Lattices in constructions of dense sphere packings and certain cryptographic schemes

Cheng, Shantian

2016

Cheng, S. (2016). Lattices in constructions of dense sphere packings and certain cryptographic schemes. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/66449>

<https://doi.org/10.32657/10356/66449>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

**LATTICES IN CONSTRUCTIONS OF DENSE SPHERE
PACKINGS AND CERTAIN CRYPTOGRAPHIC
SCHEMES**

SHANTIAN CHENG

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

2016

**LATTICES IN CONSTRUCTIONS OF DENSE SPHERE
PACKINGS AND CERTAIN CRYPTOGRAPHIC
SCHEMES**

SHANTIAN CHENG

SHANTIAN CHENG

SCHOOL OF PHYSICAL AND MATHEMATICAL SCIENCES

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

2016

I dedicate this thesis
in token of affection and gratitude
to my beloved parents
Guangmin Cheng, Bo Xu,
and fiancée
Yujie Nan.

ACKNOWLEDGEMENTS

First of all, I would like to express my deep gratitude to my supervisors, Prof. San Ling and Prof. Chaoping Xing, for their invaluable guidance and encouragement during my PhD studies. It is a great privilege for me to accomplish my PhD studies under their supervision and support. They are tremendous role models for me to learn, not only in academia but also in daily life. As their student, I will try my best to live up to their expectations and achieve more success in the future.

I am much grateful to the collaborators in my two papers in cryptography, Dr. Ta Toan Khoa Nguyen, Prof. Huaxiong Wang, and Juanyang Zhang for their precious help and support. A special thank you goes to Dr. Ta Toan Khoa Nguyen, since the cooperation with him is really a valuable experience, from which I gained a lot of knowledge in cryptography and abilities in team coordination. In addition, I wish to express my sincere gratitude to Dr. Hyung Tae Lee for his generous help in my revision of the ISPEC'15 paper, and Ivan Tjuawinata for the helpful discussions on C++ implementation of some encryption schemes.

It is also a good opportunity for me to express my appreciation and thanks to all the anonymous reviewers, who have offered great help, advice and comments on my PhD research results. I sincerely admire their profound knowledge and rigorous scientific attitude in research, and I believe their precious advice have helped improve my results.

Furthermore, I would like to gratefully thank Dr. An Yang, Dr. Anderson Siang Jing Yeo, Dr. Guizhen Zhu for their generosity in offering valuable advice in career

planning and sharing their experience in job hunting. Their genuine help offers me a broader understanding of the future career of a PhD holder.

My deepest gratitude goes to my parents, Guangmin Cheng and Bo Xu. Although they totally could not understand what I am doing everyday and I have even lost my temper several times over the phone when I was caught in deep depression and stress, they still provide unending encouragement, support and patience, which have accompanied me throughout my studies.

I am deeply indebted to my fiancée, Yujie Nan for her unwavering love. I would never have been able to finish my PhD studies without her unconditional support. She is the one for me to have and to hold. She is the reason for me to be happy, strong and positive in life. With the submission of this thesis, we can finally bring an end to the long-distance engagement and put the wedding on the agenda. It is so inspiring to conceive the image that we strive together for our common better future that I can hardly wait.

Finally I have to thank Euclid for his “τριώβολον (three pence)”.

LIST OF WORKS

Below is the list of works done during my PhD studies in Nanyang Technological University.

1. Shantian Cheng. Improvement on Asymptotic Density of Packing Families Derived from Multiplicative Lattices. *Finite Fields and Their Applications*, vol. 36, pp. 133-150. (2015)
2. Shantian Cheng. Dense Packings from Algebraic Number Fields and Codes. arXiv:1506.00419 (2015), <http://arxiv.org/abs/1506.00419> Under review.
3. Shantian Cheng, Khoa Nguyen, Huaxiong Wang. Policy-Based Signature Scheme from Lattices. *Designs, Codes and Cryptography* (2015). In press, <http://dx.doi.org/10.1007/s10623-015-0126-y>.
4. Shantian Cheng, Juanyang Zhang. Adaptive-ID Secure Revocable Identity-Based Encryption from Lattices via Subset Difference Method. In: Lopez, J., Wu, Y. (eds.) 11th International Conference on Information Security Practice and Experience (ISPEC 2015), *Lecture Notes in Computer Science*, vol. 9065, pp. 283–297. Springer International Publishing (2015)

I independently worked out the results in [1,2], while I played an active/leading role in exploring and accomplishing the results in [3,4].

