ZERO-KNOWLEDGE PROOF SYSTEMS FOR LATTICE-BASED CRYPTOGRAPHY

NANYANG TECHNOLOGICAL UNIVERSITY

ZERO-KNOWLEDGE PROOF SYSTEMS FOR LATTICE-BASED CRYPTOGRAPHY

NGUYEN TA TOAN KHOA

School of Physical and Mathematical Sciences

2013

NGUYEN TA TOAN KHOA

ZERO-KNOWLEDGE PROOF SYSTEMS FOR LATTICE-BASED CRYPTOGRAPHY

NGUYEN TA TOAN KHOA

School of Physical and Mathematical Sciences

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2013

To my late father and my late grandmother.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the guidance and support of a great many people.

First and foremost, I would like to express my profound sense of gratitude to my main supervisor, professor Wang Huaxiong, for his inspired guidance and advice, his patience, encouragement and support during my PhD years. I am deeply indebted to him for his understanding and kindness, in both academic and personal aspects.

I would like to sincerely thank my co-supervisor professor Xing Chaoping and professor Ling San for their priceless support and guidance.

I would like to give a special thank to professor Damien Stehlé for his insightful suggestions, his instructive explanations and comments. I am profoundly grateful for his enthusiasm and kindness, and am honored to have worked with him.

I want to thank my friends for helpful discussions and cooperations: Crystal Clough, Adeline Langlois, Tang Zhaohui, Reetabrata Har, Lin Fuchun, Chen Jie, Romar Dela Cruz, Rishiraj Bhattacharyya. I also want to thanks all the members of SPMS-MAS-04-20 for sharing the lab and sharing their knowledge and experience.

I would like to thank the SINGA Scholarship and the MERLION Project for their financial supports.

Last but not least, I want to gratefully thank my mother, my wife, my daughter, my sister and brother for their great love, support, encouragement, and inspiration. A very special thank to my wife's family for taking care of my daughter MiMi during the writing of this thesis.

LIST OF WORKS

Below is the list of works that have been published during my PhD studies, in chronological order.

1. Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Khoa Nguyen. Revocable Identity-Based Encryption from Lattices. In *ACISP*, volume 7372 of *Lecture Notes in Computer Science*, pages 390–403. Springer, 2012.

2. San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications. In *Public Key Cryptography* - PKC 2013, volume 7778 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2013.

3. Adeline Langlois, San Ling, Khoa Nguyen and Huaxiong Wang. Lattice-based Group Signature Scheme with Verifier-local Revocation. In *Public Key Cryptography* - PKC 2014, *Lecture Notes in Computer Science*, Springer, March 2014. To appear.

Contents

List of Figures

List of Tables

ABSTRACT

Lattice-based cryptography is one of the most active research topics in cryptography in recent years. In many cryptographic constructions based on lattice assumptions, the building block is a proof of knowledge of a solution to an instance of the Inhomogeneous Small Integer Solution (ISIS) problem. However, all known such proof systems have relatively weak security guarantees: Breaking each of these protocols is potentially easier than solving the underlying instance of the ISIS problem. As a consequence, cryptographic constructions relying on these proof systems typically inherit the sub-optimal security guarantees. Therefore, proof systems with *stronger* security guarantees are highly desirable. Such protocols are not only interesting from a theoretical point of view, but they also lead to lattice-based cryptographic constructions relying on *weaker* hardness assumptions than the contemporary schemes.

In this thesis, we construct a series of zero-knowledge proofs of knowledge with strong security guarantees and reasonable communication costs, that can find various applications in lattice-based cryptography. Our constructions rely on a simple, yet versatile and effective technique, called Decomposition-Extension. When adapting this technique to the Stern-KTX proof system (Stern '96 - Kawachi, Tanaka, Xagawa '08), we obtain a zero-knowledge proof of knowledge for the ISIS problem (in the infinity norm) with a very strong security guarantee: Breaking the protocol is as least as hard as solving the underlying ISIS instance. We then develop our technique to design the following lattice-based cryptographic constructions:

- Efficient zero-knowledge proofs of plaintext knowledge with strong security guarantees for 4 encryption schemes based on the Learning with Errors problem: Regev's scheme (Regev '05); the Dual-Regev scheme (Gentry, Peikert, Vaikuntanathan '08); the PVW scheme (Peikert, Vaikuntanathan, Waters '08); and the GHV scheme (Gentry, Halevi, Vaikuntanathan '10). Our results immediately yield 4 lattice-based interactive encryption protocols that are secure under chosen ciphertext attacks. Previously, only zero-knowledge proofs of plaintext knowledge for Regev's scheme were known, and they are relatively inefficient with rather weak security guarantees.

- A lattice-based identity-based identification scheme relying on a weaker hardness assumption than in the previous works. Furthermore, we introduce an identity-based ring identification scheme based on the worst-case hardness of lattice problems. To the best of our knowledge, this is the first such scheme.

- An improved lattice-based group signature scheme relying on relatively weak hardness assumptions, in which the signature size is logarithmic in the number of group users. Earlier lattice-based group signature schemes, which were published before 2013, rely on relatively weak hardness assumptions but have the undesirable property that the size of the signature is linear in the number of group users. A recent scheme (Laguillaumie, Langlois, Libert, Stehlé '13) achieves logarithmic signature size but it has to rely on relatively strong hardness assumptions. Our construction, thus, simultaneously achieves the good features of the existing schemes.

1. INTRODUCTION

1.1 Zero-knowledge Proof Systems

Proofs are fundamental objects in mathematics and computer science. In mathematics, one of the main goals is to decide whether statements are correct and to prove that they are so. In computer science, the famous problem "P vs. NP" essentially asks if proofs are harder to find or to verify. Seeing a *static* proof, one can verify the correctness of the statement, and can even learn more than that. We first consider the following example.

Suppose that Peggy has found a solution to some hard problem, and she wants to convince her competitor Vincent that "The problem has been solved!". Peggy can simply write down her solution, and then send it to Vincent for verifying its validity. However, a dishonest Vincent might do more than that: If the statement is true (i.e., the problem has been solved correctly), he will be capable of presenting the solution to others, or even claiming that it is his own solution. In this situation, Peggy may need another type of proof system that allows her to convince Vincent, so that the later should not learn any additional information. Such proof system must protect an honest Peggy: If the statement is true, then she should be able to convince Vincent. On the other hand, it must also be fair to an honest Vincent: He should not be convinced by a false statement, otherwise he might give up the competition. Does such proof system exist?

In 1985, Goldwasser, Micali and Rackoff [73] solved the above question by introducing *zero-knowledge proofs*, namely, proofs that reveal nothing but the validity of the statement being proved. This beautiful notion goes beyond the limits of traditional proofs with two new ingredients: interaction and randomization. A proof now is considered as an interactive protocol between two randomized parties - Peggy the prover, and Vincent the verifier. As discussed above, an interactive proof must satisfy two requirements: *completeness* that protects an honest prover, and *soundness* that protects an honest verifier. The zero-knowledge property is captured by the existence of a *simulator*. Intuitively, if a simulator, who does not have access to the real prover, can produce the transcript of the interaction between the prover and the verifier, then it hints that the conversation with the prover does not provide any additional knowledge for the verifier.

As we have seen, zero-knowledge proofs enable Peggy to convince Vincent the validity of the statement "The problem has been solved!", without revealing how it has been solved. In a *proof of knowledge* [73, 18], Peggy can also prove that she indeed *knows* how to solve the hard problem, i.e. she possesses a valid solution - a satisfying *witness* for the statement. This notion is captured by the existence of a *knowledge extractor* that can produce a witness based on the behavior of a cheating prover. Consequently, if the underlying problem is believed to be hard to solve, then the proof of knowledge is sound: Peggy cannot cheat Vincent, since otherwise, Vincent can use the knowledge extractor to solve the problem.

In the last 25 years, zero-knowledge proofs of knowledge have been extensively studied ([55, 76, 56, 125, 102],...), and have become one of the most powerful tools in public-key cryptography. These proof systems are the building blocks in many cryptographic constructions: identification schemes ([58, 55, 76, 133, 137, 85],...), group signatures ([40, 16, 27, 28, 75],...), anonymous credential systems ([37, 36, 17,

45, 44],...), interactive encryption protocols ([59, 65, 83, 72],...), and much more.

In this thesis, we are interested in zero-knowledge proofs of knowledge that are useful for lattice-based cryptography.

1.2 Lattice-based Cryptography

Lattices, or discrete subgroups of $\mathbb{R}^n$, have been studied since the 18th-19th century, pioneered by great mathematicians such as Lagrange, Gauss, Minkowski and Hermite. The computational aspect of lattices has attracted special attention over the last three decades along with the rapid development of computer science, and particularly, public-key cryptography.

The most famous computational problems on lattices are the Shortest Vector Problem (SVP_γ), the Closest Vector Problem (CVP_γ) and the Shortest Independent Vectors Problem (SIVP_γ), where γ is an approximation factor. These problems are known to be NP-hard to solve exactly (i.e., for $\gamma = 1$), or to approximate to within small factors [138, 5, 106, 53]. Moreover, it is widely believed that there is no efficient algorithm approximating lattice problems (SVP_γ, CVP_γ, SIVP_γ, among others) to within polynomial factors $\gamma = \mathsf{poly}(n)$ (where n is the lattice dimension). Indeed, the best known algorithms which run in polynomial time can only approximate lattice problems to within sub-exponential factors, while for polynomial factors, they all run in exponential time [90, 10]. Furthermore, while commonly used problems in public-key cryptography, such as the integer factorization problem and the discrete logarithm problem, can be solved efficiently by powerful quantum computers [135], currently it is not known whether there exists a quantum algorithm for solving lattice problems that performs significantly better than its non-quantum counterparts.

Therefore, it is reasonable to conjecture that:

"There is no polynomial-time (even quantum) algorithm that approximates lattice problems to within polynomial factors" [114].

Given the above conjectured hardness, lattice problems are potentially useful for building cryptographic constructions, that are resistant against the emerging quantum computers. Nevertheless, how to realize this was a big challenge until the year of 1996. The main technical difficulty was due to the gap between worst-case hardness and average-case hardness. On one hand, in complexity theory, a problem is considered hard if it is so in the worst case, i.e., for *any* instance of the problem. On the other hand, modern public-key cryptography relies on the the computational hardness of solving a *random* instance of some mathematical problem. Is it possible to base cryptography on worst-case hardness assumptions via certain type of reduction?

In 1996, Ajtai [4] discovered a ground-breaking worst-case to average-case connection. Specifically, he showed how to build a cryptographic function such that breaking a *random* instance of the function is as least as hard as approximating certain lattice problem on *any* lattice to within a small polynomial factor. Hence, for the first time ever, the provable security of a cryptographic construction can be based on the worst-case hardness of a computationally intractable problem. This fascinating result laid the first foundation for lattice-based cryptography.

The underlying hard-on-average problem in the seminal work of Ajtai was later formalized by Micciancio and Regev [113] as the Small Integer Solution (SIS) problem. Its inhomogeneous variant ISIS, which enjoys the same worst-case to average-case reduction, was subsequently defined by Gentry et al. [62]. In the ℓ_p norm, the

SIS and ISIS problems with parameters n, m, q, β are as follows: Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and a uniformly random vector $\mathbf{y} \in \mathbb{Z}_q^n$,

- SIS^p: Find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_p \leq \beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q$.
- ISIS^p: Find a vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_p \leq \beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$.

Inspired by Ajtai's one-way function, a lot of "minicrypt" constructions [80] based on the hardness of the SIS and ISIS problems have been proposed: collision-resistant hash functions [67, 120, 97], commitment scheme [85], identification schemes [115, 85, 94, 43], and signature schemes [98, 62, 95, 42, 29, 132, 96, 111]. In addition, a few provably secure encryption schemes from lattices were also introduced [8, 126, 9], but they are rather impractical.

In 2005, Regev [127] set a milestone in the development of lattice-based cryptography by introducing the Learning with Errors (LWE) problem, which enjoys a worst-case to average-case *quantum* reduction: If there is an efficient algorithm solving a *random* LWE instance, then there is an efficient quantum algorithm approximating lattice problems on *any* lattice. The LWE problem with parameters n, q, m and with noise distribution χ supported on small numbers is as follows: Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and vector $\mathbf{b} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{s}$ is uniformly random over $\mathbb{Z}_q^n$ and $\mathbf{e}$ is sampled from χ^m,

- The search version of LWE asks to find $\mathbf{s}$.
- The decision version of LWE asks to distinguish between the pair $(\mathbf{A}, \mathbf{b})$ and a uniformly random pair over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$.

The LWE problem quickly got attention of the research community, and since then it has served as the basis for a large variety of cryptographic constructions: encryption schemes that are secure under chosen plaintext attacks [127, 62, 122, 61]

and chosen ciphertext attacks [123, 118, 111], (hierarchical) identity-based encryption schemes [62, 42, 1, 2, 47], oblivious transfer scheme [122], leakage-resilient schemes [13, 11, 71], and fully homomorphic encryption schemes [35, 33, 31, 32].

As we have seen, lattice-based cryptography possesses a unique combination of attractive features:

- Provable security under worst-case hardness assumptions. This is a strong theoretical evidence that the random instances used in lattice-based cryptography are indeed hard to solve. This feature has not been seen in any other cryptographic construction.
- Conjectured resistance against quantum computers. Shor's algorithm [135] poses a future threat for cryptography based on integer factorization and discrete logarithm problems. Given the rapid development of technology nowadays, powerful quantum computers might become a reality in the near future. Among the candidates for "post-quantum cryptography", lattices are emerging as the most promising one.
- Asymptotic efficiency. The underlying average-case problems (I)SIS and LWE involve only simple operations such as choosing uniformly random integer matrices and vectors, and performing basic multiplications and additions with them. In contrast, for cryptography based on number-theoretic problems, the corresponding operations are much more expensive, such as generating huge prime numbers, and exponentiating modulo these primes.

Having these nice features, lattice-based cryptography has become one of the most trendy topics in the field, especially in the last several years. This yields interesting challenges of improving the existing schemes and constructing the new ones.

The main drawback of cryptographic constructions based on the (I)SIS and LWE problems is that they typically involve large parameters and large key sizes which are beyond practicability. Nevertheless, as noticeable improvements have been introduced in recent works ([108, 96, 77, 101]), lattice-based cryptography is moving closer to practice.

1.3 Contributions of the Thesis

Motivation. In many lattice-based cryptographic constructions ([115, 94, 136, 130, 88, 131],...), the building block is a proof of knowledge of a valid solution to an instance of the ISIS problem. In the security proof, the reduction algorithm typically uses the knowledge extractor of the underlying protocol to extract a vector that will be helpful for solving a challenge instance of the hard problem. In these cases, the "quality" of the extracted vector will affect the security guarantee of the protocol, and the hardness assumption used in the cryptographic construction built upon it.

In the scope of this thesis, we use the term "*extraction gap*" as a criterion for the quality of the extracted vector. Specifically, we say that a proof of knowledge for the ISIS^∞ or SIS^∞ problem admits extraction gap g (where $g \geq 1$) if the knowledge extractor in that protocol, at best, can produce a vector whose infinity norm is g times larger than that of the valid witness possessed by the prover. In the ideal situation when $g = 1$ (i.e., there is "no extraction gap"), the reduction algorithm would be able to use the knowledge extractor to solve the underlying ISIS or SIS instance. This reduction demonstrates that such proof system has a very strong security guarantee, because breaking the protocol is as least as hard as solving the underlying problem. However, when $g > 1$, the soundness of the proof system usually has to rely on a *stronger* hardness assumption, i.e., assumption that certain

potentially easier problem is hard to solve.

We observe that, all existing efficient proofs of knowledge for the ISIS^∞ or SIS^∞ problem ([115, 94]) admit noticeable extraction gaps. Specifically:

- For the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ problem, all known proof systems admit extraction gaps $g \geq \widetilde{\mathcal{O}}(\sqrt{n})$, where n denotes the dimension of the corresponding worst-case lattice problem. Therefore, while a valid witness vector is supposed to have infinity norm bounded by β, these schemes only guarantee that a prover using a witness $\mathbf{x} \in \mathbb{Z}^m$, such that $\|\mathbf{x}\|_\infty > g \cdot \beta$ cannot cheat the verifier. However, in these schemes, it is not guaranteed that a cheating prover using $\mathbf{x}' \in \mathbb{Z}^m$ such that $\beta < \|\mathbf{x}'\|_\infty \leq g \cdot \beta$ cannot pass the verification procedure. In other words, the soundness of these schemes has to rely on the hardness of solving the $\mathsf{ISIS}^\infty(n, m, q, g \cdot \beta)$ problem.

- For the $\mathsf{SIS}^\infty(n, m, q, \beta)$ problem, the only known proof system, which is derived from a protocol in [115], also has an extraction gap $g \geq \widetilde{\mathcal{O}}(\sqrt{n})$.

The above discussion shows that the current proof systems for the ISIS^∞ and SIS^∞ problems are sub-optimal, since breaking each of these protocols is potentially easier than solving the underlying problem. As a consequence, lattice-based cryptographic constructions using these protocols as their building blocks inherit the sub-optimal security guarantees. Therefore, zero-knowledge proof systems with stronger security guarantees (e.g. with a small constant extraction gap) are highly desirable. Such protocols are not only interesting from a theoretical point of view, but they also lead to lattice-based cryptographic constructions relying on *weaker* hardness assumptions than the contemporary schemes.

Our Contributions. In this thesis, we construct a series of statistical zero-knowledge proofs of knowledge that can find various applications in lattice-based cryptography. Our schemes rely on a simple, yet versatile and effective technique, called Decomposition-Extension. When adapting this technique to the Stern-KTX protocol [137, 85], we obtain zero-knowledge proof systems with strong security guarantees while maintaining reasonable communication costs. More generally, the technique that we developed can be used to construct zero-knowledge proofs of knowledge of linear objects (e.g., vectors, matrices) which are the secret solutions of certain given systems of modular linear equations, and which satisfy some suitable conditions (e.g., "smallness"). In other words, our technique is somewhat compatible with the properties of lattice-based cryptography (i.e., "linearity", "smallness"), and can possibly find further interesting applications.

Our main contributions are as follows.

Contribution 1: **Improved Proof Systems for the ISIS^∞ and SIS^∞ Problems** In this work, we construct statistical a zero-knowledge proof of knowledge for the ISIS^∞ problem with no extraction gap (i.e., $g = 1$). We will refer to this proof system as **SternExt**. In terms of security, **SternExt** has a very strong guarantee: Breaking the protocol is at least as hard as solving the underlying ISIS^∞ instance. Moreover, in terms of efficiency, **SternExt** compares favorably to the Micciancio-Vadhan protocol [115], which is essentially the only known efficient *zero-knowledge* proof system for the ISIS^∞ problem prior to our work. By some technical modifications, we also get a proof system for the SIS^∞ problem with a slightly weaker security guarantee, where the extraction gap is $g < 2$.

As we will sketch below, the **SternExt** proof system can find various applications in lattice-based cryptography.

Contribution 2: **Proofs of Plaintext Knowledge for LWE-based Encryption Schemes**

Proofs of plaintext knowledge (PoPK) are useful in constructing interactive encryption protocols that are secure under chosen ciphertext attacks [59, 65]. Existing zero-knowledge PoPK for LWE-based encryption schemes are somewhat unsatisfactory. First, all known such proof systems ([24, 25, 15, 51]) only target Regev's encryption scheme [127]. Furthermore, these schemes are relatively inefficient (with communication cost $\widetilde{\mathcal{O}}(n^2)$) and rely on the assumption that SIVP_γ is hard for super-polynomial approximation factors (i.e., for $\gamma = n^{\omega(1)}$). Thus, constructing efficient zero-knowledge PoPK with strong security guarantees for LWE-based encryption schemes remains an open question.

In this work, we introduced efficient statistical zero-knowledge PoPK for a variety of LWE-based encryption schemes. Moreover, the security of our proof systems relies directly on the security of the underlying encryption schemes. We achieved these nice features by adapting and generalizing the techniques used in the **SternExt** proof system. Specifically, we present an efficient zero-knowledge PoPK for Regev's scheme [127] with much stronger security guarantee than in previous works, and with communication cost $\widetilde{\mathcal{O}}(n)$. We further introduce zero-knowledge PoPK for 3 other LWE-based encryption schemes: the Dual-Regev encryption scheme [62]; the scheme introduced by Peikert, Vaikuntanathan, and Waters [122]; and the one proposed by Gentry, Halevi, and Vaikuntanathan [61]. Our constructions yield 4 interactive encryption protocols from lattices, that are secure under chosen ciphertext attacks.

Contribution 3: **Lattice-based Identity-based Identification Schemes**

An identification scheme [58] is an interactive protocol that allows a user holding a secret key to identify itself to a party holding the corresponding public key. A *ring* identification scheme [54] additionally allows a user to create a ring of users in an

ad-hoc fashion, and then *anonymously* prove its membership in such ring. We are interested in these primitives in the identity-based (ID-based) setting [134], where the public key of a user is a publicly known string representing its identity, e.g., an email address or a physical IP address.

Prior to our work, two ID-based identification schemes from lattices have been published: Stehlé et al.'s scheme [136], which is based on the hardness of $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^2)}$ in general lattices; and Rückert's scheme [130], that is based on the hardness of $\mathsf{SVP}^\infty_{\widetilde{\mathcal{O}}(n^{3.5})}$ in ideal lattices.

In this work, we rely on the **SternExt** proof system and the GPV signature scheme [62] to construct an ID-based identification scheme based on the hardness of $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^{1.5})}$ (in general lattices). Thus, our schemes relies on a *weaker* hardness assumption than the previous constructions. The scheme is only proved secure in the random oracle model because the GPV signature scheme [62] is so. Nevertheless, if we rely on lattice-based signature schemes that are secure in the standard model (e.g., [42, 29, 111]), then we can obtain secure lattice-based ID-based identification schemes the standard model.

In addition, we introduce an ID-based *ring* identification scheme based on the hardness of $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^{1.5})}$. Up to the best of our knowledge, this is the first instantiation of ID-based ring identification from lattice assumptions.

Contribution 4: **Improved Lattice-based Group Signature Scheme**

Group signatures have been an important research topic in cryptography since their introduction in 1991 by Chaum and van Heyst [46]. A secure group signature scheme must satisfy two seemingly contradictory requirements: each group user can anonymously sign documents on behalf of the whole group, but in cases of disputes, there is a tracing mechanism which can link a given signature to the identity of the misbehaving user. While numerous group signatures schemes have been proposed, only

a few of them are believed to be resistant against quantum computers.

Prior to our work, three lattice-based group signature schemes have been published. The first scheme, which is essentially the first quantum-resistant group signature, was introduced by Gordon, et al. [74] in 2010. While the scheme is of great theoretical interest, it produces signatures of size $\mathcal{O}(N)$, where N is the number of group users. In terms of efficiency, this is a noticeable disadvantage when the group is large. In 2012, Camenisch et al. [38] proposed lattice-based anonymous attribute tokens system, a primitive that can be considered as a generalization of group signature. However, in their construction, the signature size inherits the linear dependence on N from Gordon et al.'s scheme. Recently, Laguillaumie et al. [88] designed a scheme featuring signature size $\widetilde{\mathcal{O}}(\log N)$, which is the first lattice-based group signature that overcomes the linear-size barrier. The signature in their scheme is asymptotically shorter than in previous works [74, 38], but the scheme requires rather large parameters, and relies on much stronger hardness assumptions than in contemporary lattice-based cryptographic constructions.

Thus, each of the previous lattice-based group signature schemes has its own strength and weakness: those with relatively weak hardness assumptions have linear signature size, while the only known scheme with logarithmic signature size has to rely on much stronger assumptions. In this work, we adapt the **SternExt** proof system to the Bonsai tree structure [42] to construct a lattice-based group signature scheme that simultaneously achieves the good features of existing schemes. Specifically, in our construction, the size of the signature is $\widetilde{\mathcal{O}}(\log N)$ (which is comparable to that of the Laguillaumie et al.'s scheme), while the hardness assumptions are comparable to those of the Gordon et al's scheme: In the random oracle model, the anonymity of the scheme relies on the quantum hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^2)}$, and its traceability is based on the hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^{1.5})}$.

1.4 Organization of the Thesis

The rest of the thesis is organized as follows:

- In Chapter 2, we first describe the general notations, then we recall the basics of lattice-based cryptography as well as zero-knowledge proof systems, and review the cryptographic tools that will be used throughout the thesis.
- In Chapter 3, we present our proof systems for the ISIS^{∞} and SIS^{∞} problems.
- In Chapter 4, we construct zero-knowledge proofs of plaintext knowledge for 4 LWE-based encryption schemes.
- In Chapter 5, we introduce an improved identity-based identification and a new identity-based ring identification schemes from lattices.
- In Chapter 6, we construct an improved lattice-based group signature scheme.
- In Chapter 7, we conclude the thesis and state the open problems for future investigations.

In particular, the results presented in Chapter 3, Section 4.2, and Section 5.2 are based on the paper "Improved Zero-knowledge Proofs of Knowledge for the ISIS Problem, and Applications" co-authored with professors San Ling, Damien Stehlé, and Huaxiong Wang. The paper was published in the proceedings of PKC 2013. The thesis author was the primary investigator and author of this paper.

2. DEFINITIONS AND PRELIMINARIES

2.1 General Notations

Throughout the thesis, we will use the following notations:

Vectors, Matrices and Norms. We denote column vectors by bold lower-case letters (e.g., $\mathbf{x}, \mathbf{y}$), and matrices by bold upper-case letters (e.g., $\mathbf{A}, \mathbf{B}$). The concatenation of two vectors $\mathbf{x}_1 \in \mathbb{R}^m$, $\mathbf{x}_2 \in \mathbb{R}^k$ is denoted by $(\mathbf{x}_1 \| \mathbf{x}_2) \in \mathbb{R}^{m+k}$. Similarly, the concatenation of two matrices $\mathbf{A}_1 \in \mathbb{R}^{n \times m}$, $\mathbf{A}_2 \in \mathbb{R}^{n \times k}$ is denoted by $[\mathbf{A}_1 | \mathbf{A}_2] \in \mathbb{R}^{n \times (m+k)}$. We let 0^m denote the zero-vector in dimension m, and let 1^m denote the m-dimensional vector with all coordinates equal to 1. We let $\mathsf{wt}(\mathbf{x})$ denote the Hamming weight of vector $\mathbf{x}$, i.e., the number of its non-zero coordinates.

The ℓ_p norm (for $1 \leq p < \infty$) of vector $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{R}^m$, given by $(|x_1|^p + \ldots + |x_m|^p)^{1/p}$, is denoted by $\|\mathbf{x}\|_p$. The infinity norm ℓ_∞ of $\mathbf{x}$ is defined as $\|\mathbf{x}\|_\infty := \max\{|x_1|, \ldots |x_m|\}$. For simplicity, we denote the ℓ_2 norm (i.e., the Euclidean norm) by $\|\cdot\|$.

Let $\mathbf{B} = [\mathbf{b}_1 | \ldots | \mathbf{b}_m] \in \mathbb{R}^{m \times m}$, we denote by $\|\mathbf{B}\|$ the "length" of $\mathbf{B}$, given by $\max\{\|\mathbf{b}_1\|, \ldots, \|\mathbf{b}_m\|\}$. If $\mathbf{b}_1, \ldots, \mathbf{b}_m$ are linearly independent vectors, then we let $\widetilde{\mathbf{B}} = [\widetilde{\mathbf{b}}_1 | \ldots | \widetilde{\mathbf{b}}_m]$ denote the Gram-Schmidt orthogonalization of $\mathbf{B}$, defined recursively as follows: $\widetilde{\mathbf{b}}_1 = \mathbf{b}_1$, and for $i = 2, \ldots, m$, the vector $\widetilde{\mathbf{b}}_i$ is the component of $\mathbf{b}_i$ orthogonal to $\mathsf{span}(\mathbf{b}_1, \ldots, \mathbf{b}_{i-1})$.

Sets. We let the $[k]$ denote the set $\{1, \ldots, k\}$, and let $\{x_i\}_{i=1}^k$ denote the (ordered)

set $\{x_1, \ldots, x_k\}$. The symmetric group of all permutations of k elements is denoted by S_k. When S is a finite set, $y \xleftarrow{\$} \mathrm{S}$ means that y is chosen uniformly at random from S.

For integer $m \geq 1$, we let B_{3m} denote the set of all vectors $\mathbf{x} \in \{-1, 0, 1\}^{3m}$ having exactly m coordinates equal to -1; m coordinates equal to 0; and m coordinates equal to 1. By B_{2m} we denote the set of all vectors $\mathbf{x} \in \{0, 1\}^{2m}$ such that $\mathsf{wt}(\mathbf{x}) = m$.

Other Notations. We use the notation $y \hookleftarrow D$ when y is sampled from the distribution D. If x is the output of algorithm $\mathcal{A}$, then we write $x \leftarrow \mathcal{A}$. We denote by n the security parameter of cryptographic schemes in this thesis, which typically corresponds to the lattice dimension in the underlying lattice problems. Algorithms are *efficient* if they run in probabilistic polynomial time (PPT) $\mathsf{poly}(n)$. A function $\epsilon : \mathbb{N} \to \mathbb{R}_{\geq 0}$ is said *negligible* in n (denoted by $\mathsf{negl}(n)$) if it vanishes faster than the inverse of any polynomial. We say that an event happens with overwhelming probability if it happens with probability $1 - \epsilon(n)$ for some negligible function ϵ. The statistical distance between two distributions X and Y over a countable domain D is $\frac{1}{2}\sum_{d \in \mathrm{D}} |X(d) - Y(d)|$. We say that X and Y are *statistically close* (denoted by $X \approx_s Y$) if their statistical distance is negligible.

To evaluate the asymptotic running time and object sizes, we use the following standard Bachmann-Landau notations:

Notation	**Definition**
$f(n) = \mathcal{O}\big(g(n)\big)$	$\exists k > 0, \exists n_0 : \ \forall n > n_0, \lvert f(n)\rvert \leq k \cdot \lvert g(n)\rvert$
$f(n) = o\big(g(n)\big)$	$\forall k > 0, \exists n_0 : \ \forall n > n_0, \lvert f(n)\rvert \leq k \cdot \lvert g(n)\rvert$
$f(n) = \Omega\big(g(n)\big)$	$\exists k > 0, \exists n_0 : \ \forall n > n_0, \lvert f(n)\rvert \geq k \cdot \lvert g(n)\rvert$
$f(n) = \omega\big(g(n)\big)$	$\forall k > 0, \exists n_0 : \ \forall n > n_0, \lvert f(n)\rvert \geq k \cdot \lvert g(n)\rvert$
$f(n) = \Theta\big(g(n)\big)$	$f(n) = \mathcal{O}\big(g(n)\big)$ and $f(n) = \Omega\big(g(n)\big)$

In addition, we often use the "soft-O" notation: We write $f(n) = \widetilde{\mathcal{O}}(g(n))$ if $f(n) = \mathcal{O}(g(n) \cdot \log^c g(n))$ for some constant c. All logarithms are base 2.

2.2 Preliminaries on Lattices

2.2.1 Lattices

In this thesis, we will be working with full-rank lattices, which are discrete subgroups of $\mathbb{R}^n$, defined as follows:

Definition 2.2.1 (Lattice)**.** Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be linearly independent vectors in $\mathbb{R}^n$. The lattice generated by $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n] \in \mathbb{R}^{n \times n}$ is the set

$$\Lambda(\mathbf{B}) = \left\{ \mathbf{B} \cdot \mathbf{x} : \ \mathbf{x} \in \mathbb{Z}^n \right\} = \left\{ \sum_{i=1}^{n} x_i \cdot \mathbf{b}_i : \ x_i \in \mathbb{Z} \right\}$$

of all integer linear combinations of the columns of $\mathbf{B}$. The matrix $\mathbf{B}$ is called a *basis* of the lattice $\Lambda(\mathbf{B})$. The integer n is called the *dimension* of the lattice.

In dimension $n > 1$, a given lattice Λ has infinitely many bases. The quality of a basis is usually judged by its length $\|\mathbf{B}\|$. Given any basis $\mathbf{B}$ of the lattice Λ, the *determinant* $\det(\Lambda)$ of the lattice is $\sqrt{\det(\mathbf{B}^T \cdot \mathbf{B})}$. The determinant and the successive minima, defined below, are invariants of the lattice.

Definition 2.2.2 (Successive minima)**.** Let Λ be an n-dimensional lattice. For each $i \in [n]$, the i-th successive minimum λ_i^p with respect to the ℓ_p norm is the smallest number $r > 0$ such that there exist i linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_i \in \Lambda$ satisfying $\|\mathbf{b}_j\|_p \leq r$ for all $j \in [i]$.

In other words, the i-th successive minimum is the radius of the smallest ball containing i linearly independent lattice vectors. Particularly, $\lambda_1^p(\Lambda)$ is the length of the shortest non-zero vector in the lattice Λ.

2.2.2 Standard Lattice Problems

We now recall the definitions, as well as the known hardness results of standard lattice problems and their approximation versions. We denote the problems in ℓ_p norm (for $p \in \mathbb{N} \cup \{\infty\}$) by superscript p, e.g., SVP^p. When we refer to the Euclidean norm ($p = 2$), the superscript is often omitted. The approximation versions of the problems are denoted by an additional subscript γ indicating the approximation factor (which can be a function of the lattice dimension).

Definition 2.2.3 (SVP^p and SVP^p_γ)**.** Given a lattice Λ, the Shortest Vector Problem SVP^p asks to find a vector $\mathbf{v} \in \Lambda$ such that $\|\mathbf{v}\|_p = \lambda_1^p(\Lambda)$.
The approximation version of the problem, SVP^p_γ, asks to find $\mathbf{v} \in \Lambda$ such that $\|\mathbf{v}\|_p \leq \gamma \cdot \lambda_1^p(\Lambda)$.

Definition 2.2.4 (CVP^p and CVP^p_γ)**.** Given a lattice Λ, and a target vector $\mathbf{t}$, the Closest Vector Problem CVP^p asks to find a vector $\mathbf{v} \in \Lambda$ such that for any $\mathbf{x} \in \Lambda$: $\|\mathbf{v} - \mathbf{t}\|_p \leq \|\mathbf{x} - \mathbf{t}\|_p$.
The approximation version of the problem, CVP^p_γ, asks to find a vector $\mathbf{v} \in \Lambda$ such that for any $\mathbf{x} \in \Lambda$: $\|\mathbf{v} - \mathbf{t}\|_p \leq \gamma \cdot \|\mathbf{x} - \mathbf{t}\|_p$.

Definition 2.2.5 (SIVP^p and SIVP^p_γ)**.** Given an n-dimensional lattice Λ, the Shortest Independent Vectors Problem SIVP^p asks to finds n linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \Lambda$, such that $\max_i \|\mathbf{v}_i\|_p \leq \lambda_n^p(\Lambda)$.
The approximation version of the problem, SIVP^p_γ, asks to find $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \Lambda$, such that $\max_i \|\mathbf{v}_i\|_p \leq \gamma \cdot \lambda_n^p(\Lambda)$.

Apart from the above search problems, we will also consider the decision version of SVP^p_γ and CVP^p_γ, where the solver is asked to return a YES/NO decision.

Definition 2.2.6 (GapSVP^p_γ)**.** Given a gap function γ, a GapSVP^p_γ instance is a pair $(\mathbf{B}, d)$, where $\mathbf{B}$ is a basis of a lattice Λ, and $d \in \mathbb{Q}$. Return YES if there exists a

non-zero vector $\mathbf{v} \in \Lambda$, such that $\|\mathbf{v}\|_p \leq d$ (that is, $\lambda_1^p \leq d$). Return NO if for any non-zero vector $\mathbf{v} \in \Lambda$, $\|\mathbf{v}\|_p > \gamma \cdot d$ (that is $\lambda_1^p > \gamma \cdot d$).

Definition 2.2.7 (GapCVP_γ^p)**.** Given a gap function $\gamma = \gamma(n)$, a GapSVP_γ^p instance is a triple $(\mathbf{B}, \mathbf{t}, d)$, where $\mathbf{B}$ is a basis of a lattice Λ, $\mathbf{t}$ is a target vector, and $d \in \mathbb{Q}$. Return YES if there exists $\mathbf{v} \in \Lambda$, such that $\|\mathbf{v} - \mathbf{t}\|_p \leq d$. Return NO if for any vector $\mathbf{v} \in \Lambda$, $\|\mathbf{v} - \mathbf{t}\|_p > \gamma \cdot d$.

We now briefly review the known hardness results of lattice problems. The first result was provided in 1981 by van Emde Boas [138], who proved that CVP_1^p is NP-hard for any p. Arora et al. [14] later proved the NP-hardness of CVP_c^p for any constant c. The approximation factor was improved to $2^{\log^{1-\epsilon} n}$ by Dinur et al. [53]. In addition, CVP_γ is known to be at least as hard as SVP_γ for the same approximation factor [70]; and CVP_1^2 and SIVP_1^2 are proved to be equivalent under polynomial-time rank-preserving reductions [109].

The NP-hardness of SVP_1 was given by van Emde Boas [138] for the ℓ_∞ norm, and by Ajtai [5] for the ℓ_2 norm (under randomized reductions). In particular, this implies that SIVP_1 is NP-hard for these norms. Micciancio [106] later proved that for any p, and for $\gamma = \sqrt[p]{2}$, SVP_γ^p is NP-hard (under reverse unfaithful random reductions).

On the other hand, there are theoretical evidences that SVP_γ and CVP_γ are not NP-hard for large polynomial factors $\gamma = n^c$. In particular, for $\gamma = \Omega(\sqrt{n/\log n})$, both problems belong to $\mathsf{NP} \cap \mathsf{coAM}$ [66], and for factors $\gamma = \Omega(\sqrt{n})$, they belong to $\mathsf{NP} \cap \mathsf{coNP}$ [3], and thus, unless the polynomial-time hierarchy collapses, the problems cannot be NP-hard within such factors. Nevertheless, for any $\gamma = \mathsf{poly}(n)$, it is widely believed that there is no polynomial-time algorithm for solving SVP_γ, since all known algorithms, e.g. LLL [90] and its variants, run in time time $2^{\mathcal{O}(n)}$.

Moreover, as opposed to the integer factorization and discrete logarithm problems which can be solved efficiently by powerful quantum computers [135], currently it is not known whether there exists a *quantum* algorithm for solving lattice problems that performs significantly better than its non-quantum counterparts.

The discussion above leads to the following conjecture, which is the fundamental security assumption for lattice-based cryptography.

Conjecture 2.2.8 ([114])**.** *There is no polynomial-time (even quantum) algorithm that approximates lattice problems to within polynomial factors.*

2.2.3 Random q-ary Lattices and Gaussian Distributions

Random q-ary Lattices

Lattice-based cryptography is essentially based on the average-case problems SIS, ISIS and LWE, which are proved to be as least as hard as standard lattice problems (e.g., SIVP_γ), via the worst-case to average-case reductions. These average-case problems are closely connected with q-ary lattices, namely, those lattices Λ such that $q\mathbb{Z}^m \subseteq \Lambda \subseteq \mathbb{Z}^m$.

Given integers n, m, q and a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, one can define two q-ary lattices:

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

It can be checked that $\Lambda_q^{\perp}(\mathbf{A})$ and $\Lambda_q(\mathbf{A})$ are discrete subgroups of $\mathbb{Z}^m$, and thus, they are well-defined lattices.

Ideal Lattices: A special class of q-ary lattices that will often be mentioned in the thesis is the class of ideal lattices. They are structured lattices that correspond

to ideals in some polynomial ring. Basically, using ideal lattices can help to reduce the size of the public keys in lattice-based cryptographic constructions by a factor of n. For instance, to represent a q-ary (general) lattice, one typically has to use $mn \log q = \widetilde{\mathcal{O}}(n^2)$ bits, which might be unsuitable in practice. In contrast, representing an ideal lattice only costs $\widetilde{\mathcal{O}}(n)$ bits. However, the hardness of lattice problems in ideal lattices is not as well-understood as for general lattices. We refer to [97, 120, 108, 101, 100] for comprehensive discussions on ideal lattices. In the scope of this thesis, we focus only on cryptographic constructions based on general lattices.

Gaussian Distributions over Lattices.

We now recall the fundamental tool in lattice-based cryptography: Gaussian distributions over lattices (in particular, over $\mathbb{Z}^m$ and random q-ary lattices $\Lambda_q^{\perp}(\mathbf{A})$).

The Gaussian distribution over $\mathbb{R}$, parameterized by $\sigma > 0$, is defined by the density function

$$\forall x \in \mathbb{R}: \quad D_\sigma(x) = 1/\sigma \cdot \exp\big(-\pi(x/\sigma)^2\big).$$

Similarly, the m-dimensional Gaussian distribution is defined by the density function $D_\sigma(\mathbf{x}) = 1/\sigma^m \cdot \exp(-\pi(\|\mathbf{x}\|/\sigma)^2)$. We denote by $D_{\sigma,\mathbf{c}}$ the m-dimensional Gaussian distribution with center $\mathbf{c} \in \mathbb{R}^m$, i.e., $D_{\sigma,\mathbf{c}} = 1/\sigma^m \cdot \exp(-\pi(\|\mathbf{x}-\mathbf{c}\|/\sigma)^2)$.
Let $\Lambda \subseteq \mathbb{Z}^m$ be a lattice. The discrete Gaussian distribution $D_{\Lambda,\sigma,\mathbf{c}}$ is the m-dimensional Gaussian centered at $\mathbf{c}$, with support restricted to the lattice Λ. Namely, the density function of the discrete Gaussian distribution is

$$\forall \mathbf{x} \in \Lambda: \quad D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) = \frac{D_{\sigma,\mathbf{c}(\mathbf{x})}}{\sum_{\mathbf{z}\in\Lambda} D_{\sigma,\mathbf{c}}(\mathbf{z})}.$$

Gentry et al. [62] showed that the distribution $D_{\Lambda,\sigma,\mathbf{c}}$ can be efficiently sampled (to within negligible statistical distance), if σ is sufficiently large.

Lemma 2.2.9 ([62]). *Given a basis* $\mathbf{B}$ *of an* m*-dimensional lattice* Λ*, a Gaussian parameter* $\sigma \geq \|\widetilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log m})$*, and an arbitrary center* $\mathbf{c} \in \mathbb{Z}^m$*, there is a probabilistic polynomial-time algorithm that outputs a sample from a distribution statistically close to* $D_{\Lambda,\sigma,\mathbf{c}}$*.*

For simplicity, we denote $D_{\Lambda,\sigma,\mathbf{0}}$ by $D_{\Lambda,\sigma}$. We are especially interested in discrete Gaussian distribution over integers, i.e., $\Lambda = \mathbb{Z}^m$. In this case, if we let $\mathbf{x} = (x_1, \ldots, x_m) \hookleftarrow D_{\mathbb{Z}^m,\sigma}$, then each coordinate x_i is independently and identically distributed according to the 1-dimensional Gaussian distribution $D_{\mathbb{Z},\sigma}$.

We now recall several well-known facts about discrete Gaussian distributions over lattices, which will be extensively used throughout the thesis.

Lemma 2.2.10 ([120, 113, 62]). *Let* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ *such that the columns of* $\mathbf{A}$ *generate* $\mathbb{Z}_q^n$*. Let* $\mathbf{S}$ *be any basis of* $\Lambda_q^{\perp}(\mathbf{A})$*, and let* $\sigma \geq \|\widetilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log m})$*.*

1. *For* $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$*, the distribution of* $\mathbf{y} = \mathbf{A} \cdot \mathbf{x} \bmod q$ *is statistically close to uniform over* $\mathbb{Z}_q^n$*.*

2. *Fix* $\mathbf{y} \in \mathbb{Z}_q^n$*, and let* $\mathbf{t}$ *be an arbitrary vector in* $\mathbb{Z}^m$ *such that* $\mathbf{A} \cdot \mathbf{t} = \mathbf{y} \bmod q$*. Then the conditional distribution of vector* $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$ *given* $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ *is* $\mathbf{t} + D_{\Lambda_q^{\perp}(\mathbf{A}),\sigma,-\mathbf{t}}$*.*

3. *The min-entropy of* $D_{\mathbb{Z}^m,\sigma}$ *is at least* $m - 1$*.*

4. *For* $t \geq \omega(\sqrt{\log n})$ *and for* $\beta = \sigma \cdot t$*:* $\Pr_{x \hookleftarrow D_{\mathbb{Z},\sigma}}\big[\,|x| > \beta\,\big]$ *is negligible.*

2.2.4 The SIS and ISIS Problems.

In the seminal work that laid the first foundation for lattice-based cryptography [4], Ajtai introduced the Small Integer Solution (SIS) problem, which essentially asks to find a small non-zero vector in a random q-ary lattice $\Lambda_q^{\perp}(\mathbf{A})$. The problem later was formalized in [113], as follows:

Definition 2.2.11 ($\mathsf{SIS}^p(n, m, q, \beta)$, [113])**.** The Small Integer Solution problem (in the ℓ_p norm) with parameters n, m, q, β, denoted by $\mathsf{SIS}^p(n, m, q, \beta)$, is as follows: Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, find a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_p \leq \beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q$.

The SIS problem can also be interpreted as the problem of finding a small nontrivial solution of a random *homogeneous* system of modular linear equations. Gentry et al. [62] later formalized its *inhomogeneous* version, called the ISIS problem.

Definition 2.2.12 ($\mathsf{ISIS}^p(n, m, q, \beta)$, [62])**.** The Inhomogeneous Small Integer Solution problem (in the ℓ_p norm) with parameters n, m, q, β, denoted by $\mathsf{ISIS}^p(n, m, q, \beta)$, is as follows: Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, and a uniformly random vector $\mathbf{y} \in \mathbb{Z}_q^n$, find a vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_p \leq \beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$.

Parameters q and m are usually $\mathsf{poly}(n)$. The bound β should be set large enough (e.g., $\beta = \mathcal{O}(\sqrt{n \log q})$) to make sure that the problem has at least one solution. For fixed n, m and q, as β increases, SIS and ISIS become easier, and if $\beta \geq q$, then the problems are trivially easy. Ajtai proved an intriguing wosrt-case/average-case connection: for appropriate parameters setting, solving a *random* instance of the SIS^2 problem is at least as hard as approximating several lattice problems on *any* n-dimensional lattice, to within $\mathsf{poly}(n)$ factors. A series of subsequent works ([141, 107, 113, 62, 112]) provided better parameters that guarantee hardness, and

smaller approximation factor for the underlying lattice problems. In the theorem below, we state the result by Gentry et al. [62]. We note that a recent result by Micciancio and Peikert [112] ensures that the worst-case to average-case reduction still holds true for smaller parameter q, e.g., $q \geq \beta \cdot n^{\delta}$, where δ is some constant.

Theorem 2.2.13 (Hardness of SIS^2 and ISIS^2, [62])**.** *For any m, $\beta = \text{poly}(n)$, and for any integer $q \geq \beta \cdot \omega(\sqrt{n \log n})$, solving a* random *instance of the $\mathsf{SIS}^2(n, m, q, \beta)$ problem or the $\mathsf{ISIS}^2(n, m, q, \beta)$ problem with non-negligible probability is at least as hard as approximating the SIVP^2_γ problem on* any *lattice of dimension n to within certain $\gamma = \beta \cdot \widetilde{\mathcal{O}}(\sqrt{n})$ factors.*

Since SIVP^2_γ is assumed to be hard to approximate for any $\gamma = \mathsf{poly}(n)$ (see Conjecture 2.2.8), it follows that $\mathsf{SIS}^2(n, m, q, \beta)$ and $\mathsf{ISIS}^2(n, m, q, \beta)$ are believed to be hard to solve for any polynomial $\beta = \mathsf{poly}(n)$. However, in designing cryptographic schemes based on the hardness of these problems, one would expect to obtain a security guarantee for β as small as possible.

In this thesis, we will mainly work with the SIS and ISIS problems in the ℓ_∞ norm. On one hand, this restriction does not make the problems easier. Indeed, as noted in [112], there are theoretical and empirical evidences to believe that, for sufficiently small q, the SIS problem, where solutions are restricted in the ℓ_∞ norm instead of the ℓ_2 norm, may be "qualitatively harder". On the other hand, in many cryptographic constructions based on the SIS and ISIS problems, defining the range of the solutions in term of the ℓ_∞ norm rather than the ℓ_2 norm seems to be more convenient. The worst-case to average-case reduction in Theorem 2.2.14 only address the hardness of SIS^2 and ISIS^2 problems, but one can easily derive a hardness guarantee for the SIS^∞ and ISIS^∞ via the relationship between the ℓ_2 and the ℓ_∞ norms (i.e., for any vector $\mathbf{x} \in \mathbb{R}^m$, we have $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{m} \cdot \|\mathbf{x}\|_\infty$).

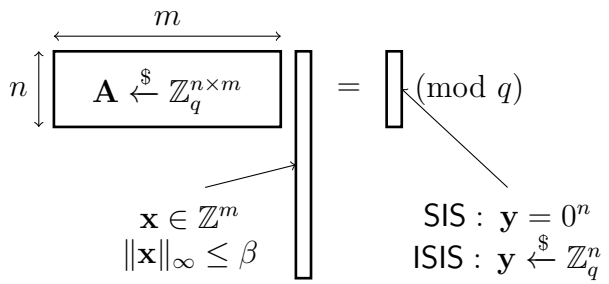

Fig. 2.1: The SIS^∞ and ISIS^∞ problems with parameters (n, m, q, β). Given $(\mathbf{A}, \mathbf{y})$, the goal is to find $\mathbf{x}$.

This result is stated in the following theorem.

Theorem 2.2.14 (Hardness of SIS^∞ and ISIS^∞, as a corollary of Theorem 2.2.13)**.** *For any m, $\beta = \mathsf{poly}(n)$, and for any integer $q \geq \beta \cdot \omega(\sqrt{n \log n})$, solving a* random *instance of the $\mathsf{SIS}^\infty(n, m, q, \beta)$ problem or the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ problem with non-negligible probability is at least as hard as approximating SIVP_γ^2 (in the ℓ_2 norm) on* any *lattice of dimension n to within certain $\gamma = \beta \cdot \widetilde{\mathcal{O}}(n)$ factors.*

Without loss of generality, throughout this thesis, we will set the ℓ_∞ norm bound β (of the SIS and ISIS solutions) as a positive *integer*. In Figure 2.1, we give an illustration of the SIS^∞ and ISIS^∞ problems with parameters (n, m, q, β).

2.2.5 The LWE *Problem.*

The Learning With Errors (LWE) problem was introduced in the seminal work of Regev [127], as the problem of decoding from random q-ary linear codes. In the special case of $q = 2$, it corresponds to a well-known problem in coding theory, namely, the Learning Parity with Noise problem. The LWE problem has two versions, the search and the decision ones, defined as follows.

Definition 2.2.15 ($\mathsf{LWE}(n, q, \chi)$, [127])**.** Let $n \geq 1$, $q \geq 2$, and let χ be a probability distribution on $\mathbb{Z}$ (or its induced distribution over $\mathbb{Z}_q$). For $\mathbf{s} \in \mathbb{Z}_q^n$, let $A_{\mathbf{s},\chi}$ be the

distribution obtained by sampling $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_q^n$ and an error $e \hookleftarrow \chi$, and outputting the pair $(\mathbf{a}, \mathbf{a}^T \cdot \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.

- The Search-LWE problem is as follows: Given m independent samples from $\mathcal{A}_{\mathbf{s},\chi}$ (for some $\mathbf{s} \in \mathbb{Z}_q^n$) , find $\mathbf{s}$ with non-negligible probability.
- The Decision-LWE problem asks to distinguish (with non-negligible advantage) m samples chosen according to $\mathcal{A}_{\mathbf{s},\chi}$ (for $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$) and m samples chosen according to the uniform distribution over $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

In many cryptographic applications, we are mainly interested in the LWE problem where the errors sampled from χ are sufficiently small, i.e., their magnitudes are bounded by a number $\beta \ll q$ with overwhelming probability.

Definition 2.2.16 (β-bounded distribution, [31, 63])**.** A distribution χ over the integers is β-bounded (denoted by $|\chi| \leq \beta$) if for any $e \hookleftarrow \chi$, $\Pr[|e| > \beta] = \mathsf{negl}(n)$.

It is usually more convenient to describe the $\mathsf{LWE}(n, q, \chi)$ problem in matrix form. Let $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, and let $\mathbf{b} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e} \bmod q$, where $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e} \hookleftarrow \chi^m$ (see Figure 2.2 for an illustration). Then the Search-LWE problem asks to find $\mathbf{s}$ given $(\mathbf{A}, \mathbf{b})$, and the Decision-LWE asks to distinguish $(\mathbf{A}, \mathbf{b})$ from a uniformly random pair over $\mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$. Stating the LWE this way, one can observe a "connection" between LWE and q-ary lattices: If χ is a β-bounded distribution, where β is sufficiently small, then $\mathbf{b}$ can be seen as a perturbed lattice point in the lattice $\Lambda_q(\mathbf{A})$. We now review several facts from the literature about the hardness of LWE.

In his ground-breaking work [127], Regev proved that, for $\chi = D_{\mathbb{Z},\alpha \cdot q}$[1], where $\alpha \in (0, 1)$ such that $\alpha \cdot q > 2\sqrt{n}$ (and thus, χ is β-bounded for $\beta \geq \alpha \cdot q \cdot \omega(\log n)$),

[1] We note that, the hardness result from [127] is for a *discretized* Gaussian distribution, which does not not produce a true discrete Gaussian. Later, a randomized rounding technique by Peikert [119] resolved this issue.

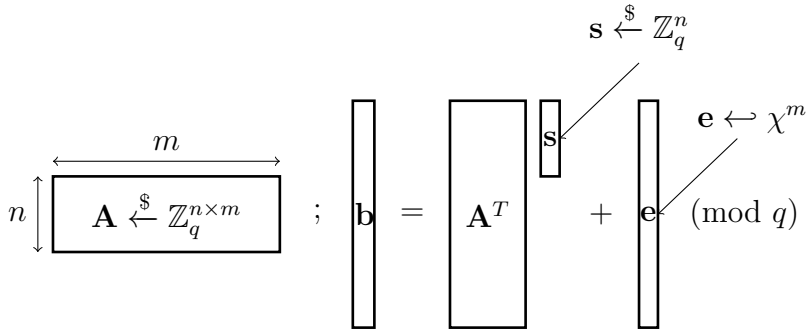

Fig. 2.2: An $\mathsf{LWE}(n, q, \chi)$ instance with m samples

the Search-LWE problem is as least as hard as *quantumly* approximating SIVP_γ^2 on n-dimensional lattices to within $\widetilde{\mathcal{O}}(n/\alpha)$ factors. Regev also showed that the Search-LWE and Decision-LWE variants are essentially *equivalent* for prime $q = \mathsf{poly}(n)$. The search-to-decision reduction was later improved in [118, 110, 111], so that it works for a wider class of moduli and error distributions. The following theorem, stated in [31, 63], summarizes the quantum hardness results of LWE from these works.

Theorem 2.2.17 (Quantum hardness of LWE, [127, 118, 110, 111])**.** *Let $q = q(n)$ be either a prime power or a product of small (i.e., size $\mathsf{poly}(n)$) distinct primes, and let $\beta \geq \sqrt{n} \cdot \omega(\log n)$. Then there exists an efficient sampleable β-bounded distribution χ such that: if there is an efficient algorithm solving the average-case problem $\mathsf{LWE}(n, q, \chi)$, then there is an efficient* quantum *algorithm that approximates SIVP_γ^2 on any n-dimensional lattice, to within certain $\gamma = \widetilde{\mathcal{O}}(n \cdot q/\beta)$ factors.*

The worst-case to average-case reduction in the above theorem is similar to the one for the SIS and ISIS problems (see Theorem 2.2.13), except for the "quantum" part. Even though lattice problems are assumed to be resistant against quantum computer (see Conjecture 2.2.8), this reduction is quite unsatisfactory. There have been several efforts to establish classical hardness for the LWE problem, most notably the results by Peikert [118] and by Brakerski et al. [34].

