $2(N_V/l + N_V/l^2 + \ldots) \approx 2N_V/(l-1)$ extra proofs. The advantage will be that we now only need to worry about the randomness growing by a factor l for the proof in Equation (5.10). We set our parameters so that $l = 30$.

One can also think of the above procedure as the authority summing up 30 votes, recommitting to the sum using fresh, small randomness, and then giving a proof that he correctly recommitted to the sum of the votes. In particular, proving that the difference between his new commitment and the sum of the 30 commitments is a commitment to 0.

5.2 Our E-Voting Scheme

5.3 Building Blocks

5.3.1 The Commitment Scheme

We will use the commitment scheme of Section 2.4.1 for messages in $\mathcal{R}_q$. We briefly recall the **CSetup**, **Com** and **Open** algorithms.

- **CSetup**(1^λ) outputs a commitment matrix $\mathbf{A} \in \mathcal{R}_q^{(n+1)\times(2n+1)}$.
- **Com**$(m \in \mathcal{R}_q)$ outputs the commitment $\mathbf{t} := \mathbf{Ar} + \begin{bmatrix} \mathbf{0}^n \\ m \end{bmatrix}$, and randomness $\mathbf{r} \leftarrow D_{\mathcal{R},\sigma}^{2n+1}$.

[1] It is also possible to do these proofs in a non-amortized fashion and only get an approximate proof. In this case, depending on the value of l, the parameters may have to be increased.

- **Open**$(\mathbf{t}, \mathbf{r}, c)$ checks that the commitment is valid and outputs $\tilde{m} := t_2 - c^{-1}\mathbf{A}_2\mathbf{r} \in \mathcal{R}_q$.

We have fixed the dimensions of $\mathbf{A}$ to $n \times (2n+1)$ which implies that the hiding property of our commitment relies on M-LWE$_{q,n,\sigma}$ and the binding property relies on M-SIS$_{q,2n+1,4B_{\mathcal{C}},B_{Com}}$ we discuss how to fix $q, \sigma, B_{\mathcal{C}}, B_{Com}$ in Section 5.5.

Embedding votes as polynomials. We will consider elections for multiple candidates in which voters output a $\{0,1\}$ vote for each candidate, i.e. if N_C is the number of candidates, a vote will be of the form $(b_1, \ldots, b_{N_C})$ with $b_i \in \{0,1\}$. As the voters will commit to their vote we will have to embed such vectors as elements of $\mathcal{R}_q$ (we could consider commitments with message space $\mathcal{R}_q^{N_C}$ but that would be very inefficient). Additionally each voter will have to output a proof that his vote is well formed, that is to say that each of the b_i is either 0 or 1, we will need an embedding that allows such proofs to go through.
To do so we will consider the splitting properties of $X^d + 1$ given in Lemma 2.2.3. Suppose there are $N_C \leq d$ candidates with N_C a power of two. We can choose a modulus q such that $X^d + 1$ splits into exactly N_C irreducible factors modulo q, let $f_1, \ldots, f_{N_C}$ be these factors. By the Chinese remainder theorem we know that:

$$\mathbb{Z}[X]/X^d + 1 \simeq \mathbb{Z}[X]/f_1(X) \times \ldots \times \mathbb{Z}[X]/f_{N_C}(X)$$

with the isomorphism:

$$\begin{array}{rccc} \phi: & \mathbb{Z}[X]/X^d + 1 & \to & \mathbb{Z}[X]/f_1(X) \times \ldots \times \mathbb{Z}[X]/f_{N_C}(X) \\ & h & \mapsto & (h \mod f_1, \ldots, h \mod f_{N_C}) \end{array}$$

Using the extended Euclidean algorithm one can efficiently obtain polynomials $g_1, \ldots, g_{N_C}$ such that $\phi(g_i) = (0, \ldots, 0, 1, 0, \ldots, 0)$ with the 1 in the i^{th} coordinate. To embed a vector $(b_1, \ldots, b_{N_C}) \in \{0,1\}^{N_C}$ in $\mathcal{R}_q$ a voter will compute the polynomial $m := \sum b_i g_i$, by linearity of ϕ this polynomial is such that $\phi(m) = (b_1, \ldots, b_{N_C})$.

5.3.2 Proof of Correct Vote

We now consider how to prove that a commitment commits to a message $m \in \mathcal{R}_q$ such that $\phi(m) \in \{0,1\}^{N_C}$. Let $\phi(m) = (m_1, \ldots, m_{N_C})$ we want to prove that $\forall i \in [N_C], m_i \in \{0,1\}$. We will construct a proof that for a fixed $i \in [N_C], m_i \in \{0,1\}$, running this proof N_C times will give the desired proof of knowledge.
Fix $i \in [N_C]$, observe that $m_i = 1 \Leftrightarrow \phi(mg_i) = (0, \ldots, 0, 1, 0, \ldots, 0) = \phi(g_i) \Leftrightarrow mg_i = g_i$ and similarly $m_i = 0 \Leftrightarrow mg_i = 0$, which is to say:

$$m_i \in \{0,1\} \Leftrightarrow mg_i \in \{0, g_i\}$$

Let $\mathbf{t} := \mathbf{Com}(m; \mathbf{r})$, we will use OR-Proof of section 3.3 to prove that either $mg_i = 0$ or $mg_i = g_i$. Let $\mathbf{A}'_i := \begin{bmatrix} \mathbf{A}_1 \\ g_i\mathbf{A}_2 \end{bmatrix}$ and $\mathbf{t}'_i = \begin{bmatrix} \mathbf{t}_1 \\ g_i t_2 \end{bmatrix}$, we have that

$$\mathbf{A}'_i\mathbf{r} = \mathbf{t}'_i \Leftrightarrow mg_i = 0$$

$$\mathbf{A}'_i\mathbf{r} = \mathbf{t}'_i - \begin{bmatrix} \mathbf{0}^n \\ g_i \end{bmatrix} \Leftrightarrow mg_i = g_i$$

Which implies that $\Pi_{OR}(\mathbf{A}'_i, \mathbf{t}'_i, \mathbf{t}'_i - \begin{bmatrix}\mathbf{0}^n \\ g_i\end{bmatrix}; \mathbf{r})$ is a Σ'-Protocol from which we can extract $\bar{\mathbf{z}} \in \mathcal{R}^{2n+1}$, $\bar{c} \in \bar{\mathcal{C}}$, $m_i \in \mathcal{R}_q$ such that $\bar{\mathbf{z}} \leq 2B_{OR}$, $\bar{c}\mathbf{t} = \mathbf{Com}(\bar{c}m_i; \bar{\mathbf{z}})$, and $mg_i \in \{0, g_i\}$. Finally by simultaneously running this proof for all $\mathbf{A}'_i$ and $\mathbf{t}'_i$ we obtain a proof that $\phi(m) \in \{0,1\}^{N_C}$.

We formally present the proof computed by the voter in Algorithm 7 and its verification algorithm in Algorithm 8. If $\sigma_V \geq 11 B_{\mathcal{C}} B_V$ and $B'_V \geq \sqrt{2d(2n+1)}\sigma_V$, then this proof of knowledge is a Σ'-Protocol for the following languages:

$$\mathfrak{R}'_{Vote} = \left\{(\mathbf{t}, \mathbf{r}, 1) \in \mathcal{R}_q^{n+1} \times \mathcal{R}^{2n+1} \times \mathcal{R} \mid \exists v \in \mathcal{R}_q, \mathbf{t} = \mathbf{Com}(v; \mathbf{r}), \|\mathbf{r}\| \leq 2B'_V, \phi(v) \in \{0,1\}^{N_C}\right\}$$
$$\mathfrak{R}'_{Vote} = \left\{(\mathbf{t}, \mathbf{r}, \bar{c}) \in \mathcal{R}_q^{n+1} \times \mathcal{R}^{2n+1} \times \mathcal{R} \mid \exists v \in \mathcal{R}_q, \bar{c}\mathbf{t} = \mathbf{Com}(\bar{c}v; \mathbf{r}), \|\mathbf{r}\| \leq 2B'_V, \phi(v) \in \{0,1\}^{N_C}\right\}$$

Algorithm 7 Π_{Vote}

Require: Public information: $\mathbf{A}, \mathbf{t}$. Private information: $\mathbf{r} \in \mathcal{R}$, $v \in \mathcal{R}_q$, s.t $\mathbf{t} = \mathbf{Com}(v; \mathbf{r})$ and $(v_1, \ldots, v_{N_C} := \phi(v) \in \{0,1\}^{N_C}$

$\begin{bmatrix}\mathbf{A}_1 \\ \mathbf{A}_2\end{bmatrix} := \mathbf{A}$; $\begin{bmatrix}\mathbf{t}_1 \\ t_2\end{bmatrix} := \mathbf{t}$
for $i \in [N_C]$ **do**
  $\mathbf{A}'_i := \begin{bmatrix}\mathbf{A}_1 \\ g_i\mathbf{A}_2\end{bmatrix}$; $\mathbf{t}'_i := \begin{bmatrix}\mathbf{t}_1 \\ g_i t_2\end{bmatrix}$
  $c_{i,1-v_i} \xleftarrow{\$} \mathcal{C}$
  $\mathbf{z}_{i,1-v_i} \leftarrow D^{2n+1}_{\mathcal{R},\sigma_V}$
  $\mathbf{w}_{i,1-v_i} := \mathbf{A}'_i \mathbf{z}_{i,1-v_i} - c_{i,1-v_i}\mathbf{t}'_i + c_{i,1-v_i}\begin{bmatrix}\mathbf{0}^n \\ (1-v_i)g_i\end{bmatrix}$
  $\mathbf{y}_{i,v_i} \leftarrow D^{2n+1}_{\mathcal{R},\sigma_V}$
  $\mathbf{w}_{i,v_i} := \mathbf{A}'_i \mathbf{y}_{i,v_i}$
$c := H(\mathbf{A}, \mathbf{t}, \mathbf{w}_{1,0}, \ldots, \mathbf{w}_{N_C,0}, \mathbf{w}_{1,1}, \ldots, \mathbf{w}_{N_C,1})$
for $i \in [N_C]$ **do**
  $c_{i,v_i} := c \ominus c_{i,1-v_i}$
  $\mathbf{z}_{i,v_i} := \mathbf{r}c_{i,v_i} + \mathbf{y}_{i,v_i}$
  if $\mathsf{Rej}(\mathbf{z}_{i,v_i}, \mathbf{r}c_{i,v_i}, \sigma_V) = 0$ **then**
    Restart
return $(c_{i,0}, c_{i,1}, \mathbf{z}_{i,0}, \mathbf{z}_{i,1})_{i \in [N_C]}$

5.3.3 Amortized Exact Zero-Knowledge Proofs

The authorities $\mathcal{A}_j$ will use amortized zero knowledge proofs for two purpose. The first will be to prove that the shares of the commitment they receive from each voter are correctly formed. Let $\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{A}_2\end{bmatrix}$ be the commitment matrix, say $\mathcal{A}_j$ has the commitments $\mathbf{t}_1, \ldots, \mathbf{t}_{N_V}$ with opening $\mathbf{r}_1, \ldots, \mathbf{r}_{N_A}$. If we rewrite $\begin{bmatrix}\mathbf{t}_{i,1} \\ t_{i,2}\end{bmatrix} := t_i$, $\mathbf{T}_1 := \left[\mathbf{t}_{1,1} \mid \ldots \mid \mathbf{t}_{N_V,1}\right]$, and $\mathbf{R} := \left[\mathbf{r}_1 \mid \ldots \mid \mathbf{r}_{N_V}\right]$, then the protocol $\Pi_{Amo2}(\mathbf{A}_1, \mathbf{T}_1; \mathbf{R})$ is a proof that each commitment is well

Algorithm 8 Verify$_{Vote}$

Require: Public information: $\mathbf{A}, \mathbf{t}, (c_{i,0}, c_{i,1}, \mathbf{z}_{i,0}, \mathbf{z}_{i,1})_{i\in[N_C]}$

$V := 1$
$\begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix} := \mathbf{A}$; $\begin{bmatrix}\mathbf{t}_1\\ t_2\end{bmatrix} := \mathbf{t}$
for $i \in [N_C]$ **do**
    $\mathbf{A}'_i := \begin{bmatrix}\mathbf{A}_1\\ g_i\mathbf{A}_2\end{bmatrix}$; $\mathbf{t}'_i := \begin{bmatrix}\mathbf{t}_1\\ g_i t_2\end{bmatrix}$
    $\mathbf{w}_{i,0} := A'_i\mathbf{z}_{i,0} - c_{i,0}\mathbf{t}'_i$
    $\mathbf{w}_{i,1} := A'_i\mathbf{z}_{i,1} - c_{i,1}\mathbf{t}'_i + c_{i,1}\begin{bmatrix}\mathbf{0}^n\\ g_i\end{bmatrix}$
$c := H(\mathbf{A}, \mathbf{t}, \mathbf{w}_{1,0}, \ldots, \mathbf{w}_{N_C,0}, \mathbf{w}_{1,1}, \ldots, \mathbf{w}_{N_C,1})$
for $i \in [N_C]$ **do**
    if $c \neq c_{i,0} \oplus c_{i,1} \vee \|\mathbf{z}_{i,0}\| > B'_V \vee \|\mathbf{z}_{i,1}\| > B'_V$ **then**
        $V \leftarrow 0$
return V

formed. The second proof of knowledge will be used during bucketing (c.f. Section 5.4.3) to prove that differences of commitments commit to zero. Say $\mathcal{A}_j$ has the commitments $\mathbf{t}_1, \ldots, \mathbf{t}_\ell$ which commit to 0 with opening $\mathbf{r}_1, \ldots, \mathbf{r}_\ell$. Let $\mathbf{T} := \left[\mathbf{t}_1 \mid \ldots \mid \mathbf{t}_\ell\right]$, and $\mathbf{R} := \left[\mathbf{r}_1 \mid \ldots \mid \mathbf{r}_\ell\right]$, then the protocol $\Pi_{Amo2}(\mathbf{A}, \mathbf{T}; \mathbf{R})$ is a proof that each commitment is well formed and commits to 0.