Contents

1. Introduction	1
1.1 Sphere Packing	2
1.2 Special Signature and Encryption Schemes	4
1.2.1 Policy-Based Signatures	6
1.2.2 Revocable Identity-Based Encryption	7
1.3 Contributions of the Thesis	8
1.4 Organization of the Thesis	11
2. Preliminaries	14
2.1 Lattices	14
2.2 Sphere Packings	16
2.3 Coding Theory	18
2.4 Some Cryptographic Tools from Lattices	19
2.4.1 Notations	19
2.4.2 Discrete Gaussian Distributions	21
2.4.3 Trapdoor Generation and Sampling Algorithms	22
2.4.4 Randomizing and Extending Control	23
2.4.5 Random Matrices in $\{-1, 1\}^{m \times k}$	24
2.4.6 The SIS Hardness Assumption	25
2.4.7 The LWE Hardness Assumption	26

3. Asymptotically Good Packing Families Derived from Multiplicative Lattices	29
3.1 Concatenation Based on Number Field $K = \mathbb{Q}(\sqrt{-3})$	29
3.2 Remarks on the Asymptotic Properties of Xing's Construction	31
3.2.1 Asymptotically Good Packing Family Derived from $\mathcal{O}_K^n$	32
3.2.2 Xing's Asymptotically Good Packing Family	33
3.3 New Method to Construct Asymptotically Good Packing Families . .	35
3.4 Concatenation with Multiplicative Lattices	37
3.4.1 Principal Lattices and Congruence Lattices	37
3.4.2 Lattice Dimension Augmentation for Full Rank Sublattices of A_{n-1}	38
3.4.3 Concatenation with Principal Lattices	39
3.4.4 Concatenation with Congruence Lattices	42
3.5 Comparison	45
4. Dense Packings from Algebraic Number Fields and Codes	47
4.1 Canonical Embedding of Algebraic Number Fields	47
4.2 Concatenation Based on General Algebraic Number Fields	50
4.2.1 Concatenation with One Code	51
4.2.2 Concatenation with a Family of Codes	54
4.2.3 Asymptotic Properties	58
4.2.4 Examples	61
5. Policy-Based Signature Schemes from Lattices	65
5.1 Policy-Based Signatures	65
5.1.1 Definitions	65
5.1.2 Security Requirements	66

5.2	Overview of Our Techniques	68
5.3	The Underlying Zero-Knowledge Argument System	71
5.3.1	Statistical Zero-Knowledge Argument Systems	71
5.3.2	A Zero-Knowledge Argument of Knowledge for the Relation R_{PBS}	72
5.4	A Policy-Based Signature Scheme from Lattices	82
5.4.1	Description of the Scheme	82
5.4.2	Analysis of the Scheme	86
5.5	An Extended Scheme Supporting Delegation	94
5.5.1	Delegatable Policy-Based Signatures	94
5.5.2	A Delegatable Policy-Based Signature Scheme from Lattices	99
6.	Revocable Identity-Based Encryption from Lattices via Subset Difference Method	110
6.1	Revocable Identity-Based Encryption	110
6.1.1	Definitions	110
6.1.2	Security Requirements	112
6.2	Overview of Our Techniques	114
6.3	Subset Difference Method	117
6.4	RIBE from Lattices via Subset Difference Method	120
6.4.1	Description of the Scheme	120
6.4.2	Correctness	125
6.4.3	Parameters	126
7.	Conclusions	129
	Bibliography	132

Appendix	147
A. Magma Code for Example 4.2.8	149
B. Proof of Lemma 5.3.3	152
C. Proof of Lemma 5.3.4	156
D. An Improved Forking Lemma	159
E. Security Analysis of Our RIBE Scheme	160
E.1 Abort-Resistant Hash Functions	160
E.2 Proof of Theorem E.0.3	162

List of Figures

5.1 A zero-knowledge argument of knowledge for the relation R_{PBS}	80
5.2 A zero-knowledge argument of knowledge for the relation R_{PBS} (cont.)	81
6.1 Example of SD scheme for $R = \{\text{id}_4, \text{id}_5, \text{id}_6, \text{id}_7\}$	119