Theorem 2.2.18 (Classical hardness of LWE, [118, 34])**.** *Assume that there exists an efficient algorithm solving the n-dimensional* LWE *problem (i.e., the secret $\mathbf{s} \in \mathbb{Z}_q^n$). Then:*

- *From [118]: For $q \geq \widetilde{\mathcal{O}}(2^{n/2})$, there is an efficient algorithm for* GapSVP *on any lattice of dimension n.*
- *From [34]: For $q = \mathsf{poly}(n)$, there is an equally efficient algorithm for* GapSVP *on any lattice of dimension $\sqrt{n}$.*

It can be seen that the reduction of [118] requires exponential q, while the reduction of [34] admits a quadratic loss in the dimension. Currently, establishing a classical hardness for LWE for commonly used parameters (as specified in Theorem 2.2.17) remains an challenging problem.

2.3 Basic Lattice-based Cryptographic Primitives

In this section, we review some basic lattice-based cryptographic primitives that will be used in the thesis. These primitives are one-way functions, commitment schemes, trapdoors for lattices, signature schemes and encryption schemes.

2.3.1 One-way Functions

The assumed hardness of the SIS, ISIS, and LWE problems implies the existence of two families of one-way functions (i.e., functions that are easy to compute, but hard to invert), which are the fundamental building blocks of lattice-based cryptography. In the following, we let $q = \mathsf{poly}(n)$, let $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, and let χ be a β-bounded distribution (e.g., uniform distribution or discrete Gaussian distribution with suitable parameters).

The ISIS Function ([4, 62]). The function is defined by $f_{\mathbf{A}}(\mathbf{x}) := \mathbf{A}\cdot\mathbf{x} \bmod q$, where the input distribution is χ^m. For commonly used parameters (e.g., $m = \mathcal{O}(n \log q)$, and $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$), this function has the following properties:

1. $f_{\mathbf{A}}$ is surjective, and the distribution of $f_{\mathbf{A}}(\mathbf{x})$ is statistically close to uniform over the range $\mathbb{Z}_q^n$ (see Lemma 2.2.10, item (1)).

2. $f_{\mathbf{A}}$ is one-way, assuming that the $\mathsf{ISIS}^{\infty}(n, m, q, \beta)$ problem is hard. Indeed, inverting $f_{\mathbf{A}}$ on a uniform output $\mathbf{y} \in \mathbb{Z}_q^n$ is syntactically equivalent to solving the $\mathsf{ISIS}^{\infty}(n, m, q, \beta)$ problem.

3. $f_{\mathbf{A}}$ is collision-resistant, assuming that the $\mathsf{SIS}^{\infty}(n, m, q, 2\beta)$ problem is hard. Indeed, a collision $\mathbf{x}_1, \mathbf{x}_2$ such that $\|\mathbf{x}_1\|_\infty, \|\mathbf{x}_2\|_\infty \leq \beta$ and $\mathbf{A}\cdot\mathbf{x}_1 = \mathbf{A}\cdot\mathbf{x}_2 \bmod q$ will imply a solution for the $\mathsf{SIS}^{\infty}(n, m, q, 2\beta)$ instance $\mathbf{A}$: Let $\mathbf{z} = \mathbf{x}_1 - \mathbf{x}_2$, then $\|\mathbf{z}\|_\infty \leq 2\beta$ and $\mathbf{A} \cdot \mathbf{z} = \mathbf{0} \bmod q$.

The LWE Function ([127]). The function is defined by $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) := \mathbf{A}^T\cdot\mathbf{s}+\mathbf{e} \bmod q$, where the input distributions are $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, and $\mathbf{e} \hookleftarrow \chi^m$. For commonly used parameters, the function has the following properties:

1. $g_{\mathbf{A}}$ is injective, and the distribution of $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e})$ is indistinguishable from uniform distribution over the range $\mathbb{Z}_q^m$, assuming that the $\mathsf{Decision\text{-}LWE}(n, q, \chi)$ problem is hard (see Theorem 2.2.17).

2. $g_{\mathbf{A}}$ is one-way, assuming that the $\mathsf{Search\text{-}LWE}(n, q, \chi)$ problem is hard. Indeed, finding $\mathbf{s}$, given $\mathbf{b} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e} \bmod q$ is syntactically equivalent to solving the $\mathsf{Search\text{-}LWE}(n, q, \chi)$ problem.

It can be seen that the ISIS and LWE function families share common properties. Interestingly, there are observations that these two fundamental function families are essentially equivalent under certain conditions [110, 112].

2.3.2 Commitment Scheme

Commitment scheme is a cryptographic primitive that allows a sender to commit a chosen value, so that the value is hidden from the receiver, but the sender has the ability to reveal it later. Commitment schemes are widely used in constructing zero-knowledge proofs: the prover commits all the information in advance in a commitment; the verifier then specifies a choice of what it wants to learn; and the prover will only reveals the information corresponding to the verifier's choice.

In particular, a commitment scheme COM works in two phases. In the committing phase, the sender, who wants to commit a value s, computes $\mathbf{c} = \mathrm{COM}(s, \rho)$, where ρ is a random string, and sends $\mathbf{c}$ to the receiver. In the revealing phase, the sender "open" $\mathbf{c}$ by revealing s and ρ, so that the receiver can compute $\mathrm{COM}(s, \rho)$ and check the validity of $\mathbf{c}$. In this thesis, we will be working with a *string* commitment scheme (i.e. the committed value is a string $s \in \{0,1\}^*$) that satisfies two security requirements: statistical hiding property and computational binding property. Roughly speaking, a commitment scheme is statistically hiding if any computationally unbounded cheating receiver cannot distinguish the commitments of two different strings. It is computationally binding if any polynomial-time cheating sender cannot change the committed string after the commitment phase. More formally, these two security notions are defined as follows.

Definition 2.3.1 ([65])**.** A statistically hiding, computationally binding string commitment scheme is a PPT algorithm $\mathrm{COM}(s, \rho)$ satisfying:

- For all $s_0, s_1 \in \{0,1\}^*$, we have (over the random coins of COM):

$$\mathrm{COM}(s_0; \cdot) \approx_s \mathrm{COM}(s_1; \cdot),$$

- For all probabilistic polynomial-time algorithm $\mathcal{A}$ returning $(s_0, \rho_0); (s_1, \rho_1)$, where $s_0 \neq s_1$, we have (over the random coins of $\mathcal{A}$):

$$\Pr[\mathrm{COM}(s_0; \rho_0) = \mathrm{COM}(s_1; \rho_1)] = \mathrm{negl}(n).$$

There are generic constructions of statistically hiding and computationally binding string commitment schemes from a family of collision-resistant hash functions (see [78, 52]). Kawachi, Tanaka and Xagawa [85] gave a more direct and simpler construction from lattices. In this thesis, we will refer to this scheme as the KTX string commitment scheme.

The KTX string commitment scheme was built upon an ISIS function $f_{\mathbf{A}}(\mathbf{x}) = \mathbf{A} \cdot \mathbf{x} \bmod q$, where $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times \bar{m}}$ and $\mathbf{x} = (s \| \rho)$, with $s, \rho \in \{0,1\}^{\bar{m}/2}$. Parameter $\bar{m}$ is set sufficiently large so that the output distribution of $f_{\mathbf{A}}$ is statistically close to uniform over the range $\mathbb{Z}_q^n$ (via the left-over hash lemma, see [127]). As mentioned in Section 2.3.1, for suitable choice of parameters, the function is collision-resistant based on the assumed hardness of the $\mathsf{SIS}^{\infty}(n, \bar{m}, q, 2)$ problem. These observations give rise to a statistically hiding and computationally binding commitment for values $s \in \{0,1\}^{\bar{m}/2}$. In [85], the authors use the Merkle-Damgard construction [105, 50] to extend the domain to $\{0,1\}^*$. The result is a string commitment scheme $\mathrm{COM} : \{0,1\}^* \times \{0,1\}^{\bar{m}/2} \to \mathbb{Z}_q^n$, whose security aspects are summarized in the following theorem.

Theorem 2.3.2 (Adapted from [85])**.** *Let* COM *be the* KTX *string commitment scheme.*

- *If* $\bar{m} > 2n(1+\delta)\log q$ *for some positive constant* δ*, then* COM *is statistically hiding.*
- *If the* $\mathsf{SIS}^{\infty}(n, \bar{m}, q, 2)$ *problem is hard, then* COM *is computationally binding.*

In this thesis, we will extensively use the KTX commitment scheme. For simplicity, sometimes we will omit the randomness ρ of the commitment.

2.3.3 Trapdoors for Lattices

At the heart of many lattice-based cryptographic constructions is an algorithm that, given $(1^n, 1^m, q)$, generates a matrix $\mathbf{A}$ statistically close to uniform over $\mathbb{Z}_q^{n\times m}$ together with a short basis $\mathbf{S}$ for the lattice $\Lambda_q^\perp(\mathbf{A})$. This fundamental algorithm, called $\mathsf{GenTrap}$, was first introduced by Ajtai [6], then was improved by Alwen and Peikert [12], and recently was simplified by Micciancio and Peikert [111]. The following theorem is adapted from the results in [12] and [111].

Theorem 2.3.3 (Adapted from [12, 111])**.** *Let $n \geq 1$, $q \geq 2$ and $m \geq 2n\log q$ be integers. There is a probabilistic polynomial-time algorithm $\mathsf{GenTrap}(1^n, 1^m, q)$ that outputs a matrix $\mathbf{A}$ statistically close to uniform over $\mathbb{Z}_q^{n\times m}$, and a basis $\mathbf{S} \in \mathbb{Z}^{m\times m}$ for the lattice $\Lambda_q^\perp(\mathbf{A})$, such that $\|\widetilde{\mathbf{S}}\| \leq L$ for certain $L = \mathcal{O}(\sqrt{n\log q})$.*

A short basis for $\Lambda_q^\perp(\mathbf{A})$ is called a lattice *trapdoor*, since the knowledge of such basis can help to efficiently invert the ISIS and LWE functions indexed by $\mathbf{A}$, as pointed out in [62].

Inverting the ISIS Function. The ISIS inverting algorithm $\mathsf{SampleD}$ is described as in the following theorem.

Theorem 2.3.4 ([62])**.** *Let $q \geq 2$ and $m \geq n$. Let $\mathbf{A}$ be a matrix in $\mathbb{Z}_q^{n\times m}$ and $\mathbf{S}$ be a basis for $\Lambda_q^\perp(\mathbf{A})$. Then for $\mathbf{y}$ in the image of $\mathbf{A}$ and $\sigma \geq \|\widetilde{\mathbf{S}}\|\omega(\sqrt{\log m})$, there is a probabilistic polynomial-time algorithm $\mathsf{SampleD}(\mathbf{S}, \mathbf{A}, \mathbf{y}, \sigma)$ that outputs $\mathbf{x} \in \mathbb{Z}^m$ sampled from the distribution $\mathrm{D}_{\mathbb{Z}^m,\sigma}$, conditioned on $\mathbf{A}\cdot\mathbf{x} = \mathbf{y} \bmod q$.*

Specifically, algorithm $\mathsf{SampleD}(\mathbf{S}, \mathbf{A}, \mathbf{y}, \sigma)$ performs the following steps:

- Compute via linear algebra a vector $\mathbf{t} \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{t} = \mathbf{y} \bmod q$. Since $\mathbf{y}$ belongs to the image of $\mathbf{A}$, such vector $\mathbf{t}$ must exist.
- Using the basis $\mathbf{S}$ of $\Lambda_q^{\perp}(\mathbf{A})$, sample $\mathbf{v} \hookleftarrow D_{\Lambda_q^{\perp}(\mathbf{A}),\sigma,-\mathbf{t}}$ (see Lemma 2.2.9).
- Return $\mathbf{x} = \mathbf{v} + \mathbf{t}$. It can be checked that $\mathbf{A} \cdot \mathbf{x} = \mathbf{A} \cdot \mathbf{v} + \mathbf{A} \cdot \mathbf{t} = \mathbf{A} \cdot \mathbf{t} = \mathbf{y} \bmod q$. Moreover, it follows from Lemma 2.2.10 (item (2)), that the conditional distribution of $\mathbf{x}$, given $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ is $D_{\mathbb{Z}^m,\sigma}$.

Inverting the LWE Function. Let $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{b} \in \mathbb{Z}_q^m)$, be an 'LWE instance', namely, there exist "secret" vector $\mathbf{s} \in \mathbb{Z}_q^n$ and "small" error vector $\mathbf{e} \in \mathbb{Z}_q^m$, such that $\mathbf{b} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e} \bmod q$. Then one can use a "short" basis $\mathbf{S}$ of $\Lambda_q^{\perp}(\mathbf{A})$ to efficiently recover $\mathbf{s}$ (and $\mathbf{e}$), as follows:

1. Compute $\mathbf{b}' = \mathbf{S}^T \cdot \mathbf{b} \bmod q = \mathbf{S}^T \cdot (\mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}) \bmod q = \mathbf{S}^T \cdot \mathbf{e} \bmod q$. Since both $\mathbf{S}$ and $\mathbf{e}$ contain only "small" entries, while $q = \mathsf{poly}(n)$ is sufficiently large, we have $\mathbf{b}' = \mathbf{S}^T \cdot \mathbf{e}$ (over the *integers*).
2. Recover $\mathbf{e}$ by computing $(\mathbf{S}^T)^{-1} \cdot \mathbf{b}'$. Having known $\mathbf{e}$, it is easy to recover $\mathbf{s}$.

Lattice trapdoors play a prominent role in many cryptographic constructions in this thesis. The 2 signature schemes that we will describe in the next section are based on trapdoors. Furthermore, the schemes presented in Chapter 5 and Chapter 6 also employ lattice trapdoors to invert the underlying ISIS and LWE functions.

2.3.4 Lattice-based Signature Schemes

In this section, we will give an overview of existing lattice-based signature schemes, and describe 2 schemes that are related to the later constructions in the thesis. First, we will recall the standard definition and security requirements of signature schemes, and the background on the random oracle model.

Signature Schemes.

Definition 2.3.5. A signature scheme $\mathcal{SS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verifiy})$ is a triple of polynomial-time (possibly probabilistic) algorithms:

- Key generation algorithm $\mathsf{KeyGen}(1^n)$: On input 1^n, where n is the security parameter, output a public verification key pk and a secret signing key sk.
- Signing algorithm $\mathsf{Sign}(\mathsf{sk}, M)$: On input sk and a message M, output a signature Σ.
- Verification algorithm $\mathsf{Verify}(\mathsf{pk}, M, \Sigma)$: On input pk, a message M, and a purported signature Σ on M, output 1 (Valid) or 0 (Invalid).

Correctness: The correctness requirement for a signature scheme $\mathcal{SS}$ is as follows: for any $(\mathsf{pk}, \mathsf{sk})$ outputted by $\mathsf{KeyGen}(1^n)$, and any message M,

$$\Pr\big[\mathsf{Verify}\big(\mathsf{pk}, M, \mathsf{Sign}(\mathsf{sk}, M)\big) = 1\big] = 1 - \mathsf{negl}(n),$$

where the probability is taken over the randomness of all algorithms.

Security Notions: The standard security notion for signature schemes is existential unforgeability (eu) under (*static* or *adaptive*) chosen message attacks (scma or acma, respectively), which roughly means that any polynomial-time adversary, after seeing signatures of messages of its choice, should not be able to output a valid signature for a new message. More precisely, depending on the attack model, the security experiment between the challenger $\mathcal{C}$ and the forger $\mathcal{F}$ is defined differently:

- The eu-scma experiment: The forger $\mathcal{F}$ specifies in advance a list of query messages $M_1, \ldots, M_Q$, for some Q. The challenger $\mathcal{C}$ then generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^n)$, and $\Sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}, M_i)$ for each $i \in [Q]$, then gives pk and $\{\Sigma_i\}_{i=1}^{Q}$ to $\mathcal{F}$. Eventually, $\mathcal{F}$ outputs a forgery (M^*, Σ^*). The challenger outputs 1 if

and only if $\mathsf{Verify}(\mathsf{pk}, M^*, \Sigma^*) = 1$ and $M^* \neq M_i$ for all $i \in [Q]$. Otherwise, it outputs 0.

- The eu-acma experiment: The challenger $\mathcal{C}$ generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^n)$, then gives pk to $\mathcal{F}$. The forger can adaptively query for signature of message M_i in the i-th query ($i \in [Q]$), and $\mathcal{C}$ answers with $\Sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}, M_i)$. Eventually, $\mathcal{F}$ outputs a forgery (M^*, Σ^*). The challenger outputs 1 if and only if $\mathsf{Verify}(\mathsf{pk}, M^*, \Sigma^*) = 1$ and $M^* \neq M_i$ for all i. Otherwise, it outputs 0.

In the eu-scma experiment (eu-acma experiment, respectively) above, the advantage of $\mathcal{F}$ is defined as the probability that $\mathcal{C}$ outputs 1, taken over the randomness of all algorithms. The scheme is called eu-scma secure (eu-acma secure, respectively), if the advantage of $\mathcal{F}$ is $\mathsf{negl}(n)$ for any polynomial-time $\mathcal{F}$.
A stronger security notion, called *strong unforgeability* (seu), additionally requires that the forger cannot come up with a new signature for a previously signed message. Namely, at the end of the experiment, the challenger outputs 1 if and only if $\mathsf{Verify}(\mathsf{pk}, M^*, \Sigma^*) = 1$, and $(M^*, \Sigma^*) \neq (M_i, \Sigma_i)$ for all $i \in [Q]$. Thus, the strongest security notion for signature schemes is seu-acma.

On the other hand, constructing signature schemes satisfying strong security notions is usually involved, and thus, a weaker notion is considered [89]: unforgeability under a *one-time* chosen message attack, i.e., in the experiment, the forger is allowed to make at most one signing query. Schemes only satisfies this weak notion is called one-time signatures.

When designing signature schemes, it is desirable to prove the security of the schemes in *the standard model*. However, in many situations, it seems impossible to obtain a security proof in the standard model. In these cases, one can alternatively consider the security of the scheme in *the random oracle model*.

The Random Oracle Model.

The random oracle model (ROM), proposed by Bellare and Rogaway [21], is a security model, in which a hash function $\mathsf{H} : \{0,1\}^* \to \{0,1\}^{\mu(n)}$ can be be modeled as a truly random function (where $\{0,1\}^{\mu(n)}$ is the output range specified in the context). More precisely, for a new query $x \in \{0,1\}^*$, H is supposed to answer with a uniformly random $y \xleftarrow{\$} \{0,1\}^{\mu(n)}$, and from this point, it has to answer with consistency, namely, $\mathsf{H}(x)$ is always y.

An interesting feature of the ROM is that: In the security proof, the reduction algorithm is allowed to *program* the random oracle, as long as the behavior of H satisfies the above requirements. Since its introduction in 1993, the ROM have become a very powerful cryptographic tool, especially for proving the security of signature schemes. In general, signature schemes, that are provably secure in the ROM, belong to one of the two groups. The first group consists of schemes following the Full Domain Hash approach [22], and the second groups are schemes obtained from an interactive proof system via the Fiat-Shamir heuristic [58].

On the other hand, the ROM is a controversial concept, as a truly random function cannot be implemented in real world. Actually, there are examples of schemes that are secure in the ROM, but become insecure if H is replaced with a real function [41]. Nevertheless, in many situations, the random oracle seems unavoidable, and achieving a security proof in the ROM is better than not modelling the security of the scheme at all. Moreover, cryptographic schemes, that are provably secure in the ROM, tend to be more efficient than their counterparts in the standard model.

Having described the fundamental concepts, we now will review lattice-based signa-

ture schemes.

Overview of Lattice-based Signature Schemes.

The development of lattice-based signature schemes was inspired by the celebrated work of Ajtai [4]. The first scheme, whose security is related to the hardness of lattice problems, is the GGH signature, proposed by Goldreich et al. [69]. Later, Hoffstein et al. [79] introduced NTRUSign - a signature scheme based on the design principle of the GGH signature, but operates in compact NTRU lattices. These earlier schemes are relatively efficient but did not come with a security proof. Subsequently, NTRUSign was partially shown insecure by Gentry and Szydlo [64], and then both NTRUSign and GGH were completely broken by Nguyen and Regev [117]. On the other hand, the last decade has witnessed a significant progress on designing provably secure lattice-based signature schemes. In general, these schemes can be divided into two categories: the ones that work without a lattice trapdoor (usually built upon a proof system), and the ones based on trapdoors.

In the first category, the earliest scheme was (implicitly) suggested by Micciancio and Vadhan [115], who constructed statistical zero-knowledge proof systems for GapSVP_γ and GapCVP_γ (see also Section 3.2.1): those proof systems yield secure identification schemes, which can be converted into secure signature schemes in the random oracle model via Fiat-Shamir heuristic [58]. Lyubashevsky and Micciancio [98] later proposed an efficient one-time signature scheme based on the hardness of approximating SVP^∞ in ideal lattices. (A full-fledged tree-based signature can be obtained via a standard transformation [104].) Subsequently, a series of work by Lyubashevsky [94, 95, 96] introduced signatures schemes based on efficient proof systems for the ISIS problem (see also Section 3.2.2). Using clever aborting and rejection sampling techniques, Lyubashevsky transformed the underlying proof sys-

tems to signature schemes in the random oracle model via Fiat-Shamir heuristic. We remark that signature schemes in this category are currently the most efficient provably secure schemes known to date, and are gradually coming closer to practice.

Lattice-based signature schemes in the second category, i.e., signatures with trapdoors, are not as efficient as their counterparts in the first category. Nevertheless, these schemes got significant attention from the community thanks to their theoretical attractiveness. In the following, we will review 2 trapdoor-based schemes that will be used in the later constructions in this thesis: the Gentry et al.'s scheme from [62], and the Cash et al.'s scheme from [42].

The GPV Signature Scheme.

In [62], Gentry, Peikert and Vaikuntanathan introduced the first signature scheme that makes use of a lattice trapdoor. In this thesis, we will refer to this scheme as the GPV signature scheme. At a high level, the authors proposed a novel abstraction called *preimage sampleable (trapdoor) functions* (PSFs), which contains the following algorithms:

- Algorithm GenTrap (see Theorem 2.3.3) generates an ISIS function $f_{\mathbf{A}}$, together with a lattice trapdoor. As mentioned in Section 2.3.1, for suitable settings, the function $f_{\mathbf{A}}$ is one-way and collision-resistant based on the assumed hardness of ISIS and SIS, respectively.

- Algorithm SampleD (see Theorem 2.3.4) allows to invert $f_{\mathbf{A}}$ on a uniform output, given the trapdoor.

As shown by Gentry et al. [62], the PSFs allow to construct Full-Domain Hash signature and identity-based encryption schemes that are secure in the ROM. In

this thesis, we are mainly interested in their signature scheme, which is described as follows.

Scheme 2.3.6 (The GPV signature scheme, [62])**.** Let $q = \mathsf{poly}(n)$, $m \geq 2n \log q$, and $\sigma = \omega(\sqrt{n \log q \log n})$. Let $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_q^n$ be a random oracle.

$\mathsf{Keygen}(1^n)$: Let $(\mathbf{A}, \mathbf{S}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$. Output $\mathsf{pk} = \mathbf{A}$ and $\mathsf{sk} = \mathbf{S}$.

$\mathsf{Sign}(\mathbf{S}, M \in \{0,1\}^*)$: Let $\mathbf{x} \leftarrow \mathsf{SampleD}(\mathbf{S}, \mathbf{A}, \mathsf{H}(M), \sigma)$. If $\|\mathbf{x}\| > \sigma\sqrt{m}$ then resample $\mathbf{x}$. Otherwise, output $\Sigma = \mathbf{x}$.

$\mathsf{Verify}(\mathbf{A}, M, \Sigma)$: Let $\Sigma = \mathbf{x}$. Output 1 if $\mathbf{A} \cdot \mathbf{x} = \mathsf{H}(M) \bmod q$ and $\|\mathbf{x}\| \leq \sigma\sqrt{m}$. Otherwise, output 0.

It can be seen that the given scheme is correct. Its security in the ROM is stated in the following theorem.

Theorem 2.3.7 (Adapted from [62])**.** *Assuming that the* $\mathsf{SIS}^2(n, m, q, 2\sigma\sqrt{m})$ *problem is hard. Then the signature described in Scheme 2.3.6 is* seu-acma *secure in the random oracle model.*

By Theorem 2.2.13, the security of the GPV signature scheme can be based on the assumed hardness of SIVP_γ^2 with $\gamma = \widetilde{\mathcal{O}}(n^{1.5})$.

In this thesis, we will use the GPV signature scheme (where the domain of the ISIS solutions is defined in terms of the ℓ_∞ norm) to construct the identity-based identification schemes in Chapter 4.

The Bonsai Signature Scheme.

In [42], Cash et al. introduced a signature scheme that makes use of a novel structure of hard random lattices called *the Bonsai tree*. In this thesis, we will refer to this scheme as the Bonsai tree signature scheme.

In the Bonsai tree structure, one is said to have *control* of a lattice, if it knows a trapdoor for the lattice. Then the interesting fact is that, one may extend its control to an arbitrary higher-dimensional extension of the lattice, but not the other way around. For instance, consider the "tree" $\mathbf{A} = [\mathbf{A}_0|\bar{\mathbf{A}}] \in \mathbb{Z}_q^{n\times(m+\bar{m})}$, where $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ is its "root". If one has a relatively short basis $\mathbf{S}_0$ for $\Lambda_q^\perp(\mathbf{A}_0)$ (i.e., it has control of the "root"), then it can efficiently obtain a basis $\mathbf{S}$ for the super-lattice $\Lambda_q^\perp(\mathbf{A})$, which automatically means that it controls the "tree" $\mathbf{A}$. Moreover, the extension algorithm preserves the quality of the basis, namely: $\|\widetilde{\mathbf{S}}\| \leq \|\widetilde{\mathbf{S}}_0\|$. This algorithm, called $\mathsf{ExtBasis}$, is summarized in the following lemma.

Lemma 2.3.8 ([42]). *Let $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ and let $\mathbf{S}_0$ be a basis of the lattice $\Lambda_q^\perp(\mathbf{A}_0)$. Let $\mathbf{A} = [\mathbf{A}_0|\bar{\mathbf{A}}] \in \mathbb{Z}_q^{n\times(m+\bar{m})}$, where $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times\bar{m}}$ is arbitrary. Then there is a deterministic polynomial-time algorithm $\mathsf{ExtBasis}(\mathbf{S}_0, \mathbf{A})$ that outputs a basis $\mathbf{S}$ of $\Lambda_q^\perp(\mathbf{A})$ such that $\|\widetilde{\mathbf{S}}\| \leq \|\widetilde{\mathbf{S}}_0\|$.*

It can be checked that, given uniformly random $\mathbf{A}_0$, and any basis $\mathbf{S}'$ of $\Lambda_q^\perp(\mathbf{A})$, there is *no* efficient algorithm that produces a relatively short basis $\mathbf{S}'_0$ for $\Lambda_q^\perp(\mathbf{A}_0)$ (otherwise, the SIS problem associated with $\mathbf{A}_0$ would be easy to solve). Therefore, the Bonsai tree structure yields a hierarchy of lattice trapdoors, where control can be extended down the hierarchy (but not up). As shown by Cash et al. [42], this structure allows to construct hash-and-sign signature and hierarchical identity-based encryption schemes in the standard model. In the scope of this thesis, we focus solely on their signature scheme, which is described as follows.

Scheme 2.3.9 (The Bonsai tree signature scheme, [42]). Let ℓ be the (hashed) message length (i.e., the output length of the function that hashes the original message). Let $q = \mathsf{poly}(n)$, $m \geq 2n\log q$, $m' = (\ell+1)\cdot m$, and $\sigma = \omega(\sqrt{n\log q \log n})$.

$\mathsf{KeyGen}(1^n)$:

1. Run $\mathsf{GenTrap}(1^n, 1^m, q)$ to obtain $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ together with a short basis $\mathbf{S}_0$ of $\Lambda_q^{\perp}(\mathbf{A}_0)$.
2. For each $(j, b) \in [\ell] \times \{0, 1\}$, pick $\mathbf{A}_j^b \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$.
3. Output $\mathsf{pk} = (\mathbf{A}_0, \mathbf{A}_1^0, \mathbf{A}_1^1, \ldots, \mathbf{A}_\ell^0, \mathbf{A}_\ell^1) \in \left(\mathbb{Z}_q^{n\times m}\right)^{2\ell+1}$, and $\mathsf{sk} = \mathbf{S}_0 \in \mathbb{Z}^{m\times m}$.

$\mathsf{Sign}(\mathsf{sk}, \mu \in \{0, 1\}^\ell)$:

1. Let $\mathbf{A}_\mu = \left[\mathbf{A}_0 | \mathbf{A}_1^{\mu[1]} | \ldots | \mathbf{A}_\ell^{\mu[\ell]}\right] \in \mathbb{Z}_q^{n\times m'}$, where for $i \in [\ell]$, $\mu[i] \in \{0, 1\}$ is the i-th bit of μ.
2. Run $\mathsf{ExtBasis}(\mathbf{S}_0, \mathbf{A}_\mu)$ to obtain a short basis $\mathbf{S}_\mu$ of $\Lambda_q^{\perp}(\mathbf{A}_\mu)$.
3. Let $\mathbf{x} \hookleftarrow \mathsf{SampleD}(\mathbf{S}_\mu, \mathbf{A}_\mu, \mathbf{0}, \sigma)$. If $\|\mathbf{x}\| = 0$ or $\|\mathbf{x}\| > \sigma\sqrt{m'}$, then resample $\mathbf{x}$.
4. Output the signature $\mathbf{x}$.

$\mathsf{Verify}(\mathsf{pk}, \mu, \mathbf{x})$: Let $\mathbf{A}_\mu$ be as above. Output 1 if and only if $0 < \|\mathbf{x}\| \leq \sigma\sqrt{m'}$ and $\mathbf{A}_\mu \cdot \mathbf{x} = \mathbf{0} \bmod q$. Otherwise, output 0.

It can be seen that the scheme is correct. Its security in the standard model is stated in the following theorem.

Theorem 2.3.10 (Adapted from [42])**.** *Let $\mathcal{F}$ be a forger mounting an* eu-scma *attack on the signature described in Scheme 2.3.9 with advantage ϵ and making at most Q queries. Then there exists a probabilistic polynomial-time algorithm $\mathcal{S}$ solving the $\mathsf{SIS}^2(n, m', q, \sigma\sqrt{m'})$ problem with probability $\epsilon' \geq \epsilon/(\ell \cdot Q) - \mathsf{negl}(n)$.*

By Theorem 2.2.13, the security of the given scheme can be based on the assumed hardness of SIVP_γ^2 with $\gamma = \sqrt{\ell} \cdot \widetilde{\mathcal{O}}(n^{1.5})$.

We note that the given signature scheme is only proved to be eu-scma secure. Nevertheless, the scheme can be adapted to achieve stronger security notions. First, it was noted in [42] that by employing a chameleon hash function [86], the scheme

can be made eu-acma secure. On the other hand, Rückert [132] later observed that the scheme is not *strongly* unforgeable: If vector $\mathbf{x}$ is a signature of μ, then it can be checked that vector $(-\mathbf{x})$ is also a valid signature of μ. As suggested in [132], by adding an extra vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$ to the public key, and outputting the signature $\mathbf{x} \hookleftarrow \mathsf{SampleD}(\mathbf{S}_\mu, \mathbf{A}_\mu, \mathbf{u}, \sigma)$, the scheme can be proved to be strongly unforgeable.

In this thesis, we will employ the Bonsai tree structure to design the lattice-based group signature scheme in Chapter 6 (Section 6.6), where we restrict the norm bound of the ISIS solutions in term of the ℓ_∞ norm.

2.3.5 Lattice-based Encryption Schemes

In this section, we will give an overview of existing lattice-based encryption schemes, and describe two LWE-based schemes related to the later construction in the thesis. First, we recall the standard definition and security notions of public-key encryption (PKE) schemes.

Encryption Schemes.

Definition 2.3.11. A public-key encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is a triple of polynomial-time (possibly probabilistic) algorithms:

- The key generation algorithm $\mathsf{KeyGen}(1^n)$: On input 1^n, where n is the security parameter, output a pair of public key and secret key $(\mathsf{pk}, \mathsf{sk})$.
- The encryption algorithm $\mathsf{Enc}(\mathsf{pk}, M)$: On input the public key pk and a message M, output a ciphertext c.
- The decryption algorithm $\mathsf{Dec}(\mathsf{sk}, c)$: On input the secret key sk and a ciphertext c, return M.

Correctness: The correctness requirement for a PKE scheme $\mathcal{E}$ is as follows: for any $(\mathsf{pk}, \mathsf{sk})$ outputted by $\mathsf{KeyGen}(1^n)$, and any message M,

$$\Pr\big[\mathsf{Dec}\big(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M)\big) = M\big] = 1 - \mathsf{negl}(n),$$

where the probability is taken over the randomness of all algorithms.

Security Notions: The standard security notion of PKE schemes is indistinguishability of ciphertexts under different attack models. Roughly speaking, it requires that any polynomial-time adversary should not be able to distinguish the ciphertexts of two distinct messages (of its choice). In chosen plaintext attacks (CPA), the adversary is not given access to the decryption oracle. In chosen ciphertext attacks ($\mathsf{CCA1}$), the adversary is allowed to query to the decryption oracle before committing the target messages. In a stronger attack notion, called $\mathsf{CCA2}$, the adversary is granted access to the decryption oracle even after it sees the target ciphertext. More formally, we consider the following security experiment $\mathbf{Exp}^{\mathsf{IND\text{-}atk}}_{\mathcal{E},\mathcal{A}}(n)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$, where $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{CCA2}\}$.

Experiment $\mathbf{Exp}^{\mathsf{IND\text{-}atk}}_{\mathcal{E},\mathcal{A}}(n)$:

- **Initialization phase:** $\mathcal{C}$ generates $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^n)$, and gives pk to $\mathcal{A}$.
- **Query phase 1:** $\mathcal{A}$ can make decryption queries if $\mathsf{atk} \in \{\mathsf{CCA1}, \mathsf{CCA2}\}$. Receiving a ciphertext c, the challenger returns $\mathsf{Dec}(\mathsf{sk}, c)$ to $\mathcal{A}$.
- **Challenge phase:** $\mathcal{A}$ outputs two messages M_0 and M_1. $\mathcal{C}$ picks $b \xleftarrow{\$} \{0, 1\}$, computes the target ciphertext $c^* = \mathsf{Enc}(\mathsf{pk}, M_b)$, and sends c^* to $\mathcal{A}$.
- **Query phase 2:** $\mathcal{A}$ can make decryption queries if $\mathsf{atk} = \mathsf{CCA2}$. Receiving a ciphertext c, if $c = c^*$, then $\mathcal{C}$ outputs 0 and halts; otherwise it returns $\mathsf{Dec}(\mathsf{sk}, c)$ to $\mathcal{A}$.

- **Guessing phase:** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $b' = b$, then $\mathcal{C}$ outputs 1, otherwise it outputs 0.

Definition 2.3.12. Let $\mathcal{A}$ be an adversary against a PKE scheme $\mathcal{E}$. Define the advantage of $\mathcal{A}$ in the above experiment as

$$\mathbf{Adv}^{\mathsf{IND\text{-}atk}}_{\mathcal{E},\mathcal{A}}(n) = \Big|\Pr\big[\mathbf{Exp}^{\mathsf{IND\text{-}atk}}_{\mathcal{E},\mathcal{A}}(n) = 1\big] - 1/2\Big|.$$

We say that $\mathcal{E}$ is $\mathsf{IND\text{-}atk}$ secure if $\mathbf{Adv}^{\mathsf{IND\text{-}atk}}_{\mathcal{E},\mathcal{A}}(n)$ is negligible for any polynomial-time adversary $\mathcal{A}$, where $\mathsf{atk} \in \{\mathsf{CPA}, \mathsf{CCA1}, \mathsf{CCA2}\}$.

Overview of Lattice-based Encryption Schemes.

The first lattice-based encryption scheme was proposed by Goldreich, Goldwasser and Halevi in [69] (henceforth, GGH). The idea is very appealing: the secret key is a "good" lattice basis and the public key is a "bad" basis of the same lattice; encryption is done by adding a short noise vector to a properly chosen lattice point and decryption consists of finding the lattice point closest to the ciphertext. Subsequently, Hoffstein et al. proposed NTRU - a cryptosystem that works with a similar principle as of GGH, but operates in lattices with special structure. Both GGH and NTRU are relatively efficient, but unfortunately, these schemes did not come with a security proof.

The first encryption scheme that is provably secure based on the worst-case hardness of a lattice problem was discovered by Ajtai and Dwork [8] in 1997. The scheme was rather impractical but it received considerable attention from the research community, and several improvements and simplified versions were introduced [68, 126, 7, 9]. It is worth to note that the security of the Ajtai-Dwork cryptosystem was based on the worst-case hardness of a special case of SVP, called unique-SVP. The hardness of this problem was not well-understood until Lyuba-

shevsky and Micciancio [99] proved in 2009 that unique-SVP is essentially equivalent to GapSVP (up to some small polynomial approximation factor).

Up to date, the most efficient provably secure encryption schemes from lattices are based on the hardness of the LWE problem (and its ring variant R-LWE). Interestingly, as observed by Regev [129], the hardness of LWE was somewhat implicitly used in Ajtai-Dwork cryptosystem [8] and its simplified form [126], which indicates that "LWE is the correct unifying idea behind all works" in lattice-based PKE schemes. In the following, we will review two basic LWE-based cryptosystems: the scheme introduced by Regev in his seminal work [127] and the one proposed by Gentry et al. [62] (as known as "Dual-Regev encryption scheme"). Variants of these two basic schemes are the building blocks in many recent lattice-based cryptographic constructions: identity-based encryption [62, 42, 1, 2, 47], oblivious transfer [122], homomorphic encryption [61, 35, 33, 31, 32], and much more.

Regev's LWE-based Encryption Scheme.

Revev's LWE-based encryption scheme [127] can be described as follows.

Scheme 2.3.13 (Regev's encryption scheme, adapted from [127])**.** Let $q = \Omega(n^2)$ be a prime, let $m = \Omega(n \log q)$, and $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$. Let χ be a β-bounded discrete Gaussian distribution.

$\mathsf{KeyGen}(1^n)$*:* Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, and $\mathbf{e} \hookleftarrow \chi^m$. Output

$$\mathsf{pk} = (\mathbf{A}, \mathbf{b} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m \quad \text{and} \quad \mathsf{sk} = \mathbf{s}.$$

$\mathsf{Enc}((\mathbf{A}, \mathbf{b}), M \in \{0,1\})$*:* Pick $\mathbf{r} \xleftarrow{\$} \{0,1\}^m$, and output the ciphertext:

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

$\mathsf{Dec}(\mathbf{s}, (\mathbf{u}, c))$: Compute $M' = c - \mathbf{s}^T \cdot \mathbf{u} \in \mathbb{Z}_q$. Output 0 if M' is closer to 0 than to $\lfloor q/2 \rfloor$ modulo q. Otherwise, output 1.

Analysis. The scheme is correct with overwhelming probability. Indeed, we have:

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

Since $\mathbf{r} \in \{0,1\}^m$, and $\|\mathbf{e}\|_\infty \leq \beta = \widetilde{\mathcal{O}}(\sqrt{n})$ with overwhelming probability, the magnitude of the term $\mathbf{e}^T \cdot \mathbf{r}$ is sufficiently smaller than $\lfloor q/4 \rfloor$. Thus, M' is closer to 0 if $M = 0$, and it is closer to $\lfloor q/2 \rfloor$ modulo q if $M = 1$. In other words, the decryption is correct, except for negligible probability.
The CPA-security of Regev's encryption scheme is stated in the following theorem.

Theorem 2.3.14 (Adapted from [127, 128])**.** *For the chosen parameters, the cryptosystem described in Scheme 2.3.13 is* $\mathsf{IND\text{-}CPA}$ *secure if the* $\mathsf{Decision\text{-}LWE}(n, q, \chi)$ *problem is hard.*

The original Regev encryption scheme described above is relatively inefficient: the public key has size $\widetilde{\mathcal{O}}(n^2)$, and the ciphertext of one bit has size $\mathcal{O}(n \log q)$. Subsequently, Kawachi et al. [84] proposed a multi-bit version of the scheme, which slightly improves the encryption blowup factor to $\mathcal{O}(n)$. A significant efficiency improvement was made later by Peikert et al. [122], who introduced an amortized technique that leads to a scheme variant with blowup factor $\mathcal{O}(1)$.

In Chapter 4 of the thesis, we will present statistical zero-knowledge proofs of plaintext knowledge for Regev's encryption scheme and its variant proposed in [122].

The Dual-Regev Encryption Scheme

Gentry et al. [62], partially motivated by the question of designing lattice-based identity encryption, introduced a "dual" variant of Regev's encryption scheme, in

which the key generation and encryption algorithms are swapped. As opposed to Regev's encryption scheme, the public keys in this dual variant are dense, i.e., every syndrome $\mathbf{u} \in \mathbb{Z}_q^n$ is a valid public key that corresponds to many equivalent decryption keys. The scheme can be described as follows.

Scheme 2.3.15 (Dual-Regev encryption scheme, adapted from [62])**.** Let $q = \Omega(n^2)$ be a prime, let $m = \Omega(n \log q)$, and $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$. Let χ be a β-bounded discrete Gaussian distribution.

$\mathsf{KeyGen}(1^n)$*:* Sample $\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{d} \hookleftarrow \chi^m$, and compute $\mathbf{u} = \mathbf{A} \cdot \mathbf{d} \bmod q$. Output $\mathsf{pk} = (\mathbf{A}, \mathbf{u})$ and $\mathsf{sk} = \mathbf{d}$.

$\mathsf{Enc}((\mathbf{A}, \mathbf{u}), M \in \{0,1\})$*:* Sample $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{e} \hookleftarrow \chi^m$, and $z \hookleftarrow \chi$. Output the ciphertext $(\mathbf{b}, c) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$, where

$$\mathbf{b} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e} \bmod q \text{ and } c = \mathbf{u}^T \cdot \mathbf{s} + z + M \cdot \lfloor q/2 \rfloor \bmod q$$

$\mathsf{Dec}(\mathbf{d}, (\mathbf{b}, c))$*:* Compute $M' = c - \mathbf{d}^T \cdot \mathbf{b} \in \mathbb{Z}_q$. Output 0 if M' is closer to 0 than to $\lfloor q/2 \rfloor$ modulo q. Otherwise, output 1.

Analysis. The scheme is correct with overwhelming probability. Indeed, we have:

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

Since the magnitudes of z and the coordinates of $\mathbf{d}, \mathbf{e}$ are all bounded by $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$ with overwhelming probability, the magnitude of the term $z - \mathbf{d}^T \cdot \mathbf{e}$ is sufficiently smaller than $\lfloor q/4 \rfloor$. Thus, M' is closer to 0 if $M = 0$, and it is closer to $\lfloor q/2 \rfloor$ modulo q if $M = 1$. In other words, the decryption is correct, except for negligible probability.

The CPA-security of the scheme is stated in the following theorem.

Theorem 2.3.16 (Adapted from [62])**.** *For the chosen parameters, the cryptosystem described in Scheme 2.3.15 is* IND-CPA *secure if the* Decision-LWE(n, q, χ) *problem is hard.*

In Chapter 4, we will also construct statistical zero-knowledge proofs of plaintext knowledge for the Dual-Regev encryption scheme, and its generalized variant proposed in [61].

2.4 Zero-knowledge Proof Systems

In this section, we first recall the standard definitions of zero-knowledge (ZK) proof systems and proofs of knowledge (PoK), then we will review the Stern-KTX zero-knowledge proof of knowledge (ZKPoK), which will serve as the basic tool in the later constructions in this thesis.

2.4.1 Definitions

Proofs are fundamental objects in mathematics and computer science. In mathematics, one of the main goals is to decide which statements are correct and to prove that they are so. In computer science, the famous problem "P vs. NP" essentially asks if proofs are harder to find or to verify. Traditionally, a proof is a static object that can be expressed in a written form so that it can be verified later. By verifying a proof, one can learn whether the statement being proven is true, and if is is, then one can transfer the given proof to others to convince them of the assertion. Thus, a traditional proof possibly reveals much more than the validity of the assertion, and this is sometimes undesirable.

Zero-knowledge Proofs. Goldwasser, Micali and Rackoff [73] introduced *zero-knowledge proofs* - a beautiful notion that goes beyond the limits of traditional proofs. Namely, they are convincing proofs that reveal nothing but the validity of the statement being proven. To achieve this goal, they first proposed two new ingredients to the notion of a proof: *interaction* and *randomization*.

A proof now is considered as an interactive protocol between two randomized parties - a prover and a verifier. After exchanging messages with the prover, the verifier will output a decision: 1 (i.e.,"Accept") if it is convinced that the statement is true; and 0 (i.e., "Reject") otherwise. In each round of a canonical protocol, the prover first sends a "commitment" to the verifier, who will send back a random "challenge". The prover then computes and sends a "response", so that the verifier checks its validity with respect to the commitment and challenge. If the statement is true, then the prover should be able to convince the verifier (*completeness*). If it is false, then it should not be able to do so (*soundness*). Indeed, in each round, a dishonest prover might correctly guess the verifier's challenge with some constant probability, and might be able to compute a valid response. However, the protocol can be repeated many times, so that the probability that prover guesses all the challenges correctly is negligible.

More formally, "statements" are viewed as strings in some fixed alphabet, and their interpretations are given by a language L specifying the "valid statements". An interactive proof system for a language $L \subseteq \{0,1\}^*$ is defined as follows.

Definition 2.4.1 ([73])**.** An *interactive proof system* for a language $L \subseteq \{0,1\}^*$ with soundness error $s \in [0,1]$ is a two-party protocol $\langle P, V \rangle$, between a PPT *verifier* V and a computationally unbounded *prover* P, which satisfies the following properties:

- Completeness : For all $x \in L$: $\Pr\big[\mathrm{out}_V[P(x) \leftrightarrow V(x)] = 1\big] = 1$,

- Soundness : For all $x \notin L$, and for any $\widehat{P}$: $\Pr\big[\text{out}_V[\widehat{P}(x) \leftrightarrow V(x)] = 1\big] \leq s$,

where $P(x) \leftrightarrow V(x)$ denotes an interaction between P and V on input x, and $\text{out}_V[\cdot]$ denotes the output of V in the specified interaction.

Remark 2.4.2. We have some remarks on the above definition:

- The completeness property states that if the statement is true, then an *honest* prover who follows the protocol should be able to convince the verifier. The completeness condition can be relaxed as

$$\Pr\big[\text{out}_V[P(x) \leftrightarrow V(x)] = 1\big] \geq 1 - c > s,$$

  where c is called the *completeness error*. We note that, all the proof systems that we will consider in this thesis have perfect completeness (i.e., $c = 0$), except for Lyubashevsky's proof systems in Section 3.2.2.

- The soundness property states that if the statement is false, then any *cheating* prover (even if it has computationally unbounded power) cannot convince the verifier with probability greater than s.

Intuitively, an interactive proof system for a language L is said to be *zero-knowledge*, if after reading the proof, the (possibly cheating) verifier cannot learn anything new from the interaction with the prover except for the fact that $x \in L$. This intuition is formulated by a *simulator*. Given an $x \in L$, if there exists an efficient simulator that can produce the transcript of the conversation between the (real) prover and the verifier, then it means that the interaction with the prover does not provide any additional knowledge for the verifier. There exist different definitions of zero-knowledge proofs with respect to different security models. For instance, ZK may be defined with respect to *honest verifiers* or *malicious verifiers*; and the ZK property

may be *perfect*, *statistical*, or *computational* (see [65, Chapter 4] for a comprehensive discussion and comparison among different definitions). In the scope of this thesis, we are mainly interested in *statistical* ZK proofs. In the following, we will give its definition with respect to a *black-box simulator*.

Given an interactive proof system $\langle P, V\rangle$ for a language L and an input x, define the *view* of V on x (denoted by $\text{View}_V[P(x) \leftrightarrow V(x)]$) to be all the messages sent from P to V and all the random bits used by V.

Definition 2.4.3 ([73, 65])**.** An interactive proof system $\langle P, V\rangle$ for a language L is called *statistical zero-knowledge* if for *any* PPT verifier $\widehat{V}$, there exists a PPT simulator $\mathcal{S}$ such that for all $x \in L$, the following conditions hold true:

1. $\Pr[\mathcal{S}^{\widehat{V}}(x) = \perp] \leq 1/2$
2. $\text{View}_{\widehat{V}}[P(x) \leftrightarrow \widehat{V}(x)] \approx_s \widetilde{\mathcal{S}}^{\widehat{V}}(x)$

where $\widetilde{\mathcal{S}}^{\widehat{V}}(x)$ denotes the output distribution of $\mathcal{S}$ having back-box access to $\widehat{V}$ on input x, conditioned on $\mathcal{S}(x) \neq \perp$.

In other words, the above definition requires that the simulator $\mathcal{S}$ outputs with probability at least $1/2$ a transcript that is statistically close to that of the real interaction between the real prover and the (possibly malicious) verifier.

Proofs of Knowledge. We have seen that in interactive proof systems, if the statement is false, then even an unbounded prover cannot get accepted. However, if the statement is true, then it is not clear in Definition 2.4.1 if *anyone* can give an accepted proof. Hence, it is desirable to consider interactive proofs in which a prover convinces a verifier that the prover indeed "knows" why a statement is true. Such proof systems are called *proofs of knowledge* [73, 18]. In order to formally define this notion, we first recall the definition of NP-relations.

Definition 2.4.4. An NP-relation $\mathrm{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ is given by a deterministic algorithm $W(\cdot,\cdot)$ that runs in time polynomial of its first input. The relation is

$$\mathrm{R} = \{(x,w) : W(x,w) \text{ accepts}\}.$$

The associated NP-language L_R is $L_\mathrm{R} = \{x : \exists w \text{ such that } W(x,w) \text{ accepts}\}$. The witness set for an $x \in \{0,1\}^*$ is defined as $\mathrm{R}(x) = \{w : W(x,w) = 1\}$.

In other words, an NP-relation is the set of statement-witness pairs (x,w) that are accepted by the verifier W.

Example 2.4.5. The relation for the $\mathsf{ISIS}^\infty(n,m,q,\beta)$ problem is defined as

$$\mathrm{R}_{\mathsf{ISIS}^\infty}(n,m,q,\beta) = \big\{((\mathbf{A},\mathbf{y}),\mathbf{z}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n \times \mathbb{Z}^m : \|\mathbf{z}\|_\infty \leq \beta \wedge \mathbf{A}\cdot\mathbf{z} = \mathbf{y} \bmod q\big\}.$$

In a PoK for $\mathrm{R}_{\mathsf{ISIS}^\infty}(n,m,q,\beta)$, given an instance $(\mathbf{A},\mathbf{y})$, the prover has to convince the verifier that it knows a witness $\mathbf{z}$ satisfying $\|\mathbf{z}\|_\infty \leq \beta$ and $\mathbf{A}\cdot\mathbf{z} = \mathbf{y} \bmod q$.

The intuition that "the prover knows a satisfying witness" is formulated by the existence of a *knowledge extractor*: if there exists a (possibly cheating) prover $\widehat{P}$ who can convinces the verifier with some probability on a statement $x \in L_\mathrm{R}$, then it should be possible to extract from $\widehat{P}$ a valid witness for x with a related probability.

Definition 2.4.6 (Adapted from [18])**.** An interactive proof system $\langle P, V\rangle$ is a *proof of knowledge* with *knowledge error* $\kappa \in [0,1]$ for an NP-relation R if there exists a PPT *knowledge extractor* $\mathcal{K}$, such that for any $x \in L_\mathrm{R}$, and for any (possibly unbounded) $\hat{P}$, for which $\hat{p}_x = \Pr\big[\mathrm{out}_V[\hat{P} \leftrightarrow V(x)] = 1\big] > \kappa$, we have

$$\Pr[\mathcal{K}^{\hat{P}} \in R(x)] \geq \mathsf{poly}(\hat{p}_x - \kappa).$$

In other words, the probability that $\mathcal{K}$ outputs a valid witness for x using its access to $\hat{P}$ is at least polynomially related to the probability $\hat{p}_x$ that $\hat{P}$ convinces the honest verifier on x, less some knowledge error.

Witness Indistinguishability. Before concluding this section, we recall a notion that is related to zero-knowledge. That is *witness indistinguishability* [57].

Consider an NP-relation $\mathrm{R} = \{(x, w) : W(x, w) \text{ accepts}\}$. Suppose that for some x, the witness set $\mathrm{R}(x)$ contains witnesses w_1 and w_2. Roughly speaking, the statistical witness indistinguishability property states that even a computationally unbounded verifier, who knows both w_1 and w_2, cannot distinguish which witness the prover uses in the interaction. The formal definition is given below.

Definition 2.4.7 ([57])**.** An interactive proof system $\langle P, V \rangle$ for a language L is called *statistical witness indistinguishable* if for *any* PPT verifier $\widehat{V}$, any fixed $x \in L$, and any $w_1, w_2 \in \mathrm{R}(x)$, the following is true:

$$\mathrm{View}_{\widehat{V}}[P(x, w_1) \leftrightarrow \widehat{V}(x)] \approx_s \mathrm{View}_{\widehat{V}}[P(x, w_2) \leftrightarrow \widehat{V}(x)].$$

Witness indistinguishability (WI) is a weaker notion than zero-knowledge. Indeed, if a proof system is (statistical) ZK then it is also (statistical) WI. However, unlike ZK, WI is preserved under parallel composition [57].

2.4.2 *The* Stern-KTX *Proof System*

In this section, we will review the Stern-KTX ZKPoK, which is the fundamental tool in this thesis. In 1996, Stern [137] introduced a three-move ZK proof system for a well-known problem in coding theory: the Syndrome Decoding (SD) problem. The SD problem, which is very similar to the ISIS problem with modulus $q = 2$, is defined as follows.

Definition 2.4.8 (SD problem)**.** Given uniformly random $\mathbf{A} \in \mathbb{Z}_2^{n \times m}$ and $\mathbf{y} \in \mathbb{Z}_2^n$. Let $w < m$ be integer. Find $\mathbf{x} \in \mathbb{Z}_2^m$, such that $\mathsf{wt}(\mathbf{x}) = w$, and $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod 2$.

In 2008, Kawachi, Tanaka and Xagawa [85], observing the similarities between the SD and ISIS problems, adapted Stern's proof system to the lattice setting to obtain a ZKPoK for a *restricted* version of the ISIS problem. We will refer to this protocol as "the Stern-KTX proof system".

Given integers n, m, q and $w < m$, the Stern-KTX proof system is a PoK for the following relation:

$$\mathrm{R}_{\text{S-KTX}} = \Big\{((\mathbf{A}, \mathbf{y}), \mathbf{x}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n \times \{0,1\}^m : \ \mathrm{wt}(\mathbf{x}) = w \wedge \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q\Big\}.$$

The proof system can be described as follows. The common input is $(\mathbf{A}, \mathbf{y})$, and the prover's auxiliary input is $\mathbf{x}$. In the protocol, the prover convinces the verifier in zero-knowledge that $\mathsf{wt}(\mathbf{x}) = w$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$. Let COM be a statistically hiding and computationally binding string commitment scheme. For simplicity, we will not explicitly write the randomness ρ of the commitment scheme.

Commitment: The prover picks a vector $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^m$, a permutation $\pi \xleftarrow{\$} \mathsf{S}_m$, and sends the commitment $\mathrm{CMT} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where:

$$\mathbf{c}_1 = \mathrm{COM}(\pi, \mathbf{A} \cdot \mathbf{r} \bmod q); \quad \mathbf{c}_2 = \mathrm{COM}(\pi(\mathbf{r})); \quad \mathbf{c}_3 = \mathrm{COM}(\pi(\mathbf{x} + \mathbf{r})).$$

Intuition: Since COM is statistically hiding, the verifier, seeing CMT, learns "nothing" about the committed values, and since COM is computationally binding, the prover, having sent CMT, cannot change what it committed.