We formally present the proof computed by the authorities in Algorithms 9 and 10, and their verification algorithms in Algorithms 11 and 12. If $k \geq (\lambda+2)/\log(2d+1)$, $\sigma_{A1} \geq 11B_{A1}\sqrt{lk}$ and $B'_{A1} \geq \sqrt{2d(2n+1)}\sigma_{A1}$, then the protocol $\Pi_{Auth,1}$ is a Σ'-Protocol for the following languages:

$$\mathfrak{R}_{Auth,1} = \left\{ \begin{array}{c} (\mathbf{t}_1, \ldots, \mathbf{t}_\ell, \mathbf{r}_1, \ldots, \mathbf{r}_l) \in \mathcal{R}_q^{(n+1)\times\ell} \times \mathcal{R}^{(2n+1)\times\ell} \\ \text{s.t. } \exists (v_1, \ldots, v_\ell) \in \mathcal{R}_q^\ell;\ \forall i \in [\ell], \mathbf{t}_i = \mathbf{Com}(v_i, \mathbf{r}_i), s_1\left(\left[\mathbf{r}_1 \mid \ldots \mid \mathbf{r}_\ell\right]\right) \leq B_{A1} \end{array} \right\}$$

$$\mathfrak{R}'_{Auth,1} = \left\{ \begin{array}{c} (\mathbf{t}_1, \ldots, \mathbf{t}_\ell, \mathbf{r}_1, \ldots, \mathbf{r}_l) \in \mathcal{R}_q^{(n+1)\times\ell} \times \mathcal{R}^{(2n+1)\times\ell} \\ \text{s.t. } \exists (v_1, \ldots, v_\ell) \in \mathcal{R}_q^\ell;\ \forall i \in [\ell], \mathbf{t}_i = \mathbf{Com}(v_i, \mathbf{r}_i), \|\mathbf{r}_i\| \leq 2B'_{A1} \end{array} \right\}$$

If $k \geq (\lambda+2)/\log(2d+1)$, $\sigma_{A2} \geq 11B_{A2}\sqrt{lk}$ and $B'_{A2} \geq \sqrt{2d(2n+1)}\sigma_{A2}$, then the protocol $\Pi_{Auth,2}$ is a Σ'-Protocol for the following languages:

$$\mathfrak{R}_{Auth,2} = \left\{ \begin{array}{c} (\mathbf{t}_1, \ldots, \mathbf{t}_\ell, \mathbf{r}_1, \ldots, \mathbf{r}_l) \in \mathcal{R}_q^{(n+1)\times\ell} \times \mathcal{R}^{(2n+1)\times\ell} \\ \text{s.t. } \forall i \in [\ell], \mathbf{t}_i = \mathbf{Com}(0, \mathbf{r}_i), s_1\left(\left[\mathbf{r}_1 \mid \ldots \mid \mathbf{r}_\ell\right]\right) \leq B_{A2} \end{array} \right\}$$

$$\mathfrak{R}'_{Auth,2} = \left\{ \begin{array}{c} (\mathbf{t}_1, \ldots, \mathbf{t}_\ell, \mathbf{r}_1, \ldots, \mathbf{r}_l) \in \mathcal{R}_q^{(n+1)\times\ell} \times \mathcal{R}^{(2n+1)\times\ell} \\ \text{s.t. } \forall i \in [\ell], \mathbf{t}_i = \mathbf{Com}(0, \mathbf{r}_i), \|\mathbf{r}_i\| \leq 2B'_{A2} \end{array} \right\}$$

Algorithm 9 $\Pi_{Auth,1}$

Require: Public information: $\mathbf{A}, \mathbf{t}_1, \ldots, \mathbf{t}_\ell$. Private information: $\mathbf{r}_1, \ldots, \mathbf{r}_\ell \in \mathcal{R}$, s.t $\exists (m_1, \ldots, m_\ell) \in \mathcal{R}_q^\ell, \mathbf{t}_i = \mathbf{Com}(v_i; \mathbf{r}_i)$.

$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} := \mathbf{A}$
for $i \in [N_C]$ **do**
  $\begin{bmatrix} \mathbf{t}_{i,1} \\ t_{i,2} \end{bmatrix} := \mathbf{t}_i$
$\mathbf{T}_1 := \begin{bmatrix} \mathbf{t}_{1,1} \mid \ldots \mid \mathbf{t}_{\ell,1} \end{bmatrix}$
$\mathbf{R} := \begin{bmatrix} \mathbf{r}_1 \mid \ldots \mid \mathbf{r}_\ell \end{bmatrix}$
$\mathbf{Y} \leftarrow D_{\mathcal{R},\sigma_{A1}}^{(2n+1)\times k}$
$\mathbf{W} = \mathbf{A}_1 \mathbf{Y}$
$\mathbf{C} := H(\mathbf{A}_1, \mathbf{T}_1, \mathbf{W}) \in \{0,1\}^{\ell \times k}$
$\mathbf{Z} := \mathbf{RC} + \mathbf{Y}$
if $\mathsf{Rej}(\mathbf{Z}, \mathbf{RC}, \sigma_{A1}) = 0$ **then**
  Restart
return $(\mathbf{Z}, \mathbf{C})$

Algorithm 10 $\Pi_{Auth,2}$

Require: Public information: $\mathbf{A}, \mathbf{t}_1, \ldots, \mathbf{t}_\ell$. Private information: $\mathbf{r}_1, \ldots, \mathbf{r}_\ell \in \mathcal{R}$, s.t $\mathbf{t}_i = \mathbf{Com}(0; \mathbf{r}_i)$.

$\mathbf{T} := \begin{bmatrix} \mathbf{t}_1 \mid \ldots \mid \mathbf{t}_\ell \end{bmatrix}$
$\mathbf{R} := \begin{bmatrix} \mathbf{r}_1 \mid \ldots \mid \mathbf{r}_\ell \end{bmatrix}$
$\mathbf{Y} \leftarrow D_{\mathcal{R},\sigma_{A2}}^{(2n+1)\times k}$
$\mathbf{W} = \mathbf{AY}$
$\mathbf{C} := H(\mathbf{A}, \mathbf{T}, \mathbf{W}) \in \{0,1\}^{\ell \times k}$
$\mathbf{Z} := \mathbf{RC} + \mathbf{Y}$
if $\mathsf{Rej}(\mathbf{Z}, \mathbf{RC}, \sigma_{A2}) = 0$ **then**
  Restart
return $(\mathbf{Z}, \mathbf{C})$

Algorithm 11 $\mathbf{Verify}_{Auth,1}$

Require: Public information: $\mathbf{A}, \mathbf{t}_1, \ldots, \mathbf{t}_\ell, \mathbf{Z}, \mathbf{C}$.

$\mathbf{T} := \begin{bmatrix} \mathbf{t}_1 \mid \ldots \mid \mathbf{t}_\ell \end{bmatrix}$
$\mathbf{W} := \mathbf{AZ} - \mathbf{TC}$
if $\mathbf{C} \neq H(\mathbf{A}, \mathbf{T}, \mathbf{W}) \ \vee \ \|\mathbf{Z}\|_{\max} > B'_{A2}$ **then**
  return 0
return 1

Algorithm 12 Verify$_{Auth,2}$

Require: Public information: $\mathbf{A}, \mathbf{t}_1, \ldots, \mathbf{t}_\ell, \mathbf{Z}, \mathbf{C}$.

$\begin{bmatrix}\mathbf{A}_1\\ \mathbf{A}_2\end{bmatrix} := \mathbf{A}$
for $i \in [N_C]$ **do**
  $\begin{bmatrix}\mathbf{t}_{i,1}\\ t_{i,2}\end{bmatrix} := \mathbf{t}_i$
$\mathbf{T}_1 := \left[\mathbf{t}_{1,1} \mid \ldots \mid \mathbf{t}_{\ell,1}\right]$
$\mathbf{W} := \mathbf{AZ} - \mathbf{T}_1\mathbf{C}$
if $\mathbf{C} \neq H(\mathbf{A}_1, \mathbf{T}_1, \mathbf{W}) \ \vee \ \|\mathbf{Z}\|_{\max} > B'_{A1}$ **then**
  return 0
return 1

5.4 Our E-Voting Scheme

5.4.1 Definitions

We base our syntax and security definitions of one-pass electronic voting schemes on that of Bernhard et al. [BCG+15], but there are some differences. First, we explicitly model a multi-authority setting where each authority independently generates its own keys. Second, ballot testing in our scheme must be performed by the authorities, who use their secret key in the process. Verification of the entire election can still be done publicly, though. As ballot testing, tallying, and verification do not require interaction with the voters, the authorities do not need to be online for the first part of the election and only need to run the ballot testing and tallying algorithms once all the ballots have been cast.

A multi-authority electronic voting scheme $\mathcal{EV}$ is a tuple (**Setup**, **ASetup**, **Vote**, **TestB**, **Tally**, **Verify**) of algorithms and protocols that are used by authorities $\mathcal{A}_1, \ldots, \mathcal{A}_{N_A}$ and voters with identities $id \in \mathbb{I}$ as follows. We consider binary elections for N_C candidates where each voter $id \in \mathbb{I}$ casts a vote $v_{id} \in \{0,1\}^{N_C}$ and the result of the election is $r = \sum_{id\in\mathbb{I}} v_{id}$. We assume that all voters and authorities have read access and authenticated append-only write access to a public bulletin board BB, meaning that entries can only be appended to the board and entries are authenticated (e.g., signed) under the writer's identity. Moreover, each voter can only write once to the bulletin board; authorities can write as often as they want.

- **Setup**(1^λ) generates trusted common parameters par.
- **ASetup**(par) is used by authority $\mathcal{A}_j$ to generate a public key pk_j and corresponding secret key sk_j.
- **Vote**$(par, pk_1, \ldots, pk_{N_A}, id, v)$ is used by voter $id \in \mathbb{I}$ to cast his vote $v \in \{0,1\}^{N_C}$. It returns a ballot b that the voter posts on the bulletin board BB.
- **TestB**$(par, pk_1, \ldots, pk_{N_A}, sk_j, b)$ allows authority $\mathcal{A}_j$ to test whether ballot b is valid or not by returning 1 or 0, respectively. The ballot is only considered valid after all N_A authorities confirm its validity on the bulletin board BB. This check can be performed as the votes come in, or only after the voting phase has ended. The tallying authorities therefore do not have to be online during the voting phase: rather than interacting

directly with the voters, the tallying authorities can obtain the ballots from the bulletin board after voting has ended and discard invalid ballots if needed.

- **Tally**$(par, pk_1, \ldots, pk_{N_A}, BB, sk_j)$ is an interactive protocol run among the authorities $\mathcal{A}_j$, $j = 1, \ldots, N_A$, at the end of which they announce the tally r and proof Π.
- **Verify**$(par, pk_1, \ldots, pk_{N_A}, BB, r, \Pi)$ can be run by anyone to check the correctness of the election result.

Correctness. Correctness guarantees that, when all parties behave honestly, all ballots are deemed valid and the result of the election is correct. Let $id_1, \ldots, id_{N_V} \in \mathbb{I}$ be voter identities and $v_1, \ldots, v_{N_V}$ be their respective votes. Let $par \xleftarrow{\$}$ **Setup**(1^λ); and $(pk_j, sk_j) \xleftarrow{\$}$ **ASetup**(par) for $j = 1, \ldots, N_A$. For $i = 1, \ldots, N_V$ and $j = 1, \ldots, N_A$ let $b_i \xleftarrow{\$}$ **Vote**$(par, pk_1, \ldots, pk_{N_A}, id_i, v_i)$, $BB[i] \leftarrow b_i$, and let (r, Π) be the outcome of the protocol when each authority $\mathcal{A}_j$ runs **Tally**$(par, pk_1, \ldots, pk_{N_A}, BB, sk_j)$, $j = 1, \ldots, N_A$. The scheme is correct if for all $i = 1, \ldots, N_V$ and $j = 1, \ldots, N_A$, the following conditions hold with overwhelming probability: $r = \sum_{i=1}^{N_V} v_i$, **TestB**$(par, pk_1, \ldots, pk_{N_A}, sk_j, b_i) = 1$, and **Verify**$(par, pk_1, \ldots, pk_{N_A}, BB, r, \Pi) = 1$.