List of Tables

3.1	Contributions from lattices and concatenated codes to asymptotic density exponent of packing families derived from principal and congruence lattices	46
4.1	Examples constructed from $\mathbb{Q}[x]/(x^4 - x^3 - x^2 + x + 1)$	63
4.2	Examples constructed from $\mathbb{Q}[x]/(x^3 + x^2 - 2x - 1)$	64
4.3	Examples constructed from $\mathbb{Q}[x]/(x^3 + x^2 - 1)$	64
4.4	Examples constructed from $\mathbb{Q}[x]/(x^6 + x^3 + 1)$	64
6.1	Comparison of our RIBE scheme with Chen et al.'s	116

ABSTRACT

In this thesis we concentrate on the usage of lattices to construct explicit sphere packings with high density, packing families with good asymptotic properties, and cryptographic schemes with special functionalities, including (delegatable) policy-based signature and revocable identity-based encryption.

First we introduce new methods using codes and lattices from number fields to construct dense packings in Euclidean spaces. Let $\omega = (-1 + \sqrt{-3})/2$. For any lattice $P \subseteq \mathbb{Z}^n$, $\mathcal{P} = P + \omega P$ is a subgroup of $\mathcal{O}_K^n$, where $\mathcal{O}_K = \mathbb{Z}[\omega] \subseteq \mathbb{C}$. As $\mathbb{C}$ is naturally isomorphic to $\mathbb{R}^2$, $\mathcal{P}$ can be regarded as a lattice in $\mathbb{R}^{2n}$. Let P be a multiplicative lattice (principal lattice or congruence lattice) introduced by Rosenbloom and Tsfasman. We concatenate a family of special codes with $t_{\mathfrak{P}}^\ell \cdot (P + \omega P)$, where $t_{\mathfrak{P}}$ is a generator of a non-zero prime ideal $\mathfrak{P}$ of $\mathcal{O}_K$. Applying this concatenation to a family of principal lattices, we obtain a new family with asymptotic density exponent $\lambda \geq -1.26532182283$, which is better than -1.87 given by Rosenbloom and Tsfasman by considering only principal lattice families. For a new family based on congruence lattices, the result is $\lambda \geq -1.26532181404$, which is better than -1.39 obtained by considering only congruence lattice families.

More generally, via the canonical $\mathbb{Q}$ -embedding of arbitrary number field K into $\mathbb{R}^{[K:\mathbb{Q}]}$, both the prime ideal $\mathfrak{p}$ and its residue field κ can be embedded as discrete subsets in $\mathbb{R}^{[K:\mathbb{Q}]}$. Thus we can concatenate the embedding image of the Cartesian product of n copies of $\mathfrak{p}$ together with the image of a length n code over κ . This

concatenation leads to a packing in the Euclidean space $\mathbb{R}^{n[K:\mathbb{Q}]}$. Moreover, we extend the single concatenation to multiple concatenations to obtain dense packings and asymptotically good packing families. For instance, with the help of **Magma**, we construct a 256-dimensional packing denser than the Barnes-Wall lattice BW_{256} .

Next we develop the recent techniques in lattice-based cryptography to construct a (delegatable) policy-based signature scheme and a revocable identity-based encryption scheme from lattice assumptions. Policy-based signature, introduced by Bellare and Fuchsbauer at PKC 2014, is a new type of digital signature in which a signer is only allowed to sign messages satisfying certain policy specified by the authority, but the signatures should not reveal the underlying policy. We adapt Langlois et al.’s zero-knowledge argument system (PKC 2014) for the Bonsai tree signature scheme (Eurocrypt 2010) to enable the prover to convince the verifier that its secret witness satisfies an additional condition. Making the protocol non-interactive via the Fiat-Shamir transformation, we obtain a policy-based signature scheme supporting polynomially many policies, which satisfies the two security requirements (simulatability and extractability) in the random oracle model. Furthermore, our construction can be efficiently extended to a delegatable policy-based signature, thanks to the hierarchical structure of the Bonsai tree.

In view of the expiration or revelation of the user’s private credential (or private key) in a realistic scenario, identity-based encryption (IBE) schemes with an efficient key revocation mechanism, or for short, revocable identity-based encryption (RIBE) schemes, become prominently significant. We present an RIBE scheme from lattices by combining two Agrawal et al.’s IBE schemes (Eurocrypt 2010) with the subset difference method. In particular, our scheme may serve as a solution to a question posed by Chen et al. (ACISP 2012).