Challenge: The verifier, receiving CMT, sends $Ch \xleftarrow{\$} \{1, 2, 3\}$ to the prover.

Response: Depending on the value of Ch, the prover responses differently:

- If $Ch = 1$, then reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. Let $\mathbf{v} = \pi(\mathbf{x})$, $\mathbf{t} = \pi(\mathbf{r})$, and send RSP $= (\mathbf{v}, \mathbf{t})$.

  Intuition: Since the permutation π was chosen uniformly at random, the only facts that the verifier, who receives $\pi(\mathbf{x})$, can learn are $\mathbf{x} \in \{0,1\}^m$ and $\mathsf{wt}(\mathbf{x}) = w$, which are exactly the facts to prove about the constraints of $\mathbf{x}$.

- If $Ch = 2$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. Let $\phi = \pi$ and $\mathbf{z} = \mathbf{x} + \mathbf{r}$. Send RSP $= (\phi, \mathbf{z})$.

  Intuition: Since $\mathbf{r}$ was chosen uniformly at random, $\mathbf{x} + \mathbf{r}$ should reveal no information about $\mathbf{x}$.

- If $Ch = 3$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. Let $\psi = \pi$, $\mathbf{s} = \mathbf{r}$, and send RSP $= (\psi, \mathbf{r})$.

Verification: Receiving RSP, the verifier proceeds as follows:

- If $Ch = 1$: Check that $\mathbf{v} \in \{0,1\}^m$; $\mathsf{wt}(\mathbf{v}) = w$; and that $\mathbf{c}_2$, $\mathbf{c}_3$ were honestly computed: $\mathbf{c}_2 = \mathrm{COM}(\mathbf{t})$; $\mathbf{c}_3 = \mathrm{COM}(\mathbf{v} + \mathbf{t})$.

- If $Ch = 2$: Check that $\mathbf{c}_1 = \mathrm{COM}(\phi, \mathbf{A} \cdot \mathbf{z} - \mathbf{y} \bmod q)$ and $\mathbf{c}_3 = \mathrm{COM}(\phi(\mathbf{z}))$.

  Intuition: If $\mathbf{z} = \mathbf{x} + \mathbf{r}$, and if the verifier is convinced that $\mathbf{A} \cdot \mathbf{z} - \mathbf{y} = \mathbf{A} \cdot \mathbf{r} \bmod q$, then he is also convinced that $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$.

- If $Ch = 3$: Check that $\mathbf{c}_1$ and $\mathbf{c}_2$ were correct: $\mathbf{c}_1 = \mathrm{COM}(\psi, \mathbf{A} \cdot \mathbf{s} \bmod q)$ and $\mathbf{c}_2 = \mathrm{COM}(\psi(\mathbf{s}))$.

In each case, the verifier outputs the decision $d = 1$ (Accept) if and only if all the conditions hold. Otherwise, it outputs $d = 0$ (Reject).

The properties of the Stern-KTX protocol are summarized in the following theorem.

Theorem 2.4.9 ([137, 85])**.** *The* Stern-KTX *proof system is a statistical* ZKPoK *for the relation* $\mathrm{R}_{\text{S-KTX}}$*, with perfect completeness, and knowledge error* $2/3$.

It can be verified that if the honest prover follows the protocol, then he always gets accepted. Therefore, the proof system has perfect completeness. It was sketched in [137] and explicitly proved in [85] that:

- If COM is statistically hiding, then one can construct a simulator (with black-box access to a cheating verifier) that produces with probability 2/3 an accepted transcript that is statistically close to that in the real interaction.

- If COM is computationally binding, and if there exists a cheating prover who convinces the verifier with probability non-negligibly larger than 2/3, then by using the standard rewinding technique, one can construct an efficient knowledge extractor which outputs $\mathbf{x}'$ such that $((\mathbf{A}, \mathbf{y}), \mathbf{x}') \in \mathrm{R}_{\text{S-KTX}}$.

In other words, the proof system is statistically ZK and a PoK with knowledge error 2/3. We note that, Stern [137] also provided a variant of the proof system with soundness error close to 1/2, but it consists of 5 moves. In 2010, Cayrel et al. [43] achieved a similar result using a modified technique. In this thesis, we focus on the 3-move version of the protocol, with soundness error 2/3 in each round. Using standard technique of protocol composition, one can repeat the protocol $t = \omega(\log n)$ times (in parallel, or in sequence) to make the soundness error negligibly small.

In summary, throughout this chapter, we have reviewed the basic definitions and constructions in lattice-based cryptography, as well as the standard notions of zero-knowledge proof systems. We have also described the basic tools to be used in the thesis. In Chapters 3, 4, 5, and 6, we will present our contributions, in comparison with the corresponding related works.

3. IMPROVED PROOF SYSTEMS FOR THE ISIS^∞ AND SIS^∞ PROBLEMS

3.1 Introduction

In many lattice-based cryptographic constructions ([115, 94, 136, 130, 88, 131],...), the building block is a proof of knowledge of a valid solution to an instance of the ISIS problem. In the security proof, the reduction algorithm typically uses the knowledge extractor of the underlying protocol to extract a vector that will be helpful for solving a challenge instance of the hard problem. In these cases, the "quality" of the extracted vector will affect the security guarantee of the protocol, and the hardness assumption used in the cryptographic construction built upon it.

We recall that, in the scope of this thesis, we use the term "*extraction gap*" as a criterion for the quality of the extracted vector. Specifically, we say that a proof of knowledge for the ISIS^∞ or SIS^∞ problem admits extraction gap g (where $g \geq 1$) if the knowledge extractor in that protocol, at best, can produce a vector whose infinity norm is g times larger than that of the valid witness possessed by the prover. In the ideal situation when $g = 1$ (i.e., there is "no extraction gap"), the reduction algorithm would be able to use the knowledge extractor to solve the underlying ISIS or SIS instance. This reduction demonstrates that such proof system has a very strong security guarantee, because breaking the protocol is as least as hard as solving the underlying problem. However, when $g > 1$, the soundness of the

proof system usually has to rely on a *stronger* hardness assumption, i.e., assumption that certain potentially easier problem is hard to solve.

We observe that, all existing efficient (zero-knowledge [115], or witness indistinguishable [94]) proofs of knowledge for the ISIS^∞ or SIS^∞ problem admit noticeable extraction gaps. Specifically:

- For the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ problem, all known proof systems admit extraction gaps $g \geq \widetilde{\mathcal{O}}(\sqrt{n})$, where n denotes the dimension of the corresponding worst-case lattice problem. Therefore, while a valid witness vector is supposed to have infinity norm bounded by β, these schemes only guarantee that a prover using a witness $\mathbf{x} \in \mathbb{Z}^m$, such that $\|\mathbf{x}\|_\infty > g \cdot \beta$ cannot cheat the verifier. However, in these schemes, it is not guaranteed that a cheating prover using $\mathbf{x}' \in \mathbb{Z}^m$ such that $\beta < \|\mathbf{x}'\|_\infty \leq g \cdot \beta$ cannot pass the verification procedure. In other words, the soundness of these schemes has to rely on the hardness of solving the $\mathsf{ISIS}^\infty(n, m, q, g \cdot \beta)$ problem.

- For the $\mathsf{SIS}^\infty(n, m, q, \beta)$ problem, the only known proof system, which is derived from a protocol in [115], also has an extraction gap $g \geq \widetilde{\mathcal{O}}(\sqrt{n})$.

The above discussion shows that the current proof systems for the ISIS^∞ and SIS^∞ problems are sub-optimal, since breaking each of these protocols is potentially easier than solving the underlying problem. As a consequence, lattice-based cryptographic constructions using these protocols as their building blocks inherit the sub-optimal security guarantees. Therefore, zero-knowledge proof systems with stronger security guarantees (e.g. with a small constant extraction gap) are highly desirable. Such protocols are not only interesting from a theoretical point of view, but they also lead to lattice-based cryptographic constructions relying on *weaker* hardness assumptions than the contemporary schemes.

In this chapter, we construct improved ZKPoK for the ISIS^∞ and SIS^∞ problems. Our schemes rely on a simple, yet versatile and effective technique, called Decomposition-Extension. When applying this technique to the Stern-KTX protocol ([137, 85], see also Section 2.4.2), we obtain ZKPoK for the ISIS^∞ problem, called **SternExt**, with a very strong security guarantee: The best way to break these systems is to solve the underlying problems. Thus, in term of security, we achieve the "optimal" solutions. Moreover, in terms of efficiency, **SternExt** compares favorably to the Micciancio-Vadhan protocol [115], which is essentially the only known efficient *zero-knowledge* proof system for the ISIS^∞ problem prior to our work. By some technical modifications, we also get a proof system for the SIS^∞ problem with a slightly weaker security guarantee, where the extraction gap is $g < 2$.

Our proof systems can find various applications in lattice-based cryptography, as we will demonstrate in the later chapters. More generally, our Decomposition-Extension technique can be used to construct ZKPoK of linear objects (e.g., vectors, matrices) which are the secret solutions of certain given systems of modular linear equations, and which satisfy some suitable conditions (e.g., "smallness"). In other words, our technique is somewhat compatible with the properties of lattice-based cryptography (i.e., "linearity", "smallness"), and can possibly find further interesting applications.

The rest of this chapter is organized as follows. In Section 3.2, we review the previous works. We then present our proof system **SternExt** for the ISIS^∞ problem in Section 3.3. In Section 3.4, we propose a modified version of **SternExt** to obtain a proof system for the SIS^∞ problem.

3.2 Previous Works

First, we will give a short overview of existing proof systems for (worst-case and average-case) lattice problems. The first interactive proofs for lattice problems was introduced by Goldreich and Goldwasser [66], who gave AM proof systems for the complement problems $\mathsf{coGapCVP}_\gamma$ and $\mathsf{coGapSVP}_\gamma$, with $\gamma = \mathcal{O}(\sqrt{n/\log n})$. Subsequently, Micciancio and Vadhan [115] proposed statistical ZK proofs for GapCVP_γ and GapSVP_γ with efficient provers, where $\gamma \geq \mathcal{O}(\sqrt{n/\log n})$. We remark that it is possible to derive proof systems for the ISIS and SIS problems from the Micciancio-Vadhan protocols (see the discussion in Section 3.3 and Section 3.4, respectively), but the obtained proof systems inherit the extraction gap $g \geq \mathcal{O}(\sqrt{n/\log n})$. Peikert and Vaikuntanathan [121] introduced the first (indeed, the only known to date) non-interactive proof systems for a variety of worst-case lattice problems. However, it seems non-trivial to derive proof systems for average-case problems from the ones given in [121].

The first proof system that explicitly addresses the ISIS problem were given by Lyubashevsky in [94], and was improved in his subsequent works [95, 96]. Being constructed using clever aborting and rejection sampling techniques, these proof systems are quite efficient. However, these protocols suffer from several limitations: they are not even zero-knowledge (they are only proved to be witness indistinguishable); every round of the protocols has a constant completeness error; and their security guarantees are relatively weak (with an extraction gap of $\widetilde{\mathcal{O}}(n)$ in [94], then was improved to $\widetilde{\mathcal{O}}(\sqrt{n})$ in [96]).

In the following section, we will review the proof systems by Micciancio and Vadhan, and the ones by Lyubashevsky.

3.2.1 The Micciancio-Vadhan Proof System

We will focus on the Micciancio-Vadhan proof system for GapCVP_γ [115]. We omit their protocol for GapSVP_γ, which employs the same idea, and refer the readers to [115].

Given a GapCVP^2_γ instance $(\mathbf{B}, \mathbf{t}, d)$, the goal of the prover is to convince the verifier in zero-knowledge that it knows a lattice vector $\mathbf{B} \cdot \mathbf{w}$, such that $\|\mathbf{t} - \mathbf{B} \cdot \mathbf{w}\| \leq d$. The idea of the proof system is basically as follows. To prove that $\mathbf{t}$ and $\mathbf{B} \cdot \mathbf{w}$ are close to each other, the prover instead proves that two small balls around $\mathbf{t}$ and $\mathbf{B} \cdot \mathbf{w}$ intersect. In the protocol, the prover picks uniformly random points from two balls, reduces them modulo $\mathbf{B}$, and sends them to the verifier. (The reduction modulo $\mathbf{B}$ results in a distribution that can be efficiently sampled without the knowledge of $\mathbf{B} \cdot \mathbf{w}$ - this is crucial to achieve the zero-knowledge property.) The prover is constrained to show that some point must belong to the intersection of two balls, which means that the centers of the balls are close to each other. Specifically, the proof system is as follows.

Scheme 3.2.1 (The Micciancio-Vadhand proof system for GapCVP_γ)**.** The common input is the triple $(\mathbf{B}, \mathbf{t}, d)$. The prover's auxiliary input is a lattice vector $\mathbf{B} \cdot \mathbf{w}$. Denote $\mathbf{t} - \mathbf{B} \cdot \mathbf{w}$ by $\mathbf{u}$. The scheme is parameterized by some integer k, which depends on the value of γ.

- **Commitment:**

  1. For $i \in [k]$, pick $c_i \xleftarrow{\$} \{0, 1\}$, and $\mathbf{r}_i \xleftarrow{\$} \mathcal{B}(0, \gamma d/2)$.
  2. Check if there exists an index i^* such that $\|\mathbf{r}_{i^*} + (2c_{i^*} - 1)\mathbf{u}\|_2 \leq \gamma d/2$. If not, set $i^* = 1$ and redefine $c_{i^*} = 0$ and $\mathbf{r}_{i^*} = \mathbf{u}/2$ to satisfy $\|\mathbf{r}_{i^*} + (2c_{i^*} - 1)\mathbf{u}\|_2 \leq \gamma d/2$.
  3. For $i \in [k]$, compute $\mathbf{m}_i = c_i \mathbf{t} + \mathbf{r}_i \bmod \mathbf{B}$.

4. Send $\mathbf{m}_1, \ldots, \mathbf{m}_k$ to the verifier.

- **Challenge:** The verifier picks $c \xleftarrow{\$} \{0,1\}$ and sends it to the prover.
- **Response:**
  - If $c = \bigoplus_i c_i$, then send the bits c_i and the lattice vectors $\mathbf{B}\cdot\mathbf{v}_i = \mathbf{m}_i - (\mathbf{r}_i + c_i\mathbf{t})$ (for $i \in [k]$) to the verifier.
  - If $c \neq \bigoplus_i c_i$, then replace c_{i^*} by $1 - c_{i^*}$ and $\mathbf{B}\cdot\mathbf{v}_{i^*}$ by $\mathbf{B}\cdot\mathbf{v}_{i^*} + (2c_{i^*} - 1)(\mathbf{t} - \mathbf{u})$, and send $c_1, \ldots, c_k$ and $\mathbf{B}\cdot\mathbf{v}_1, \ldots, \mathbf{B}\cdot\mathbf{v}_k$ to verifier.
- **Verification:** The verifier receives k bits $c_1, \ldots, c_k$ and k vectors $\mathbf{B}\cdot\mathbf{v}_1, \ldots, \mathbf{B}\cdot\mathbf{v}_k$. If $c = \bigoplus_i c_i$ and $\|\mathbf{m}_i - (\mathbf{B}\cdot\mathbf{v}_i + c_i\mathbf{t})\|_2 \leq \gamma d/2$ for all i, then output 1 (Accept). Otherwise, output 0 (Reject).

The properties of the proof system are summarized in the following theorem.

Theorem 3.2.2 ([115]). *The proof system described in Scheme 3.2.1 is a statistical zero-knowledge proof system for* GapCVP^2_γ *with perfect completeness and soundness error* $1/2$*, provided one of the following conditions hold true:*

- $\gamma = \Omega(\sqrt{n/\log n})$ *and* $k = \mathsf{poly}(n)$ *is a sufficiently large polynomial, or*
- $\gamma = \Omega(\sqrt{n})$ *and* $k = \omega(\log n)$ *is any super-logarithmic function of* n*, or*
- $\gamma = n^{0.5+\Omega(1)}$ *and* $k = \omega(1)$ *is any super-constant function of* n*.*

Furthermore, the proof system is a proof of knowledge: if there exists a cheating prover $\hat{P}$ *who convinces* V *with probability* $1/2 + \epsilon$ *on some instance* $(\mathbf{B}, \mathbf{t}, d)$ *then there exists a knowledge extractor* $\mathcal{K}$ *that outputs a lattice vector* $\mathbf{B}\cdot\mathbf{w}'$ *satisfying* $\|\mathbf{t} - \mathbf{B}\cdot\mathbf{w}'\| \leq \gamma \cdot d$ *in time* $\mathsf{poly}(n)/\epsilon$.

It can be seen that the Micciancio-Vadhan proof system admits a trade-off between its security guarantee (determined by the approximation factor γ) and its communication cost (which is linearly dependent on k). In practice, one might have to choose $k = \omega(\log n)$ (and thus, $\gamma = \Omega(\sqrt{n})$) to balance the security and efficiency requirements.

3.2.2 Lyubashevsky's Proof System

We now review some basic facts on Lyubashevsky's proof system for the ISIS problem. In [94, 95], Lyubashevsky constructed a PoK for the $\mathsf{ISIS}^\infty(m, n, q, \beta)$ problem. Namely, in the protocol, given $(\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m}, \mathbf{y} \xleftarrow{\$} \mathbb{Z}_q^n)$, the prover has to convince the verifier that it knows $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_\infty \leq \beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$. Prior to the interaction, both party agree on the following parameters: a number $r > m \cdot \beta$, and two domains

$$\begin{aligned} \mathsf{D}_{\mathbf{u}} &= \big\{\mathbf{u} \in \mathbb{Z}^m : \|\mathbf{u}\|_\infty \leq r + \beta\big\}, \\ \mathsf{D}_{\mathbf{z}} &= \big\{\mathbf{z} \in \mathbb{Z}^m : \|\mathbf{z}\|_\infty \leq r\big\}. \end{aligned}$$

One round of interaction between the prover P and the verifier V is as follows:

Commitment: P picks $\mathbf{u} \xleftarrow{\$} \mathsf{D}_{\mathbf{u}}$, and sends $\mathbf{w} = \mathbf{A} \cdot \mathbf{u} \bmod q$ to V.

Challenge: V sends a challenge $Ch \xleftarrow{\$} \{0, 1\}$ to P.

Response: P computes $\mathbf{z} = \mathbf{u} + c \cdot \mathbf{x}$ and proceeds as follows:

- If $\mathbf{z} \notin \mathsf{D}_{\mathbf{z}}$, namely $\mathbf{z}$ is not "safe", then output $\perp$ and abort.
- If $\mathbf{z} \in \mathsf{D}_{\mathbf{z}}$, namely $\mathbf{z}$ is "safe", then send the response $\mathbf{z}$.

Verification: Output 1 if and only if $\mathbf{z} \in \mathsf{D}_{\mathbf{z}}$ and $\mathbf{A} \cdot \mathbf{z} = \mathbf{y} + \mathbf{w} \bmod q$. Otherwise, output 0.

Remark 3.2.3. We have some notes on the above proof system:

- The protocol employs an interesting aborting technique: The "mask" value $\mathbf{u}$ is chosen uniformly at random from a "big" domain $\mathsf{D}_\mathbf{u}$, but the response $\mathbf{z} = \mathbf{u}+c\cdot\mathbf{x}$ is only revealed whenever it falls to the "small" domain $\mathsf{D}_\mathbf{z}$. These domain are chosen in a careful manner, so that whenever the prover does not abort, the distribution of $\mathbf{z}$ is exactly the uniform distribution over $\mathsf{D}_\mathbf{z}$, and thus it should not leak any information about the witness $\mathbf{x}$.

- The aborting technique yields two small drawbacks. Firstly, every round of the protocol admits a constant completeness error, since even an honest prover sometimes has to abort. Secondly, it is not known how to construct a transcript simulator for the protocol. Indeed, the protocol is only proved to be witness indistinguishable.

- By rewinding the cheating prover, one can construct a knowledge extractor. However, the extracted vector has infinity norm bounded by $\widetilde{\mathcal{O}}(m)\cdot\beta$, and the soundness of the protocol has to relies on the hardness of the $\mathsf{ISIS}^\infty(n, m, q, \widetilde{\mathcal{O}}(m)\cdot\beta)$ problem. In other words, there is an extraction gap $g = \widetilde{\mathcal{O}}(m)$.

In [96], Lyubashevsky proposed an improved version of the above proof system, which makes use of a clever rejection sampling algorithm. Now, $\mathbf{u}$ is not sampled from a uniform distribution, but from a discrete Gaussian distribution $D_{\mathbb{Z}^m,\sigma}$, with $\sigma = \widetilde{\mathcal{O}}(\sqrt{m})\cdot\beta$. Then the response $\mathbf{z}$ is only revealed with some carefully chosen probability. The new technique helps to reduce the extraction gap to $\widetilde{\mathcal{O}}(\sqrt{m})$.

3.3 Improved Proof System for the ISIS^∞ Problem

In this section, we will present an improved zero-knowledge proof of knowledge for the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ problem, which is associated with the relation $\mathrm{R}_{\mathsf{ISIS}^\infty}$ defined

as follows:

$$\mathrm{R}_{\mathsf{ISIS}^\infty}(n, m, q, \beta) = \big\{((\mathbf{A}, \mathbf{y}), \mathbf{x}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n \times \mathbb{Z}^m : \|\mathbf{x}\|_\infty \leq \beta \wedge \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q\big\}.$$

We first summarize the strengths and weaknesses of the existing proof systems for this relation. It follows from the discussions in Section 3.2 that there are currently two such proof systems:

- First, one might apply Lyubashevsky's proof system [94, 95, 96] to obtain an efficient PoK for $\mathrm{R}_{\mathsf{ISIS}^\infty}$ but the protocol has several noticeable limitations: it is not zero-knowledge; it has a constant completeness error in every round; and it admits an extraction gap $\widetilde{\mathcal{O}}(\sqrt{m})$.

- Alternatively, one can obtain a PoK for $\mathrm{R}_{\mathsf{ISIS}^\infty}$ by transforming the given ISIS^∞ instance into a GapCVP^∞ instance, and adapting the Micciancio-Vadhan proof system for GapCVP^2 [115] to the infinity norm. Let $\mathbf{B}$ be any basis of the lattice $\Lambda_q^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q\}$ and $\mathbf{t}$ be a vector in $\mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{t} = \mathbf{y} \bmod q$. Such $\mathbf{B}$ and $\mathbf{t}$ can be efficiently computed using linear algebra. Then run the Micciancio-Vadhan protocol for $\mathsf{GapCVP}_\gamma^\infty$ with common input $(\mathbf{B}, \mathbf{t}, \beta)$. The prover's auxiliary input is $\mathbf{e} = \mathbf{t} - \mathbf{x} \in \Lambda_q^\perp(\mathbf{A})$. We recall that the knowledge extractor in [115] is only able to output a vector $\mathbf{e}' \in \Lambda_q^\perp(\mathbf{A})$, such that $\|\mathbf{t} - \mathbf{e}'\|_\infty \leq g \cdot \beta$ for some $g \geq \widetilde{\mathcal{O}}(\sqrt{m})$. This implies that $\mathbf{x}' = \mathbf{t} - \mathbf{e}'$ is a solution to the $\mathsf{ISIS}^\infty(n, m, q, g \cdot \beta)$ problem with respect to $(\mathbf{A}, \mathbf{y})$. In other words, the obtained protocol is a ZKPoK with perfect completeness for the relation $\mathrm{R}_{\mathsf{ISIS}^\infty}$, but it inherits an extraction gap at least $\widetilde{\mathcal{O}}(\sqrt{m})$.

It follows from the discussion above that all existing proof systems for $\mathrm{R}_{\mathsf{ISIS}^\infty}$ have some limitations, most notably the fact that there is a gap between the norm of the

witness vector and the norm of the vector computed by the knowledge extractor: The latter is only guaranteed to be $g = \widetilde{\mathcal{O}}(\sqrt{m})$ times larger than the former. (For commonly used parameters, we have $g = \widetilde{\mathcal{O}}(\sqrt{m}) = \widetilde{\mathcal{O}}(\sqrt{n})$.) As a consequence, the security of these proof systems has to rely on the hardness of the $\mathsf{ISIS}^\infty(n, m, q, g \cdot \beta)$ problem. In other words, breaking these protocols is potentially easier than solving the underlying $\mathsf{ISIS}^\infty(n, m, q, \beta)$ problem. Furthermore, as we mentioned earlier, cryptographic constructions using these proof systems as the building blocks will inherit the extraction gap $g = \widetilde{\mathcal{O}}(\sqrt{n})$, which leads to relatively weak security guarantees. This hints that the existing proof systems for the ISIS^∞ problem are somewhat sub-optimal: Is it possible to design an efficient ZKPoK for ISIS^∞ with stronger security guarantee than the existing ones?

3.3.1 Our Techniques and Contributions

In this work, we reply positively to the question raised in the previous section. Specifically, we describe a statistical ZKPoK for ISIS^∞ with *no gap*, i.e., $g = 1$. On the other hand, the communication cost of our proof system compares favorably to that of the Micciancio-Vadhan protocol.

We now sketch our approach. While the Micciancio-Vadhan protocol exploits the geometric aspect of the ISIS problem, our protocol exploits its combinatorial and algebraic aspects. We first look at the Stern-KTX proof system ([137, 85], see also Section 2.4.2), which is associated with the relation

$$\mathrm{R}_{\text{S-KTX}} = \Big\{((\mathbf{A}, \mathbf{y}), \mathbf{x}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n \times \{0,1\}^m : \ \mathrm{wt}(\mathbf{x}) = w \wedge \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q\Big\}.$$

It can be seen that the restrictions on the witness $\mathbf{x}$ are very tight. As a consequence, the Stern-KTX proof system does not seem to suffice for a wide range of applica-

tions. For some cryptographic schemes that allow many users, such as identity-based identification [134] and group signature [46] schemes, the secret keys of the users are typically generated from the public keys by a trusted authority. For such schemes that rely on lattice-based hardness assumptions ([136, 130, 38, 74, 88]), this task is performed by using a secret trapdoor possessed by the trusted authority, consisting in a relatively short basis of a publicly known lattice (see Section 2.3.3). As a result, a user secret key $\mathbf{x}$ is a *general* solution to the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ problem, where β is typically $\widetilde{\mathcal{O}}(\sqrt{n})$. We observe that the Stern-KTX proof system cannot handle such general ISIS^∞ solutions.

On the positive side, the Stern-KTX proof system has the nice feature that we are looking for: Its knowledge extractor is able to output a vector with infinity norm exactly 1. We then investigate how to loosen the restrictions on the witness $\mathbf{x}$ in the Stern-KTX proof system while preserving the zero-knowledge property. We first note that the conditions

$$\mathbf{x} \in \{0,1\}^m \text{ and } \mathsf{wt}(\mathbf{x}) = w$$

are invariant under all permutations of coordinates: For $\pi \in \mathcal{S}_m$, a vector $\mathbf{x}$ satisfies those restrictions if and only if $\pi(\mathbf{x})$ also does. Thus, a witness $\mathbf{x}$ with such constraints can be verified in zero-knowledge thanks to the randomness of π. We then observe that the same statement still holds true for $\mathbf{x} \in \mathsf{B}_{3m}$, namely:

$$\text{For } \pi \in \mathcal{S}_{3m} : \mathbf{x} \in \mathsf{B}_{3m} \Leftrightarrow \pi(\mathbf{x}) \in \mathsf{B}_{3m}.$$

(We recall that by B_{3m}, we denote the set of all vectors in $\{-1, 0, 1\}^{3m}$ having exactly m coordinates equal to -1; m coordinates equal to 0; and m coordinates equal to 1.) This basic fact allows us to generalize the Stern-KTX proof system. Our generalization consists of two steps:

Step 1. *Removing the restriction on the Hamming weight.*

Specifically, we observe that a ZKPoK for the $\mathsf{ISIS}^\infty(n, m, q, 1)$ problem, which is associated with the relation

$$\mathrm{R}_{\mathsf{ISIS}^\infty_{\beta=1}}(n, m, q) = \left\{((\mathbf{A}, \mathbf{y}), \mathbf{x}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n \times \{-1, 0, 1\}^m : \ \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q\right\}$$

can be derived from the Stern-KTX scheme by the following *extensions*: For any vector $\mathbf{x} \in \{-1, 0, 1\}^m$, append $2m$ coordinates from the set $\{-1, 0, 1\}$ to $\mathbf{x}$ to obtain $\mathbf{x}^* \in \mathsf{B}_{3m}$. Next, to make the dimensions compatible, append $2m$ *zero-columns* to matrix $\mathbf{A}$ to get $\mathbf{A}^* \in \mathbb{Z}_q^{n\times 3m}$. We then have:

$$\begin{aligned} \mathbf{x}^* \in \mathsf{B}_{3m} \quad &\Leftrightarrow \quad \mathbf{x} \in \{-1, 0, 1\}^m, \\ \mathbf{A}^* \cdot \mathbf{x}^* = \mathbf{y} \bmod q \quad &\Leftrightarrow \quad \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q. \end{aligned}$$

In other words, if a verifier is convinced that $\mathbf{x}^* \in \mathsf{B}_{3m}$ and $\mathbf{A}^* \cdot \mathbf{x}^* = \mathbf{y} \bmod q$, then it is also convinced that $\mathbf{x}$ is a valid witness for the relation $\mathrm{R}_{\mathsf{ISIS}^\infty_{\beta=1}}(n, m, q)$. In Figure 3.1, we give an illustration of this extension technique.

Step 2. *Increasing the ℓ_∞ bound from 1 to β, for any positive integer β.*

Our idea is to "decompose" any vector $\mathbf{x} \in \{-\beta, \ldots, 0, \ldots, \beta\}^m$ into $p = \lfloor \lg \beta \rfloor + 1$ vectors $\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_p$ in $\{-1, 0, 1\}^m$. Specifically, given β we construct a sequence of integers $\beta_1, \ldots, \beta_p$, such that every integer in the interval $[0, \beta]$ can be efficiently expressed as a subset sum of $\beta_1, \ldots, \beta_p$. As a result, it is efficient to compute $\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_p \in \{-1, 0, 1\}^m$ such that $\mathbf{x} = \sum_{j=1}^p \beta_j \cdot \tilde{\mathbf{u}}_j$.

Next we apply the extensions of **Step 1**: Extend each $\tilde{\mathbf{u}}_j$ to $\mathbf{u}_j \in \mathsf{B}_{3m}$, and

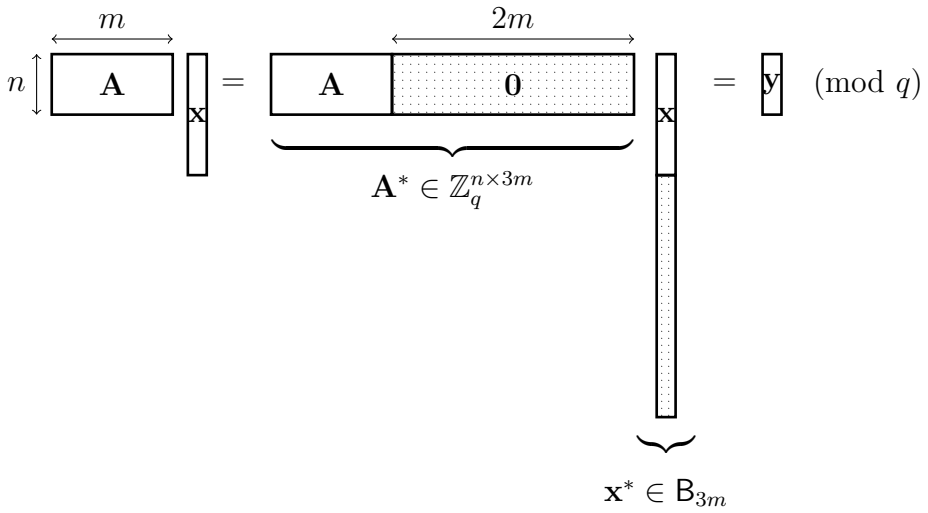

Fig. 3.1: Basic extension technique: If the verifier is convinced that $\mathbf{A}^* \cdot \mathbf{x}^* = \mathbf{y} \bmod q$ and a random permutation of $\mathbf{x}^*$ belongs to the set B_{3m}, then it is also convinced that $\mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$ and $\mathbf{x} \in \{-1, 0, 1\}^m$.

extend matrix $\mathbf{A}$ to $\mathbf{A}^* \in \mathbb{Z}_q^{n \times 3m}$. We then have:

$$\mathbf{A}^*\big(\sum_{j=1}^{p} \beta_j \cdot \mathbf{u}_j\big) = \mathbf{y} \bmod q \Leftrightarrow \mathbf{A}\mathbf{x} = \mathbf{y} \bmod q.$$

This allows us to combine p proofs for the $\mathsf{ISIS}^\infty(n, m, q, 1)$ problem into one proof for the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ problem.[1] As a result, we obtain a statistical ZKPoK for the *general* ISIS^∞ problem, that we call **SternExt**, with the following properties:

- The knowledge extractor produces an $\mathbf{x}'$ with $\|\mathbf{x}'\|_\infty \leq \beta$. In other words, the extraction gap is $g = 1$.
- The communication cost is $\log \beta \cdot \widetilde{\mathcal{O}}(n \log q)$. In particular, in most cryptographic applications, parameter q is $\mathsf{poly}(n)$, and $\log \beta \leq \log q = \widetilde{\mathcal{O}}(1)$. Thus, for commonly used parameters, one round of interaction costs $\widetilde{\mathcal{O}}(n)$ bits.

Overall, **SternExt** provides a better proof system for $\mathrm{R}_{\mathsf{ISIS}^\infty}$ in both security and

[1] This packing of proofs is akin to Jain et al.'s recent work on the Learning Parity with Noise problem [82, Section 4.2].

efficiency aspects than the one derived from the Micciancio-Vadhan protocol. We summarize the comparison among the proof systems for $\mathrm{R}_{\mathsf{ISIS}^\infty}$ in Table 3.1. The comparison data are for one round of protocol, in which case all the considered proof systems admit a constant soundness error.

Proof systems	[Lyubashevsky]	[Micciancio-Vadhan]	**SternExt**
Zero-knowledge?	✗(WI)	✓	✓
Perfect completeness?	✗	✓	✓
Norm bound in the ISIS hardness assumption	$\beta \cdot \widetilde{\mathcal{O}}(\sqrt{n})$	$\beta \cdot \widetilde{\mathcal{O}}(\sqrt{n})$	β
Communication	$\widetilde{\mathcal{O}}(n \log q)$	$\widetilde{\mathcal{O}}(n \log q)$	$\log \beta \cdot \widetilde{\mathcal{O}}(n \log q)$

Tab. 3.1: Comparison among the proof systems for $\mathrm{R}_{\mathsf{ISIS}^\infty}(n, m, q, \beta)$. See Section 3.2.1 for other security/efficiency trade-offs for the Micciancio-Vadhan protocol, and Section 3.2.2 for the discussion on Lyubashevsky's protocol.

Before presenting our proof system, we first formally describe our Decomposition-Extension technique in Section 3.3.2.

3.3.2 The Decomposition - Extension Technique

Let n, m, q, β be positive integers, and let $p = \lfloor \log \beta \rfloor + 1$. Define the sequence of integers $\beta_1, \ldots, \beta_p$ as follows:[1]

$$\beta_1 = \lceil \beta/2 \rceil,\ \beta_2 = \lceil (\beta - \beta_1)/2 \rceil,\ \beta_3 = \lceil (\beta - \beta_1 - \beta_2)/2 \rceil, \ldots, \beta_p = 1.$$

It can be checked that these integers satisfy $\sum_{j=1}^{p} \beta_j = \beta$ and their subset sums are precisely numbers between 0 and β. Moreover, any integer in this interval can be efficiently expressed as a subset sum of $\beta_1, \ldots, \beta_p$.

[1] This sequence was suggested by Daniele Micciancio. In [93], we used a sequence of powers of 2, i.e., $\beta_j = 2^{p-j}$ for each $j \in [p]$. In this thesis, we will employ that sequence in the proof system for the SIS^∞ problem (in Section 3.4).

Example 3.3.1. Let $\beta = 115$, then we have $p = \lfloor \log(115) \rfloor + 1 = 7$, and:

$$\beta_1 = 58,\ \beta_2 = 29,\ \beta_3 = 14,\ \beta_4 = 7,\ \beta_5 = 4,\ \beta_6 = 2,\ \beta_7 = 1.$$

We now construct the following procedures:

1. Elementary Decomposition. On input a vector $\mathbf{v} = (v_1, v_2, \ldots, v_m) \in \mathbb{Z}^m$ such that $\|\mathbf{v}\|_\infty \leq \beta$ the vector decomposition procedure $\mathsf{EleDec}_{m,\beta}$ outputs p vectors $\widetilde{\mathbf{w}}_1, \ldots, \widetilde{\mathbf{w}}_p \in \{-1, 0, 1\}^m$, such that $\sum_{j=1}^{p} \beta_j \cdot \widetilde{\mathbf{w}}_j = \mathbf{v}$. It works as follows:

   (a) For each $i \in [m]$, express v_i as $v_i = \beta_1 \cdot v_{i,1} + \beta_2 \cdot v_{i,2} + \ldots + \beta_p \cdot v_{i,p}$, where $v_{i,j} \in \{-1, 0, 1\}$ for all $j \in [p]$.

   (b) For each $j \in [p]$, let $\widetilde{\mathbf{w}}_j := (v_{1,j}, v_{2,j}, \ldots, v_{m,j}) \in \{-1, 0, 1\}^m$.

   (c) Output $\widetilde{\mathbf{w}}_1, \ldots, \widetilde{\mathbf{w}}_p$.

2. Elementary Extension. On input a vector $\widetilde{\mathbf{w}} \in \{-1, 0, 1\}^m$, the procedure EleExt_m extends $\widetilde{\mathbf{w}}$ to a vector $\mathbf{w} \in \mathsf{B}_{3m}$. This procedure works as follows:

   (a) Let $\lambda^{(-1)}$, $\lambda^{(0)}$ and $\lambda^{(1)}$ be the numbers of coordinates of $\widetilde{\mathbf{w}}$ that equal to -1, 0, and 1 respectively.

   (b) Pick a random vector $\widehat{\mathbf{w}} \in \{-1, 0, 1\}^{2m}$ that has exactly $(m - \lambda^{(-1)})$ coordinates -1; $(m - \lambda^{(0)})$ coordinates 0; and $(m - \lambda^{(1)})$ coordinates 1.

   (c) Output $\mathbf{w} = \left(\widetilde{\mathbf{w}} \| \widehat{\mathbf{w}}\right) \in \mathsf{B}_{3m}$.

3. Vector Decomposition and Extensions. On input vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_\infty \leq \beta$, the procedure $\mathsf{VectorDE}_{m,\beta}$ outputs p vectors $\mathbf{u}_1, \ldots, \mathbf{u}_p \in \mathsf{B}_{3m}$. This procedure works as follows:

   (a) Run $\mathsf{EleDec}_{m,\beta}(\mathbf{x})$ to obtain $\widetilde{\mathbf{u}}_1, \ldots, \widetilde{\mathbf{u}}_p \in \{-1, 0, 1\}^m$.

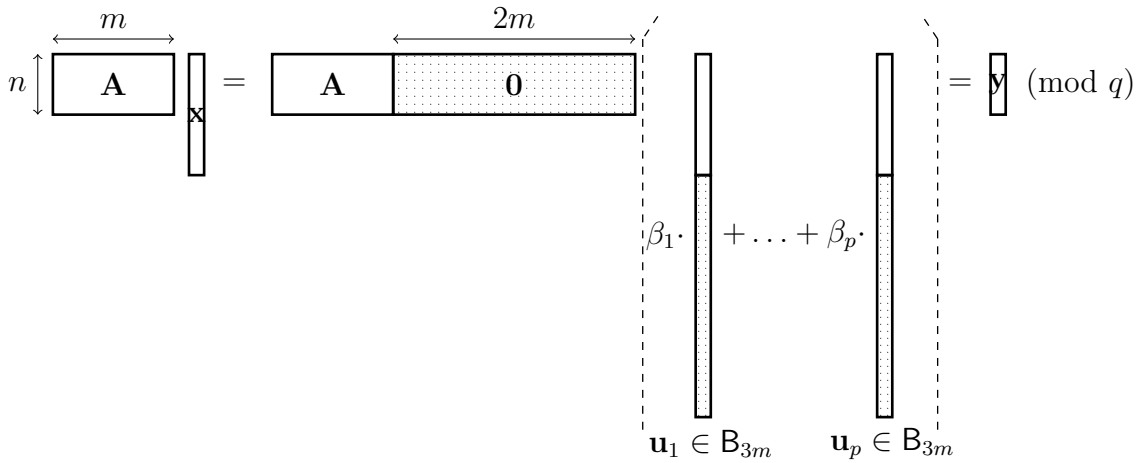

Fig. 3.2: The Decomposition-Extension technique for $\mathrm{R}_{\mathsf{ISIS}^\infty}(n, m, q, \beta)$.

(b) For each $j \in [p]$, let $\mathbf{u}_j \leftarrow \mathsf{EleExt}_m(\widetilde{\mathbf{u}}_j)$.

(c) Return $\mathbf{u}_1, \ldots, \mathbf{u}_p$.

4. Matrix Extension. On input a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the procedure $\mathsf{MatrixExt}_{n,m}$ outputs matrix $\mathbf{A}^* \in \mathbb{Z}_q^{n \times 3m}$ by appending $2m$ zero-columns to $\mathbf{A}$.

Specifically, if $\mathbf{x}$ is a solution to the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ instance $(\mathbf{A}, \mathbf{y})$, and if we let

$$\mathbf{u}_1, \ldots, \mathbf{u}_p \leftarrow \mathsf{VectorDE}_{m,\beta}(\mathbf{x}) \text{ and } \mathbf{A}^* \leftarrow \mathsf{MatrixExt}_{n,m}(\mathbf{A}),$$

then we will have $\mathbf{u}_j \in \mathsf{B}_{3m}$ for all $j \in [p]$, and $\mathbf{A}^* \cdot \left(\sum_{j=1}^{p} \beta_j \cdot \mathbf{u}_j \right) = \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q$.

In Figure 3.2 we give an illustration of this Decomposition-Extension technique. Our proof system **SternExt** is presented in Section 3.3.3.

3.3.3 The **SternExt** Proof System

Let m, q, β be positive integers, and $p = \lfloor \log \beta \rfloor + 1$. Let COM be a statistically hiding and computationally binding string commitment scheme. In this work, we will use the KTX commitment scheme from [85] (see also Section 2.3.2). For simplicity,

in the interactive protocol, we will not explicitly write the randomness ρ of COM.

The common input of the protocol is a pair $(\mathbf{A} \in \mathbb{Z}_q^{n\times m}, \mathbf{y} \in \mathbb{Z}_q^n)$ such that $\mathbf{y}$ belongs to the image of $\mathbf{A}$. The prover's auxiliary input is vector $\mathbf{x}$, such that $((\mathbf{A}, \mathbf{y}), \mathbf{x}) \in \mathrm{R}_{\mathsf{ISIS}^\infty}(n, m, q, \beta)$. Prior to the interaction, both prover and verifier form the extended matrix $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}_{n,m}(\mathbf{A})$. In addition, the prover computes $\mathbf{u}_1, \ldots, \mathbf{u}_p \leftarrow \mathsf{VectorDE}_{m,\beta}(\mathbf{x})$. In the protocol, the prover's goal is to convince the verifier in ZK that it knows $\mathbf{u}_1, \ldots, \mathbf{u}_p \in \mathsf{B}_{3m}$ such that $\mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{u}_j\big) = \mathbf{y} \bmod q$. The prover and the verifier interact as follows:

1. **Commitment.** Prover P picks p vectors $\mathbf{r}_1, \ldots, \mathbf{r}_p \xleftarrow{\$} \mathbb{Z}_q^{3m}$; p permutations $\pi_1, \ldots, \pi_p \xleftarrow{\$} \mathsf{S}_{3m}$, and sends the commitment $\mathrm{CMT} := (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\pi_1, \ldots, \pi_p, \mathbf{A}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{r}_j) \bmod q\big) \\ \mathbf{c}_2 = \mathrm{COM}\big(\pi_1(\mathbf{r}_1), \ldots, \pi_p(\mathbf{r}_p)\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\pi_1(\mathbf{u}_1 + \mathbf{r}_1), \ldots, \pi_p(\mathbf{u}_p + \mathbf{r}_p)\big) \end{cases}$$

2. **Challenge.** Receiving CMT, verifier sends a challenge $Ch \xleftarrow{\$} \{1, 2, 3\}$ to P.

3. **Response.** Prover replies as follows:

- If $Ch = 1$, then reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. For each j, let $\mathbf{v}_j = \pi_j(\mathbf{u}_j)$, and $\mathbf{w}_j = \pi_j(\mathbf{r}_j)$. Send $\mathrm{RSP} := \big(\mathbf{v}_1, \ldots, \mathbf{v}_p, \mathbf{w}_1, \ldots, \mathbf{w}_p\big)$.
- If $Ch = 2$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. For each j, let $\phi_j = \pi_j$, and $\mathbf{z}_j = \mathbf{u}_j + \mathbf{r}_j$. Send $\mathrm{RSP} := \big(\phi_1, \ldots, \phi_p, \mathbf{z}_1, \ldots, \mathbf{z}_p\big)$.
- If $Ch = 3$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. For each j, let $\psi_j = \pi_j$, and $\mathbf{s}_j = \mathbf{r}_j$. Send $\mathrm{RSP} := \big(\psi_1, \ldots, \psi_p, \mathbf{s}_1, \ldots, \mathbf{s}_p\big)$.

Verification. Receiving the response RSP, verifier V proceeds as follows:

- If $Ch = 1$: Check that $\mathbf{v}_j \in \mathsf{B}_{3m}$ for all $j \in [p]$, and

$$\begin{cases} \mathbf{c}_2 = \mathrm{COM}\big(\mathbf{w}_1, \ldots, \mathbf{w}_p\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\mathbf{v}_1 + \mathbf{w}_1, \ldots, \mathbf{v}_p + \mathbf{w}_p\big). \end{cases}$$

- If $Ch = 2$: Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\phi_1, \ldots, \phi_p, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j) - \mathbf{y} \bmod q\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\phi_1(\mathbf{z}_1), \ldots, \phi_p(\mathbf{z}_p)\big). \end{cases}$$

- If $Ch = 3$: Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\psi_1, \ldots, \psi_p, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_j) \bmod q\big) \\ \mathbf{c}_2 = \mathrm{COM}\big(\psi_1(\mathbf{s}_1), \ldots, \psi_p(\mathbf{s}_p)\big). \end{cases}$$

In each case, verifier V outputs the decision $d = 1$ (Accept) if and only if all the conditions hold. Otherwise, it outputs $d = 0$ (Reject).

Completeness. We observe that if prover P has a valid witness $\mathbf{x}$ for the relation $\mathrm{R}_{\mathsf{ISIS}^\infty}(n, m, q, \beta)$ and follows the protocol, then it always gets accepted by V. Therefore, the proof system has perfect completeness.

Communication cost. The size of the commitment scheme from [85] is $\widetilde{\mathcal{O}}(n \log q)$. If $Ch = 1$, then the size of RSP is $3pm + 3pm \log q$. If $Ch = 2$ or $Ch = 3$, then RSP consists of p vectors in $\mathbb{Z}_q^{3m}$ and p permutations. We note that in practice, instead of sending the permutations and vectors, one would send the *random seed* of the Pseudorandom Number Generator (PRNG) used to generate these data, and thus significantly reduce the communication cost. Overall, the total communication cost

of the protocol is $\log \beta \cdot \widetilde{\mathcal{O}}(n \log q)$.

3.3.4 Statistical Zero-Knowledge

We now prove that the **SternExt** proof system in Section 3.3.3 is statistically zero-knowledge, by exhibiting a transcript simulator.

Theorem 3.3.2. *If* COM *is a statistically hiding string commitment scheme, then the proof system* **SternExt** *is statistically zero-knowledge.*

Proof. Adapting the techniques of [137] and [85], we construct a simulator $\mathcal{S}$ which has black-box access to a (possibly cheating) verifier $\widehat{V}$, such that on input the public parameters $\mathbf{A}$ (and implicitly its extension $\mathbf{A}^*$) and $\mathbf{y}$, outputs with probability $2/3$ a successful transcript (i.e., an accepted interaction). Moreover, the view of $\widehat{V}$ in the simulation is statistically close to that in the real interaction. The simulator $\mathcal{S}$ begins by selecting a random $\overline{Ch} \in \{1, 2, 3\}$ (a prediction of the challenge value that $\widehat{V}$ will *not* choose), and a random tape r' of $\widehat{V}$. We note that in all the cases we consider below, by the assumption on the commitment scheme COM, the distributions of $\mathbf{c}_1^{'}, \mathbf{c}'_2, \mathbf{c}'_3$ are statistically close to the distributions of the commitments in the real interaction, and thus, the distributions of the challenge Ch from $\widehat{V}$ is also statistically close to that in the real interaction.

Case $\overline{Ch} = 1$: The simulator $\mathcal{S}$ computes $\mathbf{x}' \in \mathbb{Z}_q^m$ such that $\mathbf{A} \cdot \mathbf{x}' = \mathbf{y} \bmod q$ using linear algebra. It picks $p-1$ random vectors $\widetilde{\mathbf{u}}'_1, \ldots, \widetilde{\mathbf{u}}'_{p-1} \xleftarrow{\$} \mathbb{Z}_q^m$ and sets:

$$\widetilde{\mathbf{u}}'_p := \mathbf{x}' - \sum_{j=1}^{p-1} \beta_j \cdot \widetilde{\mathbf{u}}'_j \bmod q.$$

In other words, we have $\mathbf{x}' = \sum_{j=1}^{p} \beta_j \cdot \widetilde{\mathbf{u}}'_j \bmod q$. Now for each j, the simulator extends $\widetilde{\mathbf{u}}'_j$ to $\mathbf{u}'_j \in \mathbb{Z}_q^{3m}$ by appending $2m$ random coordinates. It then picks p

vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_p \xleftarrow{\$} \mathbb{Z}_q^{3m}$; p permutations $\pi'_1, \ldots, \pi'_p \xleftarrow{\$} \mathsf{S}_{3m}$; and uniformly random strings $\rho'_1, \rho'_2, \rho'_3$. It sends the commitment $(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\widehat{V}$, where:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\big(\pi'_1, \ldots, \pi'_p, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}'_j) \bmod q; \rho'_1\big) \\ \mathbf{c}'_2 = \mathrm{COM}\big(\pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p); \rho'_2\big) \\ \mathbf{c}'_3 = \mathrm{COM}\big(\pi'_1(\mathbf{u}'_1 + \mathbf{r}'_1), \ldots, \pi'_p(\mathbf{u}'_p + \mathbf{r}'_p); \rho'_3\big). \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ provides a transcript as follows:

- If $Ch = 1$: Output $\perp$ and halt.
- If $Ch = 2$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 2, (\pi'_1, \ldots, \pi'_p, \mathbf{u}'_1 + \mathbf{r}'_1, \ldots, \mathbf{u}'_p + \mathbf{r}'_p); \rho'_1, \rho'_3\Big).$$

- If $Ch = 3$: Output $\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 3, (\pi'_1, \ldots, \pi'_p, \mathbf{r}'_1, \ldots, \mathbf{r}'_p); \rho'_1, \rho'_2\Big)$.

Case $\overline{Ch} = 2$: The simulator $\mathcal{S}$ picks vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_p \xleftarrow{\$} \mathbb{Z}_q^{3m}$; $\mathbf{u}'_1, \ldots, \mathbf{u}'_p \xleftarrow{\$} \mathsf{B}_{3m}$; permutations $\pi'_1, \ldots, \pi'_p \xleftarrow{\$} \mathsf{S}_{3m}$; and uniformly random strings $\rho'_1, \rho'_2, \rho'_3$. It sends the commitment $(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\widehat{V}$, where:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\big(\pi'_1, \ldots, \pi'_p, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}'_j) \bmod q; \rho'_1\big) \\ \mathbf{c}'_2 = \mathrm{COM}\big(\pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p); \rho'_2\big) \\ \mathbf{c}'_3 = \mathrm{COM}\big(\pi'_1(\mathbf{u}'_1 + \mathbf{r}'_1), \ldots, \pi'_p(\mathbf{u}'_p + \mathbf{r}'_p); \rho'_3\big) \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ computes the following transcript:

- If $Ch = 1$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 1, (\pi'_1(\mathbf{u}'_1), \ldots, \pi'_p(\mathbf{u}'_p), \pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p)); \rho'_2, \rho'_3\Big).$$

- If $Ch = 2$: Output $\perp$ and halt.

- If $Ch = 3$: Output $\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 3, (\pi'_1, \ldots, \pi'_p, \mathbf{r}'_1, \ldots, \mathbf{r}'_p); \rho'_1, \rho'_2\Big)$.

Case $\overline{Ch} = 3$: The simulator picks the uniformly random vectors, permutations, and strings exactly as in the case $\overline{Ch} = 2$ above, but sends the following:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\big(\pi'_1, \ldots, \pi'_p, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot (\mathbf{u}'_j + \mathbf{r}'_j)) - \mathbf{y} \bmod q; \rho'_1\big) \\ \mathbf{c}'_2 = \mathrm{COM}\big(\pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p); \rho'_2\big) \\ \mathbf{c}'_3 = \mathrm{COM}\big(\pi'_1(\mathbf{u}'_1 + \mathbf{r}'_1), \ldots, \pi'_p(\mathbf{u}'_p + \mathbf{r}'_p); \rho'_3\big) \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ computes a transcript as follows:

- If $Ch = 1$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 1, (\pi'_1(\mathbf{u}'_1), \ldots, \pi'_p(\mathbf{u}'_p), \pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p)); \rho'_2, \rho'_3\Big).$$

- If $Ch = 2$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 2, (\pi'_1, \ldots, \pi'_p, \mathbf{u}'_1 + \mathbf{r}'_1, \ldots, \mathbf{u}'_p + \mathbf{r}'_p); \rho'_1, \rho'_3\Big).$$

- If $Ch = 3$: Output $\perp$ and halt.

We observe that the probability that the simulator outputs $\perp$ is negligibly close to 1/3. Moreover, one can check that whenever $\mathcal{S}$ does not halt, it will provide a

successful transcript, and the distribution of the transcript is statistically close to that of the prover in the real interaction. Hence, we have constructed a simulator that can successfully impersonate the honest prover with probability 2/3, and completed the proof. □

3.3.5 Proof of Knowledge

The fact that anyone can run the simulator to convince the verifier with probability 2/3 implies that the **SternExt** proof system has soundness error at least 2/3. In the following, we prove that it is indeed a proof of knowledge for the relation $\mathrm{R}_{\mathsf{ISIS}^\infty}(n, m, q, \beta)$ with knowledge error $\kappa = 2/3$.

Theorem 3.3.3. *Assume that* COM *is a computationally binding string commitment scheme. Then there exists a knowledge extractor $\mathcal{K}$ such that the following holds. If $\mathcal{K}$ has access to a cheating prover who convinces the verifier on input $(\mathbf{A}, \mathbf{y})$ with probability $2/3 + \epsilon$ for some $\epsilon > 0$ and in time T, then with overwhelming probability and in time $T \cdot \mathsf{poly}(n, m, \log q, 1/\epsilon)$, $\mathcal{K}$ outputs an $\mathbf{x}'$ such that $((\mathbf{A}, \mathbf{y}); \mathbf{x}') \in \mathrm{R}_{\mathsf{ISIS}^\infty}(n, m, q, \beta)$.*

As a corollary, **SternExt** is sound for uniformly random $(\mathbf{A}, \mathbf{y})$ under the assumption that the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ problem is hard.