Privacy. Privacy requires that an adversary who corrupts $N_A - 1$ authorities and an

$\mathsf{Exp}^{\mathrm{priv},b}_{\mathcal{A}}(\lambda)$:
  $par \leftarrow$ **Setup**(1^λ) ; $pk_1 \leftarrow$ **ASetup**(par) ; $HV := \emptyset$
  $(pk_2, \ldots, pk_{N_A}, st) \leftarrow \mathcal{A}(par, pk_1)$
  $b' \leftarrow \mathcal{A}^{\mathcal{O}, BB}(st)$
  $HV' := \{(id, v_0, v_1, b) \in HV$:
    $\forall j \in \{1, \ldots, N_A\}$: "$\mathcal{A}_j$ approves b"$\in BB\}$
  $V_0 := \sum_{(id,v_0,v_1,b) \in HV'} v_0$; $V_1 := \sum_{(id,v_0,v_1,b) \in HV'} v_1$
  If $V_0 \neq V_1$ then return $\perp$ else return b'

$\mathcal{O}$**Vote**(id, v_0, v_1):
  $b \leftarrow$ **Vote**$(par, pk_1, \ldots, pk_{N_A}, id, v_b)$
  $HV \leftarrow HV \cup \{(id, v_0, v_1, b)\}$
  $BB \leftarrow BB \| $"$id$ casts b"
  If **TestB**$(par, pk_1, \ldots, pk_{N_A}, sk_1, b) = 1$
    then $BB \leftarrow BB \|$"$\mathcal{A}_1$ approves b"
    else $BB \leftarrow BB \|$"$\mathcal{A}_1$ rejects b"

$\mathcal{O}$**Cast**(id, b):
  If "id casts b" $\in BB$ and **TestB**$(par, pk_1, \ldots, pk_{N_A}, sk_1, b) = 1$
    then $BB \leftarrow BB \|$"$\mathcal{A}_1$ approves b"
    else $BB \leftarrow BB \|$"$\mathcal{A}_1$ rejects b"

$\mathcal{O}$**Tally**:
  Run **Tally**$(par, pk_1, \ldots, pk_{N_A}, BB, sk_1)$ with $\mathcal{A}$ to obtain (r, Π)
  Return (r, Π)

Figure 5.1: Experiment for privacy

arbitrary number of voters does not learn anything more about the votes of honest voters

than what is revealed by the election result. The single-authority BPRIV notion of Bernhard et al. [BCG+15] defines this by requiring that the adversary cannot tell a bulletin board for a first set of votes with the real election result and proof from a bulletin board for a second set of votes with the same result and a simulated proof. The BPRIV notion is not easily adapted to the multi-authority setting, because the corrupt authorities would have to be involved in computing the tally for both bulletin boards. We therefore adapt the notion to require that the adversary cannot distinguish between the bulletin boards of two different sets of votes, as long as both sets of votes yield the same election result. The advantage of an adversary $\mathcal{A}$ in breaking the privacy of the electronic voting scheme $\mathcal{EV}$ is defined through the experiment of Figure 5.1 as

$$\mathsf{Adv}^{\mathrm{priv}}_{\mathcal{A}}(\lambda) \;=\; \Big|\; \Pr[\mathsf{Exp}^{\mathrm{priv},0}_{\mathcal{A}}(\lambda)=1] - \Pr[\mathsf{Exp}^{\mathrm{priv},1}_{\mathcal{A}}(\lambda)=1] \;\Big|\;,$$

where $\mathcal{A}$ is given access to all oracles in the set $\mathcal{O} = \{\mathcal{O}\mathbf{Vote}, \mathcal{O}\mathbf{Cast}, \mathcal{O}\mathbf{Tally}\}$ as well as read and append-only write access to the bulletin board BB. The $\mathcal{O}\mathbf{Vote}$ and $\mathcal{O}\mathbf{Cast}$ oracles can be queried as many times as $\mathcal{A}$ wants, but the $\mathcal{O}\mathbf{Tally}$ oracle can only be queried once.

$\mathsf{Exp}^{\mathrm{cons}}_{\mathcal{A}}(\lambda)$:
$par \xleftarrow{\$} \mathbf{Setup}(1^\lambda)\ ;\ HV \leftarrow \emptyset$
$(pk_1, \ldots, pk_{N_A}, st) \xleftarrow{\$} \mathcal{A}(par)$
$(r, \Pi) \xleftarrow{\$} \mathcal{A}^{\mathcal{O}\mathbf{Vote}, BB}(st)$
$HV' := \{(id, v, b) \in HV\ :$
$\quad \forall j \in \{1, \ldots, N_A\}\ :$ "$\mathcal{A}_j$ approves b"$\in BB\}$
$\forall k \in \{1, \ldots, N_C\},\ h_{0,k} := |\{(id, v, b) \in HV'\ :\ v[k] = 0\}|$
$\forall k \in \{1, \ldots, N_C\},\ h_{1,k} := |\{(id, v, b) \in HV'\ :\ v[k] = 1\}|$
$t \leftarrow |\{b\ :\ \forall j \in \{1, \ldots, N_A\}\ :$ "$\mathcal{A}_j$ approves b"$\in BB\}|$
If $\mathbf{Verify}(par, pk_1, \ldots, pk_{N_A}, BB, r, \Pi) = 1$
$\quad$ and $(\exists k \in \{1, \ldots, N_C\},\ r[k] < h_{1,k}$ or $r[k] > t - h_{0,k})$
$\quad$ then return 1 else return 0

$\mathcal{O}\mathbf{Vote}(id, v)$:
$b \xleftarrow{\$} \mathbf{Vote}(par, pk_1, \ldots, pk_{N_A}, id, v)$
$HV \leftarrow HV \cup (id, v, b)$
$BB \leftarrow BB \| $ "id casts b"

Figure 5.2: Experiment for consistency

Consistency. Consistency requires that the election result is correct with respect to the votes cast by voters. Bernhard et al.'s notion of strong consistency [BCG+15] requires that individual ballots can be extracted online. We relax this notion by requiring that, if an election finishes successfully, the result must be "realistic" with respect to the honestly cast votes. Meaning that for each candidate, the result must be at least the number of honest one-votes and at most the total number of votes cast minus the number of honest zero-votes. We strengthen the notion, however, by requiring that this property holds even against corrupt election authorities. Intuitively, our notion is similar to the quantitative verifiability goal of Cortier et al. [CGK+16].

Formally, the advantage of an adversary $\mathcal{A}$ in breaking the consistency of $\mathcal{EV}$ is defined

through the consistency experiment of Figure 5.2 as

$$\mathsf{Adv}^{\text{cons}}_{\mathcal{A}}(\lambda) \;=\; \Pr[\mathsf{Exp}^{\text{cons}}_{\mathcal{A}}(\lambda) = 1] \;.$$

5.4.2 The Scheme

We instantiate our voting scheme according to the definition given in Section 5.4.1. We split the tallying algorithm in two parts, this algorithm in our E-Voting definition is interactive between authorities $\mathcal{A}_1, \ldots, \mathcal{A}_{N_A}$. In our instantiation there is no need for interaction, each authority $\mathcal{A}_j$ can run an algorithm $\mathbf{Tally}_j(par, pk_1, \ldots, pk_{N_V}, BB, sk_j)$ and publish on the bulletin board its partial tally $t^{(j)}$ and proof $\pi^{A,(j)}$, anyone can then run $\mathbf{Tally}(par, pk_1, \ldots, pk_{N_V}, BB)$ to compute the total tally and final proof.

$\mathbf{Setup}(1^\lambda)$:

- generate parameters n, q, d, σ.
- $\mathbf{A} := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \leftarrow \mathbf{CSetup}(1^\lambda)$, with $\mathbf{A} \in \mathcal{R}_q^{(n+1)\times(2n+1)}$.
- output $par := n, q, d, \sigma, \mathbf{A}$

Let $\left(\mathbf{KGen}(1^\lambda), \mathbf{Enc}, \mathbf{Dec}\right)$ be a CCA-Secure public key encryption scheme, which the authorities will use to obtain the shares of the randomness of each voter.
$\mathbf{ASetup_j}(par)$:

- $(pk_j, sk_j) \leftarrow \mathbf{KGen}(1^\lambda)$
- Give sk_j to $\mathcal{A}_j$
- output pk_j

To cast a bulletin, voter i will share his vote into N_A additive shares and compute a commitment $\mathbf{t}_i^{(j)}$ for each of them. He proves that the sum of these commitments is a commitment to v_i such that$\phi(v_i) \in \{0,1\}^{N_C}$ and then encrypts the randomness $\mathbf{r}_i^{(j)}$ under the public key pk_j so that each authority can open one share of the vote. He finally posts his vote, proof, commitments, and encryptions on the bulletin board along with a signature.
$\mathbf{Vote_i}(par, pk_1, \ldots, pk_{N_A}, id_i, v_i)$:

- $v_i^{(j)} \xleftarrow{\$} \mathcal{R}_q$ s.t. $v_i = \sum_{j=1}^{N_A} v_i^{(j)}$
- $\mathbf{r}_i^{(j)} \leftarrow D_{\mathcal{R},\sigma}^{2n+1}$
- $\mathbf{t}_i^{(j)} := \mathbf{Com}\left(v_i^{(j)}; \mathbf{r}_i^{(j)}\right)$
- $\mathbf{r}_i := \sum_{j=1}^{N_A} \mathbf{r}_i^{(j)}$
- $\mathbf{t}_i := \sum_{j=1}^{N_A} \mathbf{t}_i^{(j)}$
- $\pi_i^V = \Pi_{Vote}(\mathbf{A}, \mathbf{t}_i; \mathbf{r}_i, v_i)$

- $\mathbf{e}_i^{(j)} = \mathbf{Enc}(\mathbf{r}_i^{(j)}, pk_j)$
- $b_i = \left(id_i, \pi_i^V, (\mathbf{t}_i^{(j)}, \mathbf{e}_i^{(j)})_{j\in[N_A]}\right)$
- Sign and publish b_i on the bulletin board

Before tallying the votes each authority $\mathcal{A}_j$ will check whether the bulletins have been properly cast. i.e. for each bulletin b_i, $\mathcal{A}_j$ checks the signature on b_i, the proof that $\phi(v_i) \in \{0,1\}^{N_C}$ and that the encryption of $\mathbf{r}_i^{(j)}$ under his public key decrypts to a valid randomness.
$\mathbf{TestB_{i,j}}(par, pk_1, \ldots, pk_{N_A}, sk_j, b_i)$:

- $\left(id_i, \pi_i^V, \mathbf{t}_i^{(1)}, \mathbf{e}_i^{(1)}, \ldots, \mathbf{t}_i^{(N_A)}, \mathbf{e}_i^{(N_A)}\right) := b_i$
- Check that b_i was signed by voter id_i
- $\mathbf{Verify}(\mathbf{A}, \mathbf{t}_i, \pi_i^V)$
- $\mathbf{r}_i^{(j)} := \mathbf{Dec}(\mathbf{e}_i^{(j)}, sk_j)$
- Check $\left\|\mathbf{r}_i^{(j)}\right\| \leq \sqrt{2d(2n+1)}\sigma$

Each authority $\mathcal{A}_j$ will compute its share $t^{(j)}$ of the total tally as well as a proof that $t^{(j)}$ has been computed correctly. To do so $\mathcal{A}_j$ first decrypts $\mathbf{e}_i^{(j)}$ for each bulletin b_i to recover randomness $\mathbf{r}_i^{(j)}$. He then proves that for each voter i, $\mathbf{r}_i^{(j)}$ is a valid opening of $\mathbf{t}_i^{(j)}$ and finally outputs $\mathbf{r}^{(j)}$, the sum over i of all $\mathbf{r}_i^{(j)}$ (the share $t^{(j)}$ of the final tally can be obtained by opening $\mathbf{t}^{(j)}$ the sum of the $\mathbf{t}_i^{(j)}$ using the $\mathbf{r}^{(j)}$ output by $\mathcal{A}_j$) as well as the proofs he computed. Note that w.l.o.g we will consider in all the following algorithms that all the bulletins on the bulletin board were tested and accepted by all the authorities (otherwise we can just discard the rejected bulletins and adjust N_V to the number of remaining bulletins).
$\mathbf{Tally}_j(par, pk_1, \ldots, pk_{N_A}, BB, sk_j)$

- $\left(id_i, \pi_i^V, \mathbf{t}_i^{(1)}, \mathbf{e}_i^{(1)}, \ldots, \mathbf{t}_i^{(N_A)}, \mathbf{e}_i^{(N_A)}\right) := b_i$, For $(b_i)_{i\in N_V} \in BB$
- $\forall i,\ \mathbf{r}_i^{(j)} := \mathbf{Dec}(\mathbf{e}_i^{(j)}, sk_j)$
- $\pi^{A,(j)} = \left(\pi_1^A, \ldots, \pi_{N_V}^A\right)$
  $= \Pi_{Auth1}\left(\mathbf{A}, \mathbf{t}_1^{(j)}, \ldots, \mathbf{t}_{N_V}^{(j)}, \mathbf{r}_1^{(j)}, \ldots, \mathbf{r}_{N_V}^{(j)}\right)$
- $\mathbf{r}^{(j)} = \sum_{i=1}^{N_V} \mathbf{r}_i^{(j)}$
- Sign and publish $\pi^{A,(j)}, \mathbf{r}^{(j)}$ on the bulletin board

To compute the total tally, anyone can simply recover the randomnesses $\mathbf{r}^{(j)}$ published by each authority $\mathcal{A}_j$, compute the corresponding commitment $\mathbf{t}^{(j)} = \sum \mathbf{t}_i^{(j)}$ and open it to the partial tally $t^{(j)}$. The total tally is then the sum of the partial tallies.
$\mathbf{Tally}(par, pk_1, \ldots, pk_{N_A}, BB)$