1. INTRODUCTION

Generally, a lattice is a free $\mathbb{Z}$ -module of finite rank, equipped with a positive definite symmetric bilinear form. Since the 18th century, lattices, which were treated as $\mathbb{Z}^n$ with various symmetric bilinear forms (equivalently, quadratic forms), have served as a powerful tool in mathematics [19, 54, 86]. Gauss and Lagrange used binary quadratic forms (two dimensional lattices) to prove some profound results in number theory, e.g., quadratic reciprocity and two/four square theorems. Hermite generalized the method to positive definite quadratic forms of arbitrary dimension and determined an upper bound for the minimum value of any positive definite quadratic form on non-zero lattice points. Later Minkowski formalized the study of geometric methods, especially lattices, in number theory as “geometry of numbers”, which has influenced the study in arithmetic subgroups of Lie groups and modular functions.

For studying lattices in $\mathbb{R}^n$, the notion of quadratic forms is helpful for exploring especially the arithmetic properties. There is an alternative language that uses discrete additive subgroups of $\mathbb{R}^n$ with the Euclidean norm to define lattices. The latter definition provides a geometric perspective of lattices. With the help of the Gram matrix of quadratic forms and the Cholesky decomposition of positive definite matrices, the two notions of lattices are equivalently related [43].

In this thesis, we consider the application of lattices in constructing dense sphere packings and certain cryptographic schemes. The constructions are mainly based

on the geometric properties of lattices, thus we regard lattices as discrete subgroups of $\mathbb{R}^n$ with the Euclidean norm. The formal definition will be given in Section 2.1.

1.1 *Sphere Packing*

Sphere packing is a classical problem on how to pack non-overlapping equal spheres densely in $\mathbb{R}^N$. Specifically, this problem evolves into two concrete aspects. One is how to construct packings of larger density than the record (e.g. [43, Tables 1.2-1.3]) in Euclidean spaces of a specific dimension N . The other is how to construct families of packings with dimension $N \rightarrow \infty$ such that the asymptotic density exponent has small absolute value. If the centers of the packed spheres form a discrete additive subgroup of $\mathbb{R}^n$ (lattice), we call it a lattice packing.

Sphere packing has attracted the interest of numerous mathematicians for centuries. Many methods and results from different disciplines, such as discrete geometry, combinatorics, number theory and coding theory, etc. have been involved in this problem, while some explicit fascinating dense constructions and asymptotically good packing families have been found. For a detailed survey on the development in this field, the reader may refer to the books of Cassels [34], and Conway and Sloane [43].

Sphere packings are continuous analogues of error-correcting codes in the Hamming space [49]. The basic problem of error-correcting codes is to seek the maximum size of a code given length, alphabet size and minimal Hamming distance, in other words, dense packing of points in Hamming space such that each pair of distinct points is separated at least by the minimum Hamming distance. In the digital world, error-correcting codes are widely employed in information storage and transmission, for example, the blue ray storage format and the communication between

space stations and the Earth. Based on the similarities between sphere packings and error-correcting codes, the results in sphere packings can potentially contribute to the development of error-correcting codes.

Meanwhile, some constructions of dense lattice or non-lattice packings are inspired by constructions in coding theory. For example, similar to concatenated codes, a classical packing construction idea is to concatenate proper codes with special packings in $\mathbb{Z}^n$. This method can offer new packings denser than the original ones. There are five well-known constructions that originated from this idea, which are referred to as Constructions A,B,C (due to Leech and Sloane); D (due to Bos, Conway and Sloane); and E (due to Barnes and Sloane). More details about these constructions can be found in [43, 61, 110].

Particularly, Leech and Sloane’s “Construction A” method concatenates certain binary codes together with $2 \cdot \mathbb{Z}^n$ to construct new lattice packings (see details in [43, Chapter 5]). Bachoc [12] generalized the method to construct modular lattices using codes over finite involution algebras. In Construction C [110, Chapter 5], the binary expansion of the coordinates in $\mathbb{Z}^n$ is considered. A point is a packing center if and only if its first ℓ coordinate arrays are codewords in certain ℓ binary codes respectively.