Proof. We apply the technique of [139] which relies on trees to model the probability space corresponding to the protocol execution. Suppose a cheating prover $\widehat{P}$ can convince the verifier with probability $2/3 + \epsilon$. Then by rewinding $\widehat{P}$ a number of times polynomial in $1/\epsilon$, the knowledge extractor $\mathcal{K}$ can find with overwhelming probability a node with 3 sons in the tree associated with the protocol between $\widehat{P}$ and the verifier. This node corresponds to the reception of all 3 values of the challenge. In other words, $\widehat{P}$ is able to answer correctly to all challenges for the

same commitment. Therefore, $\mathcal{K}$ can get the following relations:

$$\begin{aligned}
\mathrm{COM}\big(\phi_1,\ldots,\phi_p,\mathbf{A}^*\cdot(\sum_{j=1}^{p}\beta_j\cdot\mathbf{z}_j)-\mathbf{y}\big) &= \mathrm{COM}\big(\psi_1,\ldots,\psi_p,\mathbf{A}^*\cdot(\sum_{j=1}^{p}\beta_j\cdot\mathbf{s}_j)\big)\\
\mathrm{COM}\big(\mathbf{w}_1,\ldots,\mathbf{w}_p\big) &= \mathrm{COM}\big(\psi_1(\mathbf{s}_1),\ldots,\psi_p(\mathbf{s}_p)\big)\\
\mathrm{COM}\big(\phi_1(\mathbf{z}_1),\ldots,\phi_p(\mathbf{z}_p)\big) &= \mathrm{COM}\big(\mathbf{v}_1+\mathbf{w}_1,\ldots,\mathbf{v}_p+\mathbf{w}_p\big),
\end{aligned}$$

and $\mathbf{v}_j \in \mathsf{B}_{3m}$ for all $j \in [p]$. Since COM is computationally binding, it follows that:

$$\mathbf{A}^*\Big(\sum_{j=1}^{p}\beta_j\cdot(\mathbf{z}_j-\mathbf{s}_j)\Big)=\mathbf{y} \bmod q,$$

and for all j, we have $\phi_j=\psi_j; \mathbf{w}_j=\psi_j(\mathbf{s}_j); \mathbf{v}_j+\mathbf{w}_j=\phi_j(\mathbf{z}_j); \mathbf{v}_j\in\mathsf{B}_{3m}$. This implies that $\phi_j(\mathbf{z}_j-\mathbf{s}_j)=\mathbf{v}_j\in\mathsf{B}_{3m}$. Let $\mathbf{v}'_j:=\mathbf{z}_j-\mathbf{s}_j=\phi_j^{-1}(\mathbf{v}_j)$, then we obtain that $\mathbf{A}^*\cdot\Big(\sum_{j=1}^{p}\beta_j\cdot\mathbf{v}'_j\Big)=\mathbf{y} \bmod q$ and $\mathbf{v}'_j\in\mathsf{B}_{3m}$. Then for each $\mathbf{v}'_j$, we drop the last $2m$ coordinates to obtain $\widetilde{\mathbf{v}'_j}\in\{-1,0,1\}^m$. Now we have $\mathbf{A}\cdot\Big(\sum_{j=1}^{p}\beta_j\cdot\widetilde{\mathbf{v}'_j}\Big)=\mathbf{y} \bmod q$. Let $\mathbf{x}'=\sum_{j=1}^{p}\beta_j\cdot\widetilde{\mathbf{v}'_j}$. Then $\mathbf{A}\cdot\mathbf{x}'=\mathbf{y} \bmod q$, and

$$\|\mathbf{x}'\|_\infty\le\sum_{j=1}^{p}\beta_j\cdot\|\widetilde{\mathbf{v}'_j}\|_\infty\le\sum_{j=1}^{p}\beta_j=\beta.$$

The knowledge extractor outputs $\mathbf{x}'$, satisfying $((\mathbf{A},\mathbf{y}),\mathbf{x}')\in\mathrm{R}_{\mathsf{ISIS}^\infty}(n,m,q,\beta)$. This concludes the proof. □

The **SternExt** proof system will serve as the fundamental object in the rest of this thesis. In Section 3.4 we will adapt **SternExt** to obtain a proof system for the SIS^∞ problem. In Chapter 4, we will rely on it to construct proofs of plaintext knowledged for four LWE-based encryption schemes. In Chapter 5, **SternExt** will be used as the building block in our constructions of two lattice-based identity-based

identification schemes. In Chapter 6, we will extend it to handle a more elaborate relation, resulting in an improved lattice-based group signature scheme.

3.4 Improved Proof System for the SIS^∞ Problem

In this section, we will construct a proof system for the homogeneous $\mathsf{SIS}^\infty(n, m, q, \beta)$ problem, which is associated with the following relation:

$$\mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta) = \Big\{(\mathbf{A}, \mathbf{x}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}^m\colon\ (0 < \|\mathbf{x}\|_\infty \leq \beta) \wedge (\mathbf{A}\cdot\mathbf{x} = \mathbf{0} \bmod q)\Big\}.$$

We are interested in proof systems for the SIS^∞ problem for both theoretical and practical reasons. On the theoretical side, we want to know if SIS^∞ admits an efficient statistical ZK proof with strong security guarantee. On the practical side, a ZKPoK for $\mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta)$ will yield a secure identification schemes from SIS^∞.

We note that if $(\mathbf{A}, \mathbf{x}) \in \mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta)$ then $\mathbf{x}$ is a short non-zero vector in the lattice $\Lambda^\perp(\mathbf{A})$. To prove the knowledge of such a vector $\mathbf{x}$ in zero-knowledge, one can apply the Micciancio-Vadhan proof system for GapSVP_γ [115], which admits an extraction gap $g = \widetilde{\mathcal{O}}(\sqrt{n})$, as in the case with GapCVP_γ. Nevertheless, up to the best of our knowledge, this is the only known efficient proof system for the relation $\mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta)$. Indeed, the Lyubashevsky's proof system in Section 3.2.2 and the **SternExt** proof system in Section 3.3 do not work straightforwardly for SIS^∞. The main technical issue is that: on one hand, a valid SIS solution must be non-zero; on the other hand a cheating prover using $\mathbf{x} = 0$ can easily pass the verifications in these protocols.

We now show how to modify **SternExt** to obtain a proof system for the relation $\mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta)$, which has the following properties:

- The knowledge extractor is able to produce $\mathbf{x}' \in \mathbb{Z}^m$ such that $0 < \|\mathbf{x}'\|_\infty \leq 2\beta - 1$. In other words, the extraction gap is a small constant $g < 2$.
- The communication cost is $\log \beta \cdot \widetilde{\mathcal{O}}(n \log q)$. As mentioned earlier, in most of cryptographic applications, we have $q = \mathsf{poly}(n)$, and $\beta < q$, and thus, the asymptotic cost is $\widetilde{\mathcal{O}}(n)$.

Our modifications are as follows. Let $p = \lfloor \log \beta \rfloor + 1$ as in **SternExt**. However, instead of using the sequence $\beta_1, \ldots, \beta_p$ defined in Section 3.3.2, we redefine it as the sequence of powers of 2, namely: $\beta_1 = 2^{p-1}, \beta_2 = 2^{p-2}, \ldots, \beta_p = 2^0 = 1$.

Example 3.4.1. Let $\beta = 115$, then we have $p = \lfloor \log (115) \rfloor + 1 = 7$. Recall that for **SternExt**, the sequence is:

$$\beta_1 = 58,\ \beta_2 = 29,\ \beta_3 = 14,\ \beta_4 = 7,\ \beta_5 = 4,\ \beta_6 = 2,\ \beta_7 = 1.$$

Now, we redefine it as:

$$\beta_1 = 64,\ \beta_2 = 32,\ \beta_3 = 16,\ \beta_4 = 8,\ \beta_5 = 4,\ \beta_6 = 2,\ \beta_7 = 1.$$

It is obvious that every integer in the interval $[0, \beta]$ can be efficiently express as a subset sum of $\beta_1, \ldots, \beta_p$ (where the corresponding coefficients are exactly the bits in its binary representation). Hence, the Decomposition-Extension technique in Section 3.3.2 should work well with the new sequence.

Now, let $\mathbf{x} \in \mathbb{Z}^m$ be such that $\|\mathbf{x}\|_\infty \leq \beta$, and let $\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_p \leftarrow \mathsf{EleDec}_{m,\beta}(\mathbf{x})$, namely we have $\mathbf{x} = \sum_{j=1}^{p} 2^{p-j} \cdot \tilde{\mathbf{u}}_j$. We then observe that $\mathbf{x} = \mathbf{0}$ is equivalent to $\forall j : \tilde{\mathbf{u}}_j = \mathbf{0}$. Hence, to prevent the prover from cheating with $\mathbf{x} = \mathbf{0}$, we constrain it to prove in zero-knowledge that *at least one* of its $\tilde{\mathbf{u}}_j$'s is non-zero.

Next, observe that if $\mathbf{x} = (x_1, \ldots, x_m)$ is a valid witness for $\mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta)$, and if 2^ℓ is the highest power of 2 dividing $\mathsf{gcd}(x_1, \ldots, x_m)$, then $\mathbf{x}^* = (x_1/2^\ell, \ldots, x_m/2^\ell)$ is also a valid witness for $\mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta)$. Suppose that $\tilde{\mathbf{u}}_1^*, \ldots, \tilde{\mathbf{u}}_p^* \leftarrow \mathsf{EleDec}_{m,\beta}(\mathbf{x}^*)$. We then note that vector $\tilde{\mathbf{u}}_p^*$, whose coordinates are the least significant bits of $x_1/2^\ell, \ldots, x_m/2^\ell$, must be non-zero. To prove the knowledge of such a vector $\tilde{\mathbf{u}}_p^*$ in ZK, the prover can use the extension technique of **SternExt**, but in dimension $3m-1$ instead of dimension $3m$. More precisely, the prover appends $2m-1$ coordinates to $\tilde{\mathbf{u}}_p^*$ to get a vector $\mathbf{u}_p^*$ that has exactly m coordinates equal to 1; m coordinates equal to -1; and $m-1$ coordinates equal to 0. Seeing a permutation of $\mathbf{u}_p^*$ that has these constraints, the verifier will be convinced that the original vector $\tilde{\mathbf{u}}_p^*$ must have *at least one* coordinate equal to 1 or -1, and thus it must be non-zero!

In summary, the modified **SternExt** proof system for $\mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta)$ works as follows: The common input is a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. The auxiliary input of the prover is $\mathbf{x}$, such that $(\mathbf{A}, \mathbf{x}) \in \mathrm{R}_{\mathsf{SIS}^\infty}(n, m, q, \beta)$. Prior to the interaction, both parties append $2m-1$ and $2m$ zero-columns to matrix $\mathbf{A}$ to get matrix $\mathbf{A}_{3m-1}$, and matrix $\mathbf{A}_{3m}$, respectively. In addition, the prover performs the following steps:

1. Shifting: Map $\mathbf{x}$ to $\mathbf{x}^*$, as described above.
2. Decomposition: Let $\tilde{\mathbf{u}}_1^*, \ldots, \tilde{\mathbf{u}}_p^* \leftarrow \mathsf{EleDec}_{m,\beta}(\mathbf{x}^*)$. Namely, $\mathbf{x}^* = \sum_{j=1}^{p} 2^{p-j} \cdot \tilde{\mathbf{u}}_j^*$.
3. Extensions: Append $(2m-1)$ coordinates to $\tilde{\mathbf{u}}_p^*$ as described above to get $\mathbf{u}_p^*$, and perform the usual extension to dimension $3m$ for the other vectors: For $j \in [p-1]$, let $\mathbf{u}_j^* \leftarrow \mathsf{EleExt}_m(\tilde{\mathbf{u}}_j^*)$.

We now have

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q \Leftrightarrow \mathbf{A}_{3m} \cdot \Big(\sum_{j=1}^{p-1} 2^{p-j} \cdot \mathbf{u}_j^* \Big) + \mathbf{A}_{3m-1} \cdot \mathbf{u}_p^* = \mathbf{0} \bmod q.$$

Therefore, we can now apply the **SternExt** proof system with a small tweak: The constraints of $\mathbf{u}_p^*$ are verified using a random permutation of $3m-1$ elements. This leads to a ZK proof system for the $\mathsf{SIS}^\infty(n,m,q,\beta)$ problem.

The constructions of the simulator and the knowledge extractor follow the same principles as for the **SternExt** proof system. The main difference here is that: The knowledge extractor is only able to produce $\tilde{\mathbf{v}}'_1,\ldots,\tilde{\mathbf{v}}'_p \in \{-1,0,1\}^m$ such that

$$\tilde{\mathbf{v}}'_p \neq \mathbf{0} \quad \text{and} \quad \mathbf{A}\cdot\big(\sum_{j=1}^{p} 2^{p-j}\cdot\tilde{\mathbf{v}}'_j\big) = \mathbf{0} \bmod q.$$

If we let $\mathbf{x}' = \sum_{j=1}^{p} 2^{p-j}\cdot\tilde{\mathbf{v}}'_j$, then $\mathbf{x}' \neq \mathbf{0}$ (because $\tilde{\mathbf{v}}'_p \neq \mathbf{0}$); $\mathbf{A}\cdot\mathbf{x}' = \mathbf{0} \bmod q$; and

$$\|\mathbf{x}'\|_\infty \leq \sum_{j=1}^{p} 2^{p-j}\cdot\|\tilde{\mathbf{v}}'_j\|_\infty \leq \sum_{j=1}^{p} 2^{p-j} = 2^p - 1 = 2^{\lfloor\log\beta\rfloor+1} - 1 \leq 2\beta - 1.$$

We conclude this section by a theorem that summarizes the above discussions.

Theorem 3.4.2. *There is a statistical* ZK *proof system for* $\mathrm{R}_{\mathsf{SIS}^\infty}(n,m,q,\beta)$ *with perfect completeness, knowledge error* $2/3$*, communication cost* $\log\beta\cdot\widetilde{\mathcal{O}}(n\log q)$*, and extraction gap* $g<2$*.*

4. PROOFS OF PLAINTEXT KNOWLEDGE FOR LWE-BASED ENCRYPTION SCHEMES

4.1 Introduction

In this chapter, we are interested in zero-knowledge proofs of plaintext knowledge for LWE-based encryption schemes. Given an encryption scheme with public key pk, a proof of plaintext knowledge (PoPK) allows a sender (or prover) to convince the receiver (or verifier) that it knows the plaintext M of some ciphertext $c = \mathsf{Enc}(\mathsf{pk}, M)$. The witness used in the protocol consists of the plaintext and the encryption randomness used to generate c. A PoPK should ensure that the verifier or an eavesdropper cannot learn any additional information about M. In other words, the proof system is required to be zero-knowledge.

Proofs of plaintext knowledge are useful in constructing interactive encryption protocols that are secure under chosen ciphertext attacks. Interactive encryption protocols are used when both the sender and the receiver are online. Starting with a CPA-secure encryption scheme which has a ZKPoPK, the sender of a ciphertext c additionally participates in a PoPK with the receiver. The result is a CCA1-secure interactive encryption protocol [59, 65].

The problem of designing PoPK for lattice-based encryption schemes has received considerable attention in the last few years. Goldwasser and Kharchenko [72] adapted the Micciancio-Vadhan protocol [115] to construct a ZKPoPK for a variant

of the Ajtai-Dwork cryptosystem [8]. Xagawa and Tanaka [140] employed the Stern-KTX proof system [137, 85] to obtain ZKPoPK for NTRU [79]. These proof systems are efficient and direct, in the sense that they do not rely on general results about zero-knowledge.

Regarding the current mainstream of lattice-based encryption schemes, the existing ZKPoPK for LWE-based schemes ([127, 62, 122, 61],..) are somewhat unsatisfactory. First, all known such proof systems ([24, 25, 15, 51]) only target Regev's LWE-based encryption scheme [127]. Furthermore, they are all derived from secure Multi-Party Computation (MPC) protocols, via the IKOS transformation [81] from MPC to ZK. As a consequence, these proof systems are relatively inefficient (with communication cost $\widetilde{\mathcal{O}}(n^2)$) and rely on the assumption that SIVP_γ is hard for super-polynomial approximation factors (i.e., $\gamma = n^{\omega(1)}$). Thus, constructing efficient ZKPoPK with strong security guarantee for LWE-based encryption schemes remains an open question.

In this work, we introduced efficient statistical ZKPoPK for a variety of LWE-based encryption schemes. Moreover, the security of our proof systems relies directly on the security of the underlying encryption schemes. We achieved these nice features by adapting and generalizing the techniques used in the **SternExt** proof system in Chapter 3. Specifically, in Section 4.2, we present an improved ZKPoPK for Regev's scheme [127] with communication cost $\widetilde{\mathcal{O}}(n)$, whose security relies on the worst-case hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$. Next, in Section 4.3, we introduce a ZKPoPK for the Dual-Regev encryption scheme [62]. Finally, in Section 4.4, we construct ZKPoPK for two generalized LWE-based encryption schemes: the one introduced by Peikert, Vaikuntanathan, and Waters [122] and the one proposed by Gentry, Halevi, and Vaikuntanathan [61], which are the generalization of Regev's scheme and the Dual-Regev scheme, respectively. As mentioned above, our constructions immedi-

ately yield 4 efficient and CCA1-secure interactive encryption protocols from lattice assumptions.

4.2 Proof System for Regev's Encryption Scheme

We recall that in Regev's LWE-based encryption scheme (see Scheme 2.3.13), given the public key $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, the ciphertext is a pair $(\mathbf{u}, c) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, such that there exist encryption randomness $\mathbf{r} \in \{0,1\}^m$, and plaintext $M \in \{0,1\}$ satisfying $\mathbf{u} = \mathbf{A} \cdot \mathbf{r} \bmod q$ and $c = \mathbf{b}^T \cdot \mathbf{r} + M \cdot \lfloor q/2 \rfloor \bmod q$. Therefore, to prove the plaintext knowledge, one has to prove the possession of $\mathbf{r}$ and M, such that all the above conditions hold. More formally, a proof of plaintext knowledge (PoPK) for Regev encryption scheme is a PoK for the relation $\mathrm{R}_{\mathrm{Regev}}$ defined as follows.

Definition 4.2.1.

$$\mathrm{R}_{\mathrm{Regev}}(n,m,q) = \Big\{ \big((\mathbf{A},\mathbf{b}),(\mathbf{u},c),\mathbf{r} \| M\big) \in (\mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m) \times (\mathbb{Z}_q^n \times \mathbb{Z}_q) \times \{0,1\}^{m+1} : (\mathbf{u} = \mathbf{A}\cdot\mathbf{r} \bmod q) \wedge (c = \mathbf{b}^T \cdot \mathbf{r} + M \cdot \lfloor q/2 \rfloor \bmod q) \Big\}.$$

While all the existing PoPK for Regev's LWE-based encryption scheme are obtained via the IKOS transformation from MPC to ZK, our approach here is more direct: we observe that a zero-knowledge PoPK for $\mathrm{R}_{\mathrm{Regev}}$ can be obtained from a ZKPoK for $\mathrm{R}_{\mathsf{ISIS}^\infty}$. In particular, we form the following matrix:

$$\mathbf{A}' = \left[\begin{array}{c|c} & 0 \\ \mathbf{A} & \vdots \\ & 0 \\ \hline \mathbf{b}^{\mathrm{T}} & \lfloor q/2 \rfloor \end{array}\right] \in \mathbb{Z}_q^{(n+1)\times(m+1)},$$

and let $\mathbf{y} = (\mathbf{u} \| c) \in \mathbb{Z}_q^{n+1}$. Let $\mathbf{x} = (\mathbf{r} \| M) \in \{0,1\}^{m+1}$ be any witness of the relation $\mathrm{R}_{\mathrm{Regev}}$, then we observe that $\mathbf{A}' \cdot \mathbf{x} = \mathbf{y} \bmod q$. Therefore, vector $\mathbf{x}$ is a solution to the $\mathsf{ISIS}^{\infty}(n+1, m+1, q, 1)$ instance $(\mathbf{A}', \mathbf{y})$. In other words, we have shown that the relation $\mathrm{R}_{\mathrm{Regev}}$ can be embedded into the relation $\mathrm{R}_{\mathsf{ISIS}^{\infty}_{\beta=1}}(n+1, m+1, q)$. As shown in Section 3.3.1, by using a basic extension technique, one can construct an efficient ZKPoK for this relation (we do not need the decomposition technique here, since the infinity norm of the witness is exactly 1).

If a cheating prover succeeds in proving the knowledge of a plaintext $\mathbf{x} = (\mathbf{r} \| M)$, then we use the knowledge extractor to output a vector $\mathbf{x}' = (\mathbf{r}' \| M') \in \{0,1\}^{m+1}$. In particular, we obtain $\mathbf{r}' \in \{0,1\}^m$ such that $\mathbf{A} \cdot \mathbf{r}' = \mathbf{u} \bmod q$. Since $\mathbf{A}$ is chosen uniformly at random in $\mathbb{Z}_q^{n \times m}$, and the distribution of $\mathbf{u}$ is statistically close to uniform over $\mathbb{Z}_q^n$ (see [128, Section 5]), vector $\mathbf{r}'$ is a solution to the uniformly random $\mathsf{ISIS}^{\infty}(n, m, q, 1)$ instance $(\mathbf{A}, \mathbf{u})$. This implies, via the worst-case to average-case reduction (see Theorem 2.2.14), that the security of our ZKPoPK for Regev's LWE-based encryption scheme can be based on the assumed hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n)}$ (in the ℓ_2 norm). We summarize this result in the theorem below.

Theorem 4.2.2. *There is a statistical* ZKPoPK *for Regev encryption scheme relying on the worst-case hardness of the* $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n)}$ *problem, where each round of the protocol has perfect completeness, knowledge error* $2/3$*, and communication cost* $\widetilde{\mathcal{O}}(n \log q)$*.*

4.3 Proof System for Dual-Regev Encryption Scheme

We recall that in Dual-Regev encryption scheme (see Definition 2.3.15), given the public key $(\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^n$, the ciphertext is a pair $(\mathbf{b}, c) \in \mathbb{Z}_q^m \times \mathbb{Z}_q$, such that:

$$\exists\, \mathbf{s} \in \mathbb{Z}_q^n;\ (\mathbf{e}, z) \in \mathbb{Z}_q^m \times \mathbb{Z}_q;\ M \in \{0,1\} : \|\mathbf{e}\|_\infty \leq \beta, |z| \leq \beta \text{ and}$$
$$\mathbf{b} = \mathbf{A}^T \cdot \mathbf{s} + \mathbf{e} \bmod q;\ \ c = \mathbf{u}^T \cdot \mathbf{s} + z + M \cdot \lfloor q/2 \rfloor \bmod q.$$

Therefore, to prove the plaintext knowledge, one has to prove the possession of the objects $\mathbf{s}$, $\mathbf{e}$, z and M, such that all the above conditions hold. Observe that, if we let:

$$\mathbf{y} = (\mathbf{b} \| c) \in \mathbb{Z}_q^{m+1};\ \ \mathbf{x} = (\mathbf{e}, z) \in \mathbb{Z}_q^{m+1};\ \ \overline{\mathbf{A}} = \left[\mathbf{A} \middle| \mathbf{u}\right] \in \mathbb{Z}_q^{n\times(m+1)},$$

then we have: $\overline{\mathbf{A}}^T \cdot \mathbf{s} + \mathbf{x} + (0, \ldots, 0, \lfloor q/2 \rfloor)^T \cdot M = \mathbf{y} \bmod q$, where $(0, \ldots, 0, \lfloor q/2 \rfloor)^T$ is the column vector in dimension $m+1$, that has all entries 0 except for the last one being $\lfloor q/2 \rfloor$. To construct a ZKPoK for this relation, we first perform several preparation steps.

We first observe that the vector-scalar product $(0, \ldots, 0, \lfloor q/2 \rfloor)^T \cdot M$ can be transformed into a matrix-vector product. We consider M as the 1-dimensional vector (M), and extend it to a 2-dimensional vector $(M, 1-M) \in \{0,1\}^2$, so that it always has Hamming weight 1 for any $M \in \{0,1\}$. We then extend vector $(0, \ldots, 0, \lfloor q/2 \rfloor)^T$ to matrix $\mathbf{G} \in \{0, \lfloor q/2 \rfloor\}^{(m+1)\times 2}$ by appending 1 zero-column to it. Next, for $b \in \{0,1\}$, let δ_b denote the 2-dimensional vector $(b, 1-b)$, then we have $\{\delta_0, \delta_1\} = \mathsf{B}_2$.[1] Now, for any $M \in \{0,1\}$, we have:

$$(0, \ldots, 0, \lfloor q/2 \rfloor)^T \cdot M = \mathbf{G} \cdot \delta_M.$$

[1] We recall that for $m \geq 1$, B_{2m} denotes the set of all vectors $\mathbf{x} \in \{0,1\}^{2m}$ such that $\mathsf{wt}(\mathbf{x}) = m$ (see Section 2.1).

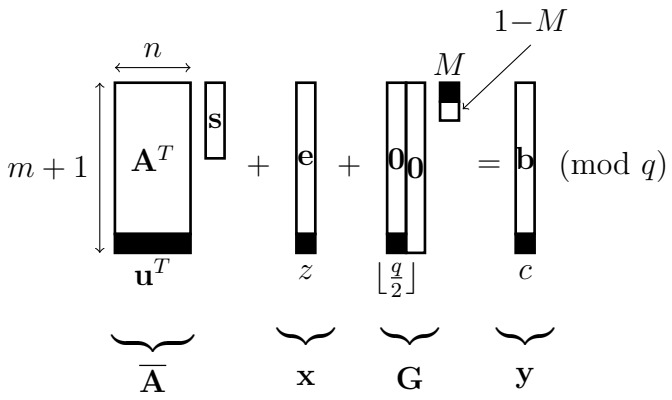

Fig. 4.1: Plaintext relation in Dual-Regev encryption scheme.

Our idea is to give vector δ_M the constraints that can be efficiently verified in zero-knowledge: in the protocol, we let the prover pick a uniformly random permutation τ of 2 elements, then send $\delta = \tau(\delta_M)$, and let the verifier check whether $\tau(\delta) \in \mathsf{B}_2$.

It follows from the above discussion that a proof of plaintext knowledge for the Dual-Regev encryption scheme can be expressed as a PoK for the relation $\mathrm{R}_{\mathrm{Dual}}$ defined below. In Figure 4.1, we give an illustration of this relation.

Definition 4.3.1.

$$\mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta) = \Big\{ \big((\overline{\mathbf{A}}, \mathbf{y}), \mathbf{s}, \mathbf{x}, \delta\big) \in \mathbb{Z}_q^{n\times(m+1)} \times \mathbb{Z}_q^{m+1} \times \mathbb{Z}_q^{n} \times \mathbb{Z}_q^{m+1} \times \mathsf{B}_2 : (\|\mathbf{x}\|_\infty \leq \beta) \wedge (\overline{\mathbf{A}}^T \cdot \mathbf{s} + \mathbf{x} + \mathbf{G} \cdot \delta = \mathbf{y} \bmod q) \Big\}$$

Next, we also express vector $\mathbf{x}$ as a matrix-vector product: $\mathbf{x} = \mathbf{I}_{m+1} \cdot \mathbf{x}$, where $\mathbf{I}_{m+1}$ is the identity matrix of order $m + 1$.[2] As in the **SternExt** proof system, when applying the Decomposition-Extension technique, in order to make the dimensions compatible, we append $2(m + 1)$ zero-columns to $\mathbf{I}_{m+1}$ to get matrix $\mathbf{I}^* \in \{0, 1\}^{(m+1)\times 3(m+1)}$.

[2] The idea of expressing $\mathbf{x}$ as $\mathbf{I}_{m+1} \cdot \mathbf{x}$ is quite natural, but helpful. This idea was first suggested by Xiang Xie (Email: xiexiangiscas@gmail.com), in a personal communication, and in a related context.

Given the above preparation steps, we now can construct a ZKPoK for the relation $\mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta)$ in the same fashion as for the **SternExt** proof system. Our protocol is as follows.

The interactive protocol. Let $p = \lfloor \log \beta \rfloor + 1$, and let $\beta_1, \ldots, \beta_p$ as defined in the **SternExt** proof system. Let COM be the string commitment scheme from [85]. The common input is $(\overline{\mathbf{A}}, \mathbf{y})$ (and implicitly $\mathbf{G}$ and $\mathbf{I}^*$, as defined above). The prover's auxiliary input is the tuple $(\mathbf{s}, \mathbf{x}, \delta)$. Prior to the interaction, the prover performs the Decomposition-Extension technique: Let $\mathbf{u}_1, \ldots, \mathbf{u}_p \leftarrow \mathsf{VectorDE}_{m+1,\beta}(\mathbf{x})$. In the protocol, it convinces the verifier that it knows $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{u}_1, \ldots, \mathbf{u}_p \in \mathsf{B}_{3(m+1)}$, and $\delta \in \mathsf{B}_2$ such that:

$$\overline{\mathbf{A}}^T \cdot \mathbf{s} + \mathbf{I}^* \cdot \sum_{j=1}^{p} \beta_j \cdot \mathbf{u}_j + \mathbf{G} \cdot \delta = \mathbf{y} \bmod q.$$

The prover P and the verifier V interact as follows:

1. **Commitment.** P samples vectors $\mathbf{g} \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{r}_1, \ldots, \mathbf{r}_p \xleftarrow{\$} \mathbb{Z}_q^{3(m+1)}$, $\mathbf{d} \xleftarrow{\$} \mathbb{Z}_q^2$; permutations $\pi_1, \ldots, \pi_p \xleftarrow{\$} \mathsf{S}_{3(m+1)}$, $\tau \xleftarrow{\$} \mathsf{S}_2$, and sends the commitment $\mathrm{CMT} := (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\pi_1, \ldots, \pi_p, \tau, \overline{\mathbf{A}}^T \cdot \mathbf{g} + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_j) + \mathbf{G} \cdot \mathbf{d} \bmod q\big) \\ \mathbf{c}_2 = \mathrm{COM}\big(\pi_1(\mathbf{r}_1), \ldots, \pi_p(\mathbf{r}_p), \tau(\mathbf{d})\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\pi_1(\mathbf{u}_1 + \mathbf{r}_1), \ldots, \pi_p(\mathbf{u}_p + \mathbf{r}_p), \tau(\delta + \mathbf{d})\big) \end{cases}$$

2. **Challenge.** Receiving CMT, V sends a challenge $Ch \xleftarrow{\$} \{1, 2, 3\}$ to P.

3. **Response.** P replies as follows:

- If $Ch = 1$, then reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. Let $\mathbf{d}_1 = \tau(\mathbf{d})$ and $\delta^{(1)} = \tau(\delta)$. For each j,

let $\mathbf{v}_j = \pi_j(\mathbf{u}_j)$, and $\mathbf{w}_j = \pi_j(\mathbf{r}_j)$.

Send RSP := $\big(\mathbf{v}_1, \ldots, \mathbf{v}_p, \mathbf{w}_1, \ldots, \mathbf{w}_p, \mathbf{d}_1, \delta^{(1)}\big)$.

- If $Ch = 2$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. Let $\tau_2 = \tau$, and $\mathbf{d}_2 = \delta + \mathbf{d}$, $\mathbf{g}_2 = \mathbf{s} + \mathbf{g}$. For each j, let $\phi_j = \pi_j$, and $\mathbf{z}_j = \mathbf{u}_j + \mathbf{r}_j$.

  Send RSP := $\big(\phi_1, \ldots, \phi_p, \tau_2, \mathbf{z}_1, \ldots, \mathbf{z}_p, \mathbf{g}_2, \mathbf{d}_2\big)$.

- If $Ch = 3$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. Let $\tau_3 = \tau$, $\mathbf{g}_3 = \mathbf{g}$, and $\mathbf{d}_3 = \mathbf{d}$. For each j, let $\psi_j = \pi_j$, and $\mathbf{s}_j = \mathbf{r}_j$.

  Send RSP := $\big(\psi_1, \ldots, \psi_p, \tau_3, \mathbf{s}_1, \ldots, \mathbf{s}_p, \mathbf{g}_3, \mathbf{d}_3\big)$.

Verification. Receiving the response RSP, verifier V proceeds as follows:

- If $Ch = 1$: Check that $\delta^{(1)} \in \mathsf{B}_2$, and $\mathbf{v}_j \in \mathsf{B}_{3(m+1)}$ for all $j \in [p]$, and

$$\begin{cases} \mathbf{c}_2 = \mathrm{COM}\big(\mathbf{w}_1, \ldots, \mathbf{w}_p, \mathbf{d}_1\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\mathbf{v}_1 + \mathbf{w}_1, \ldots, \mathbf{v}_p + \mathbf{w}_p, \delta^{(1)} + \mathbf{d}_1\big). \end{cases}$$

- If $Ch = 2$: Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\phi_1, \ldots, \phi_p, \tau_2, \overline{\mathbf{A}}^T \cdot \mathbf{g}_2 + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j) + \mathbf{G} \cdot \mathbf{d}_2 - \mathbf{y} \bmod q\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\phi_1(\mathbf{z}_1), \ldots, \phi_p(\mathbf{z}_p), \tau_2(\mathbf{d}_2)\big). \end{cases}$$

- If $Ch = 3$: Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\psi_1, \ldots, \psi_p, \tau_3, \overline{\mathbf{A}}^T \cdot \mathbf{g}_3 + \mathbf{I}^* (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_j) + \mathbf{G} \cdot \mathbf{d}_3 \bmod q) \\ \mathbf{c}_2 = \mathrm{COM}\big(\psi_1(\mathbf{s}_1), \ldots, \psi_p(\mathbf{s}_p), \tau_3(\mathbf{d}_3)\big). \end{cases}$$

In each case, verifier V outputs the decision $d = 1$ (Accept) if and only if all the conditions hold. Otherwise, he outputs $d = 0$ (Reject).

We first summarize the properties of the above proof system in the following theorem.

Theorem 4.3.2. *The given proof system is a statistical* ZKPoPK *for the Dual-Regev encryption scheme, where each round of the protocol has perfect completeness, knowledge error* $2/3$*, and communication cost* $\log \beta \cdot \widetilde{\mathcal{O}}(n \log q)$*.*

Completeness. We observe that if prover P has a valid witness $(\mathbf{s}, \mathbf{e}, \delta)$ for the relation $\mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta)$ and follows the protocol, then he always gets accepted by V. Therefore, the proof system has perfect completeness.

Communication cost. The communication cost of the present proof system is slightly larger than that of the **SternExt** proof sytem, with the introduction of new objects. The overall cost is $\log \beta \cdot \widetilde{\mathcal{O}}(n \log q)$, where the hidden constants is slightly bigger than that of **SternExt**.

Using the same techniques as for the **SternExt** proof system, we will show that the given proof system is a statistical ZKPoK for the relation $\mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta)$.

Statistical Zero-knowledge. We will prove that the given proof system is statistically zero-knowledge by exhibiting a transcript simulator.

Lemma 4.3.3. *If* COM *is a statistically hiding string commitment scheme, then the proof system for* $\mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta)$ *is statistically zero-knowledge. In particular, there exists an efficient simulator* $\mathcal{S}$*, which has black-box access to a (possibly cheating) verifier* $\widehat{V}$*, such that on input* $(\overline{\mathbf{A}}, \mathbf{y})$ *(and implicitly,* $\mathbf{G}$ *and* $\mathbf{I}^*$*), outputs with probability* $2/3$ *a successful transcript, the distribution of which is statistically close to that in the real interaction.*

Proof. We adapt the simulation technique for the **SternExt** proof system to construct a simulator satisfying the condition of the lemma. The simulator $\mathcal{S}$ begins by

selecting a random $\overline{Ch} \in \{1,2,3\}$ (a prediction of the challenge value that $\widehat{V}$ will *not* choose), and a random tape r' of $\widehat{V}$.

Case $\overline{Ch} = 1$:

Using linear algebra, $\mathcal{S}$ computes $\mathbf{s}' \in \mathbb{Z}_q^n$, $\mathbf{u}_1', \ldots, \mathbf{u}_p' \in \mathbb{Z}_q^{3(m+1)}$, $\delta' \in \mathbb{Z}_q^2$ such that:

$$\overline{\mathbf{A}}^T \cdot \mathbf{s}' + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{u}_j') + \mathbf{G} \cdot \delta' = \mathbf{y} \bmod q.$$

This can be done efficiently by picking random $\mathbf{s}', \delta, \mathbf{u}_2', \ldots, \mathbf{u}_p'$, then computing

$$\widetilde{\mathbf{u}}_1' = \mathbf{y} - \overline{\mathbf{A}}^T \cdot \mathbf{s}' - \mathbf{I}^* \sum_{j=2}^{p} \beta_j \cdot \mathbf{u}_j' - \mathbf{G} \cdot \delta' \in \mathbb{Z}_q^{m+1},$$

and appending $2(m+1)$ random coordinates to $\widetilde{\mathbf{u}}_1'$ to get $\mathbf{u}_1'$.

The simulator then picks vectors $\mathbf{g}' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{r}_1', \ldots, \mathbf{r}_p' \xleftarrow{\$} \mathbb{Z}_q^{3(m+1)}$, $\mathbf{d}' \in \mathbb{Z}_q^2$; permutations $\pi_1', \ldots, \pi_p' \xleftarrow{\$} \mathsf{S}_{3(m+1)}$, $\tau' \in \mathsf{S}_2$; and uniformly random strings $\rho_1', \rho_2', \rho_3'$. It sends the commitment $(\mathbf{c}_1', \mathbf{c}_2', \mathbf{c}_3')$ to $\widehat{V}$, where:

$$\begin{cases} \mathbf{c}_1' = \mathrm{COM}\big(\pi_1', \ldots, \pi_p', \tau', \overline{\mathbf{A}}^T \cdot \mathbf{g}' + \mathbf{I}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_j') + \mathbf{G} \cdot \mathbf{d}' \bmod q; \rho_1'\big) \\ \mathbf{c}_2' = \mathrm{COM}\big(\pi_1'(\mathbf{r}_1'), \ldots, \pi_p'(\mathbf{r}_p'), \tau'(\mathbf{d}'); \rho_2'\big) \\ \mathbf{c}_3' = \mathrm{COM}\big(\pi_1'(\mathbf{u}_1' + \mathbf{r}_1'), \ldots, \pi_p'(\mathbf{u}_p' + \mathbf{r}_p'), \tau'(\delta' + \mathbf{d}'); \rho_3'\big) \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ provides a transcript as follows:

- If $Ch = 1$: Output $\perp$ and halt.
- If $Ch = 2$: Output

$$\Big(r', (\mathbf{c}_1', \mathbf{c}_2', \mathbf{c}_3'), 2, (\pi_1', \ldots, \pi_p', \tau', \mathbf{u}_1' + \mathbf{r}_1', \ldots, \mathbf{u}_p' + \mathbf{r}_p', \mathbf{s}' + \mathbf{g}', \delta' + \mathbf{d}'); \rho_1', \rho_3'\Big).$$

- If $Ch = 3$: Output $\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 3, (\pi'_1, \ldots, \pi'_p, \tau', \mathbf{r}'_1, \ldots, \mathbf{r}'_p, \mathbf{g}', \mathbf{d}'); \rho'_1, \rho'_2\Big)$.

Case $\overline{Ch} = 2$: The simulator picks vectors $\mathbf{g}' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{r}'_1, \ldots, \mathbf{r}'_p \xleftarrow{\$} \mathbb{Z}_q^{3(m+1)}$, $\mathbf{d}' \xleftarrow{\$} \mathbb{Z}_q^2$; $\mathbf{u}'_1, \ldots, \mathbf{u}'_p \xleftarrow{\$} \mathsf{B}_{3(m+1)}$, $\delta' \xleftarrow{\$} \mathsf{B}_2$; permutations $\pi'_1, \ldots, \pi'_p \xleftarrow{\$} \mathsf{S}_{3(m+1)}$, $\tau' \xleftarrow{\$} \mathsf{S}_2$; and uniformly random strings $\rho'_1, \rho'_2, \rho'_3$. It sends the commitment $(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\widehat{V}$, where:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\big(\pi'_1, \ldots, \pi'_p, \tau', \overline{\mathbf{A}}^T \cdot \mathbf{g}' + \mathbf{I}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{r}'_j) + \mathbf{G} \cdot \mathbf{d}' \bmod q; \rho'_1\big) \\ \mathbf{c}'_2 = \mathrm{COM}\big(\pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{d}'); \rho'_2\big) \\ \mathbf{c}'_3 = \mathrm{COM}\big(\pi'_1(\mathbf{u}'_1 + \mathbf{r}'_1), \ldots, \pi'_p(\mathbf{u}'_p + \mathbf{r}'_p), \tau'(\delta' + \mathbf{d}'); \rho'_3\big) \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ computes the following transcript:

- If $Ch = 1$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 1, (\pi'_1(\mathbf{u}'_1), \ldots, \pi'_p(\mathbf{u}'_p), \pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{d}'), \tau'(\delta')); \rho'_2, \rho'_3\Big).$$

- If $Ch = 2$: Output $\perp$ and halt.

- If $Ch = 3$: Output $\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 3, (\pi'_1, \ldots, \pi'_p, \tau', \mathbf{r}'_1, \ldots, \mathbf{r}'_p, \mathbf{g}', \mathbf{d}'); \rho'_1, \rho'_2\Big)$.

Case $\overline{Ch} = 3$: The simulator picks the uniformly random vectors, permutations, and strings as in the case $\overline{Ch} = 2$ above, and additionally, $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$. It then sends the commitment $(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\widehat{V}$, where:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\big(\{\pi'_1\}_{j=1}^p, \tau', \overline{\mathbf{A}}^T \cdot (\mathbf{s}' + \mathbf{g}') + \mathbf{I}^* \cdot \sum_{j=1}^p \beta_j \cdot (\mathbf{u}'_j + \mathbf{r}'_j) - \mathbf{y} \bmod q; \rho'_1\big) \\ \mathbf{c}'_2 = \mathrm{COM}\big(\pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{d}'); \rho'_2\big) \\ \mathbf{c}'_3 = \mathrm{COM}\big(\pi'_1(\mathbf{u}'_1 + \mathbf{r}'_1), \ldots, \pi'_p(\mathbf{u}'_p + \mathbf{r}'_p), \tau'(\delta' + \mathbf{d}'); \rho'_3\big) \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ computes a transcript as follows:

- If $Ch = 1$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 1, (\pi'_1(\mathbf{u}'_1), \ldots, \pi'_p(\mathbf{u}'_p), \pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{d}'), \tau'(\delta')); \rho'_2, \rho'_3\Big).$$

- If $Ch = 2$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 2, (\pi'_1, \ldots, \pi'_p, \tau', \mathbf{u}'_1 + \mathbf{r}'_1, \ldots, \mathbf{u}'_p + \mathbf{r}'_p, \mathbf{s}' + \mathbf{g}', \delta' + \mathbf{d}'); \rho'_1, \rho'_3\Big).$$

- If $Ch = 3$: Output $\perp$ and halt.

We observe that, in every case we have considered above, since COM is statistically hiding, the distribution of the commitment from $\mathcal{S}$ and the challenge from $\widehat{V}$ are statistically close to those in the real interaction. Hence, the probability that the simulator outputs $\perp$ is negligibly close to 1/3. Moreover, one can check that whenever $\mathcal{S}$ does not halt, it will provide a successful transcript, and the distribution of the transcript is statistically close to that of the prover in the real interaction. Hence, we have constructed a simulator that can successfully impersonate the honest prover with probability 2/3, and completed the proof. □

Proof of knowledge. The following lemma states that the proof system presented in this section is a proof of knowledge for the relation $\mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta)$ with knowledge error 2/3.

Lemma 4.3.4. *Assume that* COM *is a computationally binding string commitment scheme. Then there exists a knowledge extractor $\mathcal{K}$ such that the following holds. If $\mathcal{K}$ has access to a cheating prover who convinces the verifier on input $(\overline{\mathbf{A}}, \mathbf{y})$ (and implicitly, $\mathbf{G}$ and $\mathbf{I}^*$) with probability $2/3+\epsilon$ for some $\epsilon > 0$ and in time T, then with overwhelming probability and in time $T \cdot \mathsf{poly}(n, m, \log q, 1/\epsilon)$, $\mathcal{K}$ outputs $(\mathbf{s}', \mathbf{x}', \delta')$*

such that $((\overline{\mathbf{A}}, \mathbf{y}), \mathbf{s}', \mathbf{x}', \delta') \in \mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta)$.

As a corollary, our proof system is sound if the Search-LWE problem is hard. Indeed, the tuple $(\mathbf{s}', \mathbf{x}', \delta')$ produced by the knowledge extractor in Lemma 4.3.4 can be used to recover the LWE secret of the given instance $(\mathbf{A}, \mathbf{b})$: If we let $\mathbf{e}' \in \mathbb{Z}_q^m$ be the vector obtained by dropping the last coordinate from $\mathbf{x}'$, then we have $\|\mathbf{e}'\|_\infty \leq \beta$, and $\mathbf{A}^T \cdot \mathbf{s}' + \mathbf{e}' = \mathbf{b} \bmod q$.

Proof. The proof uses the same argument as in the proof of Theorem 3.3.3. Namely, it can be shown that, in time $T \cdot \mathsf{poly}(n, m, \log q, 1/\epsilon)$, the knowledge extractor $\mathcal{K}$ can obtain with overwhelming probability 3 valid responses which correspond to all 3 challenges for the same commitment. Therefore, $\mathcal{K}$ can get the following relations:

$$\begin{gathered}
\mathrm{COM}\big(\phi_1, \ldots, \phi_p, \tau_2, \overline{\mathbf{A}}^T \cdot \mathbf{g}_2 + \mathbf{I}^* \cdot (\textstyle\sum_{j=1}^p \beta_j \cdot \mathbf{z}_j) + \mathbf{G} \cdot \mathbf{d}_2 - \mathbf{y} \bmod q\big) \\
= \mathrm{COM}\big(\psi_1, \ldots, \psi_p, \tau_3, \overline{\mathbf{A}}^T \cdot \mathbf{g}_3 + \mathbf{I}^* (\textstyle\sum_{j=1}^p \beta_j \cdot \mathbf{s}_j) + \mathbf{G} \cdot \mathbf{d}_3 \bmod q\big) \\
\mathrm{COM}\big(\mathbf{w}_1, \ldots, \mathbf{w}_p, \mathbf{d}_1\big) = \mathrm{COM}\big(\psi_1(\mathbf{s}_1), \ldots, \psi_p(\mathbf{s}_p), \tau_3(\mathbf{d}_3)\big) \\
\mathrm{COM}\big(\phi_1(\mathbf{z}_1), \ldots, \phi_p(\mathbf{z}_p), \tau_2(\mathbf{d}_2)\big) = \mathrm{COM}\big(\mathbf{v}_1 + \mathbf{w}_1, \ldots, \mathbf{v}_p + \mathbf{w}_p, \delta^{(1)} + \mathbf{d}_1\big),
\end{gathered}$$

and $\delta^{(1)} \in \mathsf{B}_2$; $\mathbf{v}_j \in \mathsf{B}_{3(m+1)}$ for all $j \in [p]$. Since COM is computationally binding, it follows that:

$$\begin{gathered}
\overline{\mathbf{A}}^T \cdot (\mathbf{g}_2 - \mathbf{g}_3) + \mathbf{I}^*\big(\sum_{j=1}^p \beta_j \cdot (\mathbf{z}_j - \mathbf{s}_j)\big) + \mathbf{G} \cdot (\mathbf{d}_2 - \mathbf{d}_3) = \mathbf{y} \bmod q; \\
\tau_2 = \tau_3;\ \ \mathbf{d}_1 = \tau_3(\mathbf{d}_3);\ \ \delta^{(1)} + \mathbf{d}_1 = \tau_2(\mathbf{d}_2);\ \ \delta^{(1)} \in \mathsf{B}_2; \\
\forall j \in [p] :\ \phi_j = \psi_j;\ \ \mathbf{w}_j = \psi_j(\mathbf{s}_j);\ \ \mathbf{v}_j + \mathbf{w}_j = \phi_j(\mathbf{z}_j);\ \ \mathbf{v}_j \in \mathsf{B}_{3(m+1)}.
\end{gathered}$$

Now, let $\mathbf{s}' = \mathbf{g_2} - \mathbf{g_3} \in \mathbb{Z}_q^n$; $\delta' = \mathbf{d}_2 - \mathbf{d}_3 = \tau_2^{-1}(\delta^{(1)})$; and for each $j \in [p]$,

let $\mathbf{v}'_j = \mathbf{z}_j - \mathbf{s}_j = \phi_j^{-1}(\mathbf{v_j})$, then we obtain that

$$\delta' \in \mathsf{B}_2;\ \forall j \in [p]:\ \mathbf{v}_j^{'} \in \mathsf{B}_{3(m+1)};$$
$$\overline{\mathbf{A}}^T \cdot \mathbf{s}' + \mathbf{I}^*\big(\sum_{j=1}^{p} \beta_j \cdot \mathbf{v}_j^{'}\big) + \mathbf{G} \cdot \delta' = \mathbf{y} \bmod q.$$

Now, for each $\mathbf{v}'_j$, we drop the last $2(m+1)$ coordinates to obtain $\widetilde{\mathbf{v}'_j} \in \{-1, 0, 1\}^{m+1}$, and we let $\mathbf{x}' = \sum_{j=1}^{p} \beta_j \cdot \widetilde{\mathbf{v}'_j}$. Then we have $\overline{\mathbf{A}}^T \cdot \mathbf{s}' + \mathbf{x}' + \mathbf{G} \cdot \delta' = \mathbf{y} \bmod q$, and

$$\|\mathbf{x}'\|_\infty \leq \sum_{j=1}^{p} \beta_j \cdot \|\widetilde{\mathbf{v}'_j}\|_\infty \leq \sum_{j=1}^{p} \beta_j = \beta.$$

The knowledge extractor then outputs $(\mathbf{s}', \mathbf{x}', \delta')$ satisfying

$$((\mathbf{A}, \mathbf{y}), \mathbf{s}', \mathbf{x}', \delta') \in \mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta).$$

This concludes the proof. □

4.4 Proof Systems for Generalized Schemes

In this section, we demonstrate how to obtain statistical zero-knowledge proofs of plaintext knowledge for two LWE-based encryption schemes:

1. The scheme introduced by Peikert, Vaikuntanathan and Waters [122], which is a generalized variant of Regev's encryption scheme.

2. The scheme proposed by Gentry, Halevi, and Vaikuntanathan [61], which is a generalized variant of Dual-Regev encryption scheme.

4.4.1 The Peikert-Vaikuntanathan-Waters Encryption Scheme

Peikert, Vaikuntanathan and Waters [122] introduced an amortized technique that leads to a generalized variant of Regev's LWE-based encryption with ciphertext blowup factor $\mathcal{O}(1)$. Below we will refer to this scheme as the PVW encryption scheme. Let us remind some facts on this scheme.

- Parameters: Security parameter n, an odd prime $q = \mathsf{poly}(\mathsf{n})$, integers $m = \mathcal{O}(n \log q)$, $\ell = \mathsf{poly}(n) \geq 1$, and k such that $1 \leq k \ll q$.
- The plaintext space is $[0, k]^\ell$.
- The public key is the pair $(\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}, \mathbf{P} \in \mathbb{Z}_q^{\ell \times m})$.
- To encrypt $\mathbf{v} \in [0, k]^\ell$, pick a vector $\mathbf{e} \xleftarrow{\$} \{0, 1\}^m$, and output the ciphertext $(\mathbf{u} \in \mathbb{Z}_q^n, \mathbf{c} \in \mathbb{Z}_q^\ell)$, where

$$\mathbf{u} = \mathbf{A} \cdot \mathbf{e} \bmod q \quad \text{and} \quad \mathbf{c} = \mathbf{P} \cdot \mathbf{e} + \lfloor \frac{q}{k} \rceil \cdot \mathbf{v} \bmod q.$$

Let n, m, q, ℓ, k be the parameters of the scheme. We define the plaintext relation for the PVW encryption scheme as follows.

Definition 4.4.1.

$$\mathrm{R}_{\mathrm{PVW}} = \Big\{((\mathbf{A}, \mathbf{P}), (\mathbf{u}, \mathbf{c}), \mathbf{e}, \mathbf{v}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{\ell \times m} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^\ell \times \{0, 1\}^m \times [0, k]^\ell : \mathbf{u} = \mathbf{A} \cdot \mathbf{e} \bmod q \wedge \mathbf{c} = \mathbf{P} \cdot \mathbf{e} + \lfloor \frac{q}{k} \rceil \cdot \mathbf{v} \bmod q \Big\}.$$

We first investigate how to express $\mathrm{R}_{\mathrm{PVW}}$ in a "more compact" form. We construct the following matrices and vectors:

- Matrix $\mathbf{G} = \left[\frac{\mathbf{A}}{\mathbf{P}}\right] \in \mathbb{Z}_q^{(n+\ell) \times m}$.

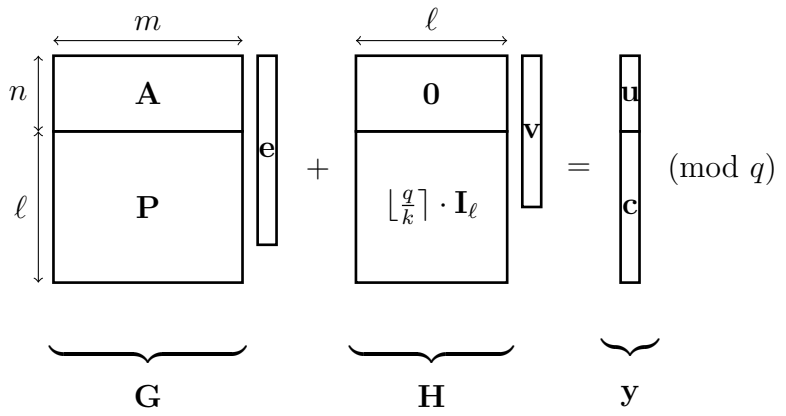

Fig. 4.2: Plaintext relation in the PVW encryption scheme.

- Matrix $\mathbf{H} = \left[\frac{0^{n\times\ell}}{\lfloor\frac{q}{k}\rceil\cdot\mathbf{I}_\ell}\right] \in \{0, \lfloor\frac{q}{k}\rceil\}^{(n+\ell)\times\ell}$, where $0^{n\times\ell}$ denotes the zero-matrix with n rows and ℓ columns, and $\mathbf{I}_\ell$ denotes the identity matrix of order ℓ.

- Vector $\mathbf{y} = (\mathbf{u}\|\mathbf{c}) \in \mathbb{Z}_q^{n+\ell}$.

We now observe that, the relation $\mathrm{R}_{\mathrm{PVW}}$ can be expressed as:

$$\mathrm{R}_{\mathrm{PVW}} = \Big\{((\mathbf{G},\mathbf{y}),\mathbf{e},\mathbf{v}) \in \mathbb{Z}_q^{(n+\ell)\times m}\times\mathbb{Z}_q^{n+\ell}\times\{0,1\}^m\times[0,k]^\ell : \mathbf{G}\cdot\mathbf{e}+\mathbf{H}\cdot\mathbf{v}=\mathbf{y} \bmod q\Big\}.$$

We illustrate this observation in Figure 4.2. To prove in ZK the possession of a witness $(\mathbf{e},\mathbf{v})$ for the relation $\mathrm{R}_{\mathrm{PVW}}$, we use the following Decomposition-Extension technique:

1. Append m coordinates to vector $\mathbf{e}$ to obtain a vector $\mathbf{e}^* \in \{0,1\}^{2m}$, such that $\mathsf{wt}(\mathbf{e}^*) = m$. In other words, $\mathbf{e}^* \in \mathsf{B}_{2m}$.