- For each authority $\mathcal{A}_j$, $j \in [N_A]$ recover $\mathbf{r}^{(j)}$ on BB

- $\forall j,\ \mathbf{t}^{(j)} := \sum_{i=1}^{N_V} \mathbf{t}_i^{(j)}$
- $\forall j,\ t^{(j)} := \mathbf{Open}(\mathbf{t}^{(j)}, \mathbf{r}^{(j)}, 1)$
- $t := \sum_1^{N_A} t^{(j)}$
- publish t

The verification algorithm can be run by anyone to check that the final tally is correct (i.e. the voting scheme is publicly verifiable). To do so one simply verifies all the proofs output by the voters and authorities and checks that the opening of the total tally has been done correctly (by computing it again).
Verify$(par, pk_1, \ldots, pk_{N_A}, BB, t)$

- For each $i \in [N_V], j \in [N_A]$, recover $\mathbf{t}_i^{(j)}, \pi_i^V, \pi_{i,j}^A$ on BB.
- $\forall i$, verify π_i^V
- $\forall i, j$, verify $\pi_{i,j}^A$
- $\forall j,\ \mathbf{t}^{(j)} := \sum_{i=1}^{N_V} \mathbf{t}_i^{(j)}$
- $\forall j,\ t^{(j)} := \mathbf{Open}(\mathbf{t}^{(j)}, \mathbf{r}^{(j)}, 1)$
- Check that $t = \sum_1^{N_A} t^{(j)}$

For correctness we need all the proofs to verify correctly, which will be true with overwhelming probability for appropriate parameters (cf. Section 5.5), we also need the test on the norm of $\mathbf{r}_i^{(j)}$ to succeed, which will be true with overwhelming probability by using Lemma 2.3.5, and we need for the commitment of the partial tallies to open correctly, i.e. we need $\left\|\mathbf{r}^{(j)}\right\| \leq B_{Com}$. We can fix the parameters so that this condition is verified, however the norm of $\mathbf{r}^{(j)}$ grows linearly with the number of voters which, as we discuss in the next section, heavily impacts the efficiency of this scheme.

Dealing with Misbehaving Authorities. A malicious authority could prevent a voter from casting his vote by claiming that the voter's ballot is invalid. Since the **TestB** algorithm requires the secret key of the authority, the authority's claim cannot be publicly verified. This situation can be improved by letting voters store the randomness used in the encryption of $\mathbf{e}_i^{(j)}$ and, in case their ballot is incorrectly claimed to be invalid, reveal $\mathbf{r}_i^{(j)}$ and the randomness to show that the authority is at fault.

5.4.3 Improved Voting Scheme

A major caveat in the scheme presented in Section 5.4.2 is that parameters grow linearly in the number of voters. Indeed for correctness a verifier needs to be able to open the sum over all voters of the commitments of the vote shares, this implies that the bound B_{Com} on the size of correct openings grows linearly in the number of voters. Increasing this bound heavily impacts the parameters of the scheme, e.g. if we fix $d = 256$ and $q \simeq 2^{31}$, then for ~ 100 bits of security we require a dimension $n = 7$ for 100 voters and $n = 12$ for 100 000 voters. This nearly doubles the commitment size, proof size and communication cost per voter (another

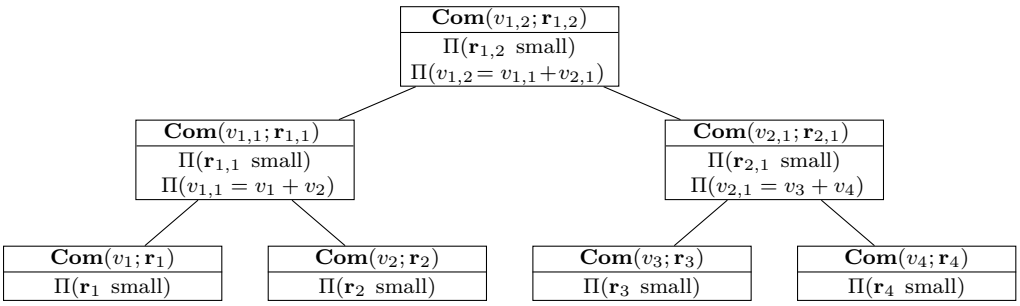

Figure 5.3: Example of the improved tallying for an authority. At each level s fresh randomnesses $\mathbf{r}_{u,s}$ are sampled and the authority commits to the sum of the votes of the previous level. The authority computes a proof that each new randomness is small and that it commits to the right value. Finally the authority publishes all the commitments and proofs as well as the opening, here $\mathbf{r}_{1,2}$, of the top level commitment which opens to the sum of the votes (i.e. $v_{1,2} = v_1 + v_2 + v_3 + v_4$).

issue with the previous scheme is that privacy is nontrivial, indeed revealing the sum of the randomnesses used makes it so that the privacy cannot be easily proven). We avoid this problem by using the fact that the authorities know the shares of many commitments and can thus create new commitments for the sum of their associated messages (this is the solution discussed in Section 5.1.2.1). e.g. Imagine authority $\mathcal{A}_j$ has received the commitments and openings $\mathbf{t}_i^{(j)} = \mathbf{Com}(v_i^{(j)}; \mathbf{r}_i^{(j)})$ from voters 1 to N_V, $\mathcal{A}_j$ can choose $l << N_V$ and compute new commitments $\mathbf{t}_{1,1}^{(j)}, \ldots, \mathbf{t}_{N_V/l,1}^{(j)}$, where $\mathbf{t}_{i,1}^{(j)} = \mathbf{Com}(\sum_{i'=l(k-1)+1}^{lk} v_{i'}^{(j)}; \mathbf{r}_{i,1}^{(j)})$ with fresh randomnesses $\mathbf{r}_{i,1}^{(j)}$ and publish these commitments on the bulletin board. Notice that $\mathbf{t}^{(j)} = \sum_{i=1}^{N_V} \mathbf{t}_i^{(j)}$ opens to the same message as $\mathbf{t}^{(j)'} = \sum_{i=1}^{N_V/l} \mathbf{t}_{i,1}^{(j)}$ (assuming both sums are valid commitments). However the randomness in the commitment $\mathbf{t}^{(j)'}$ will be approximately l times smaller than the one in $\mathbf{t}^{(j)}$ which means that $\mathbf{t}^{(j)'}$ can be a valid commitment even if $\mathbf{t}^{(j)}$ is not. By proving in zero knowledge that for $i \leq N_V/l$ the commitment $\mathbf{t}_{i,1}^{(j)}$ is valid and opens to the same value as $\sum_{i'=l(i-1)+1}^{li} \mathbf{t}_{i'}^{(j)}$ we can ensure that the scheme remains secure even if parameters are only set so that $\mathbf{t}^{(j)'}$ is valid and not $\mathbf{t}^{(j)}$. This effectively allows us to reduce B_{Com} by a factor l. This process can be iterated by summing the $\mathbf{t}_{i,1}^{(j)}$ by buckets of l and outputting new commitments $\mathbf{t}_{i,2}^{(j)}$ to the sum of the corresponding messages with fresh randomnesses, once again accompanied by a proof that each commitment is valid and opens to the same value as a sum of $\mathbf{t}_{i,1}^{(j)}$ (an example of such summations with buckets of size $l = 2$ is given in Figure 5.3).
Each authority can repeat this process until there are less than l commitments to be summed, resulting in a bound B_{Com} that grows linearly in l but remains independent of the number of voters. On the other hand this new protocol will output an overhead of $\sim N_V/(l-1)$ commitments (more precisely $N_V/l + N_V/l^2 + \ldots$ extra commitments) and 2 additional proofs (which can be amortized over) for each new commitment.
$\mathbf{Tally}_j(par, pk_1, \ldots, pk_{N_A}, BB, sk_j)$

- $\left(id_i, \pi_i^V, \mathbf{t}_1^{(j)}, \mathbf{e}_1^{(j)}, \ldots, \mathbf{t}_{N_V}^{(j)}, \mathbf{e}_{N_V}^{(j)}\right) := b_i$, For $(b_i)_{i \in N_V} \in BB$

- $\forall i \in [N_V]\,,\ \mathbf{r}_{i,0}^{(j)} := \mathbf{Dec}(\mathbf{e}_i^{(j)}, sk_j)$
- $\forall i \in [N_V]\,,\ v_{i,0}^{(j)} := \mathbf{Open}(\mathbf{t}_i^{(j)}, \mathbf{r}_i^{(j)}, 1)$
- For $s \in (1, \ldots, \lceil \log_l(N_V) \rceil)$:
  - For $u \in (1, \ldots, \lceil N_V / l^s \rceil)$:
    - $\mathbf{r}_{u,s}^{(j)} \leftarrow D_{\mathcal{R},\sigma}^{2n+1}$
    - $\mathbf{r}_{\mathbf{u},\mathbf{s}}^{(\mathbf{j})\prime} := \mathbf{r}_{u,s}^{(j)} - \sum_{x=(u-1)l+1}^{ul} \mathbf{r}_{x,s-1}^{(j)}$
    - $v_{u,s}^{(j)} := \sum_{x=(u-1)l+1}^{ul} v_{x,s-1}^{(j)}$
    - $\mathbf{t}_{u,s}^{(j)} := \mathbf{Com}\left(v_{u,s}^{(j)}; \mathbf{r}_{u,s}^{(j)}\right)$
    - $\mathbf{t}_{u,s}^{(j)\prime} := \mathbf{t}_{u,s}^{(j)} - \sum_{x=(u-1)l+1}^{ul} \mathbf{t}_{x,s-1}^{(j)}$
- $\pi^{A,(j)} = \Pi_{Auth1}\left(\mathbf{A}, \mathbf{t}_{u,s}^{(j)}; \mathbf{r}_{u,s}^{(j)}\right) \begin{cases} s \in (0, \ldots, \lceil \log_l(N_V) \rceil) \\ u \in (1, \ldots, \lceil N_V / l^s \rceil) \end{cases}$
- $\pi^{A,(j)\prime} = \Pi_{Auth2}\left(\mathbf{A}, \mathbf{t}_{u,s}^{(j)\prime}, \mathbf{r}_{u,s}^{(j)\prime}\right) \begin{cases} s \in (1, \ldots, \lceil \log_l(N_V) \rceil) \\ u \in (1, \ldots, \lceil N_V / l^s \rceil) \end{cases}$
- $\mathbf{t}_{Tot}^{(j)} = \left(\mathbf{t}_{u,s}^{(j)}\right) \begin{cases} s \in (1, \ldots, \lceil \log_l(N_V) \rceil) \\ u \in (1, \ldots, \lceil N_V / l^s \rceil) \end{cases}$
- Sign and publish $\pi^{A,(j)}, \pi^{A,(j)\prime}, \mathbf{t}_{Tot}^{(j)}, \mathbf{r}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ on the bulletin board

To compute the total tally one only needs to open the commitments $\mathbf{t}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ for each $j \in [N_A]$ as these are commitments to the partial tallies of each authority.
Tally$(par, pk_1, \ldots, pk_{N_A}, BB)$

- For $j \in [N_A]$ recover $\mathbf{t}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ and $\mathbf{r}_{1,\lceil \log_l(N_V) \rceil}^{(j)}$ on BB
- $\forall j \in [N_A]\,,\ t^{(j)} := \mathbf{Open}(\mathbf{t}_{1,\lceil \log_l(N_V) \rceil}^{(j)}, \mathbf{r}_{1,\lceil \log_l(N_V) \rceil}^{(j)}, 1)$
- $t := \sum_1^{N_A} t^{(j)}$
- publish t

To verify the election one needs to verify the proof of each user, the proof of correct opening of each $\mathbf{t}_{u,s}^{(j)}$, and the proof that $\mathbf{t}_{u,s}^{(j)\prime}$ opens to zero. The verifier can then recompute the tally and check that it has been done correctly.
Verify$(par, pk_1, \ldots, pk_{N_A}, BB, t)$

- For each $i \in [N_V]$, recover π_i^V on BB.
- $\forall i$, verify π_i^V
- $\forall j \in [N_A]\,,\ \forall (s,u) \in (1,\dots,\lceil \log_l(N_V)\rceil) \times (1,\dots,\lceil N_V/l^s\rceil)$ recover $\pi_{u,s}^{A,(j)}$ and $\pi_{u,s}^{A,(j)'}$ from BB
- $\forall j,u,s$ verify $\pi_{u,s}^{A,(j)}$ and $\pi_{u,s}^{A,(j)'}$
- For $j \in [N_A]$ recover $\mathbf{t}^{(j)}_{1,\lceil \log_l(N_V)\rceil}$ and $\mathbf{r}^{(j)}_{1,\lceil \log_l(N_V)\rceil}$ on BB
- $\forall j \in [N_A]\,,\ t^{(j)} := \mathbf{Open}(\mathbf{t}^{(j)}_{1,\lceil \log_l(N_V)\rceil}, \mathbf{r}^{(j)}_{1,\lceil \log_l(N_V)\rceil}, 1)$
- Check that $t = \sum_1^{N_A} t^{(j)}$

We prove privacy and consistency in Section 5.6 and we discuss how to set the parameters in Section 5.5.