Subsequently, Xing [109] further investigated the concatenating method. Instead of packings in $\mathbb{Z}^n$, he considered the packings in $\mathcal{O}_K^n$, where $\mathcal{O}_K = \mathbb{Z} \left[\frac{-1+\sqrt{-3}}{2} \right]$ denotes the ring of integers in the number field $\mathbb{Q}(\sqrt{-3})$. That is, for any non-zero prime ideal $\mathfrak{P}$ of $\mathcal{O}_K$, $\mathfrak{P}^n$ can be embedded as a lattice L in $\mathbb{R}^{2n}$ via the canonical $\mathbb{Q}$ -embedding of the special number field $\mathbb{Q}(\sqrt{-3})$ into $\mathbb{R}^2$. Hence any subset $\mathcal{P}$ of $\mathfrak{P}^n$ can be regarded as a packing in $\mathbb{R}^{2n}$. Then he replaced the binary expansion in Construction C by the $\mathfrak{P}$ -adic expansion, and concatenated the lattice L with some special codes over $\mathcal{O}_K/\mathfrak{P}$. This method produces several dense packings in

low dimensions attaining the best-known densities. One natural question is whether we can use arbitrary number fields instead of the special quadratic number field $\mathbb{Q}(\sqrt{-3})$ to generalize the constructing method.

For the asymptotic aspect, Minkowski gave a nonconstructive bound that asserts that there exists some packing family $\mathcal{F}$ such that the asymptotic density exponent $\lambda(\mathcal{F})$ satisfies $\lambda(\mathcal{F}) \geq -1$ (see [34, p.184]). It is of interest to construct families with $\lambda(\mathcal{F}) > -\infty$ explicitly (such families are called *asymptotically good*). Known constructive bounds for families with polynomial or exponential construction complexity in terms of dimension N are listed in the book of Litsyn and Tsfasman [106, p.628]. To the best of our knowledge, they still remain the best so far.

Xing applied his methods [109] to concatenate ℓ codes with a packing $\mathcal{P}^{(N)}$ in $\mathcal{O}_K^N$ of fixed minimum Euclidean distance. When N tends to ∞ , the number of codes $\ell \rightarrow \infty$, and the family $\{\mathcal{P}^{(N)}\}$ is not asymptotically good. However, the resulting packing family from concatenation is asymptotically good. He obtained an unconditional bound of asymptotic density exponent $\lambda \geq -1.2653$. As the concatenating method seems promising, can we further explore it?

1.2 Special Signature and Encryption Schemes

At the beginning, the applications of lattices to cryptology were negative, that is, several attacks on some cryptosystems in special settings were developed based on lattice reduction algorithms. For any lattice in $\mathbb{Z}^n$, lattice reduction algorithms try to efficiently output a basis consisting of reasonable short lattice vectors which are geometrically not far from being orthogonal to each other [86]. Lenstra, Lenstra and Lovász (LLL) [67] offered a celebrated polynomial-time algorithm which computes a so-called LLL-reduced basis of a lattice. They applied it to factor rational

polynomials in polynomial time. Later the LLL algorithm was used in public-key cryptanalysis to investigate the weaknesses in some cryptosystems, like schemes based on the knapsack problem [89], RSA cryptosystem in low public or private exponent [44], etc.

Since Ajtai pioneered a worst-case to average-case connection for some versions of the shortest lattice vector problem (SVP), lattices gradually found positive roles in cryptology, especially in developing public-key cryptosystems based on problems different from the integer factorization or discrete logarithm problem. Specifically, there are two natural average-case hardness problems: Small Integer Solution (SIS) and Learning with Errors (LWE). Ajtai [6], and Micciancio and Regev [81] showed that solving SIS is at least as hard as approximating a number of standard worst-case lattice problems, such as the shortest independent vectors problem (SIVP), to within an approximation factor that is polynomial in the lattice dimension. Regev [94] proved that under a quantum reduction, LWE enjoys similar worst-case hardness properties. That is, if there is an efficient algorithm for solving LWE, then it implies efficient quantum algorithms for approximating SIVP. Subsequently, Peikert [90] and Brakerski et al. [28] proved that under respective classical (non-quantum) reductions, the hardness of LWE is built upon the hardness of the decision version of the shortest vector problem (GapSVP). Under quantum reductions, GapSVP and SIVP are qualitatively equivalent, up to polynomial approximation factors. Moreover, there are classical reductions from GapSVP to SIVP. However, to the best of our knowledge, it remains an open problem whether a classical polynomial time reduction from SIVP to GapSVP can be obtained [28, 76].