   To make the dimensions compatible, append m zero-columns to matrix $\mathbf{G}$ to obtain matrix $\mathbf{G}^* \in \mathbb{Z}_q^{(n+\ell)\times 2m}$.

2. Let $p = \lfloor\log k\rfloor + 1$, and define the sequence of integers $k_1,\ldots,k_p$ as in

SternExt proof system, namely:

$$k_1 = \lceil k/2 \rceil,\ k_2 = \lceil (k - k_1)/2 \rceil,\ k_3 = \lceil (k - k_1 - k_2)/2 \rceil, \ldots, k_p = 1.$$

3. Decompose vector $\mathbf{v}$ to p vectors $\mathbf{v}_1, \ldots, \mathbf{v}_p \in \{0,1\}^\ell$ with respect to the sequence $k_1, \ldots, k_p$, namely, we have $\mathbf{v} = \sum_{j=1}^{p} k_j \cdot \mathbf{v}_j$. Next, append ℓ coordinates to each of the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_p$ to obtain $\mathbf{v}_1^*, \ldots, \mathbf{v}_p^* \in \mathsf{B}_{2\ell}$.

   To make the dimensions compatible, append ℓ zero-columns to matrix $\mathbf{H}$ to obtain matrix $\mathbf{H}^* \in \{0, \lfloor \frac{q}{k} \rceil\}^{(n+\ell)\times 2\ell}$.

Given the above Decomposition-Extension technique, in the protocol, the prover convinces the verifier in ZK that it knows $\mathbf{e}^* \in \mathsf{B}_{2m}$, and $\mathbf{v}_1^*, \ldots, \mathbf{v}_p^* \in \mathsf{B}_{2\ell}$ satisfying:

$$\mathbf{G}^* \cdot \mathbf{e}^* + \mathbf{H}^* \cdot (\sum_{j=1}^{p} k_j \cdot \mathbf{v}_j^*) = \mathbf{y} \bmod q.$$

This can be done by adapting the **SternExt** proof system. The protocol follows the same principle as before. We note that in this case the constraints of $\mathbf{e}^*$ are verified using a random permutation of $2m$ elements, while the constraints of the $\mathbf{v}_j^*$'s are verified using random permutations of 2ℓ elements.

In Figure 4.3, we give an illustration of the Decomposition-Extension technique described above.

In summary, we can obtain a statistical ZKPoPK for the PVW encryption scheme, such that each round of the protocol has communication cost $\widetilde{\mathcal{O}}(n) + \widetilde{\mathcal{O}}(\ell)$, and knowledge error 2/3. If a cheating prover succeeds in convincing the verifier, then one can use the knowledge extractor to extract $\mathbf{e}' \in \mathsf{B}_{2m}$, and $\mathbf{v}_1', \ldots, \mathbf{v}_p' \in \mathsf{B}_{2\ell}$

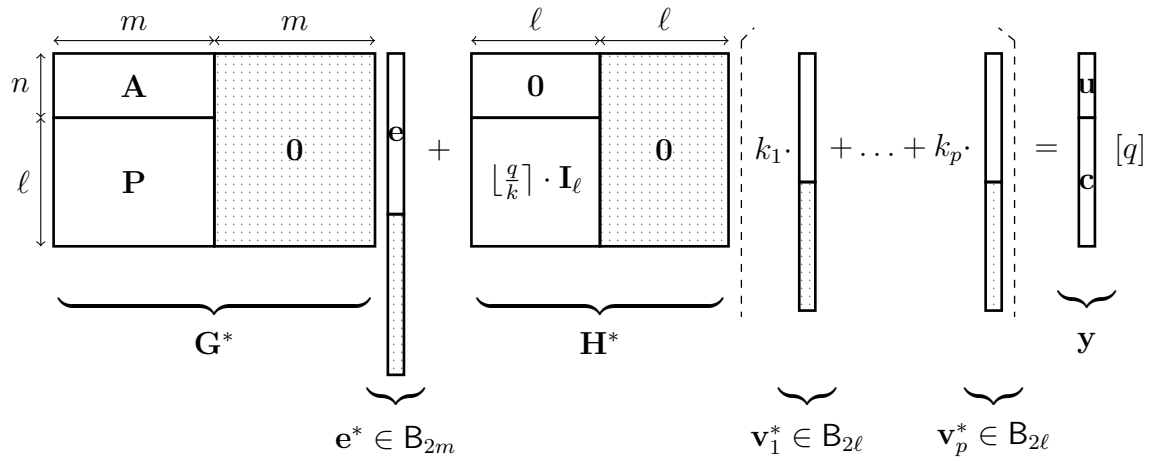

Fig. 4.3: The Decomposition-Extension technique for $\mathrm{R}_{\mathrm{PVW}}$.

satisfying:

$$\mathbf{G}^* \cdot \mathbf{e}' + \mathbf{H}^* \cdot (\sum_{j=1}^{p} k_j \cdot \mathbf{v}'_j) = \mathbf{y} \bmod q.$$

By dropping the last m coordinates from vector $\mathbf{e}'$, we obtain $\bar{\mathbf{e}} \in \{0,1\}^m$ satisfying $\mathbf{A} \cdot \bar{\mathbf{e}} = \mathbf{u} \bmod q$. Since $\mathbf{A}$ is chosen uniformly at random in $\mathbb{Z}_q^{n \times m}$ and $\mathbf{u}$ is statistically close to uniform over $\mathbb{Z}_q^n$ (by the Left-over Hash Lemma, see [128, Section 5]), vector $\bar{\mathbf{e}}$ is a valid solution to the $\mathsf{ISIS}^{\infty}(n, m, q, 1)$ instance $(\mathbf{A}, \mathbf{u})$. In other words, the soundness of the proof system can be based on the average-case hardness of the $\mathsf{ISIS}^{\infty}(n, m, q, 1)$ problem, and on the worst-case hardness of $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n)}$ (via the reduction of Theorem 2.2.14).

4.4.2 The Gentry-Halevi-Vaikuntanathan Encryption Scheme

Gentry, Halevi and Vaikuntanathan [61] constructed an LWE-based homomorphic encryption scheme that supports polynomially many additions and one multiplication of ciphertexts. Below we will refer to this scheme as the GHV encryption scheme. Let us remind some facts on the GHV encryption scheme which is a essentially a generalized variant of the Dual-Regev encryption scheme.

- Parameters: Security parameter n, an odd prime $q = \mathsf{poly}(\mathsf{n})$, integers $m = \mathcal{O}(n \log q)$, and $\beta \geq \sqrt{n} \cdot \omega(\log n)$. Also let χ be a β-bounded distribution.
- The plaintext space is $\{0,1\}^{m \times m}$.
- The public key is a matrix $\mathbf{A}$ statistically close to uniform over $\mathbb{Z}_q^{n \times m}$.
- To encrypt $\mathbf{B} \in \{0,1\}^{m \times m}$, pick $\mathbf{S} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$, sample $\mathbf{X} \hookleftarrow \chi^{m \times m}$, and output the ciphertext $\mathbf{C} \in \mathbb{Z}_q^{m \times m}$, where

$$\mathbf{C} = \mathbf{A}^T \cdot \mathbf{S} + 2 \cdot \mathbf{X} + \mathbf{B} \bmod q.$$

Let n, m, q, β be the parameters of the scheme. We define the plaintext relation for the GHV encryption scheme as follows.

Definition 4.4.2.

$$\mathrm{R}_{\mathrm{GHV}} = \Big\{((\mathbf{A}, \mathbf{C}), \mathbf{S}, \mathbf{X}, \mathbf{B}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{m \times m} \times \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^{m \times m} \times \{0,1\}^{m \times m} : \\ (\|\mathbf{X}\|_\infty \leq \beta) \wedge (\mathbf{A}^T \cdot \mathbf{S} + 2 \cdot \mathbf{X} + \mathbf{B} = \mathbf{C} \bmod q)\Big\},$$

where $\|\mathbf{X}\|_\infty$ denotes the elementwise infinity norm of matrix $\mathbf{X} \in \mathbb{Z}^{m \times m}$, defined as $\|\mathbf{X}\|_\infty = \max_{i,j \in [m]}\{|\mathbf{X}_{i,j}|\}$.

We observe that $\mathrm{R}_{\mathrm{GHV}}$ can be considered as the conjunction of m "Dual-Regev-type" relations. Therefore, to obtain a ZKPoK for $\mathrm{R}_{\mathrm{GHV}}$, we can follow the same approach as for the ZKPoK for $\mathbf{R}_{\mathrm{Dual}}$ in Section 4.3. Specifically, we use the following Decomposition-Extension technique:

1. Let $p = \lfloor \log \beta \rfloor + 1$ and define the sequence $\beta_1, \ldots, \beta_p$ as in the **SternExt** proof system.

2. Decompose matrix $\mathbf{X}$ into p matrices $\widetilde{\mathbf{X}}_1, \ldots, \widetilde{\mathbf{X}}_p \in \{-1, 0, 1\}^{m \times m}$ with respect to the sequence $\beta_1, \ldots, \beta_p$, namely, we have $\mathbf{X} = \sum_{j=1}^{p} \beta_j \cdot \widetilde{\mathbf{X}}_j$.

3. For each $j \in [p]$, append $2m$ *rows* to matrix $\widetilde{\mathbf{X}}_j$ to obtain matrix $\mathbf{X}_j^* \in \mathbb{Z}^{3m \times m}$ such that all m *columns* of $\mathbf{X}_j^*$ are elements of the set B_{3m}.

4. To make the dimensions compatible, we construct the matrix $\mathbf{G} = \left[2 \cdot \mathbf{I}_m \,\middle|\, 0^{m \times 2m}\right] \in \{0, 2\}^{m \times 3m}$, where $\mathbf{I}_m$ denotes the identity matrix of order m, and $0^{m \times 2m}$ denotes the zero-matrix with m rows and $2m$ columns.

5. Append m *rows* to matrix $\mathbf{B}$ to obtain matrix $\mathbf{B}^* \in \{0, 1\}^{2m \times m}$ such that all m *columns* of $\mathbf{B}^*$ are elements of the set B_{2m}.

6. To make the dimensions compatible, we construct the matrix $\mathbf{H} = \left[\mathbf{I}_m \,\middle|\, 0^{m \times m}\right] \in \{0, 1\}^{m \times 2m}$, obtained by appending m zero-columns to $\mathbf{I}_m$.

Given the above Decomposition-Extension technique, to prove in ZK the possesion of a valid witness for the relation $\mathrm{R}_{\mathrm{GHV}}$, the prover convinces the verifier in ZK that it knows matrix $\mathbf{S} \in \mathbb{Z}_q^{n \times m}$ and matrices $\mathbf{X}_1^*, \ldots, \mathbf{X}_p^*$ and $\mathbf{B}^*$ such that:

1. For each $j \in [p]$, all m columns of $\mathbf{X}_j^*$ are elements of B_{3m}. For convenience, we denote this event by $\mathbf{X}_j^* \in (\mathsf{B}_{3m})^m$.

2. All m columns of $\mathbf{B}^*$ are elements of B_{2m}. For convenience, we denote this event by $\mathbf{B}^* \in (\mathsf{B}_{2m})^m$.

3. The following equation holds true:

$$\mathbf{A}^T \cdot \mathbf{S} + \mathbf{G} \cdot \left(\sum_{j=1}^{p} \beta_j \cdot \mathbf{X}_j^*\right) + \mathbf{H} \cdot \mathbf{B}^* = \mathbf{C} \bmod q.$$

In Figure 4.4, we give an illustration of our Decomposition-Extension technique.

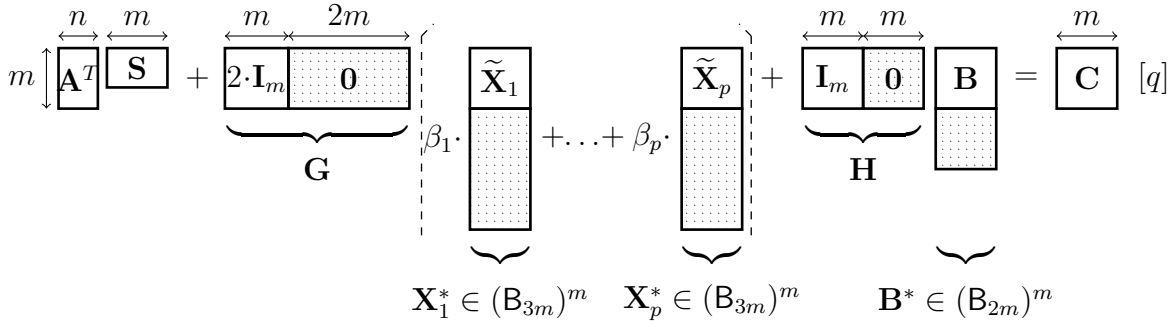

Fig. 4.4: The Decomposition-Extension technique for $\mathrm{R}_{\mathrm{GHV}}$.

We now describe the interactive protocol. For $k \in \{2m, 3m\}$, we denote by $[\mathsf{S}_k]^m$ the set of all permutations Π such that when applying to $\mathbf{D} = \big[\mathbf{d}_1 | \ldots | \mathbf{d}_m\big] \in \mathbb{Z}^{k \times m}$, it returns

$$\Pi(\mathbf{D}) = \big[\pi_1(\mathbf{d}_1) | \ldots | \pi_m(\mathbf{d}_m)\big],$$

where $\pi_1, \ldots, \pi_m$ are certain permutations of k elements.

The common input of the protocol is $(\mathbf{A}, \mathbf{C})$. In addition, both parties agree on matrices $\mathbf{G}, \mathbf{H}$ defined above. The prover's auxiliary input is a valid witness $(\mathbf{S}, \mathbf{X}, \mathbf{B})$ of the relation $\mathrm{R}_{\mathrm{GHV}}$. Prior to the interaction, the prover applies the above Decomposition-Extension technique to obtain $\mathbf{X}_1^*, \ldots, \mathbf{X}_p^*$ and $\mathbf{B}^*$. Let COM be a statistically hiding and computationally binding string commitment scheme. The protocol is as follows:

Commitment: Prover picks uniformly random matrices $\mathbf{W} \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$; $\mathbf{R}_1, \ldots, \mathbf{R}_p \xleftarrow{\$} \mathbb{Z}_q^{3m \times m}$; $\mathbf{Y} \xleftarrow{\$} \mathbb{Z}_q^{2m \times m}$; permutations $\Pi_1, \ldots, \Pi_p \xleftarrow{\$} [\mathsf{S}_{3m}]^m$; $\Gamma \xleftarrow{\$} [\mathsf{S}_{2m}]^m$; and sends the commitment $\mathrm{CMT} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\{\Pi_j\}_{j=1}^p, \Gamma, \mathbf{A}^T \cdot \mathbf{W} + \mathbf{G} \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{R}_j) + \mathbf{H} \cdot \mathbf{Y} \bmod q\big) \\ \mathbf{c}_2 = \mathrm{COM}\big(\Pi_1(\mathbf{R}_1), \ldots, \Pi_p(\mathbf{R}_p), \Gamma(\mathbf{Y})\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\Pi_1(\mathbf{X}_1^* + \mathbf{R}_1), \ldots, \Pi_p(\mathbf{X}_p^* + \mathbf{R}_p), \Gamma(\mathbf{B}^* + \mathbf{Y})\big). \end{cases}$$

Challenge: Receiving CMT, the verifier sends $Ch \xleftarrow{\$} \{1, 2, 3\}$.

Response: Receiving Ch, the prover proceeds as follows:

- If $Ch = 1$, then reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. For each $j \in [p]$, let $\mathbf{U}_j = \Pi_j(\mathbf{X}_j^*)$ and $\mathbf{V}_j = \Pi_j(\mathbf{R}_j)$. Let $\mathbf{Z}_1 = \Gamma(\mathbf{B}^*)$ and $\mathbf{Y}_1 = \Gamma(\mathbf{Y})$. Send

$$\text{RSP} = \big(\mathbf{U}_1, \ldots, \mathbf{U}_p, \mathbf{V}_1, \ldots, \mathbf{V}_p, \mathbf{Z}_1, \mathbf{Y}_1\big). \tag{4.1}$$

- If $Ch = 2$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. For each $j \in [p]$, let $\mathbf{F}_j = \mathbf{X}_j^* + \mathbf{R}_j$. Let $\mathbf{W_2} = \mathbf{S} + \mathbf{W}$ and $\mathbf{Y}_2 = \mathbf{B}^* + \mathbf{Y}$. Send

$$\text{RSP} = \big(\Pi_1, \ldots, \Pi_p, \Gamma, \mathbf{F}_1, \ldots, \mathbf{F}_p, \mathbf{Y}_2, \mathbf{W}_2\big). \tag{4.2}$$

- If $Ch = 3$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. For each $j \in [p]$, let $\mathbf{E}_j = \mathbf{R}_j$. Let $\mathbf{W}_3 = \mathbf{W}$ and $\mathbf{Y}_3 = \mathbf{Y}$. Send

$$\text{RSP} = \big(\Pi_1, \ldots, \Pi_p, \Gamma, \mathbf{E}_1, \ldots, \mathbf{E}_p, \mathbf{Y}_3, \mathbf{W}_3\big). \tag{4.3}$$

Verification: Receiving RSP, the verifier proceeds as follows.

- If $Ch = 1$: Parse RSP as in (4.1). Check that $\mathbf{Z}_1 \in (\mathsf{B}_{2m})^m$; and for all $j \in [p]$: $\mathbf{U}_j \in (\mathsf{B}_{3m})^m$; and

$$\mathbf{c}_2 = \text{COM}(\mathbf{V}_1, \ldots, \mathbf{V}_p, \mathbf{Y}_1);\ \ \mathbf{c}_3 = \text{COM}(\mathbf{U}_1 + \mathbf{V}_1, \ldots, \mathbf{U}_p + \mathbf{V}_p, \mathbf{Y}_1 + \mathbf{Z}_1).$$

- If $Ch = 2$: Parse RSP as in (4.2). Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\{\Pi_j\}_{j=1}^p, \Gamma, \mathbf{A}^T \cdot \mathbf{W}_2 + \mathbf{G} \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{F}_j) + \mathbf{H} \cdot \mathbf{Y}_2 - \mathbf{C} \bmod q\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\Pi_1(\mathbf{F}_1), \ldots, \Pi_p(\mathbf{F}_p), \Gamma(\mathbf{Y}_2)\big). \end{cases}$$

- If $Ch = 3$: Parse RSP as in (4.3). Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\{\Pi_j\}_{j=1}^p, \Gamma, \mathbf{A}^T \cdot \mathbf{W}_3 + \mathbf{G} \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{E}_j) + \mathbf{H} \cdot \mathbf{Y}_3 \bmod q\big) \\ \mathbf{c}_2 = \mathrm{COM}\big(\Pi_1(\mathbf{E}_1), \ldots, \Pi_p(\mathbf{E}_p), \Gamma(\mathbf{Y}_3)\big). \end{cases}$$

In each case, the verifier outputs 1 if and only if all the conditions hold. Otherwise, it outputs 0.

It can be checked that the above protocol has perfect completeness. For parameters $q = \mathsf{poly}(n)$, $m = \mathcal{O}(n \log q)$, and $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$, the protocol has asymptotic communication cost $\widetilde{\mathcal{O}}(n^2)$.

Furthermore, following the same approach as in the proof system for the relation $\mathrm{R}_{\mathrm{Dual}}$ (see Section 4.3), one can prove that the above protocol is statistically zero-knowledge (if COM is statistically hiding), and a proof of knowledge (if COM is computationally binding) for the relation $\mathrm{R}_{\mathrm{GHV}}$. The soundess of the proof system relies directly on the security of the GHV encryption scheme. We skip the details here, as they are very similar to those in Section 4.3.

5. LATTICE-BASED ID-BASED IDENTIFICATION SCHEMES

5.1 Introduction

Identification is a process through which one party ascertains the identity of another person or entity. Informally, an identification scheme is an interactive protocol that allows a prover holding a secret key to identify itself to a verifier holding the corresponding public key. Such schemes can find many applications in practice, most notably in access control systems, where the access privilege is determined by the user's identity. Fiat and Shamir [58] demonstrated that zero-knowledge proofs could lead to efficient identification schemes, paving the road for numerous subsequent works ([55, 76, 133, 137],...). In the scope of lattice-based cryptography, many identifications schemes have been proposed in the last decade ([115, 85, 94, 43],...).

As discussed above, identification schemes allow a person to prove that "he is himself" in order to get access to a system that he is supposed to have legitimate access right. However, in various real-life scenarios, some person may want to get access to certain system while preserving his *anonymity*. In such situations, the person may need a scheme that enables him to anonymously prove that "he belongs to a set of people having legitimate access rights". A cryptographic primitive that supports this appealing feature was introduced by Dodis et al. [54] under the name "Ad-hoc anonymous identification". We, however, will refer to this primitive as "ring identification" to emphasize the relation to its non-interactive counterpart, known as

ring signature. A ring identification scheme allows a user to form a ring of users in an ad-hoc fashion, and then anonymously prove its membership in that ring. Such schemes may find applications in anonymous access control systems in the multi-user setting, and have received noticeable attention from the community ([116, 85, 103, 26],...) since their introduction.

In traditional public-key cryptography, the public keys must be certified, stored and managed by the Public-Key Infrastructure. In 1984, Shamir [134] introduced identity-based (ID-based) cryptography, which significantly reduces the cost of public keys management: In this paradigm, the public key of a user is a publicly known string representing its identity, e.g., an email address or a physical IP address.

In this chapter, our main interests are ID-based identification and ID-based ring identification schemes from lattice assumptions. Prior to our work, two lattice-based ID-based identification schemes have been published:

1. The scheme of Stehlé et al.'s [136], which is based on the hardness of $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^2)}$ in general lattices.

2. The scheme of Rückert [130], which is based on the assumed hardness of $\mathsf{SVP}^\infty_{\widetilde{\widetilde{\mathcal{O}}}(n^{3.5})}$ in ideal lattices.

In Section 5.2, we propose an ID-based identification scheme based on the hardness of $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^{1.5})}$ (in general lattices). Thus, our scheme relies on *weaker* hardness assumption than the previous constructions.

Moreover, in Section 5.3, we introduce an ID-based ring identification scheme based on the hardness of $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^{1.5})}$. Up to the best of our knowledge, this is the first instantiation of ID-based ring identification from lattice assumptions.

5.2 Identity-based Identification

We first recall the definition and security notions of ID-based identification schemes in Section 5.2.1, then we present our construction in Section 5.2.2.

5.2.1 Definition and Security Notions

The formal definition and security notions of ID-based identification schemes were independently given by Bellare et al. [20] and Kurosawa and Heng [87] in 2004.

Definition 5.2.1 (IBI, [20, 87])**.** An identity-based identification (IBI) scheme is a tuple of four probabilistic polynomial-time algorithms $(\mathsf{MKg}, \mathsf{UKg}, P, V)$:

- The master-key generation algorithm $\mathsf{MKg}(1^n)$: On input 1^n, where n is the security parameter, output a master public and master secret key pair $(\mathsf{mpk}, \mathsf{msk})$.
- The user-key generation algorithm $\mathsf{UKg}(\mathsf{msk}, id)$: On input msk and a user identity $id \in \{0,1\}^*$, output a secret key sk_{id} for this user.
- $\langle P, V \rangle$ is an interactive protocol. The prover P takes $(\mathsf{mpk}, id, \mathsf{sk}_{id})$ as input, the verifier V takes (mpk, id) as input. At the end of the protocol, V outputs the decision 1 (accept) or 0 (reject).

In the random oracle model, algorithms UKg, P, V are given oracle access to a function H whose range may depend on mpk.

Completeness: The completeness requirement for an IBI scheme is as follows: For any $n \in \mathbb{N}$, any mpk generated by $\mathsf{MKg}(1^n)$, and sk_{id} extracted by $\mathsf{UKg}(\mathsf{msk}, id)$, the output of V after interacting with P is always 1.

Security Notions: There are three security notions of IBI schemes: security against impersonation under passive attacks, active attacks and concurrent attacks.

The capabilities of the adversary, whose goal is to impersonate a prover of its choice, depend on the attack model. In passive attacks, the adversary can obtain interaction transcripts between the real prover and a verifier. In active or concurrent attacks, the adversary can directly interact with the prover by playing the role of a cheating verifier. The different between an active attack and a concurrent attack is that: In the former case, the adversary can only have one active session at a time, while in the latter case, it can have concurrent (or parallel) active sessions with different clones of the prover.

In each of the mentioned above attack model, the IBI scheme is called secure if any polynomial-time adversary cannot impersonate the real prover with non-negligible probability (over the randomness of all algorithms). It can be seen that the strongest security notion for IBI schemes is security against impersonation under concurrent attacks [20, 87].

5.2.2 Improved Construction from Lattices

A common strategy in constructing IBI schemes consists in combining a signature scheme and a PoK in the following way: The trusted authority generates $(\mathsf{mpk}, \mathsf{msk})$ as a verification key - signing key pair of a signature scheme; Whenever a user id queries for his secret key, the authority returns sk_{id} as a signature on id; For identification, the user plays the role of the prover, and runs a PoK to prove the possession of sk_{id}. If the signature scheme is strongly secure against existential forgery under chosen message attacks, and the PoK is at least witness indistinguishable, then the resulting IBI scheme is secure against impersonation under concurrent attacks [20]. This strategy is widely used for lattice-based IBI schemes.

- Stehlé et al. [136] combined the GPV signature ([62], see also Scheme 2.3.6), and the Micciancio-Vadhan PoK ([115], see also Scheme 3.2.1) to obtain an IBI

scheme based on the hardness of the $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^2)}$ problem (in general lattices).

- Rückert [130] combined the Bonsai tree signature ([42], see also Scheme 2.3.9) and Lyubashevsky's **PoK** ([94], see also Section 3.2.2) for ideal lattices to produce an IBI scheme based on the hardness of $\mathsf{SVP}^\infty_{\widetilde{\mathcal{O}}(n^{3.5})}$ (in ideal lattices).

Following the same approach, the **SternExt** proof system (see Section 3.3) allows us to achieve better in terms of hardness assumption. Since **SternExt** is zero-knowledge, it has the witness indistinguishability property. As WI is preserved under parallel composition [57], we can repeat the protocol $\omega(\log n)$ times in parallel to obtain a WIPoK with negligible soundness error. Combining with the GPV signature scheme, we obtain a secure IBI scheme in the random oracle model with hardness assumption $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^{1.5})}$.

Our lattice-based IBI scheme $\mathcal{LIBI}$ is defined as follows.

Scheme 5.2.2 ($\mathcal{LIBI}$)**.** Let $q = \mathsf{poly}(n)$, $m \geq 2n \log q$, $\sigma = \omega(\sqrt{n \log q \log n})$. Let $\beta = \lceil \sigma \cdot t \rceil$, where $t = t(m) \geq \omega(\sqrt{\log m})$ is some fixed function. Let $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_q^n$ be a random oracle.

By Lemma 2.2.10 (item (4)), for parameters σ, β as above, the discrete Gaussian distribution $D_{\mathbb{Z},\sigma}$ is a β-bounded distribution. In the following, we will employ algorithms $\mathsf{GenTrap}$ from Theorem 2.3.3 and $\mathsf{SampleD}$ from Theorem 2.3.4.

- $\mathrm{MKg}(1^n)$: Let $(\mathbf{A}, \mathbf{S}) \leftarrow \mathsf{GenTrap}(1^n, 1^m, q)$. Output $\mathsf{mpk} = \mathbf{A}$ and $\mathsf{msk} = \mathbf{S}$.

- $\mathrm{UKg}(\mathsf{msk}, id)$: For $id \in \{0,1\}^*$, let $\mathbf{x} = \mathsf{SampleD}(\mathbf{S}, \mathbf{A}, \mathsf{H}(id), \sigma)$. If $\|\mathbf{x}\|_\infty > \beta$ (which happens with negligible probability, since the $D_{\mathbb{Z},\sigma}$ is a β-bounded distribution), then restart. Otherwise, output $\mathsf{sk}_{id} = \mathbf{x}$.

  We note that sk_{id} is the GPV signature for the message id, and is a solution to the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ instance $(\mathbf{A}, \mathsf{H}(id))$.

- $\langle P, V \rangle$: The common input is the pair $(\mathbf{A}, \mathsf{H}(id))$. The auxiliary input of P is sk_{id}. Then P and V play the roles of the prover and the verifier in the **SternExt** protocol for the $\mathsf{ISIS}^{\infty}(n, m, q, \beta)$ problem (see Section 3.3). To make the soundness error negligibly small, the protocol is repeated $\ell = \omega(\log n)$ times in parallel.

The completeness of the scheme $\mathcal{LIBI}$ follows from the perfect completeness of **SternExt**. Since the GPV signature scheme is strongly secure against existential forgery under adaptive chosen message attacks [62], and the **SternExt** protocol is a WIPoK, the scheme $\mathcal{LIBI}$ is secure against impersonation under concurrent attacks. The scheme relies on the assumed hardness of the $\mathsf{ISIS}^{\infty}(n, m, q, \beta)$ problem, where $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$. It follows from Theorem 2.2.14 that solving the $\mathsf{ISIS}^{\infty}(n, m, q, \beta)$ problem is at least as hard as solving SIVP^2_{γ} with $\gamma = \beta \cdot \widetilde{\mathcal{O}}(n) = \widetilde{\mathcal{O}}(n^{1.5})$. We summarize the above discussion in the following theorem.

Theorem 5.2.3. *The scheme $\mathcal{LIBI}$ is concurrently secure in the random oracle model if $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^{1.5})}$ is hard in the worst-case.*

Similarly, combining the **SternExt** proof system with lattice-based signature schemes that are secure in the standard model (e.g., [42, 29, 111]) we can obtain secure lattice-based IBI schemes in the standard model, with weaker hardness assumption than in the contemporary schemes.

5.3 Identity-based Ring Identification

We first recall the definition and security notions of ID-based ring identification schemes in Section 5.3.1. In Section 5.3.2 we describe the underlying proof system for our lattice-based ID-based ring identification scheme. The main scheme is presented in Section 5.3.3.

5.3.1 Definition and Security Notions

In [54], Dodis et al. gave formal definition and security notions of ring identification schemes. Subsequently, these concepts were adapted into the identity-based setting by Nguyen [116]. The presentation in this section will follow both [54] and [116].

Definition 5.3.1 (IBRI)**.** An identity-based ring identification (IBRI) scheme is a tuple of six polynomial-time algorithms $(\mathsf{MKg}, \mathsf{UKg}, \mathsf{RPKg}, \mathsf{RSKg}, P, V)$:

- The master-key generation algorithm $\mathsf{MKg}(1^n)$: On input 1^n, where n is the security parameter, output a master public and master secret key pair $(\mathsf{mpk}, \mathsf{msk})$.
- The user-key generation algorithm $\mathsf{UKg}(\mathsf{msk}, id)$: On input msk and a user identity $id \in \{0,1\}^*$, output a secret key sk_{id} for this user.
- The ring public key generation algorithm $\mathsf{RPKg}(\mathsf{mpk}, R)$: On input mpk and a ring of identities $R = (id_1, \ldots, id_N)$, output the ring public key rpk.
- The ring secret key generation algorithm $\mathsf{RSKg}(\mathsf{mpk}, R, \mathsf{sk}_{id})$: On input mpk, a ring of identities $R = (id_1, \ldots, id_N)$, and the secret key sk_{id} of an identity $id \in R$, output a ring secret key rsk.
- $\langle P(\mathsf{rpk}, \mathsf{rsk}), V(\mathsf{rpk})\rangle$ is an interactive protocol. A prover P takes $(\mathsf{rpk}, \mathsf{rsk})$ as input. A verifier V takes rpk as input. At the end of the interaction, V outputs a decision 1 (Accept) or 0 (Reject).

Completeness. The completeness requirement for an IBRI scheme is as follows: For any $n \in \mathbb{N}$, any $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{MKg}(1^n)$, any ring of identities R containing id, any ring keys $\mathsf{rpk} \leftarrow \mathsf{RPKg}(\mathsf{mpk}, R)$, $\mathsf{rsk} \leftarrow \mathsf{RSKg}(\mathsf{mpk}, R, \mathsf{UKg}(\mathsf{msk}, id))$, the decision of $V(\mathsf{rpk})$ after interacting with $P(\mathsf{rpk}, \mathsf{rsk})$ is always 1.

Security Notions. There are two security notions for an IBRI scheme: security against impersonation and anonymity [54, 116].

Security against impersonation: Informally, this security notion requires that: if someone can provide a valid ring identification transcript, then it must belong to the given ring. This requirement is modeled by an experiment $\mathbf{Exp}^{\mathsf{imp}}_{\mathcal{IBRI},\mathcal{A}}(n)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$. In a chosen ring attack, the adversary $\mathcal{A}$ can ask any prover to prove the membership in an arbitrary ring containing the him. Similar to concurrent attacks against IBI schemes, we allow the adversary to concurrently interact with clones of any provers, but prohibit it from interacting with anyone in the target ring.

The experiment is as follows. The challenger $\mathcal{C}$ generates $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{MKg}(1^n)$, and sends mpk to the adversary. During the experiment, $\mathcal{A}$ can adaptively and concurrently make queries to a transcript oracle, which takes as input an identity id and a ring $R = (id_1, \ldots, id_N)$ containing id, and returns a valid transcript of the interaction $\langle P(\mathsf{rpk}, \mathsf{rsk}), V(\mathsf{rpk})\rangle$, where $\mathsf{rpk} \leftarrow \mathsf{RKg}(\mathsf{mpk}, R)$, and $\mathsf{rsk} \leftarrow \mathsf{RSKg}(\mathsf{mpk}, R, \mathsf{UKg}(\mathsf{msk}, id))$. At some point, $\mathcal{A}$ returns a target ring of identities $R^* = (id^*_1, \ldots, id^*_{N^*})$ and plays the role of the prover in the interactive protocol $\langle P, V\rangle$, where the common input is $\mathsf{rpk} \leftarrow \mathsf{RKg}(\mathsf{mpk}, R^*)$. The challenger $\mathcal{C}$, playing the role of $V(\mathsf{rpk})$, returns 1 if the provided transcript is valid. Otherwise, it returns 0.

Definition 5.3.2. Let $\mathcal{IBRI} = (\mathsf{MKg}, \mathsf{UKg}, \mathsf{RPKg}, \mathsf{RSKg}, P, V)$ be an IBRI scheme. Define the advantage of $\mathcal{A}$ in the experiment $\mathbf{Exp}^{\mathsf{imp}}_{\mathcal{IBRI},\mathcal{A}}(n)$ as

$$\mathsf{Adv}^{\mathsf{imp}}_{\mathcal{IBRI},\mathcal{A}}(n) = \Pr\big[\mathbf{Exp}^{\mathsf{imp}}_{\mathcal{IBRI},\mathcal{A}}(n) = 1\big].$$

We say that $\mathcal{IBRI}$ is secure against impersonation if for any polynomial-time ad-

versary $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{imp}}_{\mathcal{IBRI},\mathcal{A}}(n) = \mathsf{negl}(n)$.

Anonymity against full key exposure: Informally, this security notion requires that: any adversary cannot distinguish two ring identification transcripts provided by two distinct identities in a ring, even if it knows the secret keys of all members in the ring. This requirement is modeled by the following experiment $\mathbf{Exp}^{\mathsf{anon}}_{\mathcal{IBRI},\mathcal{A}}(n)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

The challenger first generates $(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{MKg}(1^n)$, and sends mpk to the adversary. During the experiment, $\mathcal{A}$ can request the secret key sk_{id} of any identity id of his choice. At some point, $\mathcal{A}$ outputs two valid pairs $(id_0, \mathsf{sk}_{id_0})$, $(id_1, \mathsf{sk}_{id_1})$ and a ring R containing (id_0, id_1). The challenger then picks $b \xleftarrow{\$} \{0,1\}$, computes $\mathsf{rpk} \leftarrow \mathsf{RKg}(\mathsf{mpk}, R)$, and $\mathsf{rsk} \leftarrow \mathsf{RSKg}(\mathsf{mpk}, R, \mathsf{sk}_{id_b})$, and returns a transcript of $\langle P, V\rangle$ using witness rsk. At the end of the experiment $\mathcal{A}$ outputs $b' \in \{0,1\}$. The challenger outputs 1 if $b' = b$; otherwise, it outputs 0.

Definition 5.3.3. Let $\mathcal{IBRI} = (\mathsf{MKg}, \mathsf{UKg}, \mathsf{RPKg}, \mathsf{RSKg}, P, V)$ be an IBRI scheme. Define the advantage of $\mathcal{A}$ in the experiment $\mathbf{Exp}^{\mathsf{anon}}_{\mathcal{IBRI},\mathcal{A}}(n)$ as

$$\mathsf{Adv}^{\mathsf{imp}}_{\mathcal{IBRI},\mathcal{A}}(n) = \Big|\Pr\big[\mathbf{Exp}^{\mathsf{anon}}_{\mathcal{IBRI},\mathcal{A}}(n) = 1\big] - 1/2\Big|.$$

We say that $\mathcal{IBRI}$ is *unconditionally anonymous* if for any computationally unbounded adversary $\mathcal{A}$, we have $\mathsf{Adv}^{\mathsf{anon}}_{\mathcal{IBRI},\mathcal{A}}(n) = \mathsf{negl}(n)$.

Several lattice-based ring identification schemes have been proposed [85, 43, 26]. However, the identity-based setting was not considered in these works. In the following sections, we will present the first IBRI scheme from lattices, where the underlying protocol is an adaptation of the **SternExt** proof system to the context of IBRI.

5.3.2 The Underlying Proof System

In this section, we construct a ZKPoK that will be used as the building block in the lattice-based IBRI scheme in Section 5.3.3. Let n, N, m, q, β be positive integers, such that $\beta = \widetilde{\mathcal{O}}(\sqrt{n})$, $q = \mathsf{poly}(n) > \beta$, and $m \geq 2n \log q$. We consider the relation:

$$\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N) = \Big\{ \big((\mathbf{A}, \mathbf{B}), \mathbf{x}, \mathbf{e}\big) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times N} \times \mathbb{Z}^m \times \{0,1\}^N : (\|\mathbf{x}\|_\infty \leq \beta) \wedge (\mathsf{wt}(\mathbf{e}) = 1) \wedge (\mathbf{A} \cdot \mathbf{x} - \mathbf{B} \cdot \mathbf{e} = \mathbf{0} \bmod q) \Big\},$$

where $\mathsf{wt}(\mathbf{e})$ denotes the Hamming weight of $\mathbf{e}$.

Intuitively, for uniformly random $(\mathbf{A}, \mathbf{B})$, a pair $(\mathbf{x}, \mathbf{e})$ is a valid witness of the relation $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$ if there exists an index $i \in [N]$ (that is the index of the unique entry 1 in vector $\mathbf{e}$), such that $\mathbf{x}$ is a solution to the $\mathsf{ISIS}^\infty(n, m, q, \beta)$ instance $(\mathbf{A}, \mathbf{y}_i)$, where $\mathbf{y}_i$ is the i-th column of $\mathbf{B}$.

First, we note that one can possibly transform the ISIS instances $(\mathbf{A}, \mathbf{y}_i)$'s into GapCVP instances (as explained in Section 3.2.1), then express $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$ as the disjunction of N instances of GapCVP. At this point, one can combine the Micciancio-Vadhan proof system [115] for GapCVP together with the generic technique of Cramer et al. [49] to obtain a proof system for $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$. However, the resulting proof system is relatively inefficient (with communication cost $\widetilde{\mathcal{O}}(n^2) \cdot N$), and it inherits an extraction gap $g \geq \widetilde{\mathcal{O}}(\sqrt{n})$ from the Micciancio-Vadhan protocol.

In the following, we will present a more direct and efficient ZKPoK for the relation $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$, based on the **SternExt** proof system in Section 3.3. As for **SternExt**, the obtained proof system has no extraction gap, which leads to a weak hardness assumption for the resulting IBRI scheme in Section 5.3.3. Moreover, the communication cost of the protocol is only $\widetilde{\mathcal{O}}(n) + \widetilde{\mathcal{O}}(N)$. Our protocol is described

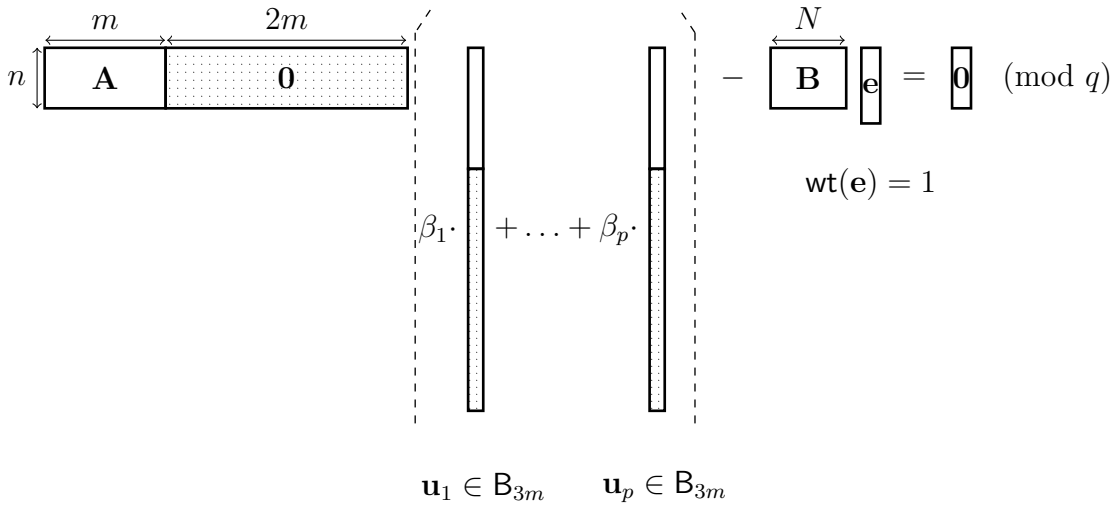

Fig. 5.1: The Decomposition-Extension technique for $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$.

as follows.

Let $p = \lfloor \log \beta \rfloor + 1$, and define the sequence $\beta_1, \ldots, \beta_p$ as in the **SternExt** proof system. Let COM be the statistically hiding and computationally binding commitment scheme from [85]. The common input of the protocol is $(\mathbf{A}, \mathbf{B})$. The prover's auxiliary input is $(\mathbf{x}, \mathbf{e})$. As for the **SternExt** protocol, we employ the Decomposition-Extension technique: Prior to the interaction, both prover and verifier form the extended matrix $\mathbf{A}^* \leftarrow \mathsf{MatrixExt}_{m,\beta}(\mathbf{A})$, and the prover additionally compute $\mathbf{u}_1, \ldots, \mathbf{u}_p \leftarrow \mathsf{VectorDE}_{m,\beta}(\mathbf{x})$. In the protocol, the prover's goal is to convince the verifier that it knows $\mathbf{u}_1, \ldots, \mathbf{u}_p \in \mathsf{B}_{3m}$ and $\mathbf{e} \in \{0,1\}^N$ such that $\mathsf{wt}(\mathbf{e}) = 1$, and

$$\mathbf{A}^* \cdot \big(\sum_{j=1}^{p} \beta_j \cdot \mathbf{u}_j \big) - \mathbf{B} \cdot \mathbf{e} = \mathbf{0} \bmod q.$$

In Figure 5.1, we give an illustration of the Decomposition-Extension technique for $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$.

The protocol follows the same principle as of **SternExt**. Here, to prove the knowledge of the additional witness $\mathbf{e}$ in zero-knowledge, we use the original "per-

mutation technique" of Stern for $\{0,1\}$ vector with fixed Hamming weight: we let the prover pick a random permutation τ of N elements, and let the verifier check whether $\tau(\mathbf{e}) \in \{0,1\}^N$ and $\mathsf{wt}(\tau(\mathbf{e})) = 1$.[1] The prover and the verifier interact as follows:

1. **Commitment.** Prover P samples vectors $\mathbf{r}_1, \ldots, \mathbf{r}_p \xleftarrow{\$} \mathbb{Z}_q^{3m}$ and $\mathbf{g} \xleftarrow{\$} \mathbb{Z}_q^N$; permutations $\pi_1, \ldots, \pi_p \xleftarrow{\$} \mathsf{S}_{3m}$, $\tau \xleftarrow{\$} \mathsf{S}_N$; and sends the commitment $\mathrm{CMT} := (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\pi_1, \ldots, \pi_p, \tau, \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}_j) - \mathbf{B} \cdot \mathbf{g} \bmod q\big) \\ \mathbf{c}_2 = \mathrm{COM}\big(\pi_1(\mathbf{r}_1), \ldots, \pi_p(\mathbf{r}_p), \tau(\mathbf{g})\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\pi_1(\mathbf{u}_1 + \mathbf{r}_1), \ldots, \pi_p(\mathbf{u}_p + \mathbf{r}_p), \tau(\mathbf{e} + \mathbf{g})\big) \end{cases}$$

2. **Challenge.** Receiving CMT, verifier sends a challenge $Ch \xleftarrow{\$} \{1,2,3\}$ to P.

3. **Response.** Prover replies as follows:

- If $Ch = 1$, then reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. For each j, let $\mathbf{v}_j = \pi_j(\mathbf{u}_j)$, and $\mathbf{w}_j = \pi_j(\mathbf{r}_j)$. Let $\mathbf{e}_1 = \tau(\mathbf{e})$ and $\mathbf{g}_1 = \tau(\mathbf{g})$.

  Send $\mathrm{RSP} := \big(\mathbf{v}_1, \ldots, \mathbf{v}_p, \mathbf{w}_1, \ldots, \mathbf{w}_p, \mathbf{e}_1, \mathbf{g}_1\big)$.

- If $Ch = 2$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. For each j, let $\phi_j = \pi_j$, and $\mathbf{z}_j = \mathbf{u}_j + \mathbf{r}_j$. Let $\tau_2 = \tau$ and $\mathbf{g_2} = \mathbf{e} + \mathbf{g}$.

  Send $\mathrm{RSP} := \big(\phi_1, \ldots, \phi_p, \mathbf{z}_1, \ldots, \mathbf{z}_p, \tau_2, \mathbf{g}_2\big)$.

- If $Ch = 3$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. For each j, let $\psi_j = \pi_j$, and $\mathbf{s}_j = \mathbf{r}_j$. Let $\tau_3 = \tau$ and $\mathbf{g}_3 = \mathbf{g}$.

  Send $\mathrm{RSP} := \big(\psi_1, \ldots, \psi_p, \mathbf{s}_1, \ldots, \mathbf{s}_p, \tau_3, \mathbf{g}_3\big)$.

[1] A random permutation of N elements was also used in the lattice-based ring identification schemes from [85, 26].

Verification. Receiving the response RSP, verifier V proceeds as follows:

- If $Ch = 1$: Check that $\mathbf{v}_j \in \mathsf{B}_{3m}$ for all $j \in [p]$; $\mathbf{e}_1 \in \{0,1\}^N$ and $\mathsf{wt}(\mathbf{e}_1) = 1$; and
$$\begin{cases} \mathbf{c}_2 = \mathrm{COM}\big(\mathbf{w}_1, \ldots, \mathbf{w}_p, \mathbf{g}_1\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\mathbf{v}_1 + \mathbf{w}_1, \ldots, \mathbf{v}_p + \mathbf{w}_p, \mathbf{e}_1 + \mathbf{g}_1\big). \end{cases}$$
- If $Ch = 2$: Check that
$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\phi_1, \ldots, \phi_p, \tau_2,\ \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j) - \mathbf{B} \cdot \mathbf{g}_2 \bmod q\big) \\ \mathbf{c}_3 = \mathrm{COM}\big(\phi_1(\mathbf{z}_1), \ldots, \phi_p(\mathbf{z}_p), \tau_2(\mathbf{g}_2)\big). \end{cases}$$
- If $Ch = 3$: Check that
$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\big(\psi_1, \ldots, \psi_p, \tau_3,\ \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{s}_j) - \mathbf{B} \cdot \mathbf{g}_3 \bmod q) \\ \mathbf{c}_2 = \mathrm{COM}\big(\psi_1(\mathbf{s}_1), \ldots, \psi_p(\mathbf{s}_p), \tau_3(\mathbf{g}_3)\big). \end{cases}$$

In each case, verifier V outputs the decision $d = 1$ (Accept) if and only if all the conditions hold. Otherwise, he outputs $d = 0$ (Reject).

Completeness. We observe that if prover P has a valid witness $(\mathbf{x}, \mathbf{e})$ for the relation $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$ and follows the protocol, then he always gets accepted by V. Therefore, the proof system has perfect completeness.

Communication cost. In comparison to the **SternExt** proof system, the present one introduces an extra vector in $\mathbb{Z}_q^N$ and a permutation of N elements. Hence, the overall communication cost is $\log \beta \cdot \widetilde{\mathcal{O}}(n \log q) + \log q \cdot \widetilde{\mathcal{O}}(N)$. If we let n and N be the primary parameters, then the asymptotic cost is $\widetilde{\mathcal{O}}(n) + \widetilde{\mathcal{O}}(N)$.

Statistical zero-knowledge. We will prove that the given proof system is statistically zero-knowledge by exhibiting a transcript simulator.

Lemma 5.3.4. *If* COM *is a statistically hiding string commitment scheme, then the proof system for* $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$ *is statistically zero-knowledge. In particular, there exists an efficient simulator* $\mathcal{S}$*, which has black-box access to a (possibly cheating) verifier* $\widehat{V}$*, such that on input* $(\mathbf{A}, \mathbf{B})$*, outputs with probability* $2/3$ *a successful transcript, the distribution of which is statistically close to that in the real interaction.*

Proof. The simulator $\mathcal{S}$ works with the same principle as the simulator in the proof of Theorem 3.3.2. It begins by selecting a random $\overline{Ch} \in \{1, 2, 3\}$ (a prediction of the challenge value that $\widehat{V}$ will *not* choose), and a random tape r' of $\widehat{V}$.

Case $\overline{Ch} = 1$: Using linear algebra, the simulator $\mathcal{S}$ computes $\mathbf{x}' \in \mathbb{Z}_q^m, \mathbf{e}' \in \mathbb{Z}_q^N$ such that $\mathbf{A} \cdot \mathbf{x}' - \mathbf{B} \cdot \mathbf{e}' = \mathbf{0} \bmod q$; and $\widetilde{\mathbf{u}}'_1, \ldots, \widetilde{\mathbf{u}}'_p \in \mathbb{Z}_q^m$ such that $\mathbf{x}' = \sum_{j=1}^{p} \beta_j \cdot \widetilde{\mathbf{u}}'_j \bmod q$. Now for each j, the simulator extends $\widetilde{\mathbf{u}}'_j$ to $\mathbf{u}'_j \in \mathbb{Z}_q^{3m}$ by appending $2m$ random coordinates. It then picks vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_p \xleftarrow{\$} \mathbb{Z}_q^{3m}, \mathbf{g}' \xleftarrow{\$} \mathbb{Z}_q^N$; permutations $\pi'_1, \ldots, \pi'_p \xleftarrow{\$} \mathsf{S}_{3m}, \tau' \xleftarrow{\$} \mathsf{S}_N$; and uniformly random strings $\rho'_1, \rho'_2, \rho'_3$. It sends the commitment $(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\widehat{V}$, where:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\big(\pi'_1, \ldots, \pi'_p, \tau',\ \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{r}'_j) - \mathbf{B} \cdot \mathbf{g}' \bmod q; \rho'_1\big) \\ \mathbf{c}'_2 = \mathrm{COM}\big(\pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{g}'); \rho'_2\big) \\ \mathbf{c}'_3 = \mathrm{COM}\big(\pi'_1(\mathbf{u}'_1 + \mathbf{r}'_1), \ldots, \pi'_p(\mathbf{u}'_p + \mathbf{r}'_p), \tau'(\mathbf{e}' + \mathbf{g}'); \rho'_3\big). \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ provides a transcript as follows:

- If $Ch = 1$: Output $\perp$ and halt.
- If $Ch = 2$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 2, (\pi'_1, \ldots, \pi'_p, \mathbf{u}'_1 + \mathbf{r}'_1, \ldots, \mathbf{u}'_p + \mathbf{r}'_p, \tau', \mathbf{e}' + \mathbf{g}'); \rho'_1, \rho'_3\Big).$$

- If $Ch = 3$: Output $\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 3, (\pi'_1, \ldots, \pi'_p, \mathbf{r}'_1, \ldots, \mathbf{r}'_p, \tau', \mathbf{g}'); \rho'_1, \rho'_2\Big)$.

Case $\overline{Ch} = 2$: The simulator $\mathcal{S}$ picks vectors $\mathbf{r}'_1, \ldots, \mathbf{r}'_p \xleftarrow{\$} \mathbb{Z}_q^{3m}$, $\mathbf{g}' \xleftarrow{\$} \mathbb{Z}_q^N$; $\mathbf{u}'_1, \ldots, \mathbf{u}'_p \xleftarrow{\$} \mathsf{B}_{3m}$; a uniform $\mathbf{e}'$ in the set $\big\{\mathbf{e} \in \{0,1\}^N : \mathsf{wt}(\mathbf{e}) = 1\big\}$; permutations $\pi'_1, \ldots, \pi'_p \xleftarrow{\$} \mathsf{S}_{3m}$, $\tau' \in \mathsf{S}_N$; and uniformly random strings $\rho'_1, \rho'_2, \rho'_3$. It sends the commitment $(\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to $\widehat{V}$, where:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\big(\pi'_1, \ldots, \pi'_p, \tau',\ \mathbf{A}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{r}'_j) - \mathbf{B} \cdot \mathbf{g}' \bmod q; \rho'_1\big) \\ \mathbf{c}'_2 = \mathrm{COM}\big(\pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{g}'); \rho'_2\big) \\ \mathbf{c}'_3 = \mathrm{COM}\big(\pi'_1(\mathbf{u}'_1 + \mathbf{r}'_1), \ldots, \pi'_p(\mathbf{u}'_p + \mathbf{r}'_p), \tau'(\mathbf{e}' + \mathbf{g}'); \rho'_3\big) \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ computes the following transcript:

- If $Ch = 1$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 1, (\pi'_1(\mathbf{u}'_1), \ldots, \pi'_p(\mathbf{u}'_p), \pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{e}'), \tau'(\mathbf{g}')); \rho'_2, \rho'_3\Big).$$

- If $Ch = 2$: Output $\perp$ and halt.
- If $Ch = 3$: Output $\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 3, (\pi'_1, \ldots, \pi'_p, \mathbf{r}'_1, \ldots, \mathbf{r}'_p, \tau', \mathbf{g}'); \rho'_1, \rho'_2\Big)$.

Case $\overline{Ch} = 3$: The simulator picks the uniformly random vectors, permutations, and strings exactly as in the case $\overline{Ch} = 2$ above, but sends the following:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\big(\pi'_1, \ldots, \pi'_p, \tau',\ \mathbf{A}^* \cdot \sum_{j=1}^p \beta_j \cdot (\mathbf{u}'_j + \mathbf{r}'_j) - \mathbf{B} \cdot (\mathbf{e}' + \mathbf{g}') \bmod q; \rho'_1\big) \\ \mathbf{c}'_2 = \mathrm{COM}\big(\pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{g}'); \rho'_2\big) \\ \mathbf{c}'_3 = \mathrm{COM}\big(\pi'_1(\mathbf{u}'_1 + \mathbf{r}'_1), \ldots, \pi'_p(\mathbf{u}'_p + \mathbf{r}'_p), \tau'(\mathbf{e}' + \mathbf{g}'); \rho'_3\big) \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$, simulator $\mathcal{S}$ computes a transcript as follows:

- If $Ch = 1$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 1, (\pi'_1(\mathbf{u}'_1), \ldots, \pi'_p(\mathbf{u}'_p), \pi'_1(\mathbf{r}'_1), \ldots, \pi'_p(\mathbf{r}'_p), \tau'(\mathbf{e}'), \tau'(\mathbf{g}')); \rho'_2, \rho'_3\Big).$$

- If $Ch = 2$: Output

$$\Big(r', (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3), 2, (\pi'_1, \ldots, \pi'_p, \mathbf{u}'_1 + \mathbf{r}'_1, \ldots, \mathbf{u}'_p + \mathbf{r}'_p, \tau', \mathbf{e}' + \mathbf{g}'); \rho'_1, \rho'_3\Big).$$

- If $Ch = 3$: Output $\perp$ and halt.