5.5 Parameters

In this section we review the bounds imposed on the parameters of our scheme by the correctness and security of the vote, and we propose concrete parameters in Table 5.2 as well as benchmarks from our implementation of the scheme.
The correctness and security of our scheme impose the following bounds on the parameters:

- Overwhelming soundness of π^V: $\binom{d}{B_{\mathcal{C}}} 2^{B_{\mathcal{C}}} > 2^{256}$
- Correctness of π^V: $B_V \geq N_A\sqrt{2d(2n+1)}\sigma$
- Zero-knowledge of π^V: $B'_V \geq 11 B_{\mathcal{C}}\sqrt{2d(2n+1)}B_V$
- Amortization of π^A and $\pi^{A'}$: $k > (\lambda+2)/\log(2d+1)$
- Correctness of π^A: $B_{A1} \geq (\sqrt{2d(2n+1)} + \sqrt{\ell})\sigma$
- Zero-knowledge of π^A: $B'_{A1} \geq 11\sqrt{2d\ell k(2n+1)}B_{A1}$
- Correctness of $\pi^{A'}$: $B_{A2} \geq (l+1)(\sqrt{2d(2n+1)} + \sqrt{\ell})\sigma$
- Zero-knowledge of $\pi^{A'}$: $B'_{A2} \geq 11\sqrt{2d\ell k(2n+1)}B_{A2}$
- Consistency of the vote (equation (1)): $2B'_V \leq B_{Com}$
- Consistency of the vote (equation (4)): $2B_{\mathcal{C}}N_A B'_{A1} \leq B_{Com}$
- Consistency of the vote (equation (7)): $2d(l+1)B'_{A1} \leq B_{Com}$
- Consistency of the vote (equation (8)): $2dB'_{A2} \leq B_{Com}$

Parameter	Notation	Value
Ring dimension	d	256
Modulus	q	$2^{31} - 2^7 - 2^5 + 1$
Module size	n	5
Commitment std deviation	σ	1
Number of voters	N_V	arbitrary
Number of authorities	N_A	4
Number of candidates	N_C	2
"Bucket" size	l	10
Amortization	ℓ	1000

Table 5.2: A possible set of parameters for our E-Voting scheme. These parameters achieve a post-quantum security of 119 bits in time and 93 bits in space.

Using the security analysis of Section 5.6, and the cryptanalysis of 2.3.4 to assess the hardness of M-LWE$_{q,n,\sigma}$ and M-SIS$_{q,2n+1n,B_{Com}}$ we set our parameters as in Table 5.2 (we arbitrarily fix the number of authorities to 4, anything larger than 2 is enough for security and this does not impact performance significantly). For this set of parameters both the M-LWE and M-SIS achieve a block size of $\beta = 450$ which corresponds to a security of 93 bits in space, 131 bits in time, and 119 bits in time for post-quantum security. Due to the improved E-voting scheme of Section 5.4.3, the number of voters does not affect the security at all and can thus be taken arbitrarily large.

We have implemented the complete voting scheme in C. The main computational problems in the scheme are the sampling of discrete Gaussian vectors and multiplication of polynomials in $\mathcal{R}_q$. For the sampling we have implemented a two-stage Knuth-Yao sampler. We have taken great care to ensure that the statistical distance between the sampled vectors and the exact discrete distribution is below 2^{-100}. This required computing the probabilities and the lookup table for the sampler with a multiprecision library. We used pari [PAR16] for this task. For the fastest possible multiplication in rings of the given form, one usually chooses the prime q in such a way that $\mathbb{Z}_q$ contains a $2d$-th root of unity. This then implies that the modulus $X^d + 1$ splits into linear factors over $\mathbb{Z}_q$ and allows for using an NTT-based multiplication algorithm. Unfortunately, the security requirements of our scheme prevent q from being chosen in this way. Instead of completely resorting to a general algorithm that works for multiplication modulo arbitrary polynomials, we have exploited the fact that for our prime q, $X^n + 1$ does in fact split into 16 factors. This allowed us to use a general multiplication algorithm only after 4 stages of NTT. We have used our own NTT implementation and the highly optimized FLINT library [HJP13] for the base case multiplication. FLINT uses a variant of Kronecker substitution for this task.

5.6 Security Analysis of the Voting Scheme

In this section we prove the privacy and consistency of our E-voting scheme as defined in Section 5.2. For privacy we consider the following advantages for an adversary $\mathcal{A}$:

- $\mathsf{Adv}^{CCA}_{\mathcal{A}}(\lambda)$ the advantage of $\mathcal{A}$ in the CCA security game of the encryption scheme.

- $\mathsf{Adv}_{\mathcal{A}}^{Hid}(\lambda)$ the advantage of $\mathcal{A}$ over the Hiding property of the commitment scheme.

Since the zero-knowledge of both the OR-Proof and the amortized proof are statistical, the probability of distinguishing between the simulator and the actual proof is less than $2^{-\lambda}$.

Theorem 5.6.1. *The advantage of any PPT adversary $\mathcal{A}$ over the privacy of our E-voting scheme is at most:*

$$\mathsf{Adv}_{\mathcal{A}}^{priv}(\lambda) \leq N_V \left(2\mathsf{Adv}_{\mathcal{A}}^{CCA}(\lambda) + \frac{l}{l-1}\mathsf{Adv}_{\mathcal{A}}^{Hid}(\lambda) + 2^{-\lambda+1}\right) + 2^{-\lambda+2}$$

Proof. We use a game based proof:
Game $\mathbf{G}_0$: In this game we run $Exp_{\mathcal{A}}^{priv,0}$ as defined in Section 5.4.1. The voting, casting and tallying oracle are run honestly by the simulator using choice bit $\beta = 0$ and thus votes $v_{0,i}$ for $i \in [N_V]$.
Game $\mathbf{G}_{1,i\leq N_V}$: In this game we modify the honest voting oracle $\mathcal{O}\mathbf{Vote}(id, v_0, v_1)$ so that when it runs $\mathbf{Vote}(par, pk_1, \ldots, pk_{N_A}, id_i, v_0)$, the OR-proof for $\mathbf{t}_i = Com(v_i; \mathbf{r}_i)$ is not done honestly but simulated. Note that when simulated the proof is independent of the randomness $\mathbf{r}_i$ and vote $v_{0,i}$. The advantage of the adversary in distinguishing between **Game** $\mathbf{G}_{1,i-1}$ and **Game** $\mathbf{G}_{1,i}$ (where we consider **Game** $\mathbf{G}_0$ as **Game** $\mathbf{G}_{1,-1}$) is zero if **Vote** is never called on id_i (i.e. id_i corresponds to a corrupted voter) and $2^{-\lambda}$ otherwise.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{1,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{1,i-1}}\right| \leq 2^{-\lambda}$$

Game $\mathbf{G}_2$: In this game we modify the tallying oracle of the first authority (the honest one) to make $\pi^{A,(1)}$ independent of the decrypted randomnesses $(\mathbf{r}_i^{(1)})_{i\in[N_V]}$. i.e. we modify the $\mathbf{Tally}_1(par, pk_1, \ldots, pk_{N_A}, BB, sk_1)$ oracle so that the proof $\pi^{A,(1)}$ is computed using the simulator of the amortized proof.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_2} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{1,N_V}}\right| \leq 2^{-\lambda}$$

Game $\mathbf{G}_3$: In this game we modify the $\mathbf{Tally}_1(par, pk_1, \ldots, pk_{N_A}, BB, sk_1)$ oracle so that the proof $\pi^{A,(1)'}$ is computed using the simulator of the amortized proof.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_3} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_2}\right| \leq 2^{-\lambda}$$

Game $\mathbf{G}_{4,i\leq N_V}$: In this game we modify the voting oracle for identity id_i so that it outputs the encryption $\mathbf{e}_i^{(1)} := \mathbf{Enc}(0, pk_1)$ instead of $\mathbf{e}_i^{(1)} := \mathbf{Enc}(\mathbf{r}_i^{(1)}, pk_1)$. The simulator also modifies the $\mathbf{Tally}_1$ oracle so that it uses $\mathbf{r}_i^{(1)}$ without decrypting $\mathbf{e}_i^{(1)}$.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{4,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{4,i-1}}\right| \leq \mathsf{Adv}_{\mathcal{A}}^{CCA}(\lambda)$$

Game $\mathbf{G}_{5,i\leq N_V}$ At this point all the values published by the oracles $\mathcal{O}\mathbf{Vote}$ and $\mathcal{O}\mathbf{Tally}_1$ are independent of the votes $(v_{0,i})_{i\in[N_V]}$ except for the commitments output by $\mathcal{O}\mathbf{Vote}$. We would like to use the hiding property of the commitment to change $\mathbf{t}_i^{(1)} = \mathbf{Com}(v_i^{(1)}, \mathbf{r}_i^{(1)})$ to $\mathbf{t}_i^{(1)} = \mathbf{Com}(v_i^{(1)} + v_{1,i} - v_{0,i}, \mathbf{r}_i^{(1)})$. Doing so implies that the commitment $\mathbf{t}_{u,s}^{(1)'}$ for $u = \lceil i/l \rceil$ and $s = 1$ will no longer be a commitment of zero but a commitment of $v_{1,i} - v_{0,i}$. This does

not matter since the proof $\pi^{A,(1)'}$ is now simulated and thus independent of the existence of a witness that $\mathbf{t}_{u,s}^{(1)'}$ commits to zero.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{5,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{5,i-1}}\right| \leq \mathsf{Adv}_{\mathcal{A}}^{Hid}(\lambda)$$

Game $\mathbf{G}_{6,1\leq s\leq\lceil\log_l(N_V)\rceil-1,1\leq u\leq\lceil N_V/l^s\rceil}$: Now that the votes have been changed from $v_{0,i}$ to $v_{1,i}$ we need to change the values of the commitments of the partial sums in order for $\mathbf{t}_{u,s}^{(1)'}$ to be commitments to zero, this will be needed to change $\pi^{A,(1)'}$ back to an honest proof. To do so we let $v_{0,u,s} = \sum_{x=(u-1)l+1}^{ul} v_{0,x,s-1}^{(j)}$ and $v_{1,u,s} = \sum_{x=(u-1)l+1}^{ul} v_{1,x,s-1}^{(j)}$ (where $v_{0,i,0} = v_{0,i}$ and $v_{1,i,0} = v_{1,i}$). We can now change the commitments $\mathbf{t}_{u,s}^{(j)} = \mathbf{Com}\left(v_{u,s}^{(j)}; \mathbf{r}_{u,s}^{(j)}\right)$ to $\mathbf{t}_{u,s}^{(j)} = \mathbf{Com}\left(v_{u,s}^{(j)} + v_{1,u,s} - v_{0,u,s}; \mathbf{r}_{u,s}^{(j)}\right)$ and the partial sums are verified.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{6,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{6,i-1}}\right| \leq \mathsf{Adv}_{\mathcal{A}}^{Hid}(\lambda)$$

Game $\mathbf{G}_{7,i\leq N_V}$: We revert the randomness encryptions to $\mathbf{e}_i^{(1)} = \mathbf{Enc}(\mathbf{r_i^{(1)}})$, this modification is consistent with the tallying scheme as the randomness used have not been modified.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{7,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{7,i-1}}\right| \leq \mathsf{Adv}_{\mathcal{A}}^{CCA}(\lambda)$$

Game $\mathbf{G}_8$: We compute the proof $\pi^{A,(1)'}$ honestly. This is possible because all commitments $\mathbf{t}_{u,s}^{(1)'}$ are commitments of zero made with the appropriate randomnesses.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_8} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{7,N_V}}\right| \leq 2^{-\lambda}$$

Game $\mathbf{G}_9$: Similarly we compute the proof $\pi^{A,(1)}$ honestly.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_9} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_8}\right| \leq 2^{-\lambda}$$

Game $\mathbf{G}_{10,i\leq N_V}$: We compute the proof π_i honestly. This is possible because $\mathbf{t}_i$ is still a commitment to either zero or one with the same randomness as before.

$$\left|\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{10,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{10,i-1}}\right| \leq 2^{-\lambda}$$

Game $\mathbf{G}_{11}$ We run $Exp_{\mathcal{A}}^{priv,1}$, this game is identical to **Game** $\mathbf{G}_{10,N_V}$.

$$\mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{11}} = \mathsf{Adv}_{\mathcal{A}}^{\mathbf{G}_{10,N_V}}$$

□

We now consider the consistency of our voting scheme. For a tighter security proof we will assume a slight modification on the algorithm $\mathbf{Tally}_j$: Rather than using only the hash of the corresponding commitments to compute the challenges for the proofs $\pi^{A,(j)}$ and $\pi^{A,(j)'}$, the authorities will hash the whole bulletin board (which among other things contains the relevant commitments). Let $H(BB) = (chl_1, \ldots, chl_{N_A})$ and $H'(BB) = (chl'_1, \ldots, chl'_{N_A})$ be these hashes, the j^{th} authority will then use chl_j and chl'_j as challenges for his proofs $\pi^{A,(j)}$ and $\pi^{A,(j)'}$. In doing so we guarantee that we can extract witnesses for all N_A proofs $\pi^{A,(j)}$ in one rewinding of the random oracle $\mathcal{O}_H$.