Based on the above facts, most current lattice-based cryptographic schemes are established based on SIS or LWE directly. For instance, the SIS problem has played a role as the basis for one-way [6] and collision-resistant hash functions [55], iden-

tification schemes [62, 72, 82], and digital signatures [23, 33, 53, 74, 80], etc. The LWE problem has served as the foundation for secure public-key encryption schemes against both chosen-plaintext [69, 91, 94] and chosen-ciphertext [80, 90, 92] attacks, oblivious transfer [91], identity-based encryption [2, 4, 33, 53], various forms of leakage-resilient cryptography (e.g., [8, 10, 56]), fully homomorphic encryption [25, 26, 29] (following the pioneering work of Gentry [52]), etc. All these schemes enjoy provable security under worst-case hardness assumptions and asymptotic efficiency. Moreover, they are conjectured to resist quantum computers.

In this thesis, we use lattices to construct special cryptosystems, including a (delegatable) policy-based signature scheme based on SIS, and a revocable identity-based encryption scheme based on LWE.

1.2.1 Policy-Based Signatures

Policy-based signature (PBS), introduced by Bellare and Fuchsbauer [14], is a new type of privacy-enhanced digital signature in which a signer is only allowed to sign messages that satisfy certain policy specified by the authority, but the signatures should not reveal the underlying policy. Having these nice features, PBS may find applications in a corporate environment, where the authority may want to allow the employees to sign some documents on behalf of the corporation, but with internal restrictions on the signing capability of different employees based on their positions or functions. For example, the employees in the IT office can only sign the budget on IT related supplies, while the employees in the marketing office can only sign the budget on branding. From the theoretical point of view, PBS also attracts attention, because it can serve as an umbrella notion that implies many existing types of signatures, including group signatures [36], ring signatures [95], mesh signatures [22], attribute-based signatures [78], and signatures of knowledge [35].

In their pioneering work, Bellare and Fuchsbauer defined two strong security notions for PBS: simulatability and extractability. They also provided generic constructions of secure policy-based signatures, and a concrete instantiation based on pairings. Furthermore, they extended the PBS framework to support delegation of signing privileges - a useful feature in hierarchical settings. For instance, the executive Alice can issue the manager Bob a key that allows Bob to sign a contract with Company X. Using the key, Bob can further issue the employee Tom another key that allows Tom to sign the contract with Company X in one week. As a result, there are two policies associated to Tom's key, one is "to sign the contract with Company X", the other is "to sign the contract in one week". This feature can be realized by delegatable policy-based signature (DPBS).

In recent years, many standard signature schemes from lattices have been proposed, in both the standard model [23, 33, 48, 75, 80] and the random oracle model [47, 53, 73, 74]. Lattice-based signatures with privacy-enhanced features also receive a lot of attention, including blind signatures [97], ring signatures [27], and especially, group signatures [32, 57, 64, 65, 71, 87]. However, no construction of policy-based signatures from lattices has been known so far. Can we enrich the field by providing such a scheme?

1.2.2 Revocable Identity-Based Encryption

Since Shamir [102] proposed the concept of identity-based encryption (IBE), in which the sender may use the recipient's identity as a public key to encrypt a message, it was a longstanding open problem to construct a practical IBE scheme. Later, Boneh and Franklin [17] proposed the first practical IBE scheme using pairings. Since then, IBE has become extensively attractive from both the theoretical and practical perspectives. As a result, many IBE schemes have been proposed based

on various tools and assumptions such as quadratic residue modulo a hard-to-factor integer [18, 42], and lattices [2, 4, 53] as well as pairings [16, 107, 108].

In the practical use of IBE, a user's private key may be expired or revealed for various reasons. In such situations, an efficient key revocation mechanism is needed to keep the whole system stable and secure. To handle this issue, various solutions have been presented based on pairings. The first revocable IBE (RIBE) scheme was proposed by Boneh and Franklin [17], but their suggestion is inefficient in terms of the workload of the key authority. Later, Boldyreva et al. [15] presented the first selective-ID secure scalable RIBE scheme by combining a fuzzy IBE scheme [99] and the complete subtree (CS) method [83]. The first adaptive-ID secure scalable RIBE scheme was proposed by Libert and Vergnaud [68].

Subsequently, Seo and Emura [101] presented an enhanced security model of RIBE by considering one special realistic threat, which is called the decryption key exposure attack, and provided a concrete construction based on pairings. Recently, Lee et al. [66] succeeded in constructing an adaptively secure RIBE scheme based on pairings using the subset difference (SD) method.