We observe that, in every case we have considered above, since COM is statistically hiding, the distribution of the commitment from $\mathcal{S}$ and the challenge from $\widehat{V}$ are statistically close to those in the real interaction. Hence, the probability that the simulator outputs $\perp$ is negligibly close to 1/3. Moreover, one can check that whenever $\mathcal{S}$ does not halt, it will provide a successful transcript, and the distribution of the transcript is statistically close to that of the prover in the real interaction. Hence, we have constructed a simulator that can successfully impersonate the honest prover with probability 2/3, and completed the proof. □

Proof of knowledge. The following lemma states that the proof system presented in this section is a proof of knowledge for the relation $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$ with knowledge error 2/3.

Lemma 5.3.5. *Assume that* COM *is a computationally binding string commitment scheme. Then there exists a knowledge extractor $\mathcal{K}$ such that the following holds. If $\mathcal{K}$ has access to a cheating prover who convinces the verifier on input $(\mathbf{A}, \mathbf{B})$ with probability $2/3 + \epsilon$ for some $\epsilon > 0$ and in time T, then with overwhelming probability and in time $T \cdot \mathsf{poly}(n, m, N, \log q, 1/\epsilon)$, $\mathcal{K}$ outputs $(\mathbf{x}', \mathbf{e}')$ such that $((\mathbf{A}, \mathbf{B}), \mathbf{x}', \mathbf{e}') \in$*

$\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$.

Proof. The proof uses the same argument as in the proof of Theorem 3.3.3. Namely, it can be shown that, in time $T \cdot \mathsf{poly}(n, m, N, \log q, 1/\epsilon)$, the knowledge extractor $\mathcal{K}$ can obtain with overwhelming probability 3 valid responses which correspond to all 3 challenges for the same commitment. Therefore, $\mathcal{K}$ can get the following relations:

$$\begin{aligned}
\mathrm{COM}\big(\{\phi_j\}_{j=1}^{p}, \tau_2,\ \mathbf{A}^*(\sum_{j=1}^{p}\beta_j\mathbf{z}_j) - \mathbf{B}\mathbf{g}_2\big) &= \mathrm{COM}\big(\{\psi_j\}_{j=1}^{p}, \tau_3,\ \mathbf{A}^*(\sum_{j=1}^{p}\beta_j\mathbf{s}_j) - \mathbf{B}\mathbf{g}_3\big) \\
\mathrm{COM}\big(\mathbf{w}_1, \ldots, \mathbf{w}_p, \mathbf{g}_1\big) &= \mathrm{COM}\big(\psi_1(\mathbf{s}_1), \ldots, \psi_p(\mathbf{s}_p), \tau_3(\mathbf{g}_3)\big) \\
\mathrm{COM}\big(\phi_1(\mathbf{z}_1), \ldots, \phi_p(\mathbf{z}_p), \tau_2(\mathbf{g}_2)\big) &= \mathrm{COM}\big(\mathbf{v}_1 + \mathbf{w}_1, \ldots, \mathbf{v}_p + \mathbf{w}_p, \mathbf{e}_1 + \mathbf{g}_1\big),
\end{aligned}$$

and $\mathbf{v}_j \in \mathsf{B}_{3m}$ for all $j \in [p]$; $\mathbf{e}_1 \in \{0,1\}^N$, $\mathsf{wt}(\mathbf{e}_1) = 1$. Since COM is computationally binding, it follows that:

$$\begin{gathered}
\mathbf{A}^*\big(\sum_{j=1}^{p}\beta_j \cdot (\mathbf{z}_j - \mathbf{s}_j)\big) - \mathbf{B} \cdot (\mathbf{g}_2 - \mathbf{g}_3) = \mathbf{0} \bmod q; \\
\tau_2 = \tau_3;\ \ \mathbf{g}_1 = \tau_3(\mathbf{g}_3);\ \ \mathbf{e}_1 + \mathbf{g}_1 = \tau_2(\mathbf{g}_2);\ \ \mathbf{e}_1 \in \{0,1\}^N,\ \mathsf{wt}(\mathbf{e}_1) = 1; \\
\forall j \in [p]:\ \phi_j = \psi_j;\ \ \mathbf{w}_j = \psi_j(\mathbf{s}_j);\ \ \mathbf{v}_j + \mathbf{w}_j = \phi_j(\mathbf{z}_j);\ \ \mathbf{v}_j \in \mathsf{B}_{3m}.
\end{gathered}$$

Let $\mathbf{e}' := \mathbf{g}_2 - \mathbf{g}_3 = \tau_2^{-1}(\mathbf{e}_1)$, and for all $j \in [p]$, let $\mathbf{v}'_j := \mathbf{z}_j - \mathbf{s}_j = \phi_j^{-1}(\mathbf{v_j})$, then we obtain that

$$\begin{gathered}
\mathbf{e}' \in \{0,1\}^N;\ \ \mathsf{wt}(\mathbf{e}') = 1;\ \ \forall j \in [p]:\ \mathbf{v}'_j \in \mathsf{B}_{3m}; \\
\mathbf{A}^* \cdot \big(\sum_{j=1}^{p}\beta_j \cdot \mathbf{v}'_j\big) - \mathbf{B} \cdot \mathbf{e}' = \mathbf{0} \bmod q.
\end{gathered}$$

Now, for each $\mathbf{v}'_j$, we drop the last $2m$ coordinates to obtain $\widetilde{\mathbf{v}}'_j \in \{-1, 0, 1\}^m$, and

we let $\mathbf{x}' = \sum_{j=1}^{p} \beta_j \cdot \widetilde{\mathbf{v}}'_j$. Then we have $\mathbf{A} \cdot \mathbf{x}' - \mathbf{B} \cdot \mathbf{e}' = \mathbf{0} \bmod q$, and

$$\|\mathbf{x}'\|_\infty \leq \sum_{j=1}^{p} \beta_j \cdot \|\widetilde{\mathbf{v}}'_j\|_\infty \leq \sum_{j=1}^{p} \beta_j = \beta.$$

The knowledge extractor then outputs $(\mathbf{x}', \mathbf{e}')$ satisfying

$$((\mathbf{A}, \mathbf{B}), \mathbf{x}', \mathbf{e}') \in \mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N).$$

This concludes the proof. □

Since the given proof system is statistically zero-knowledge, it is also statistically witness indistinguishable. Since witness indistinguishability is preserved under parallel composition [57], by repeating the protocol $t = \omega(\log n)$ times in parallel, one can obtain a statistical witness indistinguishable proof of knowledge with negligible soundness error.

5.3.3 Construction from Lattices

In this section, we present our construction of a lattice-based IBRI scheme, based on the GPV signature scheme [62] and the proof system for $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$ in Section 5.3.2. Our scheme, called $\mathcal{LIBRI}$, is as follows:

Scheme 5.3.6 ($\mathcal{LIBRI}$)**.** Let n be a security parameter, and q, m, σ, β be functions of n, where:

- The modulus $q = \mathsf{poly}(n)$.
- The matrix dimension $m \geq 2n \log q$,
- The Gaussian parameter $\sigma = \omega(\sqrt{n \log q \log n})$,

- The norm bound $\beta = \lceil \sigma \cdot f \rceil = \widetilde{\mathcal{O}}(\sqrt{n})$, for some fixed function $f \geq \omega(\sqrt{\log m})$.

By Lemma 2.2.10 (item (4)), for parameters σ, β as above, the discrete Gaussian distribution $D_{\mathbb{Z},\sigma}$ is a β-bounded distribution.

Let $t = \omega(\log n)$ be the number of protocol repetitions, and $\mathsf{H} : \{0,1\}^* \to \mathbb{Z}_q^n$ be the random oracle used in the GPV signature. As for the IBI scheme in Section 5.2.2, the secret key of identity $id \in \{0,1\}^*$ is defined as a GPV signature of the message id. In the scheme, we will employ algorithms $\mathsf{GenTrap}$ from Theorem 2.3.3 and $\mathsf{SampleD}$ from Theorem 2.3.4.

- $\mathsf{MKg}(1^n)$: Run $\mathsf{GenTrap}(1^n, 1^m, q)$ to obtain matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a trapdoor $\mathbf{T}$. Output $\mathsf{mpk} = \mathbf{A}$, and $\mathsf{msk} = \mathbf{T}$.

- $\mathsf{UKg}(\mathsf{msk}, id)$: For an identity $id \in \{0,1\}^*$, let $\mathsf{sk}_{id} \leftarrow \mathsf{SampleD}(\mathbf{T}, \mathbf{A}, \mathsf{H}(id), \sigma)$. If $\|\mathsf{sk}_{id}\|_\infty > \beta$ (which happens only with negligible probability, since $D_{\mathbb{Z},\sigma}$ is a β-bounded distribution), then restart. Otherwise, output $\mathsf{sk}_{id} \in \mathbb{Z}^m$ as the secret key for identity id.

  We remark that the algorithms $\mathsf{MKg}(1^n)$ and $\mathsf{UKg}(\mathsf{msk}, id)$ are performed by a trusted authority, and they are the same as in the IBI scheme in Section 5.2.2.

- $\mathsf{RPKg}(\mathsf{mpk}, R)$: Let $R = (id_1, \ldots, id_N)$. The algorithm computes $\mathbf{B} \in \mathbb{Z}_q^{n \times N}$, where for each $i \in [N]$, the i-th column of matrix $\mathbf{B}$ is vector $\mathsf{H}(id_i) \in \mathbb{Z}_q^n$. It outputs $\mathsf{rpk} = (\mathbf{A}, \mathbf{B})$.

- $\mathsf{RSKg}(\mathsf{mpk}, R, \mathsf{sk}_{id})$: Let $R = (id_1, \ldots, id_N)$ and let k be the index of id in R, i.e., $id = id_k$. Let $\mathbf{e} \in \{0,1\}^N$, such that $\mathsf{wt}(\mathbf{e}) = 1$, and the only entry 1 in vector $\mathbf{e}$ is the k-th one. The algorithm outputs $\mathsf{rsk} = (\mathsf{sk}_{id}, \mathbf{e})$.

- $\langle P(\mathsf{rpk}, \mathsf{rsk}), V(\mathsf{rpk}) \rangle$: Let $\mathsf{rpk} = (\mathbf{A}, \mathbf{B}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times N}$, and $\mathsf{rsk} = (\mathbf{x}, \mathbf{e})$. By

construction, one has

$$\begin{cases} \mathbf{x} \in \mathbb{Z}^m \text{ and } \|\mathbf{x}\|_\infty \leq \beta; \\ \mathbf{e} \in \{0,1\}^N \text{ and } \mathsf{wt}(\mathbf{e}) = 1; \\ \mathbf{A} \cdot \mathbf{x} - \mathbf{B} \cdot \mathbf{e} = \mathbf{0} \bmod q. \end{cases}$$

In other words, we have $\big((\mathbf{A}, \mathbf{B}), (\mathbf{x}, \mathbf{e})\big) \in \mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$. The prover $P((\mathbf{A}, \mathbf{B}), \mathbf{x}, \mathbf{e})$ and the verifier $V(\mathbf{A}, \mathbf{B})$ execute a WIPoK for the relation $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$. This is done by repeating the ZK proof system in Section 5.3.2 in parallel $t = \omega(\log n)$ times to make the soundness error negligibly small.

Completeness. The scheme $\mathcal{LIBRI}$ has perfect completeness. It follows directly from the perfect completeness of the underlying proof system.

Security against impersonation. The following theorem states that in the random oracle model, the scheme $\mathcal{LIBRI}$ is secure against impersonation, based on the hardness of the $\mathsf{SIS}_{n,m,q,2\beta}$ problem.

Theorem 5.3.7. *In the random oracle model, if there exists a polynomial-time adversary $\mathcal{A}$ having non-negligible advantage in the experiment $\mathsf{Exp}^{\mathsf{imp}}_{\mathcal{LIBRI},\mathcal{A}}(n)$, then there exists a polynomial-time algorithm solving the $\mathsf{SIS}^{\infty}(n, m, q, 2\beta)$ problem with non-negligible probability.*

It follows from Theorem 5.3.7 and Theorem 2.2.14 that the scheme $\mathcal{LIBRI}$ is secure in the random oracle model based on the worst-case hardness of SIVP^2_γ, with $\gamma = 2\beta \cdot \widetilde{\mathcal{O}}(n) = \widetilde{\mathcal{O}}(n^{1.5})$.

Proof. We construct a polynomial-time algorithm $\mathcal{S}$ attacking the $\mathsf{SIS}^{\infty}(n, m, q, 2\beta)$ problem. Without loss of generality, we assumes that COM is computationally bind-

ing (since an adversary breaking the computational binding property of COM can be used to solve the $\mathsf{SIS}^{\infty}(n, m, q, 2\beta)$ problem, see Theorem 2.3.2). By programming the random oracle H, algorithm $\mathcal{S}$ simulates the challenger in the experiment $\mathsf{Exp}^{\mathsf{imp}}_{\mathcal{IBRI},\mathcal{A}}(n)$ as follows:

- $\mathcal{S}$ receives a challenge $\mathsf{SIS}^{\infty}(n, m, q, 2\beta)$ instance $\mathbf{A}$ which is uniformly random over $\mathbb{Z}_q^{n\times m}$. It wins the challenge if it can produce a non-zero vector $\mathbf{x} \in \mathbb{Z}^m$ such that $\|\mathbf{x}\|_\infty \leq 2\beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q$.

- Observe that by Theorem 2.3.3, the distribution of $\mathbf{A}$ is statistically close to that in the real scheme. Algorithm $\mathcal{S}$ sets $\mathsf{mpk} := \mathbf{A}$ and sends it to the adversary $\mathcal{A}$.

- $\mathcal{A}$ can adaptively and concurrently query the interaction transcript of any identity id contained in any ring R. To generate the ring public key, it must query to the random oracle for $\mathsf{H}(id)$. $\mathcal{S}$ handles the query by sampling a vector $\mathbf{x}_{id} \hookleftarrow \mathrm{D}_{\mathbb{Z}^m,\sigma}$, recording $\mathbf{x}_{id}$, programming $\mathsf{H}(id) := \mathbf{A} \cdot \mathbf{x}_{id} \bmod q$, and returning $\mathsf{H}(id)$, which is statistically close to uniform over $\mathbb{Z}_q^n$ (by Lemma 2.2.10, item (1)). If $\mathcal{A}$ ever repeats the same query, then $\mathcal{S}$ returns with consistency.

  Now, to answer the query for the interaction transcript of $id \in R$, algorithm $\mathcal{S}$ lets $(\mathbf{A}, \mathbf{B}) \leftarrow \mathsf{RPKg}(\mathbf{A}, R)$, $(\mathbf{x}, \mathbf{e}) \leftarrow \mathsf{RSKg}(\mathbf{A}, R, \mathbf{x}_{id})$, and executes the WIPoK for the relation $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$ with common input $(\mathbf{A}, \mathbf{B})$ and witness $(\mathbf{x}, \mathbf{e})$.

- At some point, $\mathcal{A}$ returns a target ring of identities $R^* = (id_1^*, \ldots, id_{N^*}^*)$ and an interaction transcript, such that the honest verifier $V(\mathbf{A}, \mathbf{B}^*)$, where $\mathbf{B}^*$ is produced by $\mathsf{RPKg}(\mathbf{A}, R^*)$, accepts with non-negligible probability.

Since the underlying protocol is a proof of knowledge with negligible soundness error

for the relation $\mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$, algorithm $\mathcal{S}$ can run the knowledge extractor in Lemma 5.3.7 to obtain in polynomial time and with overwhelming probability a pair $(\mathbf{x}^*, \mathbf{e}^*)$ such that $((\mathbf{A}, \mathbf{B}^*), \mathbf{x}^*, \mathbf{e}^*) \in \mathrm{R}_{\mathrm{LIBRI}}(n, m, q, \beta, N)$.

Now let k be the index of the only entry 1 of vector e, and let $\mathbf{y}_k$ be the k-th column of matrix $\mathbf{B}^*$. We have that:

$$\|\mathbf{x}^*\|_\infty \leq \beta \text{ and } \mathbf{A} \cdot \mathbf{x}^* = \mathbf{y}_k = \mathsf{H}(id_k^*) = \mathbf{A} \cdot \mathbf{x}_{id_k^*} \bmod q,$$

where $\mathbf{x}_{id_k^*}$ is the vector sampled and recorded by $\mathcal{S}$ while answering the random oracle query for $\mathsf{H}(id_k^*)$. Furthermore, by Lemma 2.2.10 (item (3)), the min-entropy of $\mathbf{x}^*$ given $\mathbf{y}_k$ is $\omega(\log n)$. Thus, we have $\mathbf{x}^* \neq \mathbf{x}_{id_k^*}$, except for probability $\mathsf{negl}(n)$. Now, let $\mathbf{x} := \mathbf{x}^* - \mathbf{x}_{id_k^*} \in \mathbb{Z}^m$, then $\|\mathbf{x}\|_\infty \leq \beta + \beta = 2\beta$, and $\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q$.

Algorithm $\mathcal{S}$ outputs $\mathbf{x}$, which is a solution to the challenge $\mathsf{SIS}^(n, m, q, 2\beta)$ instance $\mathbf{A}$. This concludes the proof. □

Anonymity against full key exposure. The unconditional anonymity of the scheme follows directly from the statistical witness indistinguishability property of the underlying protocol. In the experiment $\mathbf{Exp}^{\mathsf{anon}}_{\mathcal{LIBRI},\mathcal{A}}(n)$, the challenger first generates $(\mathbf{A}, \mathbf{T}) \leftarrow \mathsf{MKg}(1^n)$ and sends $\mathbf{A}$ to the adversary $\mathcal{A}$. Whenever $\mathcal{A}$ asks for the secret key of an identity id, the challenger returns $\mathsf{sk}_{id} \leftarrow \mathsf{UKg}(\mathbf{T}, id)$. At some point, $\mathcal{A}$ outputs two valid pairs $(id_0, \mathsf{sk}_{id_0})$, $(id_1, \mathsf{sk}_{id_1})$ and a ring R containing (id_0, id_1). The challenger then picks $b \xleftarrow{\$} \{0, 1\}$, computes $(\mathbf{A}, \mathbf{B}) \leftarrow \mathsf{RKg}(\mathbf{A}, R)$, and $(\mathbf{x}_b, \mathbf{e}_b) \leftarrow \mathsf{RSKg}(A, R, \mathsf{sk}_{id_b})$, and returns a transcript of $\langle P, V \rangle$ using witness $(\mathbf{x}_b, \mathbf{e}_b)$. At the end of the experiment $\mathcal{A}$ outputs $b' \in \{0, 1\}$. The statistical witness indistinguishability property of the protocol ensures that $b' = b$ with probability $1/2 + \mathsf{negl}(n)$ for any polynomially unbounded adversary $\mathcal{A}$. In other words, $\mathcal{A}$ has negligible advantage in the experiment. Therefore, we have the following theorem.

Theorem 5.3.8. *Assume that* COM *is a statistically hiding string commitment scheme. Then, for the chosen parameters, the scheme* $\mathcal{LIBRI}$ *is unconditional anonymous against full key exposure.*

In summary, we have constructed an ID-based ring identification scheme, which is secure in the random oracle model if $\mathsf{SIVP}^2_{\widetilde{\mathcal{O}}(n^{1.5})}$ is hard in the worst-case. To the best of our knowledge, this is the first ID-based ring identification scheme from lattice assumptions.

6. IMPROVED LATTICE-BASED GROUP SIGNATURE SCHEME

6.1 Introduction

Group signatures have been an important research topic in public-key cryptography since their introduction in 1991 by Chaum and van Heyst [46]. In these schemes, all the potential users form a group, where each user can anonymously issue a signature on behalf of the whole group (*anonymity*). On the other hand, in cases of disputes, there is a tracing mechanism which can link a given signature to the identity of the misbehaving user (*traceability*). These two seemingly contradictory features allow group signatures to find applications in various real-life scenarios, such as anonymous online communications, digital right management, e-commerce systems, and much more. Over the last two decades, many group signature schemes with different security models and different level of efficiency have been proposed ([48, 40, 16, 19, 27, 28, 23, 75, 92, 91], ...).

The state-of-the-art group signature schemes (e.g., [16, 27, 28],...) are very efficient, but they all rely on the hardness of factoring integers or solving discrete logarithms. As shown by Shor [135], these schemes would become insecure once powerful quantum computers become a reality. This gives rise to an interesting research challenge: designing efficient quantum-resistant group signature schemes. Up to the best of our knowledge, three quantum-resistant group signature schemes

have been published [74, 38, 88], all of which are based on lattice assumptions. As we will describe in Section 6.3, each of these schemes has there own strength and weakness. This gives room for improvements.

In this chapter, we will present a new lattice-based group signature scheme which combines the good features of the existing ones. We first recall the definition and security notions of group signature schemes in Section 6.2. We then review the previous works in Section 6.3 and summarize our contribution and high-level techniques in Section 6.4. Next, we construct the underlying interactive proof system in Section 6.5. Finally, we present our group signature in Section 6.6.

6.2 Definitions and Security Notions

The presentation in this section follows the work of Chaum and van Heyst [46] which introduced group signatures; the work of Bellare et al. [19] which formalized the notions of *full anonymity* (also known as CCA-anonymity) and *full traceability*; and the work of Boneh et al. [27] which suggested the relaxed notion of *weak anonymity* (also known as CPA-anonymity).

Definition 6.2.1. A group signature scheme $\mathcal{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ is a tuple of four polynomial-time algorithms:

- $\mathsf{KeyGen}(1^n, 1^N)$: The randomized key generation algorithm takes as input $1^n, 1^N$, where $n \in \mathbb{N}$ is the security parameter and $N \in \mathbb{N}$ is the number of group users, and outputs a triple $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$, where gpk is the group public key; gmsk is the group manager's secret key; and $\mathsf{gsk} = \mathsf{gsk}[1] \dots \mathsf{gsk}[N]$ is an N-vector of keys, such that $\mathsf{gsk}[i]$ is the secret key for the group user with index $i \in [N]$.

- $\mathsf{Sign}(\mathsf{gsk}[i], M)$: The randomized signing algorithm takes as input a secret signing

key $\mathsf{gsk}[i]$ for some $i \in [N]$, and a message M, and returns a group signature Σ of M.

- $\mathsf{Verify}(\mathsf{gpk}, M, \Sigma)$: The deterministic verification algorithm takes as input the group public key gpk, a message M, a purported signature Σ of M, and returns either 1 (Valid) or 0 (Invalid).

- $\mathsf{Open}(\mathsf{gmsk}, M, \Sigma)$: The deterministic opening algorithm takes as input the group manager's secret key gmsk, a message M, a signature Σ of M, and returns an index $i \in [N]$, or $\perp$ (to indicate failure).

Correctness. The correctness requirement for a group signature scheme is as follows. For all $n, N \in \mathbb{N}$, all $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$ produced by $\mathsf{KeyGen}(1^n, 1^N)$, all $i \in [N]$, and all $M \in \{0,1\}^*$,

$$\mathsf{Verify}\big(\mathsf{gpk}, M, \mathsf{Sign}(\mathsf{gsk}[i], M)\big) = 1 \quad \text{and} \quad \mathsf{Open}\big(\mathsf{gmsk}, M, \mathsf{Sign}(\mathsf{gsk}[i], M)\big) = i.$$

In other words, it is required that legitimate signatures are always accepted and the opening algorithm correctly returns the identity of the signer from a legitimate signature.

Compactness. In practice, if the sizes of the keys and signatures in a group signature depend linearly on the number of group users, then it would be inefficient to implement the scheme for large groups. On the other hand, it was pointed out in [19] and [88] that "a polynomial dependency of these sizes on $\log N$ can be shown to be unavoidable", and that "no group signature scheme can provide signatures of bit-size $o(\log N)$" because "such signatures would be impossible to open". In other words, asymptotically these sizes are expected to be at least $\mathsf{poly}(\log N)$. To evaluate the efficiency of group signature schemes, Bellare et al. [19] introduced the criterion

of *compactness*.

Definition 6.2.2. A group signature scheme $\mathcal{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ is called compact if there exist polynomials $p_1(\cdot,\cdot)$ and $p_2(\cdot,\cdot,\cdot)$ such that

$$|\mathsf{gpk}|, |\mathsf{gmsk}|, |\mathsf{gsk}[i]| \leq p_1(n, \log N) \text{ and } |\Sigma| \leq p_2(n, \log N, |M|)$$

for all n, N, all $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{KeyGen}(1^n, 1^N)$, all $i \in [N]$, all $M \in \{0,1\}^*$, and all $\Sigma \leftarrow \mathsf{Sign}(\mathsf{gsk}[i], M)$.

Security Notions. A secure group signature scheme must satisfy the anonymity and traceability requirements defined below.

Anonymity. Anonymity requires that any adversary, who does not possess the group manager's secret key, should not be able to determine the identity of the user given its signature. Bellare et al. [19] introduced the notion of *full anonymity*, where the adversary has strong attack capabilities: it is given the secret keys of all group users, and is granted access to an *opening oracle*. In other words, this adversary is allowed to collude with all group users, and to see the results of openings of all signatures (except for the target one). Boneh et al. [27] later proposed a relaxed notion, called *weak anonymity*, where the adversary cannot query the opening oracle. We note that full anonymity and weak anonymity are often called "CCA-anonymity" and "CPA-anonymity", respectively, since their difference is similar to that between the notions of CCA and CPA security for PKE schemes.

More formally, consider the following anonymity experiment $\mathbf{Exp}^{\mathsf{t\text{-}anon}}_{\mathcal{GS},\mathcal{A}}(n, N)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$, where $\mathsf{t} \in (\mathsf{weak}, \mathsf{full})$.

Experiment $\mathbf{Exp}^{\mathsf{t\text{-}anon}}_{\mathcal{GS},\mathcal{A}}(n, N)$:

- **Initialization Phase:** The challenger $\mathcal{C}$ runs the key generation algorithm

$\mathsf{KeyGen}(1^n, 1^N)$ to obtain $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$, then it gives $(\mathsf{gpk}, \mathsf{gsk})$ to $\mathcal{A}$.

- **Query phase 1:** If $\mathsf{t} = \mathsf{full}$, then $\mathcal{A}$ can make queries to the opening oracle. On input a message M and a signature Σ, the oracle returns $\mathsf{Open}(\mathsf{gmsk}, M, \Sigma)$ to $\mathcal{A}$.

- **Challenge phase:** $\mathcal{A}$ outputs two distinct identities $i_0, i_1 \in [N]$ and a message M^*. The challenger picks a coin $b \xleftarrow{\$} \{0,1\}$, computes the target signature $\Sigma^* = \mathsf{Sign}(\mathsf{gsk}[i_b], M^*)$ and sends Σ^* to $\mathcal{A}$.

- **Query phase 2:** If $\mathsf{t} = \mathsf{full}$, then the adversary $\mathcal{A}$ can make queries to the opening oracle. On input (M, Σ), if $(M, \Sigma) = (M^*, \Sigma^*)$, then the challenger outputs 0 and halts; otherwise it returns $\mathsf{Open}(\mathsf{gmsk}, M, \Sigma)$ to $\mathcal{A}$.

- **Guessing phase:** Finally, $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$. If $b' = b$, then $\mathcal{C}$ outputs 1, otherwise it outputs 0.

Definition 6.2.3. Let $\mathcal{A}$ be an adversary against the anonymity of a group signature scheme $\mathcal{GS}$. Define the advantage of $\mathcal{A}$ in the above experiment as

$$\mathbf{Adv}^{\mathsf{t\text{-}anon}}_{\mathcal{GS},\mathcal{A}}(n, N) = \Big|\Pr\big[\mathbf{Exp}^{\mathsf{t\text{-}anon}}_{\mathcal{GS},\mathcal{A}}(n, N) = 1\big] - 1/2\Big|.$$

We say that $\mathcal{GS}$ is weakly anonymous (*resp.* fully anonymous) if for all polynomial $N(\cdot)$ and all polynomial-time adversaries $\mathcal{A}$, the function $\mathbf{Adv}^{\mathsf{weak\text{-}anon}}_{\mathcal{GS},\mathcal{A}}(n, N)$ (*resp.* $\mathbf{Adv}^{\mathsf{full\text{-}anon}}_{\mathcal{GS},\mathcal{A}}(n, N)$) is negligible in the security parameter n.

Traceability. *Full traceability* [19] requires that all signatures, even those produced by a coalition of group users *and* the group manager, can be traced back to a member of the coalition. This notion is formally defined by the following traceability experiment $\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n, N)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Experiment $\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n, N)$:

- **Initialization Phase:** The challenger $\mathcal{C}$ runs algorithm $\mathsf{KeyGen}(1^n, 1^N)$ to obtain $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$, then it sets $CU \leftarrow \emptyset$ and gives $(\mathsf{gpk}, \mathsf{gmsk})$ to $\mathcal{A}$.
- **Query Phase:** The adversary $\mathcal{A}$ can make the following queries adaptively, and in any order:
  - Secret key query: On input $i \in [N]$, the challenger adds i to CU, and returns $\mathsf{gsk}[i]$ to $\mathcal{A}$.
  - Signing query: On input i, M, the challenger returns $\mathsf{Sign}(\mathsf{gsk}[i], M)$.
- **Challenge Phase:** $\mathcal{A}$ outputs a message M, and a signature Σ. The challenger proceeds as follows:

  If $\mathsf{Verify}(\mathsf{gpk}, M, \Sigma) = 0$ then return 0. If $\mathsf{Open}(\mathsf{gmsk}, M, \Sigma) = \perp$ then return 1.

  If $\exists i \in [N]$ such that the following are true then return 1, else return 0:

  1. $\mathsf{Open}(\mathsf{gmsk}, M, \Sigma) = i \notin CU$,
  2. $\mathcal{A}$ has never make a signing query for i, M.

Definition 6.2.4. Let $\mathcal{A}$ be an adversary against the traceability of a group signature scheme $\mathcal{GS}$. Define the advantage of $\mathcal{A}$ in the above experiment as

$$\mathbf{Adv}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n, N) = \Pr\big[\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n, N) = 1\big].$$

We say that $\mathcal{GS}$ is fully traceable if for all polynomial $N(\cdot)$ and all polynomial-time adversaries $\mathcal{A}$, the function $\mathbf{Adv}^{\mathsf{trace}}_{\mathcal{GS},\mathcal{A}}(n, N)$ is negligible in the security parameter n.

6.3 Previous Works

The first lattice-based group signature scheme, which is essentially the first quantum-resistant one, was introduced by Gordon, Katz and Vaikuntanathan [74] in 2010. Below, we will refer to this scheme as the GKV scheme. At a high level, the GKV scheme follows a common strategy in constructing group signatures, which makes use of three basic cryptographic ingredients: a secure signature scheme, a secure encryption scheme and a ZK (or WI) proof system. Specifically, the corresponding ingredients in [74] are: the GPV signature [62], a variant of the Dual-Regev encryption [62], and the Micciancio-Vadhan proof system [115][1]. The scheme relies on relatively weak hardness assumptions: It is CPA-anonymous assuming the quantum hardness of SIVP_γ, where γ can be as small as $\widetilde{\mathcal{O}}(n^2)$; And it is traceable if $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^{1.5})}$ is hard. While the GKV scheme is of great theoretical interest, it produces signatures of size $\widetilde{\mathcal{O}}(n^2) \cdot N$. In terms of efficiency, this is a noticeable disadvantage when the group is large, e.g., group of all employees of a big company.

In 2012, Camenisch, Neven and Rückert [38] proposed lattice-based anonymous attribute tokens system, a primitive that can be considered as a generalization of group signature. We will refer to their scheme as the CNR scheme. The CNR scheme follows a similar approach as of the GKV scheme, but it achieves the CCA-anonymity notion (i.e., full anonymity) by employing an additional cryptographic ingredient: a strongly unforgeable one-time signature. However, in their construction, the signature size inherits the linear dependence on N of the GKV signature, i.e., its bit-size is still $\widetilde{\mathcal{O}}(n^2) \cdot N$. In addition, the traceability of the scheme is only guaranteed if SIVP_γ is hard for slightly larger approximation factor, i.e., $\gamma = \widetilde{\mathcal{O}}(n^2)$.

Recently, Laguillaumie, Langlois, Libert, and Stehlé [88] designed a scheme fea-

[1] More precisely, the proof system in [74] is for the disjunction of N instances of GapCVP_γ, obtained by applying the technique of Cramer et al. [49] to the Micciancio-Vadhan protocol.

turing signature size $\widetilde{\mathcal{O}}(n \cdot \log N)$, which is the first lattice-based group signature that overcomes the linear-size barrier. In particular, the scheme is compact, in the sense of Definition 6.2.2. We will refer to this scheme as the LLLS scheme. The basic variant of the LLLS scheme only satisfies the CPA-anonymity notion, but CCA-anonymity can also be achieved by using a strongly secure one-time signature. Instead of relying on the GPV signature as in the GKV and CNR schemes, the LLLS scheme operates in the "mixing" structure of hard random lattices, which is used in the Boyen signature [29]. Each group user is identified by a string $d \in \{0,1\}^{\ell}$, where $\ell = \lceil \log N \rceil$, and is given a trapdoor of the mixing lattice determined by d. To sign a message, the user first generates a Boyen signature of his identity d, and encrypts (part of) this signature using two different variants of the Dual-Regev encryption. Next, the user generates $\ell + 2$ non-interactive ZKPoK (obtained via Fiat-Shamir heuristic) linking together in a sophisticated fashion to prove that he knows a Boyen signature for certain $d \in \{0,1\}^{\ell}$ and that the ciphertexts he provided are well-formed. The resulting group signature is asymptotically shorter than in previous works [74, 38], but the scheme requires rather large parameters, e.g., $q = \widetilde{\mathcal{O}}(n^8 \cdot \sqrt{\log N} + \log N)$. As a consequence, the anonymity of the scheme has to rely on the quantum hardness of SIVP_{γ}, with $\gamma = \sqrt{\log N} \cdot \widetilde{\mathcal{O}}(n^{8.5})$, while its traceability is based on the hardness of $\mathsf{SIVP}_{\sqrt{\log N} \cdot \widetilde{\mathcal{O}}(n^{7.5})}$. Although SIVP_{γ} is assumed to be hard for any $\gamma = \mathsf{poly}(n)$, these hardness assumptions are much stronger than in contemporary lattice-based cryptographic constructions.

It follows from the above discussions that each of the existing lattice-based group signature schemes has its own strength and weakness: those with relatively weak hardness assumptions (i.e., the GKV and CNR schemes) have signature size linear in N, while the only known scheme with logarithmic signature size (i.e., the LLLS scheme) has to rely on much stronger assumptions. This raises an interesting

question: Is it possible to construct a lattice-based group signature scheme that simultaneously achieves the good features of existing schemes?

6.4 Our Contribution and Techniques

6.4.1 Our Contribution

In this work, we construct a lattice-based group signature scheme that simultaneously achieve the good features of existing schemes. Specifically, in our construction, the bit-size of the signature is $\widetilde{\mathcal{O}}(n \cdot \log N)$ (which is comparable to that of the LLLS scheme), while the hardness assumptions are comparable to those of the GKV scheme: the anonymity of the scheme relies on the quantum hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^2)}$, and its traceability is based on the hardness of $\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^{1.5})}$. Moreover, as we will show in the next section, our signature is syntactically very simple.

In Table 6.1, we give a comparison among known lattice-based group signature schemes.

Scheme	GKV	CNR	LLLS	This work
Bit-size of the signature	$\widetilde{\mathcal{O}}(n^2) \cdot N$	$\widetilde{\mathcal{O}}(n^2) \cdot N$	$\widetilde{\mathcal{O}}(n \cdot \log N)$	$\widetilde{\mathcal{O}}(n \cdot \log N)$
Quantum hardness assumption for Anonymity	$\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^2)}$	$\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^2)}$	$\mathsf{SIVP}_{\sqrt{\log N} \cdot \widetilde{\mathcal{O}}(n^{8.5})}$	$\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^2)}$
Hardness assumption for Traceability	$\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^{1.5})}$	$\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^2)}$	$\mathsf{SIVP}_{\sqrt{\log N} \cdot \widetilde{\mathcal{O}}(n^{7.5})}$	$\mathsf{SIVP}_{\widetilde{\mathcal{O}}(n^{1.5})}$

Tab. 6.1: Comparison among lattice-based group signature schemes. For the GKV, CNR and LLLS schemes, the approximation factors γ in the SIVP_γ hardness assumptions are estimated based on the parameters given in the corresponding works [74, 38, 88]. The GKV scheme and the one presented in this work only satisfy the CPA-anonymity notion, while the CNR and LLLS schemes achieve CCA-anonymity.

Open Problems. Similar to the GKV scheme, our scheme only satisfies the CPA-anonymity notion suggested by Boneh et al. [27]. Upgrading it into a CCA-anonymous group signature is an interesting target for future investigation. A possible solution towards this direction is to employ a strongly secure one-time signature, as in the CNR and LLLS schemes.

It is also important to note that all 4 schemes considered in Table 6.1 are only proven secure in the random oracle model: They all rely on the Fiat-Shamir heuristic to remove interaction in the underlying interactive protocols. Thus, designing lattice-based group signature schemes that are secure in the standard model remains an open question. This is closely related to the challenging problem of constructing non-interactive zero-knowledge proofs without the random oracle for average-case lattice problems ISIS and LWE.

6.4.2 Overview of Our Techniques

At a high level, our group signature makes use of three lattice-based cryptographic ingredients: the Bonsai tree signature [42] (see also Scheme 2.3.9); a variant of the Dual-Regev encryption [62, 61]; and a special ZKPoK that combines an adaptation of the **SternExt** proof system (see Section 3.3) to the Bonsai tree structure, and an adaptation of the PoPK for the Dual-Regev encryption scheme (see Section 4.3).

In our scheme, the signature of an arbitrary message M is syntactically very simple: It is just a pair $(\mathbf{c}, \pi)$, where:

- $\mathbf{c}$ is an encryption of the signer's identity.
- π is a non-interactive zero-knowledge proof, obtained via Fiat-Shamir heuristic from an interactive protocol, in which the verifier is convinced in zero-

knowledge that:

The prover knows the secret key of certain group user,

whose identity encrypts to $\mathbf{c}$.

Specifically, we consider a group of $N = 2^\ell$ users, where each user is identified by a string $d \in \{0,1\}^\ell$ denoting the binary representation of his index in the group[2]. Let n, m, q, β, k be integers (to be determined later). Our scheme operates within the structure of a *Bonsai tree* of hard random lattices ([42], see also Section 2.3.9), namely, a uniformly random matrix $\mathbf{A} = \left[\mathbf{A}_0 \middle| \mathbf{A}_0^1 \middle| \mathbf{A}_1^0 \middle| \mathbf{A}_1^1 \middle| \ldots \middle| \mathbf{A}_\ell^0 \middle| \mathbf{A}_\ell^1\right] \in \mathbb{Z}_q^{n \times (2\ell+1)m}$, and a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$. Initially, the group user with identity $d = d[1] \ldots d[\ell] \in \{0,1\}^\ell$ is issued a Bonsai signature of his identity, that is a small vector $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$, such that $\|\mathbf{z}\|_\infty \leq \beta$ and $\mathbf{A}_d \cdot \mathbf{z} = \mathbf{u} \bmod q$, where $\mathbf{A}_d = \left[\mathbf{A}_0 \middle| \mathbf{A}_1^{d[1]} \middle| \ldots \middle| \mathbf{A}_\ell^{d[\ell]}\right] \in \mathbb{Z}_q^{n \times (\ell+1)m}$ - a "subtree" defined by d. In other words, $\mathbf{z}$ is a solution to the $\mathsf{ISIS}^\infty(n, (\ell+1)m, q, \beta)$ instance $(\mathbf{A}_d, \mathbf{u})$.

To prove that he "knows the secret key of group user with identity d" without leaking $\mathbf{z}$, the user can run a ZKPoK for $\mathrm{R}_{\mathsf{ISIS}^\infty}(n, (\ell+1)m, q, \beta)$ with common input $(\mathbf{A}_d, \mathbf{u})$, using $\mathbf{z}$ as the witness. As we have seen in Chapter 3, there are several proof systems for this task, among which **SternExt** provides the strongest security guarantee.

At this stage, one can obtain a secure identity-based identification scheme (see Section 5.2), but it is insufficient for our purposes: to achieve anonymity, the group user also has to *hide* his identity d, and hence the matrix $\mathbf{A}_d$ should not be explicitly given. This raises an interesting question: If the verifier does not know $\mathbf{A}_d$, how

[2] This is a common strategy in constructing *compact* group signature schemes. The technique was first introduced in the Boyen-Waters group signature [30], and was first adapted to lattice-based group signatures in the LLLS scheme [88].

could he be convinced that $\mathbf{A}_d \cdot \mathbf{z} = \mathbf{u} \bmod q$?

To address the above question, we introduce the following extension: we add ℓ *suitable zero-blocks* of size m to vector $\mathbf{z}$ to obtain the extended vector

$$\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \mathbb{Z}^{(2\ell+1)m},$$

where the added zero-blocks are $\mathbf{x}_1^{1-d[1]}, \ldots, \mathbf{x}_\ell^{1-d[\ell]}$. We then have $\|\mathbf{x}\|_\infty \leq \beta$, and $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$. Namely, $\mathbf{x}$ is a solution to the $\mathsf{ISIS}^\infty(n, (2\ell+1)m, q, \beta)$ instance given by the *whole Bonsai tree*, with an additional constraint: for each $i \in [\ell]$, one of the two blocks $\mathbf{x}_i^0, \mathbf{x}_i^1$ must be zero, where the arrangement of the zero-blocks is determined by d. To prove in zero-knowledge the possession of such a vector $\mathbf{x}$, we employ the **SternExt** proof system, together with a "*one-time pad*" technique for hiding the user identity d.

The technique is as follows. In each round of the protocol, the user samples a fresh uniformly random $e \in \{0,1\}^\ell$ and permutes the blocks of $\mathbf{x}$ to obtain the permuted vector $\mathbf{v}$, whose zero-blocks are arranged according to $d \oplus e$ (where $\oplus$ denotes the bit XOR operation). Depending on the verifier's challenge, the user later will either reveal e, or reveal $d \oplus e$ and show that $\mathbf{v}$ has the correct shape determined by $d \oplus e$. Since $d \oplus e$ is uniformly random over $\{0,1\}^\ell$, the user identity d is completely hidden. Combining this technique with **SternExt**, we obtain a protocol in which the prover is able to convince the verifier that he "knows the secret key of certain group user".

The next step is to link the above protocol with the encryption layer. Our idea is as follows. Let k_0 be some positive integer (to be determined later). To encrypt identity $d \in \{0,1\}^\ell$, we first map it to vector

$$\mathbf{y} = (\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \ldots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1) \in \{0,1\}^{(2\ell+1)k_0}$$

via a *bijection*, where $\mathbf{y}$ consists of $2\ell + 1$ blocks of size k_0 such that: for each $i \in [\ell]$, the block $\mathbf{y}_i^{1-d[i]}$ is zero-block. It is important to note that the shape of $\mathbf{y}$ is determined by d in the same fashion as the shape of vector $\mathbf{x}$ given above. We then encrypt $\mathbf{y}$ using a variant of the Dual-Regev encryption scheme [62] to get the ciphertext $\mathbf{c}$.

Adapting the techniques presented in Chapter 4, we can construct a statistical ZKPoPK for the underlying encryption scheme. Combining this proof system with the one obtained before, we will get an interactive protocol, in which, given $((\mathbf{A}, \mathbf{u}), \mathbf{c})$ (and the public key of the encryption scheme), the user is able to convince the verifier in zero-knowledge that he knows $\mathbf{x}, \mathbf{y}, \mathbf{d}$ (and the encryption randomness) such that:

1. $\|\mathbf{x}\| \leq \beta$ and $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$.

2. $\mathbf{y}$ encrypts to $\mathbf{c}$.

3. $\mathbf{x}$ and $\mathbf{y}$ simultaneously have the correct shape determined by d. This is done by simultaneously applying the "one-time pad" technique above for $\mathbf{x}$ and $\mathbf{y}$. Namely, the user permutes $\mathbf{x}$ and $\mathbf{y}$ by the same permutation, and shows that the two permuted vectors have the same shape.

The obtained interactive protocol is repeated many times to make the soundness error negligibly small. Then it is transformed into a non-interactive proof π via the Fiat-Shamir heuristic which replaces the verifier by a random oracle that takes the message to be signed as part of its input.

As a result, we obtain a group signature scheme, where the signature for message $M \in \{0,1\}^*$ is a pair $(\mathbf{c}, \pi)$. To open the signature, the group manager uses the decryption key to decrypt $\mathbf{c}$ and recover d. The correctness of the scheme follows

from the completeness of the underlying protocol and the correctness of the encryption scheme. Furthermore, in the random oracle model, its CPA-anonymity relies on the CPA-security of the encryption scheme and the zero-knowledge property of the underlying protocol, while its traceability is based on the fact that the underlying protocol is a proof of knowledge.

6.5 The Underlying Proof System

6.5.1 Preparation

Parameters.

We use the following parameters for the protocol in Section 6.5, and for the group signature scheme in Section 6.6.

- A security parameter n.
- A maximum expected number of group users $N = 2^\ell \in \mathsf{poly}(n)$.
- A prime modulus $q = \widetilde{\mathcal{O}}(n^{1.5})$.
- A dimension $m \geq 2n \log q$.
- A dimension $k \geq 2n \log q$ such that $k = (2\ell + 1)k_0$ for some integer k_0.
- A Gaussian parameter $\sigma = \omega(\sqrt{n \log q \log n})$.
- An integer $\beta = \lceil \sigma \cdot f \rceil$, where $f \geq \omega(\sqrt{\log m})$ is some fixed function . For such β, the Gaussian distribution $\chi = D_{\mathbb{Z},\sigma}$ is β-bounded, by Lemma 2.2.10 (item (4)).
- Given β, let $p = \lfloor \log \beta \rfloor + 1$, and define the sequence $\beta_1, \ldots, \beta_p$ as in the **SternExt** proof system.

Some Specific Sets.

In our construction, we will work with vectors and permutations that possess special structures. For convenience, we propose the notations of some specific sets, that will be extensively used for describing the scheme.

1. Given a binary string $d = d[1] \dots d[\ell] \in \{0,1\}^\ell$, we propose the following notations:

   - $\mathsf{Secret}(d)$: The set of all vectors $\mathbf{x} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \mathbb{Z}^{(2\ell+1)m}$ consisting of $2\ell+1$ blocks of size m, such that $\|\mathbf{x}\|_\infty \leq \beta$, and the following ℓ blocks are zero-blocks 0^m: $\mathbf{x}_1^{1-d[1]}, \dots, \mathbf{x}_\ell^{1-d[\ell]}$. Specifically, $\mathsf{Secret}(d)$ will serve as the set of all the potential valid secret keys of group user with index d in the scheme.

   - $\mathsf{SecretExt}(d)$: The set of all vectors

   $$\mathbf{z} = \left(\mathbf{z}_0 \| \mathbf{z}_1^0 \| \mathbf{z}_1^1 \| \dots \| \mathbf{z}_\ell^0 \| \mathbf{z}_\ell^1\right) \in \{-1,0,1\}^{(2\ell+1)3m}$$

   consisting of $2\ell+1$ blocks of size $3m$, such that the $\ell+1$ blocks $\mathbf{z}_0, \mathbf{z}_1^{d[1]}, \dots, \mathbf{z}_\ell^{d[\ell]}$ are elements of B_{3m}, and the remaining ℓ blocks $\mathbf{z}_1^{1-d[1]}, \dots, \mathbf{z}_\ell^{1-d[\ell]}$ are zero-blocks 0^{3m}. Specifically, assuming that $\mathbf{x} \in \mathsf{Secret}(d)$, then after applying the Decomposition-Extension technique (see below) to $\mathbf{x}$, one will obtain $\lfloor \log \beta \rfloor + 1$ vectors $\mathbf{z}_j \in \mathsf{SecretExt}(d)$.

   - Let $\mathbf{y} = (\mathbf{y}_0 \| \mathbf{y}_1^0 \| \mathbf{y}_1^1 \| \dots \| \mathbf{y}_\ell^0 \| \mathbf{y}_\ell^1)$ be a $\{0,1\}$ vector in dimension $k = (2\ell+1) \cdot k_0$, which consists of $2\ell+1$ blocks of size k_0. We say that $\mathbf{y} = \mathsf{IdExt}(d)$ if the followings hold: $\mathbf{y}_0 = 1^{k_0}$ and for all $i \in [\ell]$: $\mathbf{y}_i^{d[i]} = 1^{k_0}$; $\mathbf{y}_i^{1-d[i]} = 0^{k_0}$.

     In the scheme, to encrypt d, we first map it to $\mathsf{IdExt}(d)$. It can be checked that, there is a bijection between the set $\{0,1\}^\ell$, and the set $\left\{\mathsf{IdExt}(d) \,\middle|\, d \in \{0,1\}^\ell\right\}$.

2. Given a vector $\mathbf{z} = \left(\mathbf{z}_0 \| \mathbf{z}_1^0 \| \mathbf{z}_1^1 \| \dots \| \mathbf{z}_\ell^0 \| \mathbf{z}_\ell^1\right) \in \mathbb{Z}^{(2\ell+1)3m}$ consisting of $2\ell+1$ blocks of

size $3m$, we define the following two sets of permutations of $\mathbf{z}$:

- The set $\mathcal{S}$ of all permutations that keep the arrangement of the blocks. Specifically, if $\pi \in \mathcal{S}$, then

$$\pi(\mathbf{z}) = \big(\tau_0(\mathbf{z}_0)\|\tau_1^0(\mathbf{z}_1^0)\|\tau_1^1(\mathbf{z}_1^1)\|\ldots\|\tau_\ell^0(\mathbf{z}_\ell^0)\|\tau_\ell^1(\mathbf{z}_\ell^1)\big),$$

  where $\tau_0, \tau_1^0, \tau_1^1, \ldots, \tau_\ell^0, \tau_\ell^1$ are certain permutations of $3m$ elements.

- The set $\mathcal{T} = \{\mathrm{T}_e \big| \, e \in \{0,1\}^\ell\}$, where for $e = e[1]\ldots e[\ell]$, $\mathrm{T}_e \in \mathcal{T}$ *rearranges* the blocks as follows:

$$\mathrm{T}_e(\mathbf{z}) = \big(\mathbf{z}_0\|\mathbf{z}_1^{e[1]}\|\mathbf{z}_1^{1-e[1]}\|\ldots\|\mathbf{z}_\ell^{e[\ell]}\|\mathbf{z}_\ell^{1-e[\ell]}\big).$$

3. Given a vector $\mathbf{y} = (\mathbf{y}_0\|\mathbf{y}_1^0\|\mathbf{y}_1^1\|\ldots\|\mathbf{y}_\ell^0\|\mathbf{y}_\ell^1) \in \mathbb{Z}_q^k$ consisting of $2\ell+1$ blocks of size k_0, we define the set $\mathcal{T}' = \{\mathrm{T}'_e \big| \, e \in \{0,1\}^\ell\}$ of permutations of $\mathbf{y}$, such that for $e = e[1]\ldots e[\ell]$, $\mathrm{T}'_e \in \mathcal{T}'$ *rearranges* the blocks of $\mathbf{y}$ as follows:

$$\mathrm{T}'_e(\mathbf{y}) = \big(\mathbf{y}_0\|\mathbf{y}_1^{e[1]}\|\mathbf{y}_1^{1-e[1]}\|\ldots\|\mathbf{y}_\ell^{e[\ell]}\|\mathbf{y}_\ell^{1-e[\ell]}\big).$$

In particular, given $d, e \in \{0,1\}^\ell$, $\pi \in \mathcal{S}$; $\mathbf{z} \in \mathbb{Z}^{(2\ell+1)3m}$, and a k-dimensional vector $\mathbf{y}$, it can be checked that:

$$\begin{cases} \mathbf{z} \in \mathsf{SecretExt}(d) \Leftrightarrow \pi(\mathbf{z}) \in \mathsf{SecretExt}(d) \Leftrightarrow \mathrm{T}_e \circ \pi(\mathbf{z}) \in \mathsf{SecretExt}(d \oplus e), \\ \mathbf{y} = \mathsf{IdExt}(d) \Leftrightarrow \mathrm{T}'_e(\mathbf{y}) = \mathsf{IdExt}(d \oplus e). \end{cases} \tag{6.1}$$

The Decomposition-Extension Technique

Adapting the Decomposition-Extension technique in Section 3.3.2, we construct the following procedures:

- WITNESS DECOMPOSITION AND EXTENSION. On input $\mathbf{x} \in \mathsf{Secret}(d)$ for some $d = d[1] \dots d[\ell] \in \{0,1\}^{\ell}$, the procedure $\mathsf{WitnessDE}$ outputs p vectors $\mathbf{z}_1, \dots \mathbf{z}_p \in \mathsf{SecretExt}(d)$. This procedure works as follows:

  1. Write vector $\mathbf{x}$ as the concatenation of $2\ell + 1$ blocks of size m, namely: $\mathbf{x} = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\big)$.
  2. Run $\mathsf{VectorDE}_{m,\beta}$ on each of the $\ell + 1$ blocks $\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \dots, \mathbf{x}_\ell^{d[\ell]}$ to obtain $(\ell + 1)p$ vectors in B_{3m}, denoted respectively by
  $$\{\mathbf{w}_{0,j}\}_{j=1}^{p}, \{\mathbf{w}_{1,j}^{d[1]}\}_{j=1}^{p}, \dots, \{\mathbf{w}_{\ell,j}^{d[\ell]}\}_{j=1}^{p}.$$
  3. Create ℓp *zero-vectors* of dimension $3m$, and denote them by
  $$\{\mathbf{w}_{1,j}^{1-d[1]}\}_{j=1}^{p}, \dots, \{\mathbf{w}_{\ell,j}^{1-d[\ell]}\}_{j=1}^{p}.$$
  4. For each $j \in [p]$, let $\mathbf{z}_j = \big(\mathbf{w}_{0,j} \| \mathbf{w}_{1,j}^0 \| \mathbf{w}_{1,j}^1 \| \dots \| \mathbf{w}_{\ell,j}^0 \| \mathbf{w}_{\ell,j}^1\big)$.
  5. Output $\mathbf{z}_1, \dots, \mathbf{z}_p \in \mathsf{SecretExt}(d)$.

- MATRIX EXTENSION. On input $\mathbf{A} \in \mathbb{Z}_q^{n \times (2\ell+1)m}$, the following procedure $\mathsf{GS\text{-}MatrixExt}$ outputs $\mathbf{A}^* \in \mathbb{Z}_q^{n \times (2\ell+1)3m}$:

  1. Write $\mathbf{A}$ as the concatenation of $2\ell + 1$ component-matrices in $\mathbb{Z}_q^{n \times m}$:
  $$\mathbf{A} = [\mathbf{A}_0 \big| \mathbf{A}_1^0 \big| \mathbf{A}_1^1 \big| \dots \big| \mathbf{A}_\ell^0 \big| \mathbf{A}_\ell^1].$$
  2. Append $2m$ *zero-columns* to each of the component-matrices to obtain $\overline{\mathbf{A}}_0, \overline{\mathbf{A}}_1^0, \overline{\mathbf{A}}_1^1, \dots, \overline{\mathbf{A}}_\ell^0, \overline{\mathbf{A}}_\ell^1$, respectively.
  3. Output $\mathbf{A}^* = [\overline{\mathbf{A}}_0 \big| \overline{\mathbf{A}}_1^0 \big| \overline{\mathbf{A}}_1^1 \big| \dots \big| \overline{\mathbf{A}}_\ell^0 \big| \overline{\mathbf{A}}_\ell^1]$.