Theorem 5.6.2. *Let $\mathcal{A}$ be an adversary with non negligible advantage $\mathsf{Adv}^{cons}_{\mathcal{A}}(\lambda) = \varepsilon$ in experiment $\mathsf{Exp}^{cons}_{\mathcal{A}}(\lambda)$. Using $\mathcal{A}$ we construct an extractor $\mathcal{E}$ who breaks the binding property of* **Com** *in expected time $1/\varepsilon + negl(\lambda)$*

Proof. We will assume that when $\mathcal{A}$ succeeds in $\mathsf{Exp}^{cons}_{\mathcal{A}}(\lambda)$ all the bulletins on BB were accepted by all the authorities, we can make this assumption because any bulletin that was not accepted is effectively discarded (the authorities do not include it in their amortized proofs nor in the final tally). We can thus use N_V as the number of accepted tallies. $\mathcal{E}$ starts by running $\mathcal{A}$ until it succeeds in $\mathsf{Exp}^{cons}_{\mathcal{A}}(\lambda)$, i.e. until it outputs a bulletin board BB that verifies correctly and such that $\exists k \in N_C, r[k] < h_{1,k}$ or $r[k] > N_V - h_{0,k}$. Using the soundness of the proofs $\pi^{A,(j)}$, $\mathcal{E}$ rewinds $\mathcal{A}$ and obtains witnesses $\hat{\mathbf{r}}_i^{(j)}$ and messages $\hat{v}_i^{(j)}$, for $i \leq N_V$ and $j \leq N_A$, such that:

$$\mathbf{t}_i^{(j)} = \mathbf{Com}(\hat{v}_i^{(j)}; \hat{\mathbf{r}}_i^{(j)}) \tag{1}$$

Suppose there exists $i \leq N_V$ such that $\hat{v}_i := \sum_{j=1}^{N_A} \hat{v}_i^{(j)} \mod q$ is such that $\phi(\hat{v}_i) \notin \{0,1\}^{N_C}$, $\mathcal{E}$ runs the soundness extractor for π_i^V and obtains $\bar{\mathbf{r}}_i$, $\bar{c} \in \mathcal{R}$ and $\bar{v}_i \in \{0,1\}$ such that:

$$\bar{c}\mathbf{t}_i = \mathbf{Com}(\bar{c}\bar{v}_i, \bar{\mathbf{r}}_i) \tag{2}$$

By summing equations (1) over $j \in [N_A]$ and multiplying them by $\bar{c}$, we obtain the following:

$$\bar{c}\mathbf{t}_i = \mathbf{Com}(\bar{c}\hat{v}_i; \bar{c}\sum_{j=1}^{N_A} \hat{\mathbf{r}}_i^{(j)}) \tag{4}$$

Since we assumed that $\phi(\hat{v}_i) \notin \{0,1\}^{N_C}$ and we know $\phi(\bar{v}_i) \in \{0,1\}^{N_C}$, if we have $\|\bar{\mathbf{r}}_i\| \leq B_{Com}$ and $\left\|\bar{c}\sum_{j=1}^{N_A} \hat{\mathbf{r}}_i^{(j)}\right\| \leq B_{Com}$ (which we will ensure in Section 5.5), then $\mathcal{E}$ has successfully opened $\bar{c}\mathbf{t}_i$ to two different messages and thus broken the binding property of **Com**.
We can now assume that for every $i \leq N_V$, $\phi(\hat{v}_i) \in \{0,1\}^{N_C}$. For $s \in [\lceil \log_l N_V \rceil]$, $u \in [\lceil N_V/l^s \rceil]$ and $j \in [N_A]$, let $\hat{v}_{u,s}^{(j)} = \sum_{x=(u-1)l+1}^{ul} \hat{v}_{x,s-1}^{(j)}$ (where $\hat{v}_{u,0}^{(j)} := \hat{v}_u^{(j)}$), let $\mathbf{t}_{u,s}^{(j)\prime} = \mathbf{t}_{u,s}^{(j)} - \sum_{x=(u-1)l+1}^{ul} \mathbf{t}_{x,s-1}^{(j)}$. $\mathcal{E}$ runs the soundness extractor for $\pi^{A,(j)\prime}$ and obtains $\hat{\mathbf{r}}_{u,s}^{(j)\prime}$ such that $\mathbf{t}_{u,s}^{(j)\prime} = \mathbf{Com}(0, \hat{\mathbf{r}}_{u,s}^{(j)\prime})$.
Using the extraction for $\pi^{A,(j)}$ we already have $\hat{\mathbf{r}}_{u,s}^{(j)}$ and messages $m_{u,s}^{(j)}$ such that:

$$\mathbf{t}_{u,s}^{(j)} = \mathbf{Com}(m_{u,s}^{(j)}; \hat{\mathbf{r}}_{u,s}^{(j)}) \tag{5}$$

Now suppose that for all u, s, j we have $m_{u,s}^{(j)} = \hat{v}_{u,s}^{(j)}$ this implies that the ciphertext $\mathbf{t}_{1,\lceil \log_l N_V \rceil}^{(j)}$ has an extraction:

$$\mathbf{t}_{1,\lceil \log_l N_V \rceil}^{(j)} = \mathbf{Com}(\hat{v}_{1,\lceil \log_l N_V \rceil}^{(j)}; \hat{\mathbf{r}}_{1,\lceil \log_l N_V \rceil}^{(j)}) \tag{6}$$

By construction of $\hat{v}_{u,s}^{(j)}$ we have $\hat{v}_{1,\lceil \log_l N_V \rceil}^{(j)} = \sum_{i=1}^{N_V} \hat{v}_i^{(j)}$. Since the bulletin board verifies correctly we know that $\mathbf{t}_{1,\lceil \log_l N_V \rceil}^{(j)}$ opens to plaintext $v^{(j)}$ such that $r = \sum_{j=1}^{N_V} v^{(j)}$ by the

binding property of **Com** we have that $v^{(j)} = \sum_{i=1}^{N_V} \hat{v}_i^{(j)}$ and thus:

$$\begin{aligned} r &= \sum_j v^{(j)} \\ &= \sum_j \sum_i \hat{v}_i^{(j)} \\ &= \sum_i \hat{v}_i \\ &= \sum_{i \in HV'} v_i + \sum_{i \in CV'} \hat{v}_i \\ &= h_1 + \sum_{i \in CV'} \hat{v}_i \end{aligned}$$

Since we have shown that for all $i \leq N_V$, $\phi(\hat{v}_i) \in \{0,1\}^{N_C}$ this implies that $h_{1,k} \leq r[k] \leq N_V - h_{0,k}$ which contradicts the fact that $\mathcal{A}$ wins experiment $\mathsf{Exp}_{\mathcal{A}}^{\text{cons}}(\lambda)$. We have thus shown that there exist u, v, j such that $m_{u,s}^{(j)} \neq \hat{v}_{u,s}^{(j)}$, i.e. one of the partial sum does not commit to the proper value.
Fix a $j \leq N_A$ for which there exist such a commitment and let u, s be the smallest such triple (in lexicographic order). In particular this implies that $s \geq 1$ (as we have proven that all $\mathbf{t}_i^{(j)}$ open to $\hat{v}_i^{(j)}$) and that for $x \in ((u-1)l+1, ul)$ we have the following witness extracted from $\pi^{A,(j)}$:

$$\mathbf{t}_{x,s-1}^{(j)} = \mathbf{Com}(\hat{v}_{x,s-1}^{(j)}, \hat{\mathbf{r}}_{x,s-1}^{(j)}) \tag{7}$$

By summing equation (7) over $x \in ((u-1)l+1, ul)$ and subtracting the extraction for $\mathbf{t}_{u,s}^{(j)}$ we obtain:

$$\begin{aligned} \mathbf{t}_{u,s}^{(j)\prime} &= \mathbf{t}_{u,s}^{(j)} - \sum_{x=(u-1)l+1}^{ul} \mathbf{t}_{x,s-1}^{(j)} \\ &= \mathbf{Com}\left(m_{u,s}^{(j)} - \hat{v}_{u,s}^{(j)}; \hat{\mathbf{r}}_{u,s}^{(j)} - \sum_{x=(u-1)l+1}^{ul} \hat{\mathbf{r}}_{x,s-1}^{(j)} \right) \end{aligned} \tag{8}$$

From the extraction of $\pi^{A,(j)\prime}$ we had $\mathbf{t}_{u,s}^{(j)\prime} = \mathbf{Com}(0, \hat{\mathbf{r}}_{u,s}^{(j)\prime})$. We know that $m_{u,s}^{(j)} \neq \hat{v}_{u,s}^{(j)}$, which implies that if $\left\| \hat{\mathbf{r}}_{u,s}^{(j)} - \sum_{x=(u-1)l+1}^{ul} \hat{\mathbf{r}}_{x,s-1}^{(j)} \right\| \leq B_{Com}$ and $\left\| \hat{\mathbf{r}}_{u,s}^{(j)\prime} \right\| \leq B_{Com}$ (which we ensure in Section 5.5) then we have found two distinct openings for $\mathbf{t}_{u,s}^{(j)\prime}$ and broken the binding property of **Com**. □

Chapter 6

Conclusion and Open Questions

6.1 Conclusion

Throughout this thesis we have studied the "Fiat-Shamir with Aborts" technique for constructing lattice-based zero-knowledge proofs of knowledge. We have put forward new constructions which increase the expressiveness as well as the practicality of proofs of knowledge. We have used these protocols as building blocks to create efficient lattice-based privacy protocols.

We proposed proofs of knowledge for the disjunction of NP Languages over lattices, bridging the gap with similar number theoretic constructions. To do so we observed that well chosen challenge spaces can be endowed with an "ad-hoc" group law, allowing for generic OR-Proofs constructions to be applied.

We constructed proofs of knowledge for subset membership for commitments, i.e. proofs that a commitment opens to a message in a fixed subset of the message space. To do so we studied the properties of the automorphisms of polynomial rings and have shown that the so-called Galois automorphisms can be used in conjunction with zero-knowledge proofs in order to show that committed message belong to subrings of variable dimensions. Through this use of the structure of polynomial rings we obtained proofs of subset membership with constant proof size, and maybe more importantly for privacy-preserving applications: these proofs use subsets containing messages that have inverse in the ring.

Another approach taken was to improve the state of the art in amortized zero-knowledge. While previous works already achieved exact amortized zero-knowledge proofs with constant overhead and polynomial slack, they required either to be amortized over more than thousands of equations or were very intensive in running time both for the prover and verifier. We presented a new amortized proof that has substantially the same overhead and slack but can be used with as few as 12 equations. Surprisingly the protocol we presented was conceptually much simpler than previous works.

Building upon our proofs for subset membership we constructed a group signature with size independent of the group size. While such a result already existed, previous constructions were not practical as they required the repetition of a proof of knowledge with constant soundness. In contrast our construction focuses on efficiency and obtains signatures that are barely larger than $500KB$.

Lastly we proposed what is to our knowledge the first efficient lattice-based e-voting scheme. Building upon our OR-proofs and amortized proofs we constructed a voting scheme that boasts small ballot size, around $30KB$, and very efficient complexity with algorithms that

run in a matter of milliseconds.

6.2 Open Questions

Question 6.1. *Can we obtain exact proofs of knowledge with overwhelming soundness without any repetition or amortization?*

Such a proof of knowledge would be the holy grail of lattice-based zero knowledge. However obtaining overwhelming soundness requires a large challenge space, and it is difficult to imagine how to obtain such a challenge space without either increasing the number of equations proven or obtaining a knowledge extractor whose output depends on the challenges.

Question 6.2. *Can we obtain proofs without slack?*

While the presence of polynomial slack in lattice-based zero knowledge is not terribly inconvenient, having proofs without slack would allow for smaller parameters and more efficient constructions. An idea towards this goal would be the following: Given $\mathbf{As} = \mathbf{t}$ with $\mathbf{s}$ a binary vector, compute a regular lattice based zero-knowledge proof (possible approximate or amortized) with polynomial slack, and add an extra proof that that $\mathbf{s}$ is binary. Taking inspiration from proofs for arithmetic circuits on could sample a random vector $\mathbf{v}$ and prove that $\langle \mathbf{v} \circ \mathbf{s}, \mathbf{1} - \mathbf{s} \rangle = 0$

Question 6.3. *Can we obtain smaller group sizes for our group signature?*

The way we construct the group in our group signature inherently forces the group size to be at least q_2. We could maybe obtain a smaller group size if on top of showing that id is invariant under a set of automorphism we also showed that it+is small.

Question 6.4. *How can we adapt our e-voting scheme to different types of election?*

For now our e-voting scheme computes the result of an election as the sum of votes in $\{0,1\}^{N_C}$, however we could consider different types or votes or tallies. For example we could prove that voter voted for exactly one candidate by doing an or proof over the possible votes $(1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, etc... However if we want elections where voters vote for exactly two candidates the same method will give proofs of size quadratic in the number of candidates. We could search for better ways to encode votes or prove properties about them.