However, as a counterpart of pairings, lattices have not been extensively used to construct RIBE schemes. To the best of our knowledge, in terms of RIBE schemes, only Chen et al. [37] offered a selective-ID secure RIBE scheme from lattices. Besides, the problem of constructing adaptive-ID secure RIBE schemes from lattices remains to be solved.

1.3 Contributions of the Thesis

In this thesis we concentrate on the use of lattices to construct explicit sphere packings with high density, packing families with good asymptotic properties, and

cryptographic schemes with special functionalities, including (delegatable) policy-based signature and revocable identity-based encryption. Our main contributions are as follows:

Contribution 1. We further explore the concatenating method of Xing to obtain another method to construct asymptotically good packing families. Compared with Xing’s construction, we employ asymptotically good packing families and concatenate finitely many codes to them. The number of codes remains finite as the packing dimension $N \rightarrow \infty$.

Explicitly, we apply the generalized concatenating method to Rosenbloom and Tsfasman’s multiplicative lattices in function fields (see [96]), and we get two asymptotically good families with bounds $\lambda \geq -1.26532182282$ (principal lattice case) and $\lambda \geq -1.26532181404$ (congruence lattice case), while the bounds for multiplicative lattice families provided in [96] are -1.87 and -1.39 respectively. Hence our construction improves the asymptotic density of packing families derived from multiplicative lattices. The result is published in [39].

Contribution 2. We extend Xing’s method to a general level, i.e., we employ ideals in $\mathcal{O}_K$ instead of $\mathbb{Z} \left[\frac{-1+\sqrt{-3}}{2} \right]$, where $\mathcal{O}_K$ denotes the ring of integers in an arbitrary number field K . Suppose the extension degree of K over $\mathbb{Q}$ is m . Minkowski interpreted the elements in K as points in the m -dimensional Euclidean space $\mathbb{R}^m$. The interpretation is called “the Minkowski Theory” in algebraic number theory (see [84, Section I.5] or [106, Section 5.3]).

By Minkowski’s interpretation, we can use the canonical $\mathbb{Q}$ -embedding to construct lattices in $\mathbb{R}^m$ from ideals in $\mathcal{O}_K$. In this way, for a non-zero prime ideal $\mathfrak{p}$ of $\mathcal{O}_K$, the Cartesian product of n copies of $\mathfrak{p}$ can be embedded as a lattice in $\mathbb{R}^{mn}$. Meanwhile, the codes defined over $\mathcal{O}_K/\mathfrak{p}$ with length n can also be embedded as finite subsets of $\mathbb{R}^{mn}$. Then we can proceed with the concatenating method as

Construction A on these two kinds of subsets of $\mathbb{R}^{mn}$ to construct dense packings. For instance, we construct a 256-dimensional packing with center density δ satisfying $\log_2 \delta \geq 208.09$, which is larger than 192 of the Barnes-Wall lattice BW_{256} (see the table of dense packings in [43, Table 1.3]). Furthermore, for different choices of number fields and prime ideals, we also provide several asymptotically good packing families. The result is given in [38].

Contribution 3. We introduce the first policy-based signature (PBS) scheme from lattice assumptions. To the best of our knowledge, this is also the first concrete instantiation of PBS that is believed to be quantum-resistant. Moreover, our construction can easily be extended to support the desirable delegation feature. Our scheme satisfies strong security notions in the random oracle model. Specifically, the scheme is extractable based on the assumed hardness of the Small Integer Solution (SIS) problem, and it is simulatable against arbitrary (even computationally unbounded) adversaries.

Towards our goal, we develop the argument system given by Langlois et al. [65], so that the prover can convince the verifier in zero-knowledge that the prover knows a secret valid message-signature pair for the Bonsai tree signature [33], *and* the secret message satisfies some additional relation. Apart from PBS, such protocols potentially can be used as building blocks in designing various privacy-enhanced cryptographic constructions, e.g., group signatures [36], anonymous credentials [31], compact e-cash [30], etc. Hence, we believe that our technical approach is of independent interest in the field of lattice-based cryptography. The result is published in [40].

Contribution 4. We present the first adaptive-ID secure revocable identity-based encryption scheme from lattices via the subset difference (SD) method in the standard model. To this end, we employ two adaptive-ID secure identity-based

encryption (IBE) schemes for an identity and a time, respectively, proposed by Agrawal et al. [2]. Then, we combine them with the SD method for the revocation and use a secret sharing scheme utilized in fuzzy IBE proposed by Agrawal et al. [5] to share one common random vector in the SD method.