In particular, let $\{\mathbf{z}_j\}_{j=1}^{p} \leftarrow \mathsf{WitnessDE}(\mathbf{x})$ and $\mathbf{A}^* \leftarrow \mathsf{GS\text{-}MatrixExt}(\mathbf{A})$ then we have $\mathbf{A} \cdot \mathbf{x} = \mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j) \bmod q$. Therefore, in the protocol in Section 6.5, in order to prove that $\mathbf{x} \in \mathsf{Secret}(d)$ for some $d \in \{0,1\}^\ell$, and $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$, one can instead prove that $\mathbf{A}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j) = \mathbf{u} \bmod q$, and that

$$\forall j \in [p], \forall \pi_j \in \mathcal{S}, \forall e \in \{0,1\}^\ell : \mathrm{T}_e \circ \pi(\mathbf{z}_j) \in \mathsf{SecretExt}(d \oplus e), \tag{6.2}$$

where (6.2) follows from the fact that $\mathbf{z}_j \in \mathsf{SecretExt}(d)$ for all $j \in [p]$, and from (6.1).

6.5.2 The Interactive Protocol

The underlying proof system for our lattice-based group signature scheme is a ZKPoK for the relation $\mathrm{R}_{\mathrm{LGS}}$ defined as follows.

Definition 6.5.1.

$$\mathrm{R}_{\mathrm{LGS}} = \Big\{ \big((\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c}), \mathbf{x}, \mathbf{y}, \mathbf{e}, \mathbf{s}, d\big) \in (\mathbb{Z}_q^{n \times (2\ell+1)m} \times \mathbb{Z}_q^{n \times k} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^k) \times \mathbb{Z}^{(2\ell+1)m} \times$$
$$\times \{0,1\}^k \times \mathbb{Z}_q^k \times \mathbb{Z}_q^n \times \{0,1\}^\ell : (\mathbf{x} \in \mathsf{Secret}(d)) \wedge (\mathbf{y} = \mathsf{IdExt}(d)) \wedge (\|\mathbf{e}\|_\infty \leq \beta)$$
$$(\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q) \wedge (\mathbf{B}^T \cdot \mathbf{s} + 2 \cdot \mathbf{e} + \mathbf{y} = \mathbf{c} \bmod q) \Big\}.$$

The Interactive Protocol.

In the interactive protocol, the common input is $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c})$, and prover's auxiliary input is $(\mathbf{x}, \mathbf{y}, \mathbf{e}, \mathbf{s}, d)$. Roughly speaking, there are 3 facts that prover has to convince the verifier in zero-knowledge simultaneously:

1. The prover knows $\mathbf{x} \in \mathbb{Z}^{(2\ell+1)m}$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$. This can be done by using the **SternExt** proof system for the ISIS^∞ problem (in Section 3.3). Prior to the interaction, both prover and verifier create the extended matrix $\mathbf{A}^* \leftarrow \mathsf{GS\text{-}MatrixExt}(\mathbf{A})$; while the prover performs the Decomposition-

Extension technique in Section 6.5.1 on $\mathbf{x}$: Let $\{\mathbf{z}_j\}_{j=1}^{p} \leftarrow \mathsf{WitnessDE}(\mathbf{x})$. In the protocol, the prover convinces the verifier that $\mathbf{A}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{z}_j) = \mathbf{u} \bmod q$.

2. The prover knows $\mathbf{s}$, $\mathbf{e}$, and $\mathbf{y}$ such that $\mathbf{B^T} \cdot \mathbf{s} + \mathbf{2} \cdot \mathbf{e} + \mathbf{y} = \mathbf{c} \bmod q$ and $\|\mathbf{e}\|_\infty \leq \beta$. This can be done by adapting the technique used in the proof of plaintext knowledge for the Dual-Regev encryption scheme (see Section 4.3). To realize this, assume that the common input implicitly includes the matrix $\mathbf{E}^* = \left[2 \cdot \mathbf{I}_k \,\middle|\, 0^{k \times 2k}\right] \in \{0,2\}^{k \times 3k}$, where $\mathbf{I}_k$ is the identity matrix of order k, and $0^{k \times 2k}$ denotes the zero-matrix with k rows and $2k$ columns. Prior to the interaction, the prover applies the Decomposition-Extension technique in Section 3.3.2 to vector $\mathbf{e}$: Let $\{\mathbf{f}_j\}_{j=1}^{p} \leftarrow \mathsf{VectorDE}_{k,\beta}(\mathbf{e})$. In the protocol, the prover convinces the verifier that

$$\mathbf{B}^T \cdot \mathbf{s} + \mathbf{E}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{f}_j) + \mathbf{y} = \mathbf{c} \bmod q \quad \text{and} \quad \forall j \in [p] : \mathbf{f}_j \in \mathsf{B}_{3k}.$$

3. $\mathbf{x} \in \mathsf{Secret}(d)$ and $\mathbf{y} = \mathsf{IdExt}(d)$ for some (secret) $d \in \{0,1\}^\ell$. Namely, the following conditions must simultaneously hold: $\|\mathbf{x}\|_\infty \leq \beta$; $\|\mathbf{y}\|_\infty = 1$; and $\mathbf{x}$ and $\mathbf{y}$ have the "same internal structure", i.e., their zero-blocks are arranged in the same fashion, determined by some string d, while the non-zero blocks of $\mathbf{y}$ are all 1^{k_0}. It follows from (6.1) and (6.2), that, it can be done by proving that: $\forall j \in [p], \forall \pi_j \in \mathcal{S}, \forall e \in \{0,1\}^\ell$:

$$\mathrm{T}_e \circ \pi_j(\mathbf{z}_j) \in \mathsf{SecretExt}(d \oplus e) \text{ and } \mathrm{T}'_e(\mathbf{y}) = \mathsf{IdExt}(d \oplus e);$$

Putting everything together, we obtain a proof system for the relation $\mathrm{R_{LGS}}$. Let COM be the KTX string commitment scheme [85], which is statistically hiding and computationally binding. The protocol is as follows:

1. **Commitment.** Prover P samples

$$\begin{cases} e \in \{0,1\}^{\ell};\ \mathbf{g} \xleftarrow{\$} \mathbb{Z}_q^n;\ \mathbf{r}_{\mathbf{y}} \xleftarrow{\$} \mathbb{Z}_q^k; \\ \mathbf{r}_{\mathbf{z}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{z}}^{(p)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m};\ \mathbf{r}_{\mathbf{f}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{f}}^{(p)} \xleftarrow{\$} \mathbb{Z}_q^{3k}; \\ \pi_1, \ldots, \pi_p \xleftarrow{\$} \mathcal{S};\ \sigma_1, \ldots, \sigma_p \xleftarrow{\$} \mathsf{S}_{3k}, \end{cases}$$

and sends the commitment $\mathrm{CMT} := (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, where

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\Big(e, \{\pi_j\}_{j=1}^p, \{\sigma_j\}_{j=1}^p, \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{\mathbf{z}}^{(j)}\big) \bmod q, \\ \qquad\qquad \mathbf{B}^T \cdot \mathbf{g} + \mathbf{E}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{\mathbf{f}}^{(j)}\big) + \mathbf{r}_{\mathbf{y}} \bmod q\Big) \\ \mathbf{c}_2 = \mathrm{COM}\Big(\{\mathrm{T}_e \circ \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p,\ \{\sigma_j(\mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_e(\mathbf{r}_{\mathbf{y}})\Big) \\ \mathbf{c}_3 = \mathrm{COM}\Big(\{\mathrm{T}_e \circ \pi_j(\mathbf{z}_j + \mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p,\ \{\sigma_j(\mathbf{f}_j + \mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_e(\mathbf{y} + \mathbf{r}_{\mathbf{y}})\Big). \end{cases}$$

2. **Challenge.** Receiving CMT, V sends a challenge $Ch \xleftarrow{\$} \{1,2,3\}$ to P.

3. **Response.** Prover P replies as follows:

   - If $Ch = 1$, then reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. Let $d_1 = d \oplus e$, $\mathbf{v}_{\mathbf{y}} = \mathrm{T}'_e(\mathbf{y})$; $\mathbf{w}_{\mathbf{y}} = \mathrm{T}'_e(\mathbf{r}_{\mathbf{y}})$, and for each $j \in [p]$ let

$$\begin{cases} \mathbf{v}_{\mathbf{z}}^{(j)} = \mathrm{T}_e \circ \pi_j(\mathbf{z}_j);\ \mathbf{w}_{\mathbf{z}}^{(j)} = \mathrm{T}_e \circ \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)}) \\ \mathbf{v}_{\mathbf{f}}^{(j)} = \sigma_j(\mathbf{f}_j);\ \mathbf{w}_{\mathbf{f}}^{(j)} = \sigma_j(\mathbf{r}_{\mathbf{f}}^{(j)}). \end{cases}$$

     Send

$$\mathrm{RSP} = \big(d_1, \mathbf{v}_{\mathbf{y}}, \mathbf{w}_{\mathbf{y}}, \{\mathbf{v}_{\mathbf{z}}^{(j)}\}_{j=1}^p, \{\mathbf{w}_{\mathbf{z}}^{(j)}\}_{j=1}^p, \{\mathbf{v}_{\mathbf{f}}^{(j)}\}_{j=1}^p, \{\mathbf{w}_{\mathbf{f}}^{(j)}\}_{j=1}^p\big). \tag{6.3}$$

- If $Ch = 2$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. Let

$$\begin{cases} d_2 = e;\ \mathbf{g}_2 = \mathbf{s} + \mathbf{g};\ \mathbf{t}_\mathbf{y} = \mathbf{y} + \mathbf{r}_\mathbf{y} \\ \forall j \in [p]:\ \phi_j = \pi_j;\ \rho_j = \sigma_j;\ \mathbf{t}_\mathbf{z}^{(j)} = \mathbf{z}_j + \mathbf{r}_\mathbf{z}^{(j)};\ \mathbf{t}_\mathbf{f}^{(j)} = \mathbf{f}_j + \mathbf{r}_\mathbf{f}^{(j)}. \end{cases}$$

  Send

$$\text{RSP} = \big(d_2, \mathbf{g}_2, \mathbf{t}_\mathbf{y}, \{\phi_j\}_{j=1}^p, \{\rho_j\}_{j=1}^p, \{\mathbf{t}_\mathbf{z}^{(j)}\}_{j=1}^p, \{\mathbf{t}_\mathbf{f}^{(j)}\}_{j=1}^p\big). \tag{6.4}$$

- If $Ch = 3$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. Let

$$\begin{cases} d_3 = e;\ \mathbf{g}_3 = \mathbf{g};\ \mathbf{h}_\mathbf{y} = \mathbf{r}_\mathbf{y} \\ \forall j \in [p]:\ \psi_j = \pi_j;\ \xi_j = \sigma_j;\ \mathbf{h}_\mathbf{z}^{(j)} = \mathbf{r}_\mathbf{z}^{(j)};\ \mathbf{h}_\mathbf{f}^{(j)} = \mathbf{r}_\mathbf{f}^{(j)}. \end{cases}$$

  Send

$$\text{RSP} = \big(d_3, \mathbf{g}_3, \mathbf{h}_\mathbf{y}, \{\psi_j\}_{j=1}^p, \{\xi_j\}_{j=1}^p, \{\mathbf{h}_\mathbf{z}^{(j)}\}_{j=1}^p, \{\mathbf{h}_\mathbf{f}^{(j)}\}_{j=1}^p\big). \tag{6.5}$$

Verification. Receiving the response RSP, verifier V performs the following checks:

- If $Ch = 1$: Parse RSP as in (6.3). Check that $\mathbf{v}_\mathbf{y} = \mathsf{IdExt}(d_1)$; and for all $j \in [p]$: $\mathbf{v}_\mathbf{f}^{(j)} \in \mathsf{B}_{3k}$, $\mathbf{v}_\mathbf{z}^{(j)} \in \mathsf{SecretExt}(d_1)$; and

$$\begin{cases} \mathbf{c}_2 = \text{COM}\Big(\{\mathbf{w}_\mathbf{z}^{(j)}\}_{j=1}^p,\ \{\mathbf{w}_\mathbf{f}^{(j)}\}_{j=1}^p,\ \mathbf{w}_\mathbf{y}\Big) \\ \mathbf{c}_3 = \text{COM}\Big(\{\mathbf{v}_\mathbf{z}^{(j)} + \mathbf{w}_\mathbf{z}^{(j)}\}_{j=1}^p,\ \{\mathbf{v}_\mathbf{f}^{(j)} + \mathbf{w}_\mathbf{f}^{(j)}\}_{j=1}^p,\ \mathbf{v}_\mathbf{y} + \mathbf{w}_\mathbf{y}\Big) \end{cases}$$

- If $Ch = 2$: Parse RSP as in (6.4). Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\Big(d_2, \{\phi_j\}_{j=1}^p, \{\rho_j\}_{j=1}^p,\ \mathbf{A}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{t_z}^{(j)}) - \mathbf{u} \bmod q, \\ \qquad\qquad \mathbf{B}^T \cdot \mathbf{g}_2 + \mathbf{E}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{t_f}^{(j)}) + \mathbf{t_y} - \mathbf{c} \bmod q\Big) \\ \mathbf{c}_3 = \mathrm{COM}\Big(\{\mathrm{T}_{d_2} \circ \phi_j(\mathbf{t}_\mathbf{z}^{(j)})\}_{j=1}^p,\ \{\rho_j(\mathbf{t}_\mathbf{f}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_{d_2}(\mathbf{t_y})\Big). \end{cases}$$

- If $Ch = 3$: Parse RSP as in (6.5). Check that

$$\begin{cases} \mathbf{c}_1 = \mathrm{COM}\Big(d_3, \{\psi_j\}_{j=1}^p, \{\xi_j\}_{j=1}^p, \mathbf{A}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{h_z}^{(j)}) \bmod q, \\ \qquad\qquad \mathbf{B}^T \cdot \mathbf{g}_3 + \mathbf{E}^* \cdot (\sum_{j=1}^p \beta_j \cdot \mathbf{h_f}^{(j)} + \mathbf{h_y} \bmod q) \\ \mathbf{c}_2 = \mathrm{COM}\Big(\{\mathrm{T}_{d_3} \circ \psi_j(\mathbf{h}_\mathbf{z}^{(j)})\}_{j=1}^p,\ \{\xi_j(\mathbf{h}_\mathbf{f}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_{d_3}(\mathbf{h_y})\Big). \end{cases}$$

In each case, verifier V outputs the decision 1 (Accept) if and only if all the conditions hold. Otherwise, he outputs 0 (Reject).

We first summarize the properties of the above proof system in the following theorem.

Theorem 6.5.2. *The given proof system is a statistical* ZKPoK *for the relation* $\mathrm{R_{LGS}}$*, where each round of the protocol has perfect completeness, knowledge error* $2/3$*, and communication cost* $p \cdot \log q \cdot (\widetilde{\mathcal{O}}(\ell \cdot m) + \widetilde{\mathcal{O}}(k))$.

Completeness. We observe that, if the honest prover, who has a valid witness for the relation $\mathrm{R_{LGS}}$, follows the protocol, then he is always accepted by the verifier. Therefore, the proof system has perfect completeness.

Communication cost. The commitment CMT has bit-size $3n \log q$. The size of the response RSP is bounded by the sizes of a string in $\{0,1\}^\ell$; two vectors in $\mathbb{Z}_q^k$; $2p$ vectors in $\mathbb{Z}_q^{(2\ell+1)3m}$; $2p$ vectors in $\mathbb{Z}_q^{3k}$; p permutations of $(2\ell+1)3m$ elements and p

permutations of $3k$ elements. Hence, the overall cost is $p \cdot \log q \cdot (\widetilde{\mathcal{O}}(\ell \cdot m) + \widetilde{\mathcal{O}}(k))$. If we evaluate the efficiency of the protocol based on two primary parameters ℓ and n, then, given our parameters setting, the interaction transcript has bit-size $\widetilde{\mathcal{O}}(n \cdot \ell)$.

Statistical Zero-knowledge

We now prove that the given proof system is statistically zero-knowledge.

Lemma 6.5.3. *If* COM *is a statistically hiding string commitment scheme, then the given proof system for* $\mathrm{R}_{\mathrm{LGS}}$ *is statistically zero-knowledge. In particular, there exists an efficient simulator, which has black-box access to a (possibly cheating) verifier* $\widehat{V}$*, such that on input* $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c})$ *(and implicitly,* $\mathbf{E}^*$*), outputs with probability* $2/3$ *a successful transcript, the distribution of which is statistically close to that in the real interaction.*

Proof. We adapt the simulation technique for the **SternExt** proof system to construct a simulator satisfying the conditions of the lemma. The simulator begins by selecting a random $\overline{Ch} \in \{1, 2, 3\}$ (a prediction of the challenge value that $\widehat{V}$ will *not* choose), and a random tape r' of $\widehat{V}$.

Case $\overline{Ch} = 1$: The simulator proceeds as follows:

1. Compute $\mathbf{z}'_1, \ldots, \mathbf{z}'_p \in \mathbb{Z}_q^{(2\ell+1)3m}$ such that $\mathbf{A}^* \cdot (\sum_{j=1}^{p} \cdot \beta_j \cdot \mathbf{z}'_j) = \mathbf{u} \bmod q$. This can efficiently be done using linear algebra.

2. Pick $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$, $\mathbf{f}'_1, \ldots, \mathbf{f}'_p \xleftarrow{\$} \mathbb{Z}_q^{3k}$, and compute $\mathbf{y}' = \mathbf{c} - \mathbf{B}^T \cdot \mathbf{s}' - \mathbf{E}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{f}'_j) \bmod q$, so that $\mathbf{B}^T \cdot \mathbf{s}' + \mathbf{E}^* \cdot (\sum_{j=1}^{p} \beta_j \cdot \mathbf{f}'_j) + \mathbf{y}' = \mathbf{c} \bmod q$.

3. Sample

$$\begin{cases} \rho'_1, \rho'_2, \rho'_3 \xleftarrow{\$} \{0,1\}^n;\ d', e \in \{0,1\}^\ell;\ \mathbf{g} \xleftarrow{\$} \mathbb{Z}_q^n;\ \mathbf{r}_{\mathbf{y}} \xleftarrow{\$} \mathbb{Z}_q^k; \\ \mathbf{r}_{\mathbf{z}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{z}}^{(p)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m};\ \mathbf{r}_{\mathbf{f}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{f}}^{(p)} \xleftarrow{\$} \mathbb{Z}_q^{3k}; \\ \pi_1, \ldots, \pi_p \xleftarrow{\$} \mathcal{S};\ \sigma_1, \ldots, \sigma_p \xleftarrow{\$} \mathsf{S}_{3k}, \end{cases}$$

and send the commitment $\mathrm{CMT} := (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$, where

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\Big(e, \{\pi_j\}_{j=1}^p, \{\sigma_j\}_{j=1}^p, \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{\mathbf{z}}^{(j)}\big) \bmod q, \\ \qquad\qquad \mathbf{B}^T \cdot \mathbf{g} + \mathbf{E}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{\mathbf{f}}^{(j)}\big) + \mathbf{r}_{\mathbf{y}} \bmod q; \rho'_1\Big) \\ \mathbf{c}'_2 = \mathrm{COM}\Big(\{\mathrm{T}_e \circ \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p,\ \{\sigma_j(\mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_e(\mathbf{r}_{\mathbf{y}}); \rho'_2\Big) \\ \mathbf{c}'_3 = \mathrm{COM}\Big(\{\mathrm{T}_e \circ \pi_j(\mathbf{z}'_j + \mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p,\ \{\sigma_j(\mathbf{f}'_j + \mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_e(\mathbf{y}' + \mathbf{r}_{\mathbf{y}}); \rho'_3\Big). \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$:

- If $Ch = 1$: Output $\perp$ and abort.
- If $Ch = 2$: Set

$$\mathrm{RSP} := (d' \oplus e, \mathbf{s}' + \mathbf{g}, \mathbf{y}' + \mathbf{r}_{\mathbf{y}}, \{\pi_j\}_{j=1}^p, \{\sigma_j\}_{j=1}^p, \{\mathbf{z}'_j + \mathbf{r}_{\mathbf{z}}^{(j)}\}_{j=1}^k, \{\mathbf{f}'_j + \mathbf{r}_{\mathbf{f}}^{(j)}\}_{j=1}^p),$$

  and outputs $(r'; \mathrm{CMT}; 2; \mathrm{RSP}; \rho'_1, \rho'_3)$.

- If $Ch = 3$: Set $\mathrm{RSP} := (e, \mathbf{g}, \mathbf{r}_{\mathbf{y}}, \{\pi_j\}_{j=1}^p, \{\sigma_j\}_{j=1}^p, \{\mathbf{r}_{\mathbf{z}}^{(j)}\}_{j=1}^p, \{\mathbf{r}_{\mathbf{f}}^{(j)}\}_{j=1}^p)$, and output $\big(r'; \mathrm{CMT}; 3; \mathrm{RSP}; \rho'_1, \rho'_2\big)$.

Case $\overline{Ch} = 2$: The simulator samples

$$\begin{cases} \rho'_1, \rho'_2, \rho'_3 \xleftarrow{\$} \{0,1\}^n;\ d', e \in \{0,1\}^\ell;\ \mathbf{g} \xleftarrow{\$} \mathbb{Z}_q^n;\ \mathbf{r}_{\mathbf{y}} \xleftarrow{\$} \mathbb{Z}_q^k; \\ \mathbf{z}'_1, \ldots, \mathbf{z}'_p \xleftarrow{\$} \mathsf{SecretExt}(d');\ \mathbf{r}_{\mathbf{z}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{z}}^{(p)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1)\cdot 3m}; \\ \mathbf{f}'_1, \ldots, \mathbf{f}'_p \xleftarrow{\$} \mathsf{B}_{3k};\ \mathbf{r}_{\mathbf{f}}^{(1)}, \ldots, \mathbf{r}_{\mathbf{f}}^{(p)} \xleftarrow{\$} \mathbb{Z}_q^{3k}; \\ \pi_1, \ldots, \pi_p \xleftarrow{\$} \mathcal{S};\ \sigma_1, \ldots, \sigma_p \xleftarrow{\$} \mathsf{S}_{3k}. \end{cases}$$

It then sets $\mathbf{y}' = \mathsf{IdExt}(d')$, and sends the commitment $\mathrm{CMT} := (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$, where

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\Big(e, \{\pi_j\}_{j=1}^p, \{\sigma_j\}_{j=1}^p, \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{\mathbf{z}}^{(j)}\big) \bmod q, \\ \qquad\qquad \mathbf{B}^T \cdot \mathbf{g} + \mathbf{E}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot \mathbf{r}_{\mathbf{f}}^{(j)}\big) + \mathbf{r}_{\mathbf{y}} \bmod q; \rho'_1\Big) \\ \mathbf{c}'_2 = \mathrm{COM}\Big(\{\mathrm{T}_e \circ \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p,\ \{\sigma_j(\mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_e(\mathbf{r}_{\mathbf{y}}); \rho'_2\Big) \\ \mathbf{c}'_3 = \mathrm{COM}\Big(\{\mathrm{T}_e \circ \pi_j(\mathbf{z}'_j + \mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p,\ \{\sigma_j(\mathbf{f}'_j + \mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_e(\mathbf{y}' + \mathbf{r}_{\mathbf{y}}); \rho'_3\Big). \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$:

- If $Ch = 1$: Let RSP be the tuple:

  $$(d' \oplus e, \mathrm{T}'_e(\mathbf{y}'), \mathrm{T}'_e(\mathbf{r}_{\mathbf{y}}), \{\mathrm{T}_e \circ \pi_j(\mathbf{z}'_j)\}_{j=1}^p, \{\mathrm{T}_e \circ \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p, \{\sigma_j(\mathbf{f}'_j)\}_{j=1}^p, \{\sigma_j(\mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p),$$

  and output $\big(r'; \mathrm{CMT}; 1; \mathrm{RSP}; \rho'_2, \rho'_3\big)$.

- If $Ch = 2$: Output $\perp$ and abort.

- If $Ch = 3$: Set $\mathrm{RSP} := (e, \mathbf{g}, \mathbf{r}_{\mathbf{y}}, \{\pi_j\}_{j=1}^p, \{\sigma_j\}_{j=1}^p, \{\mathbf{r}_{\mathbf{z}}^{(j)}\}_{j=1}^p, \{\mathbf{r}_{\mathbf{f}}^{(j)}\}_{j=1}^p)$, and output $\big(r'; \mathrm{CMT}; 3; \mathrm{RSP}; \rho'_1, \rho'_2\big)$.

Case $\overline{Ch} = 3$: The simulator proceeds the preparation as in the case $\overline{Ch} = 2$ above, and additionally picks $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$. Then it sends the commitment $\mathrm{CMT} := (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$,

where:

$$\begin{cases} \mathbf{c}'_1 = \mathrm{COM}\Big(e, \{\pi_j\}_{j=1}^p, \{\sigma_j\}_{j=1}^p, \mathbf{A}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot (\mathbf{z}'_j + \mathbf{r}_{\mathbf{z}}^{(j)})\big) - \mathbf{u} \bmod q, \\ \qquad\qquad \mathbf{B}^T \cdot (\mathbf{s}' + \mathbf{g}) + \mathbf{E}^* \cdot \big(\sum_{j=1}^p \beta_j \cdot (\mathbf{f}'_j + \mathbf{r}_{\mathbf{f}}^{(j)})\big) + (\mathbf{y}' + \mathbf{r}_{\mathbf{y}}) - \mathbf{c}' \bmod q; \rho'_1\Big) \\ \mathbf{c}'_2 = \mathrm{COM}\Big(\{\mathrm{T}_e \circ \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p,\ \{\sigma_j(\mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_e(\mathbf{r}_{\mathbf{y}}); \rho'_2\Big) \\ \mathbf{c}'_3 = \mathrm{COM}\Big(\{\mathrm{T}_e \circ \pi_j(\mathbf{z}'_j + \mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p,\ \{\sigma_j(\mathbf{f}'_j + \mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p,\ \mathrm{T}'_e(\mathbf{y}' + \mathbf{r}_{\mathbf{y}}); \rho'_3\Big). \end{cases}$$

Receiving a challenge Ch from $\widehat{V}$:

- If $Ch = 1$: Let RSP be the tuple

$$(d' \oplus e, \mathrm{T}'_e(\mathbf{y}'), \mathrm{T}'_e(\mathbf{r}_{\mathbf{y}}), \{\mathrm{T}_e \circ \pi_j(\mathbf{z}'_j)\}_{j=1}^p, \{\mathrm{T}_e \circ \pi_j(\mathbf{r}_{\mathbf{z}}^{(j)})\}_{j=1}^p, \{\sigma_j(\mathbf{f}'_j)\}_{j=1}^p, \{\sigma_j(\mathbf{r}_{\mathbf{f}}^{(j)})\}_{j=1}^p),$$

  and output $\big(r'; \mathrm{CMT}; 1; \mathrm{RSP}; \rho'_2, \rho'_3\big)$.

- If $Ch = 2$: Set

$$\mathrm{RSP} := (d' \oplus e, \mathbf{s}' + \mathbf{g}, \mathbf{y}' + \mathbf{r}_{\mathbf{y}}, \{\pi_j\}_{j=1}^p, \{\sigma_j\}_{j=1}^p, \{\mathbf{z}'_j + \mathbf{r}_{\mathbf{z}}^{(j)}\}_{j=1}^k, \{\mathbf{f}'_j + \mathbf{r}_{\mathbf{f}}^{(j)}\}_{j=1}^p),$$

  and output $(r'; \mathrm{CMT}; 2; \mathrm{RSP}; \rho'_1, \rho'_3)$.

- If $Ch = 3$: Output $\perp$ and abort.

We observe that, in every case we have considered above, since COM is statistically hiding, the distribution of the commitment CMT and the challenge Ch from $\widehat{V}$ are statistically close to those in the real interaction. Hence, the probability that the simulator outputs $\perp$ is negligibly close to 1/3. Moreover, one can check that whenever the simulator does not halt, it will provide a successful transcript, and the

distribution of the transcript is statistically close to that of the prover in the real interaction. Hence, we have constructed a simulator that can successfully impersonate the honest prover with probability 2/3, and completed the proof. □

Proof of Knowledge.

The following lemma states that the given proof system is a proof of knowledge for the relation $\mathrm{R}_{\mathrm{LGS}}$ with knowledge error 2/3.

Lemma 6.5.4. *Assume that* COM *is a computationally binding string commitment scheme. Then there exists a knowledge extractor $\mathcal{K}$ such that the following holds. If $\mathcal{K}$ has access to a cheating prover who convinces the verifier on input $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c})$ (and implicitly, $\mathbf{E}^*$) with probability $2/3 + \epsilon$ for some $\epsilon > 0$ and in time T, then with overwhelming probability and in time $T \cdot \mathsf{poly}(n, m, k, \log q, 1/\epsilon)$, $\mathcal{K}$ outputs $(\mathbf{x}', \mathbf{y}', \mathbf{e}', \mathbf{s}', d')$ such that:*

$$((\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c}), \mathbf{x}', \mathbf{y}', \mathbf{e}', \mathbf{s}', d') \in \mathrm{R}_{\mathrm{LGS}}.$$

Proof. The proof uses the same argument as in the **SternExt** proof system (see Section 3.3.5). Namely, it can be shown that, in time $T \cdot \mathsf{poly}(n, m, k, \log q, 1/\epsilon)$, the knowledge extractor $\mathcal{K}$ can obtain with overwhelming probability 3 valid responses which correspond to all 3 challenges for the same commitment. Therefore, $\mathcal{K}$ can get the following relations:

1. $\mathrm{COM}\big(d_2, \{\phi_j\}_{j=1}^p, \{\rho_j\}_{j=1}^p, \mathbf{A}^*(\sum_{j=1}^p \beta_j \mathbf{t}_{\mathbf{z}}{}^{(j)}) - \mathbf{u}, \mathbf{B}^T\mathbf{g}_2 + \mathbf{E}^*(\sum_{j=1}^p \beta_j \mathbf{t}_{\mathbf{f}}{}^{(j)}) + \mathbf{t}_{\mathbf{y}} - \mathbf{c}\big) = \mathrm{COM}\big(d_3, \{\psi_j\}_{j=1}^p, \{\xi_j\}_{j=1}^p, \mathbf{A}^*(\sum_{j=1}^p \beta_j \mathbf{h}_{\mathbf{z}}{}^{(j)}), \mathbf{B}^T\mathbf{g}_3 + \mathbf{E}^*(\sum_{j=1}^p \beta_j \mathbf{h}_{\mathbf{f}}{}^{(j)} + \mathbf{h}_{\mathbf{y}})\big)$;
2. $\mathrm{COM}\big(\{\mathbf{w}_{\mathbf{z}}^{(j)}\}_{j=1}^p; \{\mathbf{w}_{\mathbf{f}}^{(j)}\}_{j=1}^p; \mathbf{w}_{\mathbf{y}}\big) = \mathrm{COM}\big(\{\mathrm{T}_{d_3} \circ \psi_j(\mathbf{h}_{\mathbf{z}}^{(j)})\}_{j=1}^p, \{\xi_j(\mathbf{h}_{\mathbf{f}}^{(j)})\}_{j=1}^p, \mathrm{T}'_{d_3}(\mathbf{h}_{\mathbf{y}})\big)$;
3. $\mathrm{COM}\big(\{\mathrm{T}_{d_2} \circ \phi_j(\mathbf{t}_{\mathbf{z}}^{(j)})\}_{j=1}^p, \{\rho_j(\mathbf{t}_{\mathbf{f}}^{(j)})\}_{j=1}^p, \mathrm{T}'_{d_2}(\mathbf{t}_{\mathbf{y}})\big) = \mathrm{COM}\big(\{\mathbf{v}_{\mathbf{z}}^{(j)} + \mathbf{w}_{\mathbf{z}}^{(j)}\}_{j=1}^p; \{\mathbf{v}_{\mathbf{f}}^{(j)} + \mathbf{w}_{\mathbf{f}}^{(j)}\}_{j=1}^p; \mathbf{v}_{\mathbf{y}} + \mathbf{w}_{\mathbf{y}}\big)$;

4. $\mathbf{v_y} = \mathsf{IdExt}(d_1)$; and for all $j \in [p]$: $\mathbf{v_f}^{(j)} \in \mathsf{B}_{3k}$, $\mathbf{v_z}^{(j)} \in \mathsf{SecretExt}(d_1)$.

Since COM is computationally binding, it then follows that:

$$\begin{cases} d_2 = d_3; \mathbf{w_y} = \mathrm{T}'_{d_3}(\mathbf{h_y}); \mathbf{v_y} + \mathbf{w_y} = \mathrm{T}'_{d_2}(\mathbf{t_y}); \mathbf{v_y} = \mathsf{IdExt}(d_1); \\ \forall j \in [p] : \phi_j = \psi_j; \mathbf{w_z}^{(j)} = \mathrm{T}_{d_3} \circ \psi_j(\mathbf{h_z}^{(j)}); \mathrm{T}_{d_2} \circ \phi_j(\mathbf{t_z}^{(j)}) = \mathbf{v_z}^{(j)} + \mathbf{w_z}^{(j)}; \\ \mathbf{v_z}^{(j)} \in \mathsf{SecretExt}(d_1);\ \rho_j = \xi_j; \mathbf{w_f}^{(j)} = \xi_j(\mathbf{h_f}^{(j)}); \mathbf{v_f}^{(j)} + \mathbf{w_f}^{(j)} = \rho_j(\mathbf{t_f}^{(j)}); \mathbf{v_f}^{(j)} \in \mathsf{B}_{3k}; \\ \mathbf{A}^*(\sum_{j=1}^p \beta_j {\mathbf{t_z}}^{(j)}) - \mathbf{u} = \mathbf{A}^*(\sum_{j=1}^p \beta_j {\mathbf{h_z}}^{(j)}) \bmod q; \\ \mathbf{B}^T\mathbf{g}_2 + \mathbf{E}^*(\sum_{j=1}^p \beta_j {\mathbf{t_f}}^{(j)}) + \mathbf{t_y} - \mathbf{c} = \mathbf{B}^T\mathbf{g}_3 + \mathbf{E}^*(\sum_{j=1}^p \beta_j {\mathbf{h_f}}^{(j)}) + \mathbf{h_y} \bmod q. \end{cases}$$

Now, the knowledge extractor proceeds as follows:

1. Let $d' = d_1 \oplus d_2$.

2. Let $\mathbf{y}' = \mathbf{t_y} - \mathbf{h_y}$. Then one has $\mathrm{T}'_{d_2}(\mathbf{y}') = \mathbf{v_y} = \mathsf{IdExt}(d_1)$, and it follows from (6.1) that $\mathbf{y}' = \mathsf{IdExt}(d_1 \oplus d_2) = \mathsf{IdExt}(d')$.

3. For each $j \in [p]$, let $\mathbf{z}'_j = \mathbf{t_z}^{(j)} - \mathbf{h_z}^{(j)}$. Then one has $\mathrm{T}_{d_2} \circ \phi_j(\mathbf{z}'_j) = \mathbf{v_z}^{(j)} \in \mathsf{SecretExt}(d_1)$, and, again, it follows from (6.1) that $\mathbf{z}'_j \in \mathsf{SecretExt}(d')$. On the other hand, $\mathbf{A}^*(\sum_{j=1}^p \beta_j \cdot \mathbf{z}'_j) = \mathbf{u} \bmod q$. Let $\widehat{\mathbf{x}}' = \sum_{j=1}^p \beta_j \cdot \mathbf{z}'_j \in \mathbb{Z}^{(2\ell+1)3m}$, then we have $\|\widehat{\mathbf{x}}'\|_\infty \leq \sum_{j=1}^p \beta_j \cdot \|\mathbf{z}'_j\|_\infty = \sum_{j=1}^p \beta_j = \beta$. Now, consider $\widehat{\mathbf{x}}'$ as a concatenation of $2\ell+1$ blocks of size $3m$, and drop the last $2m$ coordinates in each of the blocks to obtain $\mathbf{x}' = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1)m}$. We observe that, $\mathbf{A} \cdot \mathbf{x}' = \mathbf{u} \bmod q$, and $\|\mathbf{x}'\|_\infty \leq \beta$. Moreover, for all $i \in [\ell]$, the block $\mathbf{x}_i^{1-d'[i]}$ is zero-block 0^m. In other words, we have $\mathbf{x}' \in \mathsf{Secret}(d')$.

4. Let $\mathbf{s}' = \mathbf{g}_2 - \mathbf{g}_3$, and for all $j \in [p]$, let $\mathbf{f}'_j = \mathbf{t}_{\mathbf{f}}^{(j)} - \mathbf{h}_{\mathbf{f}}^{(j)}$. Then $\rho_j(\mathbf{f}'_j) = \mathbf{v}_{\mathbf{f}}^{(j)} \in \mathsf{B}_{3k}$, which means that $\mathbf{f}'_j \in \mathsf{B}_{3k}$. On the other hand, the following equation holds true:

$$\mathbf{B}^T \cdot \mathbf{s}' + \mathbf{E}^*(\sum_{j=1}^{p} \beta_j \cdot \mathbf{f}'_j) + \mathbf{y}' = \mathbf{c} \bmod q.$$

Let $\widehat{\mathbf{e}}' = \sum_{j=1}^{p} \beta_j \cdot \mathbf{f}'_j \in \mathbb{Z}_q^{3k}$, then $\|\widehat{\mathbf{e}}'\|_\infty \leq \sum_{j=1}^{p} \beta_j \cdot \|\mathbf{f}'_j\|_\infty = \sum_{j=1}^{p} \beta_j = \beta$. Now, if we drop the last $2k$ coordinates of $\widehat{\mathbf{e}}'$ to obtain $\mathbf{e}' \in \mathbb{Z}_q^k$, then $\|\mathbf{e}'\|_\infty \leq \beta$. Moreover, one has: $\mathbf{B}^T \cdot \mathbf{s}' + 2 \cdot \mathbf{e}' + \mathbf{y}' = \mathbf{c} \bmod q$.

The knowledge extractor then outputs $(\mathbf{x}', \mathbf{y}', \mathbf{e}', \mathbf{s}', d')$ as described above. It is easy to check that

$$((\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c}), \mathbf{x}', \mathbf{y}', \mathbf{e}', \mathbf{s}', d') \in \mathrm{R}_{\mathrm{LGS}}.$$

This concludes the proof. □

6.6 An Improved Lattice-based Group Signature Scheme

In this section, we present our group signature scheme $\mathcal{LGS}$. For convenience of presentation, we used the following modified notations:

1. Instead of indexing a group user by a number $\mathrm{id} \in \{1, \ldots, N\}$ as defined in Section 6.2, we equivalently use a binary string $d \in \{0,1\}^\ell$, which is the binary representation of $(\mathrm{id} - 1)$.

2. The secret key of the user is then denoted by $\mathsf{gsk}[d]$, instead of $\mathsf{gsk}[\mathrm{id}]$.

6.6.1 Description of the Scheme

The scheme $\mathcal{LGS}$ uses the parameters specified in Section 6.5.1, and a random oracle $\mathsf{H} : \{0,1\}^* \to \{1,2,3\}^t$, for some $t = \omega(\log n)$.

Key Generation Algorithm. The randomized algorithm $\mathsf{KeyGen}(1^n, 1^N)$ performs the following steps:

1. Run $\mathsf{GenTrap}(1^n, 1^m, q)$ to obtain a matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$, together with a short basis $\mathbf{S}$ of $\Lambda_q^{\perp}(\mathbf{A}_0)$.

2. For $i \in [\ell]$ and $b \in \{0,1\}$, sample $\mathbf{A}_i^b \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$, and define the matrix

$$\mathbf{A} = \left[\mathbf{A}_0 \middle| \mathbf{A}_1^0 \middle| \mathbf{A}_1^1 \middle| \ldots \middle| \mathbf{A}_\ell^0 \middle| \mathbf{A}_\ell^1\right] \in \mathbb{Z}_q^{n\times(2\ell+1)\cdot m}.$$

3. Run $\mathsf{GenTrap}(1^n, 1^k, q)$ to output a matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times k}$, together with a short basis $\mathbf{T}$ of $\Lambda_q^{\perp}(\mathbf{B})$.

4. Sample a vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$.

5. For group user with index $d = d[1]\ldots d[\ell] \in \{0,1\}^\ell$, perform the following steps:

   (a) Sample ℓ vectors $\mathbf{z}_1, \ldots, \mathbf{z}_\ell \hookleftarrow D_{\mathbb{Z}^m,\sigma}$. If there exists some $i \in [\ell]$, such that $\|\mathbf{z}_i\|_\infty > \beta$, then resample $\mathbf{z}_i$.

   (b) Compute vector $\hat{\mathbf{u}} = \mathbf{u} - \sum_{i=1}^{\ell} \mathbf{A}_i^{d[i]} \cdot \mathbf{z}_i \bmod q$.

   (c) Sample $\mathbf{x}_0 \hookleftarrow \mathsf{SampleD}\big(\mathbf{S}, \mathbf{A}_0, \hat{\mathbf{u}}, \sigma\big)$. If $\|\mathbf{x}_0\|_\infty > \beta$, then resample $\mathbf{x}_0$.

   (d) Define the secret signing key $\mathsf{gsk}[d]$ of the user as:

$$\mathsf{gsk}[d] = \big(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\big) \in \mathbb{Z}^{(2\ell+1)m},$$

   where for each $i \in [\ell]$: $\mathbf{x}_i^{d[i]} = \mathbf{z}_i$ and $\mathbf{x}_i^{1-d[i]}$ is the zero-vector 0^m.

6. Output the group public key $\mathsf{gpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$, the group manager's secret key $\mathsf{gmsk} = \mathbf{T}$, and the vector of users' secret keys $\mathsf{gsk} = \big(\{\mathsf{gsk}[d]\}_{d\in\{0,1\}^\ell}\big)$.

Remark 6.6.1. We have some observations on the behaviour of the above key generation algorithm $\mathsf{KeyGen}(1^n, 1^N)$:

- By Theorem 2.3.3, for parameter setting $m \geq 2n \log q$ (*resp.* $k \geq 2n \log q$), algorithm $\mathsf{GenTrap}(1^n, 1^m, q)$ (*resp.* $\mathsf{GenTrap}(1^n, 1^k, q)$) runs in polynomial time, and the distribution of the obtained matrix $\mathbf{A}_0$ (*resp.*, matrix $\mathbf{B}$) is statistically close to uniform over $\mathbb{Z}_q^{n \times m}$ (*resp.* $\mathbb{Z}_q^{n \times k}$). As a result, the distribution of the group public key $\mathsf{gpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$ is statistically close to uniform over $\mathbb{Z}_q^{n \times (2\ell+1)m} \times \mathbb{Z}_q^{n \times k} \times \mathbb{Z}_q^n$. In particular, the pair $(\mathbf{A}, \mathbf{u})$ resembles the Bonsai tree structures, while matrix $\mathbf{B}$ and its associated trapdoor $\mathbf{T}$ will be used for the encrypting and decrypting procedures in the signing and opening algorithms, respectively.

- By Theorem 2.3.4, for parameter setting $\sigma = \omega(\sqrt{n \log q \log n})$, the algorithm $\mathsf{SampleD}\big(\mathbf{S}, \mathbf{A}_0, \hat{\mathbf{u}}, \sigma\big)$ runs in polynomial time and produces $\mathbf{x}_0$ sampled from the distribution $D_{\mathbb{Z}^m, \sigma}$. We note that, for each $i \in [\ell]$, the block-vector $\mathbf{z}_i$ is also sampled from $D_{\mathbb{Z}^m, \sigma}$. Since $D_{\mathbb{Z}, \sigma}$ is a β-bounded distribution, the event that vectors $\mathbf{x}_0, \mathbf{z}_1, \ldots, \mathbf{z}_\ell$ have to be resampled only happens with negligible probability. Therefore, algorithm $\mathsf{KeyGen}(1^n, 1^N)$ runs in polynomial time.

- By construction, any valid user secret key $\mathbf{x} \in \mathbb{Z}^{(2\ell+1)m}$ satisfies $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$, and $\|\mathbf{x}\|_\infty \leq \beta$, and the zero-blocks of $\mathbf{x}$ are arranged according to the index of the user. In other words, one can set the requirements for a valid user key $\mathbf{x}$ as follows:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q \text{ and } \mathbf{x} \in \mathsf{Secret}(d) \text{ for some } d \in \{0,1\}^\ell.$$

Signing Algorithm. Given $\mathsf{gpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$, to sign a message $M \in \{0,1\}^*$ using the secret key $\mathsf{gsk}[d]$, the user runs the the randomized algorithm $\mathsf{Sign}(\mathsf{gsk}[d], M)$, which perform the following steps:

1. Let $\mathbf{x} = \mathsf{gsk}[d]$ then $\mathbf{x} \in \mathsf{Secret}(d)$, and $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$.

2. Let $\mathbf{y} = \mathsf{IdExt}(d) \in \{0,1\}^k$, and encrypt $\mathbf{y}$ by picking $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, sampling $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^k,\sigma}$, then computing the ciphertext $\mathbf{c} = \mathbf{B}^T \cdot \mathbf{s} + 2 \cdot \mathbf{e} + \mathbf{y} \bmod q$.

3. Using witness $(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{e}, d)$ and common input $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c})$ (and M), generate a non-interactive proof π for the relation $\mathrm{R}_{\mathrm{LGS}}$. This is done by repeating $t = \omega(\log n)$ times the protocol in Section 6.5, then making it non-interactive with the Fiat-Shamir heuristic, as follows:

(a) For $j \in [t]$, compute the 'commitment' $\mathrm{CMT}^{(j)}$.

(b) Invoke the random oracle: Let $\mathrm{CH} = \big(\{Ch^{(j)}\}_{j=1}^t\big) = \mathsf{H}\big(\{\mathrm{CMT}^{(j)}\}_{j=1}^t, M, \mathbf{c}\big)$.

(c) For $j \in [t]$, compute the 'response' $\mathrm{RSP}^{(j)}$ with respect to $(\mathrm{CMT}^{(j)}, Ch^{(j)})$.

(d) Return

$$\pi = \big(\{\mathrm{CMT}^{(j)}\}_{j=1}^t,\ \{Ch^{(j)}\}_{j=1}^t,\ \{\mathrm{RSP}^{(j)}\}_{j=1}^t\big). \tag{6.6}$$

4. Output the group signature: $\Sigma = \big(\mathbf{c}, \pi\big)$.

Verification Algorithm. On input $\mathsf{gpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$, a message $M \in \{0,1\}^*$, and a purported group signature $\Sigma = (\mathbf{c}, \pi)$ on M, the verifier runs the deterministic algorithm $\mathsf{Verify}(\mathsf{gpk}, \Sigma, M)$, which performs the following steps:

1. Parse π as in (6.6).

2. Invoke H to check if $\big(Ch^{(1)}, \ldots, Ch^{(t)}\big) = \mathsf{H}\big(\{\mathrm{CMT}^{(j)}\}_{j=1}^t, M, \mathbf{c}\big)$.

3. For $j = 1$ to t, run the verification of the protocol from Section 6.5 with common input $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c})$ to check the validity of $\mathrm{RSP}^{(j)}$ with respect to $\mathrm{CMT}^{(j)}$ and $Ch^{(j)}$. If any of the verification conditions does not hold, then output 0 and terminate.

4. Output 1.

Opening Algorithm. On input the group manager's secret key $\mathsf{gmsk} = \mathbf{T}$, a message M, and a signature $\Sigma = (\mathbf{c}, \pi)$, the opening algorithm $\mathsf{Open}(\mathbf{T}, M, \Sigma)$ performs the following steps:

1. Decrypting $\mathbf{c}$: First compute $\mathbf{c}' = \mathbf{T}^T \cdot \mathbf{c} \bmod q$, then return $\mathbf{y} = \mathbf{T}^{-T} \cdot \mathbf{c}' \bmod 2$.

2. If $\mathbf{y} = \mathsf{IdExt}(d)$ for some $d \in \{0,1\}^\ell$, then output d. Otherwise, output $\perp$.

6.6.2 Analysis of the Scheme

Correctness

We observe that, for all $(\mathbf{A}, \mathbf{B}, \mathbf{u}) \leftarrow \mathsf{KeyGen}(1^n, 1^N)$, and for all $d \in \{0,1\}^\ell$, the secret key $\mathbf{x} = \mathsf{gsk}[d]$ satisfies $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$, and $\mathbf{x} \in \mathsf{Secret}(d)$. Furthermore, given $(\mathbf{x}, d)$, the signer can always generate $(\mathbf{y}, \mathbf{s}, \mathbf{e})$, and compute $\mathbf{c}$ such that

$$((\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c}), \mathbf{x}, \mathbf{y}, \mathbf{e}, \mathbf{s}, d) \in \mathrm{R}_{\mathrm{LGS}}.$$

The completeness of the underlying proof system then implies that, for any message $M \in \{0,1\}^*$, the obtained signature Σ is always valid.

On the other hand, if $\Sigma = (\mathbf{c}, \pi)$ is a legitimate signature generated by an honest user with index d, then we have $\mathbf{c} = \mathbf{B}^T \cdot \mathbf{s} + 2 \cdot \mathbf{e} + \mathbf{y} \bmod q$, where $\|\mathbf{e}\|_\infty \leq \beta$, and $\mathbf{y} = \mathsf{IdExt}(d)$. In this case, the opening algorithm $\mathsf{Open}(\mathbf{T}, M, \Sigma)$ will output d with overwhelming probability. Indeed, we have:

$$\mathbf{c}' = \mathbf{T}^T\ \mathbf{c} \bmod q = \mathbf{T}^T(\mathbf{B}^T\ \mathbf{s} + 2\mathbf{e} + \mathbf{y}) \bmod q) = \mathbf{T}^T(2\mathbf{e} + \mathbf{y}) \bmod q.$$

Since $\mathbf{T}$ is a trapdoor basis with small entries, and both $\mathbf{e}$ and $\mathbf{y}$ are small, while q

is set to be sufficiently large, we have $\mathbf{c}' = \mathbf{T}^T(2\mathbf{e} + \mathbf{y})$ (over *integers*). Hence, multiplying it by the inverse of $\mathbf{T}^T$, and reducing modulo 2 allows to compute $\mathbf{y} = \mathsf{IdExt}(d)$, and thus, to recover d (via the bijection between the set $\{0,1\}^\ell$, and the set $\big\{\mathsf{IdExt}(d) \,\big|\, d \in \{0,1\}^\ell\big\}$.)

Efficiency

It can be checked that all 4 component-algorithms in the proposed group signature scheme can be implemented in polynomial time. For the given parameters setting, the group public key $(\mathbf{A}, \mathbf{B}, \mathbf{u})$ has bit-size $\big((2\ell+1)m + k + 1\big)n \log q = \ell \cdot \mathcal{O}(n^2)$. The signature contains a ciphertext $\mathbf{c} \in \mathbb{Z}_q^k$, and a proof π, which roughly costs $t = \omega(\log n)$ times the communication cost $\widetilde{\mathcal{O}}(n \cdot \ell)$ of the underlying protocol. Overall, the signature has bit-size $\widetilde{\mathcal{O}}(n \cdot \ell) = \widetilde{\mathcal{O}}(n \cdot \log N)$. In particular, the group signature scheme is compact, in the sense of Definition 6.2.2.

Anonymity

We will prove that the proposed group signature scheme $\mathcal{LGS}$ is CPA anonymous assuming the hardness of the decision LWE problem.

Theorem 6.6.2. *Assume that* COM *is a statistically hiding string commitment scheme. Let χ be the Gaussian distribution $D_{\mathbb{Z},\sigma}$. If the $\mathsf{LWE}(n, q, \chi)$ problem is hard, then the group signature $\mathcal{LGS}$ is* CPA*-anonymous.*

Proof. We consider a sequence of hybrid experiments $\mathbf{G}_0^{(b)}$, $\mathbf{G}_1^{(b)}$ and $\mathbf{G}_2$, where $b \in \{0,1\}$. The experiments are defined as follows:

Experiment $\mathbf{G}_0^{(b)}$: For each b, $\mathbf{G}_0^{(b)}$ is the anonymity experiment $\mathbf{Exp}^{\mathsf{weakly\text{-}anon}}_{\mathcal{LGS},\mathcal{A}}(n, N)$ considered in Definition 6.2.3, where the challenger picks the coin b.

1. Initialization: The challenger $\mathcal{C}$ runs $\mathsf{KeyGen}(1^n, 1^N)$ to obtain $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$, then it gives $\mathsf{gpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$ and the vector of all user secret keys gsk to $\mathcal{A}$.

2. Challenge: The adversary $\mathcal{A}$ outputs two distinct identities $d_0, d_1 \in \{0,1\}^\ell$ and a message M^*. The challenger computes a legitimate signature Σ^* using the secret key $\mathsf{gsk}[d_b]$, namely:

(a) $\mathbf{x} = \mathsf{gsk}[d_b] \in \mathsf{Secret}(d_b)$, and $\mathbf{y} = \mathsf{IdExt}(d_b) \in \{0,1\}^k$.

(b) Compute the ciphertext $\mathbf{c}$: Pick $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^n$, sample $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^k,\sigma}$, and then compute $\mathbf{c} = \mathbf{B}^T \cdot \mathbf{s} + 2 \cdot \mathbf{e} + \mathbf{y} \bmod q$.

(c) Generate a non-interactive proof π for the relation $\mathrm{R_{LGS}}$ where the common input is $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c})$ (and M^*), using witness $(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{e}, d_b)$, as in the scheme.

(d) Output $(\mathbf{c}, \pi)$.

Experiment $\mathbf{G}_1^{(b)}$: In this experiment, we introduce the following modifications with respect to experiment $\mathbf{G}_0^{(b)}$:

1. In Step (2b), we change how the ciphetext $\mathbf{c}$ is computed. Specifically, the challenger sample $\mathbf{z} \xleftarrow{\$} \mathbb{Z}_q^k$, and set $\mathbf{c} = \mathbf{z} + \mathbf{y} \bmod q$.

2. In Step (2c), the challenger simulates the proof π, given common input $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c})$ (and M^*).

   We first demonstrate how the proof π can be simulated in the random oracle model. The simulator, which is allowed to program the random oracle H, proceeds as follows:

   1. For each $j \in [t]$, pick a uniformly random 'fake challenge' $\overline{Ch}^{(j)} \in \{1,2,3\}$, and prepare the 'commitment' $\mathrm{CMT}^{(j)}$ according to the value of $\overline{Ch}^{(j)}$. Then pick a uniformly random 'real challenge' $Ch^{(j)} \in \{1,2,3\} \setminus \{\overline{Ch}^{(j)}\}$.

2. Let $\mathrm{CH} = \big(\{Ch^{(j)}\}_{j=1}^{t}\big)$, and program the random oracle:

$$\mathsf{H}\big(\{\mathrm{CMT}^{(j)}\}_{j=1}^{t}, M^*, \mathbf{c}\big) = \mathrm{CH}.$$

3. Prepare the 'response' $\mathrm{RSP}^{(1)}, \ldots, \mathrm{RSP}^{(t)}$.

4. Output $\pi = \big(\{\mathrm{CMT}^{(j)}\}_{j=1}^{t}, \{Ch^{(j)}\}_{j=1}^{t}, \{\mathrm{RSP}^{(j)}\}_{j=1}^{t}\big)$.

We note that, for each $j \in [t]$, the 'real challenge' $Ch^{(j)}$ is uniformly distributed in $\{1, 2, 3\}$, which satisfies the requirement on the output of the random oracle. Furthermore, the 'commitment' $\mathrm{CMT}^{(j)}$ and the 'response' $\mathrm{RSP}^{(j)}$ are prepared in the same way as in the proof of Lemma 6.5.3. Therefore, the statistical zero-knowledge property of the underlying proof system implies that the obtained proof π is a valid proof, and its distribution is statistically close to that of the legitimate proof produced in Step (2c) of experiment $\mathbf{G}_1^{(b)}$.