Notation

General Mathematical Notations

$:=$	"Is defined as"
$\mathbb{N}$	The set of natural numbers
$\mathbb{Z}$	The set of integers
$\mathbb{Q}$	The set of rationals
$\mathbb{R}$	The set of reals
log	Base 2 logarithm
$\mathbf{v}$	A (column) vector
$\mathbf{M}$	A matrix
$\mathcal{R}$	The ring $\mathbb{Z}[X]/X^d + 1$
$\langle \cdot, \cdot \rangle$	The inner-product
$\lfloor \cdot \rceil$	The rounding function
$\lvert S \rvert$	Size of a set
$\lVert \cdots \rVert_{p,\infty,\max}$	Norms
$s_(\mathbf{M})$	The matrix operator norm
$D_{\mathcal{R},\sigma}$	A discrete gaussian
o, O, ω, Ω	Asymptotic notations
σ	A standard deviation
ξ	An automorphism of $\mathcal{R}$
δ_0	The root-Hermite Factor
β	A block-size
$\mathcal{C}$	A challenge set
$\bar{\mathcal{C}}$	Set of differences (excluding 0) of $\mathcal{C}$
q	A prime modulus
λ	The security parameter

Notations Specific to Chapter 2

ρ_σ	A continuous Gaussian
$\mathcal{L}$	A lattice
$\lambda_i(\mathcal{L})$	The i^{th} minimum of $\mathcal{L}$
$\det(\mathcal{L})$	The determinant of $\mathcal{L}$
$\mathrm{Vol}(\mathcal{L})$	The volume of $\mathcal{L}$
$\bar{B}$	A closed ball

Notations Specific to Chapter 3

$\mathcal{P}$	A prover
$\mathcal{V}$	A verifier
$\mathfrak{R}$	A binary relation
$\mathfrak{L}$	An NP Language

Abbreviations

Primitives	
AKE	Authenticated Key Exchange
KEM	Key Encapsulation Mechanism
PKE	Public Key Encryption
ZKPoK	Zero-Knowledge Proof of Knowledge
Algorithms	
BKZ	Block Korkin-Zolotarev
GPV	Gentry-Peikert-Vaikuntanathan
GSO	Gram-Schmidt Orthogonalization
LLL	Lenstra-Lenstra-Lovász
Computational Problems	
BDD	Bounded Distance Decoding
CVP	Closest Vector Problem
ISIS	Inhomogeneous Short Integer Solution
LWE	Learning With Errors
SVP	Shortest Vector Problem
SIS	Short Integer Solution

List of Illustrations

Figures

Tables

Personal Publications

[BBC+18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. "Sub-Linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits". In: *In submission* (2018) (cit. on p. 4).

[BPMW16] Florian Bourse, Rafaël del Pino, Michele Minelli, and Hoeteck Wee. "FHE Circuit Privacy Almost for Free". In: *CRYPTO 2016, Part II*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9815. LNCS. Springer, Heidelberg, Aug. 2016, pp. 62–89. DOI: 10.1007/978-3-662-53008-5_3 (cit. on p. 4).

[PL17] Rafaël del Pino and Vadim Lyubashevsky. "Amortization with Fewer Equations for Proving Knowledge of Small Secrets". In: *CRYPTO 2017, Part III*. Ed. by Jonathan Katz and Hovav Shacham. Vol. 10403. LNCS. Springer, Heidelberg, Aug. 2017, pp. 365–394 (cit. on pp. 3, 4, 38, 66).

[PLNS17] Rafaël del Pino, Vadim Lyubashevsky, Gregory Neven, and Gregor Seiler. "Practical Quantum-Safe Voting from Lattices". In: *ACM CCS 17*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. ACM Press, Oct. 2017, pp. 1565–1581 (cit. on p. 3).

[PLP16] Rafaël del Pino, Vadim Lyubashevsky, and David Pointcheval. "The Whole is Less Than the Sum of Its Parts: Constructing More Efficient Lattice-Based AKEs". In: *SCN 16*. Ed. by Vassilis Zikas and Roberto De Prisco. Vol. 9841. LNCS. Springer, Heidelberg, Aug. 2016, pp. 273–291. DOI: 10.1007/978-3-319-44618-9_15 (cit. on p. 4).

[PLS18] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. "Zero-Knowledge Proofs of Automorphism Stability and Applications to Lattice-Based Privacy Protocols". In: *In submission* (2018) (cit. on pp. 3, 34, 35).

Bibliography

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. "Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE". In: *CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. LNCS. Springer, Heidelberg, Aug. 2010, pp. 98–115 (cit. on pp. 4, 44, 45, 47).

[Adi08] Ben Adida. "Helios: Web-based Open-Audit Voting". In: *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA*. 2008, pp. 335–348. URL: http://www.usenix.org/events/sec08/tech/full_papers/adida/adida.pdf (cit. on p. 66).

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. "Post-quantum Key Exchange - A New Hope". In: *USENIX*. 2016, pp. 327–343 (cit. on pp. 13, 15).

[AFLT12] Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi. "Tightly-Secure Signatures from Lossy Identification Schemes". In: *EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Heidelberg, Apr. 2012, pp. 572–590 (cit. on p. 68).

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *28th ACM STOC*. ACM Press, May 1996, pp. 99–108 (cit. on pp. 2, 12).

[Ajt99] Miklós Ajtai. "Generating Hard Instances of the Short Basis Problem". In: *ICALP 99*. Ed. by Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen. Vol. 1644. LNCS. Springer, Heidelberg, July 1999, pp. 1–9. DOI: 10.1007/3-540-48523-6_1 (cit. on p. 15).

[AP14] Jacob Alperin-Sheriff and Chris Peikert. "Faster Bootstrapping with Polynomial Error". In: *CRYPTO 2014, Part I*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8616. LNCS. Springer, Heidelberg, Aug. 2014, pp. 297–314. DOI: 10.1007/978-3-662-44371-2_17 (cit. on p. 4).

[BBC+18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. "Sub-Linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits". In: *In submission* (2018) (cit. on p. 4).

[BCG+15] David Bernhard, Véronique Cortier, David Galindo, Olivier Pereira, and Bogdan Warinschi. "SoK: A Comprehensive Analysis of Game-Based Ballot Privacy Definitions". In: *2015 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2015, pp. 499–516. DOI: 10.1109/SP.2015.37 (cit. on pp. 65, 76, 78).

[BCK+14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. "Better Zero-Knowledge Proofs for Lattice Encryption and Their Application to Group Signatures". In: *ASIACRYPT 2014, Part I*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8873. LNCS. Springer, Heidelberg, Dec. 2014, pp. 551–572. DOI: 10.1007/978-3-662-45611-8_29 (cit. on pp. 8, 25, 26).

[BCN17] Cecilia Boschini, Jan Camenisch, and Gregory Neven. "Relaxed Lattice-Based Signatures with Short Zero-Knowledge Proofs". In: *IACR Cryptology ePrint Archive* 2017 (2017), p. 1123. URL: http://eprint.iacr.org/2017/1123 (cit. on pp. 44, 45).

[BDGL16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. "New directions in nearest neighbor searching with applications to lattice sieving". In: *27th SODA*. Ed. by Robert Krauthgamer. ACM-SIAM, Jan. 2016, pp. 10–24. DOI: 10.1137/1.9781611974331.ch2 (cit. on p. 15).

[BDLN16] Carsten Baum, Ivan Damgård, Kasper Green Larsen, and Michael Nielsen. "How to Prove Knowledge of Small Secrets". In: *CRYPTO 2016, Part III*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9816. LNCS. Springer, Heidelberg, Aug. 2016, pp. 478–498. DOI: 10.1007/978-3-662-53015-3_17 (cit. on pp. 3, 4, 38, 66).

[BDOP16] Carsten Baum, Ivan Damgård, Sabine Oechsner, and Chris Peikert. *Efficient Commitments and Zero-Knowledge Protocols from Ring-SIS with Applications to Lattice-based Threshold Cryptosystems*. Cryptology ePrint Archive, Report 2016/997. http://eprint.iacr.org/2016/997. 2016 (cit. on pp. 3, 38).

[BG14] Shi Bai and Steven D. Galbraith. "An Improved Compression Technique for Signatures Based on Learning with Errors". In: *CT-RSA 2014*. Ed. by Josh Benaloh. Vol. 8366. LNCS. Springer, Heidelberg, Feb. 2014, pp. 28–47. DOI: 10.1007/978-3-319-04852-9_2 (cit. on pp. 3, 38).

[BG93] Mihir Bellare and Oded Goldreich. "On Defining Proofs of Knowledge". In: *CRYPTO'92*. Ed. by Ernest F. Brickell. Vol. 740. LNCS. Springer, Heidelberg, Aug. 1993, pp. 390–420 (cit. on p. 27).

[BKLP15] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. "Efficient Zero-Knowledge Proofs for Commitments from Learning with Errors over Rings". In: *ESORICS 2015, Part I*. Ed. by Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl. Vol. 9326. LNCS. Springer, Heidelberg, Sept. 2015, pp. 305–325. DOI: 10.1007/978-3-319-24174-6_16 (cit. on pp. 3, 38).

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. "Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions". In: *EUROCRYPT 2003*. Ed. by Eli Biham. Vol. 2656. LNCS. Springer, Heidelberg, May 2003, pp. 614–629 (cit. on p. 49).

[BN06] Mihir Bellare and Gregory Neven. "Multi-signatures in the plain public-Key model and a general forking lemma". In: *ACM CCS 06*. Ed. by Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati. ACM Press, Oct. 2006, pp. 390–399 (cit. on p. 63).

[Boy10] Xavier Boyen. "Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More". In: *PKC 2010*. Ed. by Phong Q. Nguyen and David Pointcheval. Vol. 6056. LNCS. Springer, Heidelberg, May 2010, pp. 499–517 (cit. on p. 44).

[BPMW16] Florian Bourse, Rafaël del Pino, Michele Minelli, and Hoeteck Wee. "FHE Circuit Privacy Almost for Free". In: *CRYPTO 2016, Part II*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9815. LNCS. Springer, Heidelberg, Aug. 2016, pp. 62–89. DOI: 10.1007/978-3-662-53008-5_3 (cit. on p. 4).

[CD09] Ronald Cramer and Ivan Damgård. "On the Amortized Complexity of Zero-Knowledge Protocols". In: *CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. LNCS. Springer, Heidelberg, Aug. 2009, pp. 177–191 (cit. on p. 38).

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols". In: *CRYPTO'94*. Ed. by Yvo Desmedt. Vol. 839. LNCS. Springer, Heidelberg, Aug. 1994, pp. 174–187 (cit. on pp. 67, 68).

[CDXY17] Ronald Cramer, Ivan Damgård, Chaoping Xing, and Chen Yuan. "Amortized Complexity of Zero-Knowledge Proofs Revisited: Achieving Linear Soundness Slack". In: *EUROCRYPT 2017, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Heidelberg, May 2017, pp. 479–500 (cit. on pp. 3, 4, 38, 66).

[CFSY96] Ronald Cramer, Matthew K. Franklin, Berry Schoenmakers, and Moti Yung. "Multi-Autority Secret-Ballot Elections with Linear Work". In: *EUROCRYPT'96*. Ed. by Ueli M. Maurer. Vol. 1070. LNCS. Springer, Heidelberg, May 1996, pp. 72–83 (cit. on p. 67).

[CGGI16] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. "A Homomorphic LWE Based E-voting Scheme". In: *PQCrypto*. Vol. 9606. Lecture Notes in Computer Science. Springer, 2016, pp. 245–265 (cit. on p. 66).

[CGK+16] Véronique Cortier, David Galindo, Ralf Küsters, Johannes Mueller, and Tomasz Truderung. "SoK: Verifiability Notions for E-Voting Protocols". In: *2016 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2016, pp. 779–798. DOI: 10.1109/SP.2016.52 (cit. on p. 78).

[CL03] Jan Camenisch and Anna Lysyanskaya. "A Signature Scheme with Efficient Protocols". In: *SCN 02*. Ed. by Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano. Vol. 2576. LNCS. Springer, Heidelberg, Sept. 2003, pp. 268–289 (cit. on p. 44).

[Cra97] Ronald Cramer. "Modular design of secure yet practical cryptographic protocols". 1997 (cit. on p. 26).

[Dam10] Ivan Damgård. *On Σ-protocols*. http://www.cs.au.dk/~ivan/Sigma.pdf. 2010 (cit. on pp. 27, 32, 38).

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. "Lattice Signatures and Bimodal Gaussians". In: *CRYPTO 2013, Part I*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Heidelberg, Aug. 2013, pp. 40–56. DOI: 10.1007/978-3-642-40041-4_3 (cit. on pp. 3, 38).

[DFS04] Ivan Damgård, Serge Fehr, and Louis Salvail. "Zero-Knowledge Proofs and String Commitments Withstanding Quantum Attacks". In: *CRYPTO 2004*. Ed. by Matthew Franklin. Vol. 3152. LNCS. Springer, Heidelberg, Aug. 2004, pp. 254–272 (cit. on p. 68).

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. "Efficient Identity-Based Encryption over NTRU Lattices". In: *ASIACRYPT 2014, Part II*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8874. LNCS. Springer, Heidelberg, Dec. 2014, pp. 22–41. DOI: 10.1007/978-3-662-45608-8_2 (cit. on p. 16).

[DM15] Léo Ducas and Daniele Micciancio. "FHEW: Bootstrapping Homomorphic Encryption in Less Than a Second". In: *EUROCRYPT 2015, Part I*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9056. LNCS. Springer, Heidelberg, Apr. 2015, pp. 617–640. DOI: 10.1007/978-3-662-46800-5_24 (cit. on p. 44).

[DPSZ12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. "Multiparty Computation from Somewhat Homomorphic Encryption". In: *CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. LNCS. Springer, Heidelberg, Aug. 2012, pp. 643–662 (cit. on pp. 3, 4, 38).