We now prove that, under the assumption that the decision $\mathsf{LWE}(n, q, \chi)$ problem is hard, the 'fake ciphertext' $\mathbf{c}$ generated in experiment $\mathbf{G}_1^{(b)}$ is computationally indistinguishable from the legitimate $\mathbf{c}$ produced by Step (2b) in experiment $\mathbf{G}_0^{(b)}$. Assuming by contradictory that the adversary $\mathcal{A}$ can distinguish between the real ciphertext and the 'fake ciphertext' with non-negligible probability. Adapting the technique of [74] and [88], we construct a polynomial-time LWE distinguisher $\mathcal{D}$, that has non-negligible advantage, as follows.

The distinguisher $\mathcal{D}$ takes as input a challenge LWE instance $(\mathbf{B}', \mathbf{z}') \in \mathbb{Z}_q^{n \times k} \times \mathbb{Z}_q^k$, where $\mathbf{B}'$ is uniformly random and $\mathbf{z}'$ is either uniformly random or of the form $\mathbf{z}' = \mathbf{B}'^T \cdot \mathbf{s}' + \mathbf{e}'$ for $\mathbf{s}' \xleftarrow{\$} \mathbb{Z}_q^n$ and $\mathbf{e}' \hookleftarrow D_{\mathbb{Z}^k,\sigma}$. We first define a modified $\mathsf{KeyGen}(1^n, 1^N)$ algorithm in which instead of generating the statistically close to uniform matrix $\mathbf{B}$ using $\mathsf{GenTrap}(1^n, 1^k, q)$, we set $\mathbf{B} = \mathbf{B}'$. We note that, under this modification,

the keys $\mathsf{gpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$ and gsk, which are given to the adversary, are statistically close to those produced by the genuine $\mathsf{KeyGen}(1^n, 1^N)$ algorithm. We then consider the following two intermediate experiments, where the only difference is the way we handle the ciphertext $\mathbf{c}$.

- $\mathbf{Exp}_{\mathsf{LWE}}$: $\mathbf{c} = 2 \cdot \mathbf{z}' + \mathbf{y}\ [q]$, where $\mathbf{z}'$ is uniformly random. Let $\mathbf{z} = 2 \cdot \mathbf{z}' \bmod q$, then $\mathbf{z}$ is uniformly distributed over $\mathbb{Z}_q^k$, because 2 is invertible modulo q. It then follows that experiment $\mathbf{Exp}_{\mathsf{LWE}}$ is statistically close to experiment $\mathbf{G}_1^{(b)}$.

- $\mathbf{Exp}_{\mathsf{rand}}$: $\mathbf{c} = 2 \cdot \mathbf{z}' + \mathbf{y}\ [q]$, where $\mathbf{z}'$ is of the form $\mathbf{z}' = \mathbf{B'}^T \cdot \mathbf{s}' + \mathbf{e}'$. Let $\mathbf{s} = 2 \cdot \mathbf{s}' \bmod q$, then $\mathbf{s}$ is uniformly distributed over $\mathbb{Z}_q^n$. We then have $\mathbf{c} = \mathbf{B'}^T \cdot \mathbf{s} + 2 \cdot \mathbf{e}' + \mathbf{y} \bmod q$. In other words, experiment $\mathbf{Exp}_{\mathsf{rand}}$ is statistically close to experiment $\mathbf{G}_0^{(b)}$.

The above discussions imply that, if the adversary $\mathcal{A}$ can distinguish experiments $\mathbf{G}_0^{(b)}$ and $\mathbf{G}_1^{(b)}$ with advantage ϵ, then it can distinguish D_{LWE} and D_{rand} with advantage $\epsilon - \mathsf{negl}(n)$. The distinguisher $\mathcal{D}$ then can use $\mathcal{A}$ to solve the challenge LWE instance with advantage $\epsilon - \mathsf{negl}(n)$. This leads to the following lemma:

Lemma 6.6.3. *If the $\mathsf{LWE}(n, q, \chi)$ problem is hard, then for each $b \in \{0, 1\}$, experiments $\mathbf{G}_0^{(b)}$ and $\mathbf{G}_1^{(b)}$ are (computationally) indistinguishable.*

Experiment $\mathbf{G}_2^{(b)}$: In this experiment, we make the outputted signature Σ^* independent of the coin b, while preserving the statistical closeness to experiment $\mathbf{G}_1^{(b)}$. In particular, we make the following modification with respect to $\mathbf{G}_1^{(b)}$: Instead of sampling $\mathbf{z} \xleftarrow{\$} \mathbb{Z}_q^k$ and outputting $\mathbf{c} = \mathbf{z} + \mathbf{y} \bmod q$, we simply sample $\mathbf{c} \xleftarrow{\$} \mathbb{Z}_q^k$. Since $\mathbf{c}$ is uniformly distributed, it is statistically close to that produced in experiment $\mathbf{G}_1^{(b)}$. Thus, we have the following lemma.

Lemma 6.6.4. *For each $b \in \{0,1\}$, experiment $\mathbf{G}_1^{(b)}$ and $\mathbf{G}_2$ are statistically indistinguishable.*

Combining the results of Lemma 6.6.3 and Lemma 6.6.4, we obtain a sequence of indistinguishable experiments: $\mathbf{G}_0^{(0)}$, $\mathbf{G}_1^{(0)}$, $\mathbf{G}_2$, $\mathbf{G}_1^{(1)}$, $\mathbf{G}_0^{(1)}$. Since experiment $\mathbf{G}_2$ is independent of the coin b, the advantage of $\mathcal{A}$ in this experiment is 0. As a result, its advantage in experiments $\mathbf{G}_0^{(0)}$ and $\mathbf{G}_0^{(1)}$ is negligible. In other words, we have:

$$\mathbf{Adv}^{\mathsf{weakly\text{-}anon}}_{\mathcal{LGS},\mathcal{A}}(n,N) = \mathsf{negl}(n).$$

This concludes the proof. □

Combining the results of Theorem 6.6.2 and Theorem 2.2.17, we obtain that: the CPA-anonymity of our scheme can be based on the quantum hardness of SIVP_γ with $\gamma = n \cdot q/\beta = \widetilde{\mathcal{O}}(n^2)$.

Traceability

To prove the full traceability of our group signature in the random oracle model, we need a *forking lemma*, which roughly says that if an adversary $\mathcal{A}$ can forge a signature with non-negligible probability, then by replaying $\mathcal{A}$ many times with the same random tape but different inputs, one can obtain other forgeries with high probability. The lemma stated below is adapted from the Improved Forking Lemma by Pointcheval and Vaudenay [124, Lemma 7] for the case of 3-fork.

Lemma 6.6.5 (Forking Lemma, Adapted from [124])**.** *Let $\mathcal{SS}$ be a signature scheme with security parameter n. Let $\mathcal{A}$ be a probabilistic polynomial-time algorithm whose input only consists of public data and which can ask q_{H} queries to the random oracle, where $q_{\mathsf{H}} > 0$. Assume that, within the time bound T, algorithm $\mathcal{A}$ produces, with*

probability ϵ, a valid signature $(\mathrm{CMT}, Ch, \mathrm{RSP}_1)$ *of message M. Then, within time* $32 \cdot T \cdot q_{\mathsf{H}}/\epsilon$, *and with probability $\epsilon' > 1/2$, a replay of $\mathcal{A}$ outputs* 3 *valid signatures of message M:*

$$(\mathrm{CMT}, Ch_1, \mathrm{RSP}_1);\ (\mathrm{CMT}, Ch_2, \mathrm{RSP}_2);\ (\mathrm{CMT}, Ch_3, \mathrm{RSP}_3),$$

such that Ch_1, Ch_2, Ch_3 are pairwise distinct.

We now prove that, in the random oracle model, our group signature scheme is traceable if the $\mathsf{ISIS}^{\infty}(n, (\ell+1) \cdot m, q, \beta)$ is hard.

Theorem 6.6.6. *Assume that* COM *is a computationally binding string commitment scheme. If there exists a traceability adversary $\mathcal{A}$ against the scheme $\mathcal{LGS}$, which has advantage ϵ and running time T, then there exists an algorithm $\mathcal{F}$ that solves the* $\mathsf{ISIS}^{\infty}(n, (\ell+1) \cdot m, q, \beta)$ *problem with success probability* $\epsilon' > \left(1 - (7/9)^t\right) \cdot \frac{1}{2N}$, *and running time* $T' = 32 \cdot T \cdot q_{\mathsf{H}}/(\epsilon - 3^{-t}) + \mathsf{poly}(n, N)$, *where q_{H} is the number of queries to the random oracle* $\mathsf{H} : \{0,1\}^* \to \{1,2,3\}^t$.

Proof. Assume that there exists an adversary $\mathcal{A}$ running in time T, such that its advantage in the experiment $\mathbf{Exp}^{\mathsf{trace}}_{\mathcal{LGS},\mathcal{A}}(n, N)$ is ϵ. We use $\mathcal{A}$ to construct an algorithm $\mathcal{F}$ solving the $\mathsf{ISIS}^{\infty}(n, (\ell+1) \cdot m, q, \beta)$ problem, which works as follows:
Challenge: Algorithm $\mathcal{F}$ is given a matrix $\mathbf{C} = \left[\mathbf{C}_0 \middle| \mathbf{C}_1 \middle| \ldots \middle| \mathbf{C}_\ell\right] \in \mathbb{Z}_q^{n \times (\ell+1) \cdot m}$ and a vector $\mathbf{u} \in \mathbb{Z}_q^n$ chosen uniformly at random. $\mathcal{F}$ wins the challenge if it can produce vector $\mathbf{z} \in \mathbb{Z}^{(\ell+1) \cdot m}$ such that $\|\mathbf{z}\|_\infty \leq \beta$ and $\mathbf{C} \cdot \mathbf{z} = \mathbf{u} \bmod q$.
Setup: Algorithm $\mathcal{F}$ performs the following steps:

1. Run $\mathsf{GenTrap}(1^n, 1^m, q)$ algorithm ℓ times, and let the outputs be

$$\big((\mathbf{F}_1, \mathbf{R}_1), (\mathbf{F}_2, \mathbf{R}_2), \ldots, (\mathbf{F}_\ell, \mathbf{R}_\ell)\big).$$

2. Pick a target index $d^* = d^*[1] \dots d^*[\ell] \xleftarrow{\$} \{0,1\}^\ell$, and define

$$\mathbf{A} = \left[\mathbf{A}_0 \middle| \mathbf{A}_1^0 \middle| \mathbf{A}_1^1 \middle| \dots \middle| \mathbf{A}_\ell^0 \middle| \mathbf{A}_\ell^1\right] \in \mathbb{Z}_q^{n \times (2\ell+1)\cdot m},$$

where $\mathbf{A}_0 = \mathbf{C}_0$, and for each $i \in [\ell]$: $\mathbf{A}_i^{d^*[i]} = \mathbf{C}_i$ and $\mathbf{A}_i^{1-d^*[i]} = \mathbf{F}_i$.

3. Generate the secret key for each user with index $d \neq d^*$, where $d = d[1] \dots d[\ell]$, as follows:

   - Let $d[b]$ $(1 \leq b \leq \ell)$ be the first bit from the left where $d[b] \neq d^*[b]$. Since $d \neq d^*$, such b must exist. It follows that $\mathbf{A}_b^{d[b]} = \mathbf{A}_b^{1-d^*[b]} = \mathbf{F}_b$.
   - Sample ℓ vectors

$$\mathbf{x}_0, \mathbf{x}_1^{d[1]}, \dots, \mathbf{x}_{b-1}^{d[b-1]}, \mathbf{x}_{b+1}^{d[b+1]}, \dots, \mathbf{x}_\ell^{d[\ell]} \hookleftarrow D_{\mathbb{Z}^m,\sigma},$$

     and let

$$\mathbf{t}^{(d)} = \mathbf{u} - \left(\mathbf{A}_0 \cdot \mathbf{x}_0 + \sum_{i \in [\ell], i \neq b} \mathbf{A}_i^{d[i]} \cdot \mathbf{x}_i^{d[i]}\right) \bmod q.$$

   - Sample $\mathbf{x}_b^{d[b]} \hookleftarrow \mathsf{SampleD}(\mathbf{R}_b, \mathbf{F}_b, \mathbf{t}^{(d)}, \sigma)$.
   - For each $i \in [\ell]$, let $\mathbf{x}_i^{1-d[i]} = \mathbf{0}^m$, then let

$$\mathbf{x}^{(d)} = \left(\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \dots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1\right) \in \mathbb{Z}^{(2\ell+1)\cdot m}.$$

     If the very rare event that $\|\mathbf{x}^{(d)}\|_\infty > \beta$ happens, then repeat the sampling. Otherwise, set $\mathsf{gsk}[d] = \mathbf{x}^{(d)}$.

4. Run $\mathsf{GenTrap}(1^n, 1^k, q)$ to output matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times k}$ together with a trapdoor $\mathbf{T}$.

5. Let $\mathsf{gpk} = (\mathbf{A}, \mathbf{B}, \mathbf{u})$, and $\mathsf{gmsk} = \mathbf{T}$.

We note that, by construction, the distribution of $(\mathsf{gpk}, \mathsf{gmsk})$ is statistically close to that of the real scheme, and for any $d \in \{0,1\}^\ell \setminus \{d^*\}$, the user secret key $\mathsf{gsk}[d]$ satisfies: $\mathsf{gsk}[d] \in \mathsf{Secret}(d)$ and $\mathbf{A} \cdot \mathsf{gsk}[d] = \mathbf{u} \bmod q$. Moreover, the choice of the target index d^* is hidden from the adversary.

Algorithm $\mathcal{F}$ then sets $CU \leftarrow \emptyset$ and gives $((\mathbf{A}, \mathbf{B}, \mathbf{u}), \mathbf{T})$ to $\mathcal{A}$.

Queries: The adversary $\mathcal{A}$ can adaptively make corruption and signing queries. Algorithm $\mathcal{F}$ answers the queries of $\mathcal{A}$ as follows:

- Corruption queries: If $\mathcal{A}$ queries the secret key of group user with index d, for any $d \in \{0,1\}^\ell \setminus \{d^*\}$, then $\mathcal{F}$ adds d to the set of corrupted users CU, and returns $\mathsf{gsk}[d]$. If $\mathcal{A}$ ever asks for $\mathsf{gsk}[d^*]$, then $\mathcal{F}$ aborts.

- Signatures queries: We consider two cases:

– If $\mathcal{A}$ queries for signatures of user with index d^* on arbitrary message M, then $\mathcal{F}$ encrypts vector $\mathbf{y}^* = \mathsf{Secret}(d^*)$ to obtain a ciphertext $\mathbf{c}^*$, then simulates a non-interactive proof π^* for the relation $\mathrm{R}_{\mathrm{LGS}}$ with common input $(\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c}^*)$ (and M), and return $(\mathbf{c}^*, \pi^*)$.

– If $\mathcal{A}$ queries signature of user with index $d \in \{0,1\}^\ell \setminus \{d^*\}$ on arbitrary message M, then $\mathcal{F}$ returns the legitimate signature $\Sigma = \mathsf{Sign}\big(\mathsf{gpk}, \mathsf{gsk}[d], M\big)$.

In both cases, queries to the random oracle H are handled by returning uniformly random values in $\{1,2,3\}^t$ in a consistent manner. For each $\kappa \leq q_{\mathsf{H}}$, we let r_κ denote the answer to the κ-th query.

Forgery: Eventually, $\mathcal{A}$ outputs a message M', and a forged group signature

$$\Sigma' = \big(\mathbf{c}', \{\mathrm{CMT}^{(j)}\}_{j=1}^t, \{Ch^{(j)}\}_{j=1}^t, \{\mathrm{RSP}^{(j)}\}_{j=1}^t\big),$$

such that $\mathsf{Verify}(\mathsf{gpk}, \Sigma', M') = 1$, and the opening algorithm $\mathsf{Open}(\mathbf{T}, M', \Sigma')$ either fails, or returns an index $d' \notin CU$, and $\mathcal{A}$ has never make a signing query for d', M'.

We observe that, with overwhelming probability, $\mathcal{A}$ must have queried H on input $\big(M', \{\mathrm{CMT}^{(j)}\}_{j=1}^{t}, \mathbf{c}'\big)$, as otherwise, the probability that $(Ch^{(1)}, \ldots, Ch^{(t)}) = \mathsf{H}\big(M', \{\mathrm{CMT}^{(j)}\}_{j=1}^{t}, \mathbf{c}'\big)$ is at most 3^{-t}. Therefore, with probability at least $\epsilon - 3^{-t}$, there exists certain $\kappa' \leq q_{\mathsf{H}}$ such that the κ'-th oracle query involves the tuple $\big(M', \{\mathrm{CMT}^{(j)}\}_{j=1}^{t}, \mathbf{c}'\big)$. Now, algorithm $\mathcal{F}$ picks κ' as the target forking point and replays $\mathcal{A}$ many times with the same random tape and input as in the original run. In each rerun, for the first $\kappa' - 1$ queries, $\mathcal{A}$ is given the same answers $r_1, \ldots, r_{\kappa'-1}$ as in the initial run, but from the κ'-th query onwards, $\mathcal{F}$ replies with fresh random values $r_{\kappa'}^{'}, \ldots, r_{q_{\mathsf{H}}}^{'} \xleftarrow{\$} \{1,2,3\}^t$. The forking lemma (Lemma 6.6.5) implies that, with probability larger than $1/2$, algorithm $\mathcal{F}$ can obtain a 3-fork involving the tuple $\big(M', \{\mathrm{CMT}^{j}\}_{j=1}^{t}, \mathbf{c}'\big)$ after at most $32 \cdot q_{\mathsf{H}}/(\epsilon - 3^{-t})$ executions of $\mathcal{A}$. Now, let the answers of $\mathcal{F}$ with respect to the 3-fork branches be:

$$r_{\kappa',1} = (Ch_1^{(1)}, \ldots, Ch_1^{(t)});\ r_{\kappa',2} = (Ch_2^{(1)}, \ldots, Ch_2^{(t)});\ r_{\kappa',3} = (Ch_3^{(1)}, \ldots, Ch_3^{(t)}).$$

A simple calculation shows that:

$$\Pr\big[\exists j \in \{1, \ldots, t\} :\ \{Ch_1^{(j)}, Ch_2^{(j)}, Ch_3^{(j)}\} = \{1,2,3\}\big] = 1 - (7/9)^t.$$

Conditioned on the existence of such index j, one parses the 3 forgeries corresponding to the fork branches to obtain $\big(\mathrm{RSP}_1^{(j)}, \mathrm{RSP}_2^{(j)}, \mathrm{RSP}_3^{(j)}\big)$. They turn out to be 3 valid responses with respect to all 3 challenges for the same commitment $\mathrm{CMT}^{(j)}$. Since COM is assumed to be computationally-binding, we can apply the extraction

technique in the proof of Lemma 6.5.4 to extract $(\mathbf{x}', \mathbf{y}', \mathbf{e}', \mathbf{s}', d')$ such that:

$$((\mathbf{A}, \mathbf{B}, \mathbf{u}, \mathbf{c}'), \mathbf{x}', \mathbf{y}', \mathbf{e}', \mathbf{s}', d') \in \mathrm{R}_{\mathrm{LGS}}.$$

In particular, we have $\mathbf{x}' \in \mathsf{Secret}(d')$ and $\mathbf{A} \cdot \mathbf{x}' = \mathbf{u} \bmod q$. Now consider two cases:

- If $d' \neq d^*$, which happens with probability at most $\frac{N-1}{N}$ (as the choice of d^* was hidden from $\mathcal{A}$), then algorithm $\mathcal{F}$ aborts and declares failure.
- If $d' = d^*$, which happens with probability at least $1/N$, then write vector $\mathbf{x}'$ as a concatenation of $2\ell+1$ blocks of size m, namely, $\mathbf{x}' = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \ldots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1)$, and let $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$ be the vector obtained by removing the zero-blocks $\mathbf{x}_1^{1-d^*[1]}, \ldots, \mathbf{x}_\ell^{1-d^*[\ell]}$ from $\mathbf{x}'$. It turns out that $\|\mathbf{z}\|_\infty \leq \beta$, and

  $$\mathbf{C} \cdot \mathbf{z} = \mathbf{A} \cdot \mathbf{x}' = \mathbf{u} \bmod q.$$

  In this case, $\mathcal{F}$ then outputs $\mathbf{z}$, which is a valid solution to the challenge $\mathsf{ISIS}^\infty(n, (\ell+1) \cdot m, q, \beta)$ instance $(\mathbf{C}, \mathbf{u})$.

We observe that the probability that $\mathcal{F}$ does not abort is at least $1/N$, and conditioned on not aborting, $\mathcal{F}$ can solve the $\mathsf{ISIS}^\infty(n, (\ell+1) \cdot m, q, \beta)$ problem with probability larger than $1/2 \cdot \left(1 - (7/9)^t\right)$ in time

$$T \cdot 32 \cdot q_{\mathsf{H}} / (\epsilon - 3^{-t}) + \mathsf{poly}(n, N).$$

This concludes the proof. □

The results of Theorem 2.2.14 and Theorem 6.6.6 imply that our group signature scheme is fully traceable in the random oracle model, based on the worst-case hardness of SIVP_γ^2 with $\gamma = \beta \cdot \widetilde{\mathcal{O}}(n) = \widetilde{\mathcal{O}}(n^{1.5})$.

7. CONCLUSION AND OPEN PROBLEMS

Throughout the thesis, we have developed a versatile Decomposition-Extension technique for constructing Stern-type ZKPoK for various relations arising from lattice-based cryptography. Starting from the **SternExt** proof system for the ISIS^∞ problem, we have modified the technique to get a ZK proof system for the SIS^∞ problem. Then we have adapted the technique to obtain a variety of lattice-based cryptographic construction with strong security guarantees: 4 ZKPoPK for 4 different LWE-based encryption schemes; an improved identity-based identification scheme and a new identity-based ring identification scheme; and finally an improved group signature scheme.

More generally, the technique developed in this thesis can be used to construct ZKPoK of linear objects (e.g., vectors, matrices) which are the secret solutions of some given systems of modular linear equations, and which satisfy some suitable constraints (e.g., "smallness"). In other words, our technique is somewhat compatible with the properties of lattice-based cryptography ("linearity", "smallness"), and can possibly find further interesting applications.

To summarize and generalize what we have studied throughout the thesis, we consider the following abstract relation. Let q, d, k, ℓ be some integers, and for

each $i \in [k]$, let $\mathsf{S}_i \subseteq \mathbb{Z}_q^{\ell_i}$ be some publicly known set. Define:

$$\mathrm{R} = \Big\{((\mathbf{M}_1, \ldots, \mathbf{M}_k, \mathbf{v}), \mathbf{z}_1, \ldots, \mathbf{z}_k) \in (\mathbb{Z}_q^{d \times \ell_1} \times \ldots \times \mathbb{Z}_q^{d \times \ell_k} \times \mathbb{Z}_q^d) \times (\mathbb{Z}_q^{\ell_1} \times \ldots \times \mathbb{Z}_q^{\ell_k}) : \\ (\mathbf{z}_1 \in \mathsf{S}_1) \wedge \ldots \wedge (\mathbf{z}_k \in \mathsf{S}_k) \wedge (\sum_{i=1}^{k} \mathbf{M}_i \cdot \mathbf{z}_i = \mathbf{v} \bmod q)\Big\}.$$

We observe that the above abstract relation subsumes all but one of the relations considered in this thesis. Indeed, we have:

1. In the proof system for the $\mathsf{ISIS}^{\infty}(n, m, q, 1)$ in Section 3.3.1, the relation was:

   $$\mathrm{R}_{\mathsf{ISIS}^{\infty}_{\beta=1}}(n, m, q) = \Big\{((\mathbf{A}, \mathbf{y}), \mathbf{x}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n \times \{-1, 0, 1\}^m : \ \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q\Big\}$$

   In this case, we have: $d = n$, $k = 1$, $\ell_1 = m$; $\mathbf{M_1} = \mathbf{A}$, $\mathbf{v} = \mathbf{y}$; and $\mathsf{S}_1 = \{-1, 0, 1\}^m$.

   Technique: Since each coordinate of $\mathbf{x}$ has 3 possible values, we added 2 "fake" values to hide the real one. As a result, the dimension was extended to $3m$.

2. In the proof system for $\mathsf{ISIS}^{\infty}(n, m, q, \beta)$ in Section 3.3, the relation was:

   $$\mathrm{R}_{\mathsf{ISIS}^{\infty}}(n, m, q, \beta) = \big\{((\mathbf{A}, \mathbf{y}), \mathbf{x}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n \times \mathbb{Z}^m : \|\mathbf{x}\|_{\infty} \leq \beta \wedge \mathbf{A} \cdot \mathbf{x} = \mathbf{y} \bmod q\big\}.$$

   In this case, we have $d = n$, $k = 1$, $\ell_1 = m$; $\mathbf{M_1} = \mathbf{A}$, $\mathbf{v} = \mathbf{y}$;

   $\mathsf{S}_1 = \{\mathbf{x} \in \mathbb{Z}_q^m : \ \|\mathbf{x}\|_{\infty} \leq \beta\}$.

   Technique: We decomposed vector $\mathbf{x}$ into $p = \lfloor \log \beta \rfloor + 1$ vectors in $\{-1, 0, 1\}^m$, and combined p proofs for $\mathrm{R}_{\mathsf{ISIS}^{\infty}}(n, m, q, \beta)$ into one proof.

3. In the proof system for $\mathsf{SIS}^{\infty}(n, m, q, \beta)$ in Section 3.4, the relation was:

   $$\mathrm{R}_{\mathsf{SIS}^{\infty}}(n, m, q, \beta) = \Big\{(\mathbf{A}, \mathbf{x}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}^m \colon \ (0 < \|\mathbf{x}\|_{\infty} \leq \beta) \wedge (\mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q)\Big\}.$$

In this case, we have $d = n$, $k = 1$, $\ell_1 = m$; $\mathbf{M_1} = \mathbf{A}$, $\mathbf{v} = \mathbf{0}$;

$\mathsf{S}_1 = \{\mathbf{x} \in \mathbb{Z}_q^m : 0 < \|\mathbf{x}\|_\infty \leq \beta\}$.

Technique: We used binary decomposition. To prove that $\mathbf{x} \neq \mathbf{0}$, we repeatedly divided it by 2 until at least one of its coordinate is odd, which means that the least significant bit of that coordinate must be non-zero. The extension corresponding to the least significant bit was only to dimension $3m - 1$, to prevent cheating with $\mathbf{x} = \mathbf{0}$.

4. The plaintext relation $\mathrm{R}_{\mathrm{Regev}}$ in Section 4.2 is essentially an $\mathsf{ISIS}^\infty(n+1, m+1, q, 1)$ relation.

5. The plaintex relation $\mathrm{R}_{\mathrm{Dual}}$ in Section 4.3:

$$\mathrm{R}_{\mathrm{Dual}}(n, m, q, \beta) = \Big\{ \big((\overline{\mathbf{A}}, \mathbf{y}), \mathbf{s}, \mathbf{x}, M\big) \in \mathbb{Z}_q^{n\times(m+1)} \times \mathbb{Z}_q^{m+1} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^{m+1} \times \{0,1\} : (\|\mathbf{x}\|_\infty \leq \beta) \wedge (\overline{\mathbf{A}}^T \cdot \mathbf{s} + \mathbf{x} + (0, \ldots, 0, \lfloor q/2 \rfloor)^T \cdot M = \mathbf{y} \bmod q) \Big\}$$

In this case, we have: $d = m + 1$, $k = 3$, $\ell_1 = n$, $\ell_2 = m + 1$, $\ell_3 = 1$; $\mathbf{M}_1 = \overline{\mathbf{A}}^T$, $\mathbf{M}_2 = \mathbf{I}_{m+1}$, $\mathbf{M}_3 = (0, \ldots, 0, \lfloor q/2 \rfloor)^T \in \mathbb{Z}_q^{(m+1)\times 1} = \mathbb{Z}_q^{d\times 1}$, $\mathbf{v} = \mathbf{y}$;

$\mathsf{S}_1 = \mathbb{Z}_q^n$, $\mathsf{S}_2 = \{\mathbf{x} \in \mathbb{Z}_q^{m+1} : \|\mathbf{x}\|_\infty \leq \beta\}$; $\mathsf{S}_3 = \{0, 1\}$.

Technique: Since there is no restriction on the set S_1, we did not have to perform Decomposition-Extension technique for $\mathbf{s} = \mathbf{z}_1$. Since the restriction on the set S_2 is "smallness", we performed Decomposition-Extension, as for $\mathrm{R}_{\mathsf{ISIS}^\infty}$. And we performed extension with respect to M: Here M has 2 possible values, we added 1 "fake" values to hide the real one, namely, we extended M to vector $(M, 1 - M)$, and added 1 zero-column to matrix $\mathbf{M}_3$ to make the dimensions compatible.

6. The plaintext relation $\mathrm{R}_{\mathrm{PVW}}$ in Section 4.4.1:

$$\mathrm{R}_{\mathrm{PVW}} = \Big\{((\mathbf{G},\mathbf{y}),\mathbf{e},\mathbf{v}) \in \mathbb{Z}_q^{(n+\ell)\times m} \times \mathbb{Z}_q^{n+\ell} \times \{0,1\}^m \times [0,k]^\ell : \mathbf{G}\cdot\mathbf{e}+\mathbf{H}\cdot\mathbf{v}=\mathbf{y} \bmod q\Big\}.$$

In this case, we have: $d = n+\ell$, $k = 2$, $\ell_1 = m$, $\ell_2 = \ell$; $\mathbf{M}_1 = \mathbf{G}$, $\mathbf{M}_2 = \mathbf{H}$; $\mathsf{S}_1 = \{0,1\}^m$, $\mathsf{S}_2 = \{0,1,\ldots,k\}^\ell$.

Technique: Each coordinate of $\mathbf{e}$ has 2 possible values. Hence, we extended it to dimension $2m$. For vector $\mathbf{v} \in \{0,1,\ldots,k\}^\ell$, we decomposed it to $\lfloor \log k \rfloor + 1$ vectors in $\{0,1\}^\ell$, and applied the extension technique, to dimension 2ℓ.

7. The plaintext relation $\mathrm{R}_{\mathrm{GHV}}$ in Section 4.4.2 is essentially the conjunction of m "Dual-Regev-type" relations. We combined m proofs into one proof.

8. The relation $\mathrm{R}_{\mathrm{LIBRI}}$ for the lattice-based identity-based ring identification scheme in Section 5.3.2:

$$\mathrm{R}_{\mathrm{LIBRI}}(n,m,q,\beta,N) = \Big\{\big((\mathbf{A},\mathbf{B}),\mathbf{x},\mathbf{e}\big) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^{n\times N} \times \mathbb{Z}^m \times \{0,1\}^N : (\|\mathbf{x}\|_\infty \le \beta) \wedge (\mathsf{w}(\mathbf{e}) = 1) \wedge (\mathbf{A}\cdot\mathbf{x} - \mathbf{B}\cdot\mathbf{e} = \mathbf{0} \bmod q)\Big\},$$

In this case, we have: $d = n$, $k = 2$, $\ell_1 = m$, $\ell_2 = N$; $\mathbf{M}_1 = \mathbf{A}$, $\mathbf{M}_2 = \mathbf{B}$; $\mathsf{S}_1 = \{\mathbf{x} \in \mathbb{Z}_q^m : \|\mathbf{x}\|_\infty \le \beta\}$, $\mathsf{S}_2 = \{\mathbf{e} \in \{0,1\}^N : \mathsf{wt}(\mathbf{e}) = 1\}$.

Technique: We handled $\mathbf{x}$ by the Decomposition-Extension technique, while the constraint of $\mathbf{e}$ was verified by a random permutation of N elements, as usual.

Generally, for the given abstract relation, we could prove in zero-knowledge the knowledge of $\mathbf{z}_i \in \mathsf{S}_i$ if S_i is either $\mathbb{Z}_q^{\ell_i}$ or a "suitable" subset of $\mathbb{Z}_q^{\ell_i}$, such as $[-\beta,\beta]^{\ell_i}$; $[0,\beta]^{\ell_i}$; a set of small cardinality (e.g., $\{0,1\}$); and a set of vectors with given Hamming weight.

On the other hand, the underlying relation of our group signature scheme was rather sophisticated, and was not subsumed by the above abstract relation. We have:

$$\mathrm{R_{LGS}} = \Big\{ \big((\mathbf{A},\mathbf{B},\mathbf{u},\mathbf{c}),\mathbf{x},\mathbf{y},\mathbf{e},\mathbf{s},d\big) \in (\mathbb{Z}_q^{n\times(2\ell+1)m} \times \mathbb{Z}_q^{n\times k} \times \mathbb{Z}_q^n \times \mathbb{Z}_q^k) \times \mathbb{Z}^{(2\ell+1)m} \times \\ \times\{0,1\}^k \times \mathbb{Z}_q^k \times \mathbb{Z}_q^n \times \{0,1\}^\ell : (\mathbf{x} \in \mathsf{Secret}(d)) \wedge (\mathbf{y} = \mathsf{IdExt}(d)) \wedge (\|\mathbf{e}\|_\infty \leq \beta) \\ (\mathbf{A}\cdot\mathbf{x} = \mathbf{u} \bmod q) \wedge (\mathbf{B}^T\cdot\mathbf{s} + 2\cdot\mathbf{e} + \mathbf{y} = \mathbf{c} \bmod q) \Big\}.$$

This can be considered as the conjunction of an ISIS-type relation and a "Dual-Regev-type" relation, with one additional condition linking them together: the ISIS solution $\mathbf{x}$ and the plaintext $\mathbf{y}$ in the Dual-Regev encryption must simultaneously have the shape determined by some $d \in \{0,1\}^\ell$. Specifically, the conditions on $\mathbf{x}$ can be expressed as:

$$(\mathbf{x}_0 \in [-\beta,\beta]^m) \bigwedge_{i\in[\ell]} \Big(\big((\mathbf{x}_i^0 = 0^m) \wedge (\mathbf{x}_i^1 \in [-\beta,\beta]^m)\big) \vee \big((\mathbf{x}_i^0 \in [-\beta,\beta]^m) \wedge (\mathbf{x}_i^1 = 0^m)\big) \Big).$$

To handle this, we combined the Decomposition-Extension technique (to prove the smallness of $\mathbf{x}$) and a "one-time-pad" technique (to prove the internal structure of $\mathbf{x}$), which essentially permutes the blocks $\mathbf{x}_i^0, \mathbf{x}_i^1$. Namely, we applied a composition of permutations of different types on the witness.

At a high level, the proof system for $\mathrm{R_{LGS}}$ hints that our technique can handle more complicated conditions apart from $(\mathbf{z}_1 \in \mathsf{S}_1) \wedge \ldots \wedge (\mathbf{z}_k \in \mathsf{S}_k)$. Adapting the technique for conditions such as $(\mathbf{z}_1 \in \mathsf{S}_1) \vee \ldots \vee (\mathbf{z}_k \in \mathsf{S}_k)$ could possibly lead to other interesting cryptographic applications. This is a subject for further investigations.

In addition, there are open questions with respect to some of the schemes we have constructed in the thesis.

Chapter 3: We have obtained interactive ZKPoK for the ISIS^∞ and SIS^∞ problems.

When we needed to make them non-interactive, we had to rely on the Fiat-Shamir heuristic, and the resulting non-interactive proofs is only proven secure in the random oracle model. Designing non-interactive ZKPoK without the random oracle for these problems is an interesting open question. Such proof systems would possibly lead to trapdoor-free lattice-based signature schemes and lattice-based group signature schemes secure in the standard model, and much more. A recent result by Garg, Gentry and Halevi [60] might be a possible direction towards solving this open question.

Chapter 4: We have constructed ZKPoPK for 4 LWE-based encryption schemes, which yields interactive encryption protocols secure against chosen ciphertext attacks. On the other hand, there are interesting primitives related to our construction: verifiable encryption and decryption schemes [39]. A verifiable encryption scheme for the Ajtai-Dwork cryptosystem was introduced by Goldwasser and Kharchenko [72], but verifiable decryption scheme for Ajtai-Dwork cryptosystem was left as an open question. Designing verifiable encryption and decryption schemes from LWE is a subject of our further development.

Chapter 5: The ring identification scheme that we introduced has communication cost linear in the size N of the ring, which could be a disadvantage if the ring is large. Reducing the communication cost (e.g., to $\widetilde{\mathcal{O}}(\log N)$) is an open question. In pairing-based schemes, this is typically done by employing an accumulator. It is not clear whether an analogous primitive can be realized in the lattice setting.

Chapter 6: Our group signature scheme so far only satisfies the CPA-anonymity notion. An open problem is how to upgrade it to a CCA-anonymous group signature. As shown in the previous works [38, 88], an additional cryptographic ingredient, such as a strongly secure one-time signature, might be useful in this context.

BIBLIOGRAPHY

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient Lattice (H)IBE in the Standard Model. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, 2010.

[2] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 98–115. Springer, 2010.

[3] Dorit Aharonov and Oded Regev. Lattice Problems in $\mathsf{NP} \cap \mathsf{coNP}$. *Journal of the ACM*, 52(5):749–765, 2005.

[4] Miklós Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract). In *STOC*, pages 99–108. ACM, 1996.

[5] Miklós Ajtai. The Shortest Vector Problem in ℓ_2 is NP-hard for Randomized Reductions (Extended Abstract). In *STOC*, pages 10–19. ACM, 1998.

[6] Miklós Ajtai. Generating Hard Instances of the Short Basis Problem. In *ICALP*, volume 1644 of *Lecture Notes in Computer Science*, pages 1–9. Springer, 1999.

[7] Miklós Ajtai. Representing Hard Lattices with $\mathcal{O}(n \log n)$ Bits. In *STOC*, pages 94–103. ACM, 2005.

[8] Miklós Ajtai and Cynthia Dwork. A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence. In *STOC*, pages 284–293. ACM, 1997.

[9] Miklós Ajtai and Cynthia Dwork. The First and Fourth Public-Key Cryptosystems with Worst-Case/Average-Case Equivalence. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(097), 2007.

[10] Miklós Ajtai, Ravi Kumar, and D. Sivakumar. A Sieve Algorithm for the Shortest Lattice Vector Problem. In *STOC*, pages 601–610. ACM, 2001.

[11] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous Hardcore Bits and Cryptography against Memory Attacks. In *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009.

[12] Joël Alwen and Chris Peikert. Generating Shorter Bases for Hard Random Lattices. *Theory Comput. Syst.*, 48(3):535–553, 2011.

[13] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.

[14] Sanjeev Arora, László Babai, Jacques Stern, and Z. Sweedyk. The Hardness of Approximate Optima in Lattices, Codes, and Systems of Linear Equations. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[15] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE. In *EURO-*

CRYPT, volume 7237 of *Lecture Notes in Computer Science*, pages 483–501. Springer, 2012.

[16] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A Practical and Provably Secure Coalition-Resistant Group Signature Scheme. In *CRYPTO*, volume 1880 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2000.

[17] Mira Belenkiy, Jan Camenisch, Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Hovav Shacham. Randomizable Proofs and Delegatable Anonymous Credentials. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 108–125. Springer, 2009.

[18] Mihir Bellare and Oded Goldreich. On Defining Proofs of Knowledge. In *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pages 390–420. Springer, 1992.

[19] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions. In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 614–629. Springer, 2003.

[20] Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security Proofs for Identity-Based Identification and Signature Schemes. *Journal of Cryptology*, 22(1):1–61, 2009.

[21] Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[22] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. In *EUROCRYPT*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer, 1996.

[23] Mihir Bellare, Haixia Shi, and Chong Zhang. Foundations of Group Signatures: The Case of Dynamic Groups. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 136–153. Springer, 2005.

[24] Rikke Bendlin and Ivan Damgård. Threshold Decryption and Zero-Knowledge Proofs for Lattice-Based Cryptosystems. In *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 201–218. Springer, 2010.

[25] Rikke Bendlin, Ivan Damgård, Claudio Orlandi, and Sarah Zakarias. Semi-homomorphic Encryption and Multiparty Computation. In *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 169–188. Springer, 2011.

[26] Slim Bettaieb and Julien Schrek. Improved Lattice-Based Threshold Ring Signature Scheme. In *PQCrypto*, volume 7932 of *Lecture Notes in Computer Science*, pages 34–51. Springer, 2013.

[27] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short Group Signatures. In *CRYPTO*, volume 3152 of *Lecture Notes in Computer Science*, pages 41–55. Springer, 2004.

[28] Dan Boneh and Hovav Shacham. Group Signatures with Verifier-local Revocation. In *ACM Conference on Computer and Communications Security*, pages 168–177. ACM, 2004.

[29] Xavier Boyen. Lattice Mixing and Vanishing Trapdoors: A Framework for

Fully Secure Short Signatures and More. In *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 499–517. Springer, 2010.

[30] Xavier Boyen and Brent Waters. Compact Group Signatures Without Random Oracles. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444. Springer, 2006.

[31] Zvika Brakerski. Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, 2012.

[32] Zvika Brakerski, Craig Gentry, and Shai Halevi. Packed Ciphertexts in LWE-based Homomorphic Encryption. In *Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2013.

[33] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) Fully Homomorphic Encryption without Bootstrapping. In *ITCS*, pages 309–325. ACM, 2012.

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

[35] Zvika Brakerski and Vinod Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. In *FOCS*, pages 97–106. IEEE, 2011.

[36] Jan Camenisch and Thomas Groß. Efficient Attributes for Anonymous Credentials. *ACM Trans. Inf. Syst. Secur.*, 15(1):4, 2012.

[37] Jan Camenisch and Anna Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation.

In *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer, 2001.

[38] Jan Camenisch, Gregory Neven, and Markus Rückert. Fully Anonymous Attribute Tokens from Lattices. In *SCN*, volume 7485 of *Lecture Notes in Computer Science*, pages 57–75. Springer, 2012.

[39] Jan Camenisch and Victor Shoup. Practical Verifiable Encryption and Decryption of Discrete Logarithms. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 126–144. Springer, 2003.

[40] Jan Camenisch and Markus Stadler. Efficient Group Signature Schemes for Large Groups (Extended Abstract). In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 410–424. Springer, 1997.

[41] Ran Canetti, Oded Goldreich, and Shai Halevi. The Random Oracle Methodology, Revisited. *Journal of the ACM*, 51(4):557–594, 2004.

[42] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai Trees, or How to Delegate a Lattice Basis. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010.

[43] Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. A Lattice-Based Threshold Ring Signature Scheme. In *LATINCRYPT*, volume 6212 of *Lecture Notes in Computer Science*, pages 255–272. Springer, 2010.

[44] Melissa Chase, Markulf Kohlweiss, Anna Lysyanskaya, and Sarah Meiklejohn. Malleable Signatures: Complex Unary Transformations and Delegatable Anonymous Credentials. *IACR Cryptology ePrint Archive*, 2013:179, 2013.

[45] Melissa Chase and Anna Lysyanskaya. On Signatures of Knowledge. In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 78–96. Springer, 2006.

[46] David Chaum and Eugène van Heyst. Group Signatures. In *EUROCRYPT*, volume 547 of *Lecture Notes in Computer Science*, pages 257–265. Springer, 1991.

[47] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Khoa Nguyen. Revocable Identity-Based Encryption from Lattices. In *ACISP*, volume 7372 of *Lecture Notes in Computer Science*, pages 390–403. Springer, 2012.

[48] Lidong Chen and Torben P. Pedersen. New Group Signature Schemes (Extended Abstract). In *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 171–181. Springer, 1994.

[49] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187. Springer, 1994.

[50] Ivan Damgård. A Design Principle for Hash Functions. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.

[51] Ivan Damgård and Adriana López-Alt. Zero-Knowledge Proofs with Low Amortized Communication from Lattice Assumptions. In *SCN*, volume 7485 of *Lecture Notes in Computer Science*, pages 38–56. Springer, 2012.

[52] Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the Existence

of Statistically Hiding Bit Commitment Schemes and Fail-Stop Signatures. *Journal of Cryptology*, 10(3):163–194, 1997.

[53] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to Within Almost-Polynomial Factors is NP-Hard. *Combinatorica*, 23(2):205–243, 2003.

[54] Yevgeniy Dodis, Aggelos Kiayias, Antonio Nicolosi, and Victor Shoup. Anonymous Identification in Ad Hoc Groups. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 609–626. Springer, 2004.

[55] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-Knowledge Proofs of Identity. *Journal of Cryptology*, 1(2):77–94, 1988.

[56] Uriel Feige and Adi Shamir. Zero Knowledge Proofs of Knowledge in Two Rounds. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 526–544. Springer, 1989.

[57] Uriel Feige and Adi Shamir. Witness Indistinguishable and Witness Hiding Protocols. In *STOC*, pages 416–426. ACM, 1990.

[58] Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[59] Zvi Galil, Stuart Haber, and Moti Yung. Symmetric Public-Key Encryption. In *CRYPTO*, volume 218 of *Lecture Notes in Computer Science*, pages 128–137. Springer, 1985.

[60] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate Multilinear Maps

from Ideal Lattices. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2013.

[61] Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. A Simple BGN-Type Cryptosystem from LWE. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 506–522. Springer, 2010.

[62] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC*, pages 197–206. ACM, 2008.

[63] Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based. *IACR Cryptology ePrint Archive*, 2013:340, 2013.

[64] Craig Gentry and Michael Szydlo. Cryptanalysis of the Revised NTRU Signature Scheme. In *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 299–320. Springer, 2002.

[65] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

[66] Oded Goldreich and Shafi Goldwasser. On the Limits of Nonapproximability of Lattice Problems. *Journal of Computer and System Sciences*, 60(3):540–563, 2000.

[67] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Collision-free hashing from lattice problems. *Electronic Colloquium on Computational Complexity (ECCC)*, 3(42), 1996.

[68] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Eliminating Decryption Errors in the Ajtai-Dwork Cryptosystem. In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 105–111. Springer, 1997.

[69] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-Key Cryptosystems from Lattice Reduction Problems. In *CRYPTO*, volume 1294 of *Lecture Notes in Computer Science*, pages 112–131. Springer, 1997.

[70] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and Jean-Pierre Seifert. Approximating Shortest Lattice Vectors is not Harder than Approximating Closest Lattice Vectors. *Inf. Process. Lett.*, 71(2):55–61, 1999.

[71] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the Learning with Errors Assumption. In *ICS*, pages 230–240. Tsinghua University Press, 2010.

[72] Shafi Goldwasser and Dmitriy Kharchenko. Proof of Plaintext Knowledge for the Ajtai-Dwork Cryptosystem. In *TCC*, volume 3378 of *Lecture Notes in Computer Science*, pages 529–555. Springer, 2005.

[73] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[74] S. Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. A Group Signature Scheme from Lattice Assumptions. In *ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 395–412. Springer, 2010.

[75] Jens Groth. Fully Anonymous Group Signatures Without Random Oracles.

In *ASIACRYPT*, volume 4833 of *Lecture Notes in Computer Science*, pages 164–180. Springer, 2007.

[76] Louis C. Guillou and Jean-Jacques Quisquater. A "Paradoxical" Indentity-Based Signature Scheme Resulting from Zero-Knowledge. In *CRYPTO*, volume 403 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 1988.

[77] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems. In *CHES*, pages 530–547, 2012.

[78] Shai Halevi and Silvio Micali. Practical and Provably-Secure Commitment Schemes from Collision-Free Hashing. In *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 201–215. Springer, 1996.

[79] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In *ANTS*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.

[80] Russell Impagliazzo. A Personal View of Average-Case Complexity. In *Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society, 1995.

[81] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. Zero-knowledge from Secure Multiparty Computation. In *STOC*, pages 21–30. ACM, 2007.

[82] Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and Efficient Zero-Knowledge Proofs from Learning Parity with Noise.

In *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 663–680. Springer, 2012.

[83] Jonathan Katz. Efficient and Non-malleable Proofs of Plaintext Knowledge and Applications. In *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 211–228. Springer, 2003.

[84] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Multi-bit Cryptosystems Based on Lattice Problems. In *Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 315–329. Springer, 2007.

[85] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, 2008.

[86] Hugo Krawczyk and Tal Rabin. Chameleon Signatures. In *NDSS*. The Internet Society, 2000.

[87] Kaoru Kurosawa and Swee-Huay Heng. From Digital Signature to ID-based Identification/Signature. In *Public Key Cryptography*, volume 2947 of *Lecture Notes in Computer Science*, pages 248–261. Springer, 2004.

[88] Fabien Laguillaumie, Adeline Langlois, Benoît Libert, and Damien Stehlé. Lattice-Based Group Signatures with Logarithmic Signature Size. *IACR Cryptology ePrint Archive*, 2013:308, 2013.

[89] Leslie Lamport. Constructing Digital Signatures from One-way Functions. Technical Report CSL-98, SRI International, 1979.

[90] Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and László Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[91] Benoît Libert, Thomas Peters, and Moti Yung. Group Signatures with Almost-for-Free Revocation. In *CRYPTO*, volume 7417 of *Lecture Notes in Computer Science*, pages 571–589. Springer, 2012.

[92] Benoît Libert, Thomas Peters, and Moti Yung. Scalable Group Signatures with Revocation. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 609–627. Springer, 2012.

[93] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications. In *Public Key Cryptography*, volume 7778 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2013.

[94] Vadim Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *Public Key Cryptography*, volume 4939 of *Lecture Notes in Computer Science*, pages 162–179. Springer, 2008.

[95] Vadim Lyubashevsky. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, 2009.

[96] Vadim Lyubashevsky. Lattice Signatures without Trapdoors. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, 2012.

[97] Vadim Lyubashevsky and Daniele Micciancio. Generalized Compact Knapsacks Are Collision Resistant. In *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, 2006.

[98] Vadim Lyubashevsky and Daniele Micciancio. Asymptotically Efficient Lattice-Based Digital Signatures. In *TCC*, volume 4948 of *Lecture Notes in Computer Science*, pages 37–54. Springer, 2008.

[99] Vadim Lyubashevsky and Daniele Micciancio. On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2009.

[100] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On Ideal Lattices and Learning with Errors over Rings. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010.

[101] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A Toolkit for Ring-LWE Cryptography. In *EUROCRYPT*, volume 7881 of *Lecture Notes in Computer Science*, pages 35–54. Springer, 2013.

[102] Ueli M. Maurer. Unifying Zero-Knowledge Proofs of Knowledge. In *AFRICACRYPT*, volume 5580 of *Lecture Notes in Computer Science*, pages 272–286. Springer, 2009.

[103] Carlos Aguilar Melchor, Pierre-Louis Cayrel, Philippe Gaborit, and Fabien Laguillaumie. A New Efficient Threshold Ring Signature Scheme Based on Coding Theory. *IEEE Transactions on Information Theory*, 57(7):4833–4842, 2011.

[104] Ralph C. Merkle. A Certified Digital Signature. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer, 1989.

[105] Ralph C. Merkle. One Way Hash Functions and DES. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.

[106] Daniele Micciancio. The Shortest Vector in a Lattice is Hard to Approximate to within Some Constant. *Electronic Colloquium on Computational Complexity (ECCC)*, 5(16), 1998.

[107] Daniele Micciancio. Almost Perfect Lattices, the Covering Radius Problem, and Applications to Ajtai's Connection Factor. *SIAM Journal on Computing*, 34(1):118–169, 2004.

[108] Daniele Micciancio. Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions. *Computational Complexity*, 16(4):365–411, 2007.

[109] Daniele Micciancio. Efficient Reductions among Lattice Problems. In *SODA*, pages 84–93. SIAM, 2008.

[110] Daniele Micciancio and Petros Mol. Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions. In *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 465–484. Springer, 2011.

[111] Daniele Micciancio and Chris Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 700–718. Springer, 2012.

[112] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with Small Parameters. *IACR Cryptology ePrint Archive*, 2013:69, 2013.

[113] Daniele Micciancio and Oded Regev. Worst-Case to Average-Case Reductions Based on Gaussian Measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[114] Daniele Micciancio and Oded Regev. Lattice-based Cryptography. In *Post-Quantum Cryptography*, pages 147–191. Springer, 2009.

[115] Daniele Micciancio and Salil P. Vadhan. Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 282–298. Springer, 2003.

[116] Lan Nguyen. Accumulators from Bilinear Pairings and Applications. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 275–292. Springer, 2005.

[117] Phong Q. Nguyen and Oded Regev. Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 271–288. Springer, 2006.

[118] Chris Peikert. Public-key Cryptosystems from the Worst-case Shortest Vector Problem: Extended Abstract. In *STOC*, pages 333–342. ACM, 2009.

[119] Chris Peikert. An Efficient and Parallel Gaussian Sampler for Lattices. In *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 80–97. Springer, 2010.

[120] Chris Peikert and Alon Rosen. Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices. In *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 145–166. Springer, 2006.

[121] Chris Peikert and Vinod Vaikuntanathan. Noninteractive Statistical Zero-Knowledge Proofs for Lattice Problems. In *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 536–553. Springer, 2008.

[122] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A Framework for Efficient and Composable Oblivious Transfer. In *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008.

[123] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *STOC*, pages 187–196. ACM, 2008.

[124] D. Pointcheval and S. Vaudenay. On Provable Security for Digital Signature Algorithms. Technical Report LIENS-96-17, Laboratoire d'Informatique de Ecole Normale Superieure, 1997.

[125] Charles Rackoff and Daniel R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

[126] Oded Regev. New Lattice-based Cryptographic Constructions. In *STOC*, pages 407–416. ACM, 2003.

[127] Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In *STOC*, pages 84–93. ACM, 2005.

[128] Oded Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *Journal of the ACM*, 56(6), 2009.

[129] Oded Regev. The Learning with Errors Problem (Invited Survey). In *IEEE Conference on Computational Complexity*, pages 191–204. IEEE Computer Society, 2010.

[130] Markus Rückert. Adaptively Secure Identity-Based Identification from Lattices without Random Oracles. In *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 345–362. Springer, 2010.

[131] Markus Rückert. Lattice-Based Blind Signatures. In *ASIACRYPT*, volume 6477 of *Lecture Notes in Computer Science*, pages 413–430. Springer, 2010.

[132] Markus Rückert. Strongly Unforgeable Signatures and Hierarchical Identity-Based Signatures from Lattices without Random Oracles. In *PQCrypto*, volume 6061 of *Lecture Notes in Computer Science*, pages 182–200. Springer, 2010.

[133] Claus-Peter Schnorr. Efficient Identification and Signatures for Smart Cards. In *CRYPTO*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989.

[134] Adi Shamir. Identity-Based Cryptosystems and Signature Schemes. In *CRYPTO*, volume 196 of *Lecture Notes in Computer Science*, pages 47–53. Springer, 1984.

[135] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[136] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient Public Key Encryption Based on Ideal Lattices. In *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 617–635. Springer, 2009.

[137] Jacques Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

[138] Peter van Emde Boas. Another NP-complete Partition Problem and the Complexity of Computing Short Vectors in a Lattice. Technical Report 81-04, Math Inst., University of Amsterdam, Amsterdam, 1981. Available at author's home page.

[139] Pascal Véron. Improved Identification Schemes based on Error-correcting Codes. *Appl. Algebra Eng. Commun. Comput.*, 8(1):57–69, 1996.

[140] Keita Xagawa and Keisuke Tanaka. Zero-Knowledge Protocols for NTRU: Application to Identification and Proof of Plaintext Knowledge. In *ProvSec*, volume 5848 of *Lecture Notes in Computer Science*, pages 198–213. Springer, 2009.

[141] Jin yi Cai and Ajay Nerurkar. Approximating the SVP to within a Factor $(1 + 1/\dim^{\text{epsilon}})$ is NP-hard under Randomized Reductions. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(59), 1997.