[FHF07] Ariel J. Feldman, J. Alex Halderman, and Edward W. Felten. "Security Analysis of the Diebold AccuVote-TS Voting Machine". In: *EVT*. USENIX Association, 2007 (cit. on p. 66).

[FS87] Amos Fiat and Adi Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *CRYPTO'86*. Ed. by Andrew M. Odlyzko. Vol. 263. LNCS. Springer, Heidelberg, Aug. 1987, pp. 186–194 (cit. on p. 27).

[Gen09] Craig Gentry. "Fully homomorphic encryption using ideal lattices". In: *41st ACM STOC*. Ed. by Michael Mitzenmacher. ACM Press, May 2009, pp. 169–178 (cit. on pp. 2, 66).

[GH07] Rop Gonggrijp and Willem-Jan Hengeveld. "Studying the Nedap/Groenendaal ES3B Voting Computer: A Computer Security Perspective". In: *2007 USENIX/ACCURATE Electronic Voting Technology Workshop, EVT'07, Boston, MA, USA, August 6, 2007*. 2007. URL: https://www.usenix.org/conference/evt-07/studying-nedapgroenendaal-es3b-voting-computer-computer-security-perspective (cit. on p. 66).

[GLP12] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. "Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems". In: *CHES 2012*. Ed. by Emmanuel Prouff and Patrick Schaumont. Vol. 7428. LNCS. Springer, Heidelberg, Sept. 2012, pp. 530–547 (cit. on pp. 3, 38).

[GN08] Nicolas Gama and Phong Q. Nguyen. "Predicting Lattice Reduction". In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Heidelberg, Apr. 2008, pp. 31–51 (cit. on p. 14).

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. “Trapdoors for hard lattices and new cryptographic constructions”. In: *40th ACM STOC*. Ed. by Richard E. Ladner and Cynthia Dwork. ACM Press, May 2008, pp. 197–206 (cit. on p. 15).

[HJP13] W. Hart, F. Johansson, and S. Pancratz. *FLINT: Fast Library for Number Theory*. Version 2.4.0, http://flintlib.org. 2013 (cit. on p. 85).

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. “NTRU: A Ring-Based Public Key Cryptosystem”. In: *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*. 1998, pp. 267–288. DOI: 10.1007/BFb0054868. URL: https://doi.org/10.1007/BFb0054868 (cit. on p. 2).

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. “Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems”. In: *ASIACRYPT 2008*. Ed. by Josef Pieprzyk. Vol. 5350. LNCS. Springer, Heidelberg, Dec. 2008, pp. 372–389 (cit. on p. 2).

[KY16] Shuichi Katsumata and Shota Yamada. “Partitioning via Non-linear Polynomial Functions: More Compact IBEs from Ideal Lattices and Bilinear Maps”. In: *ASIACRYPT 2016, Part II*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10032. LNCS. Springer, Heidelberg, Dec. 2016, pp. 682–712. DOI: 10.1007/978-3-662-53890-6_23 (cit. on p. 44).

[Laa15] Thijs Laarhoven. “Sieving for Shortest Vectors in Lattices Using Angular Locality-Sensitive Hashing”. In: *CRYPTO 2015, Part I*. Ed. by Rosario Gennaro and Matthew J. B. Robshaw. Vol. 9215. LNCS. Springer, Heidelberg, Aug. 2015, pp. 3–22. DOI: 10.1007/978-3-662-47989-6_1 (cit. on p. 15).

[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. “Factoring Polynomials with Rational Coefficients”. In: *Math. Ann.* 261 (1982), pp. 515–534 (cit. on p. 13).

[LLNW16] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. “Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures Without Trapdoors”. In: *EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Heidelberg, May 2016, pp. 1–31. DOI: 10.1007/978-3-662-49896-5_1 (cit. on pp. 44, 45).

[LM06] Vadim Lyubashevsky and Daniele Micciancio. “Generalized Compact Knapsacks Are Collision Resistant”. In: *ICALP 2006, Part II*. Ed. by Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener. Vol. 4052. LNCS. Springer, Heidelberg, July 2006, pp. 144–155 (cit. on p. 12).

[LMPR08] Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. “SWIFFT: A Modest Proposal for FFT Hashing”. In: *FSE 2008*. Ed. by Kaisa Nyberg. Vol. 5086. LNCS. Springer, Heidelberg, Feb. 2008, pp. 54–72 (cit. on p. 3).

[LN17] Vadim Lyubashevsky and Gregory Neven. “One-Shot Verifiable Encryption from Lattices”. In: *EUROCRYPT 2017, Part I*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10210. LNCS. Springer, Heidelberg, May 2017, pp. 293–323 (cit. on pp. 3, 38, 43, 45, 48, 50, 53, 55).

[LNSW13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. "Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications". In: *PKC 2013*. Ed. by Kaoru Kurosawa and Goichiro Hanaoka. Vol. 7778. LNCS. Springer, Heidelberg, Feb. 2013, pp. 107–124. DOI: 10.1007/978-3-642-36362-7_8 (cit. on p. 2).

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. LNCS. Springer, Heidelberg, May 2010, pp. 1–23 (cit. on pp. 2, 12).

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "A Toolkit for Ring-LWE Cryptography". In: *EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, Heidelberg, May 2013, pp. 35–54. DOI: 10.1007/978-3-642-38348-9_3 (cit. on p. 22).

[LS15] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptography* 75.3 (2015), pp. 565–599. DOI: 10.1007/s10623-014-9938-4. URL: https://doi.org/10.1007/s10623-014-9938-4 (cit. on p. 12).

[LS17] Vadim Lyubashevsky and Gregor Seiler. *Partially Splitting Rings for Faster Lattice-Based Zero-Knowledge Proofs*. Cryptology ePrint Archive, Report 2017/523. http://eprint.iacr.org/2017/523. 2017 (cit. on pp. 8, 9).

[Lyu08] Vadim Lyubashevsky. "Towards practical lattice-based cryptography". PhD thesis. UC San Diego, 2008 (cit. on p. 30).

[Lyu09] Vadim Lyubashevsky. "Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures". In: *ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Heidelberg, Dec. 2009, pp. 598–616 (cit. on pp. 3, 25).

[Lyu12] Vadim Lyubashevsky. "Lattice Signatures without Trapdoors". In: *EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Heidelberg, Apr. 2012, pp. 738–755 (cit. on pp. 3, 11, 66).

[Mic02] Daniele Micciancio. "Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions from Worst-Case Complexity Assumptions". In: *43rd FOCS*. IEEE Computer Society Press, Nov. 2002, pp. 356–365 (cit. on pp. 2, 12).

[MP12] Daniele Micciancio and Chris Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. LNCS. Springer, Heidelberg, Apr. 2012, pp. 700–718 (cit. on p. 15).

[NZZ15] Phong Q. Nguyen, Jiang Zhang, and Zhenfeng Zhang. "Simpler Efficient Group Signatures from Lattices". In: *PKC 2015*. Ed. by Jonathan Katz. Vol. 9020. LNCS. Springer, Heidelberg, Mar. 2015, pp. 401–426. DOI: 10.1007/978-3-662-46447-2_18 (cit. on pp. 44, 45).

[PAR16] PARI. *PARI/GP version 2.9.0*. available from http://pari.math.u-bordeaux.fr/. The PARI Group. Univ. Bordeaux, 2016 (cit. on p. 85).

[Pei10] Chris Peikert. “An Efficient and Parallel Gaussian Sampler for Lattices”. In: *CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. LNCS. Springer, Heidelberg, Aug. 2010, pp. 80–97 (cit. on p. 15).

[PL17] Rafaël del Pino and Vadim Lyubashevsky. “Amortization with Fewer Equations for Proving Knowledge of Small Secrets”. In: *CRYPTO 2017, Part III*. Ed. by Jonathan Katz and Hovav Shacham. Vol. 10403. LNCS. Springer, Heidelberg, Aug. 2017, pp. 365–394 (cit. on pp. 3, 4, 38, 66).

[PLNS17] Rafaël del Pino, Vadim Lyubashevsky, Gregory Neven, and Gregor Seiler. “Practical Quantum-Safe Voting from Lattices”. In: *ACM CCS 17*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. ACM Press, Oct. 2017, pp. 1565–1581 (cit. on p. 3).

[PLP16] Rafaël del Pino, Vadim Lyubashevsky, and David Pointcheval. “The Whole is Less Than the Sum of Its Parts: Constructing More Efficient Lattice-Based AKEs”. In: *SCN 16*. Ed. by Vassilis Zikas and Roberto De Prisco. Vol. 9841. LNCS. Springer, Heidelberg, Aug. 2016, pp. 273–291. DOI: 10.1007/978-3-319-44618-9_15 (cit. on p. 4).

[PLS18] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. “Zero-Knowledge Proofs of Automorphism Stability and Applications to Lattice-Based Privacy Protocols”. In: *In submission* (2018) (cit. on pp. 3, 34, 35).

[PR06] Chris Peikert and Alon Rosen. “Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices”. In: *TCC 2006*. Ed. by Shai Halevi and Tal Rabin. Vol. 3876. LNCS. Springer, Heidelberg, Mar. 2006, pp. 145–166 (cit. on p. 12).

[RD17] Scytl R&D. *Swiss Online Voting Protocol*. https://www.post.ch/-/media/post/evoting/dokumente/swiss-post-online-voting-protocol.pdf. 2017 (cit. on p. 66).

[Reg05] Oded Regev. “On lattices, learning with errors, random linear codes, and cryptography”. In: *37th ACM STOC*. Ed. by Harold N. Gabow and Ronald Fagin. ACM Press, May 2005, pp. 84–93 (cit. on pp. 2, 12, 22).

[Sch87] Claus-Peter Schnorr. “A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms”. In: *Theor. Comput. Sci.* 53 (1987), pp. 201–224 (cit. on p. 13).

[SS11] Damien Stehlé and Ron Steinfeld. “Making NTRU as Secure as Worst-Case Problems over Ideal Lattices”. In: *EUROCRYPT 2011*. Ed. by Kenneth G. Paterson. Vol. 6632. LNCS. Springer, Heidelberg, May 2011, pp. 27–47 (cit. on p. 12).

[Ste88] Jacques Stern. “A method for finding codewords of small weight”. In: *Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings*. 1988, pp. 106–113. DOI: 10.1007/BFb0019850. URL: https://doi.org/10.1007/BFb0019850 (cit. on p. 34).

[Ste94] Jacques Stern. “A New Identification Scheme Based on Syndrome Decoding”. In: *CRYPTO’93*. Ed. by Douglas R. Stinson. Vol. 773. LNCS. Springer, Heidelberg, Aug. 1994, pp. 13–21 (cit. on pp. 2, 66).

[Unr17] Dominique Unruh. “Post-quantum Security of Fiat-Shamir”. In: *ASIACRYPT 2017, Part I*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10624. LNCS. Springer, Heidelberg, Dec. 2017, pp. 65–95 (cit. on p. 68).

Résumé

Le chiffrement à base de réseaux euclidiens a connu un grand essor durant les vingt dernières années. Autant grâce à l'apparition de nouvelles primitives telles que le chiffrement complètement homomorphe, que grâce à l'amélioration des primitives existantes, comme le chiffrement á clef publique ou les signatures digitales, qui commencent désormais à rivaliser avec leurs homologues fondés sur la théorie des nombres. Cela dit les preuves à divulgation nulle de connaissance, bien qu'elles représentent un des piliers des protocols de confidentialité, n'ont pas autant progressé, que ce soit au niveau de leur expressivité que de leur efficacité.

Cette thèse s'attelle dans un premier temps à améliorer l'état de l'art en matière de preuves à divulgation nulle de connaissance. Nous construisons une preuve d'appartenance à un sous ensemble dont la taille est indépendante de l'ensemble en question. Nous construisons de même une preuve de connaissance amortie qui est plus efficace et plus simple que toutes les constructions qui la précèdent.

Notre second propos est d'utiliser ces preuves à divulgation nulle de connaissance pour construire de nouvelles primitives cryptographiques. Nous concevons une signature de groupe dont la taille est indépendante du groupe en question, ainsi qu'un schéma de vote électronique hautement efficace, y compris pour des élections à grand échelle.

Mots Clés

cryptographie, réseaux euclidiens, preuves de connaissance, signatures de groupe, vote électronique.

Abstract

Lattice based cryptography has developed greatly in the last two decades, both with new and stimulating results such as fully-homomorphic encryption, and with great progress in the efficiency of existing cryptographic primitives like encryption and signatures which are becoming competitive with their number theoretic counterparts. On the other hand, even though they are a crucial part of many privacy-based protocols, zero-knowledge proofs of knowledge are still lagging behind in expressiveness and efficiency.

The first goal of this thesis is to improve the quality of lattice-based proofs of knowledge. We construct new zero-knowledge proofs of knowledge such as a subset membership proof with size independent of the subset. We also work towards making zero-knowledge proofs more practical, by introducing a new amortized proof of knowledge that subsumes all previous results.

Our second objective will be to use the proofs of knowledge we designed to construct novel and efficient cryptographic primitives. We build a group signature whose size does not depend on the size of the group, as well as a practical and highly scalable lattice-based e-voting scheme.

Keywords

cryptography, lattices, zero knowledge, group signatures, e-voting.