We demonstrate that our construction achieves adaptive-ID security in the standard model under the **LWE** assumption. Our security proof follows Libert and Vergnaud’s security model [68] with a slight modification that uses a game which expresses a strong privacy property, called *indistinguishable from random*, as in [2]. As a result, we provide a solution to a question posed by Chen et al. [37]. The result is presented in [41].

1.4 Organization of the Thesis

The thesis is organized into seven chapters. In Chapter 1, we first describe the roles of lattices in the development of sphere packing and cryptology, together with the problems derived from Xing’s concatenating constructions [109] and the lack of lattice-based constructions of (delegatable) policy-based signature and revocable identity-based encryption. Then we briefly introduce the contributions of this thesis on the corresponding problems.

In Chapter 2, we provide the formal definitions and necessary properties of lattices, sphere packing densities, codes, and some cryptographic tools from lattices including the discrete Gaussian distribution, trapdoor generation and sampling algorithms, randomizing and extending control, **SIS** and **LWE** hardness assumptions, etc. They play important roles in our constructions introduced in the subsequent chapters.

In Chapter 3, we first recall the concatenation based on the special quadratic

number field $K = \mathbb{Q}(\sqrt{-3})$. Then we give some remarks on Xing's construction [109] in comparison with the basic concatenation with $\mathcal{O}_K^n$. The general description of our new construction comes thereafter, and as an application, we apply the new method on multiplicative lattices to get asymptotically good packing families. In the end, we compare the results and conclude our contribution.

In Chapter 4, we first review the canonical $\mathbb{Q}$ -embedding of algebraic number fields and the properties of the embedding images of ideals. Then we describe our generalized concatenating method based on the embedding images of prime ideals and codes. Several examples of dense packings and asymptotically good packing families for different choices of number fields and prime ideals are provided. The detailed numerical results are provided in Tables 4.1 - 4.4.

Since several lattice-based techniques are integrated in the construction of our cryptographic schemes, we give an overview of our constructions after the formal definition of the respective schemes (policy-based signature in Chapter 5 and revocable identity-based encryption in Chapter 6) to make the description clearer. In Chapter 5, after the overview, we introduce a zero-knowledge argument of knowledge for an NP-relation, which is an important building block in the signing phase of our construction. Subsequently, our lattice-based construction of policy-based signature scheme is provided in detail. Finally, we extend our construction to a scheme supporting signing key delegation.

Chapter 6 has a similar structure as Chapter 5. After the formal definition of the revocable identity-based encryption scheme and the overview of our techniques, we introduce the subset difference method, which serves as the key assigning framework of our construction. Then we describe our lattice-based construction of revocable identity-based encryption scheme in detail.

Chapter 7 acts as a short summary of our results, and a remaining problem

is briefly introduced for future investigations. Meanwhile, we list several possible relevant topics for consideration. To keep the main body of the thesis balanced and complete, the **Magma** code, proofs of some lemmas and theorem, as well as a description of an improved Forking Lemma, are included in the Appendix.

2. PRELIMINARIES

2.1 Lattices

Generally, a lattice is a pair (L, b) , where L is a free $\mathbb{Z}$ -module of finite rank and $b : L \times L \rightarrow \mathbb{R}$ is a positive definite symmetric bilinear form (see [13]). Particularly, we use the following formal definition (see [43, 79, 84]).

Definition 2.1.1. A lattice L is an additive subgroup of $\mathbb{R}^n$ that can be expressed as

$$L = \mathbb{Z}\mathbf{b}_1 + \cdots + \mathbb{Z}\mathbf{b}_m$$

with $\mathbb{R}$ -linearly independent vectors $\mathbf{b}_1, \dots, \mathbf{b}_m$ of $\mathbb{R}^n$. We say the rank of L is m and its dimension is n . If $m = n$, the lattice L is called a full-rank lattice in $\mathbb{R}^n$. (Note that as a Euclidean vector space, $\mathbb{R}^n$ has a natural positive definite symmetric bilinear form, that is, the dot product $\langle, \rangle : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$. The Euclidean distance (or Euclidean norm) of a vector $\mathbf{v} = (a_1, \dots, a_n) \in \mathbb{R}^n$ is defined as $\|\mathbf{v}\| := \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle} = \sqrt{a_1^2 + \cdots + a_n^2}$. The dot product can be restricted to the subset L of $\mathbb{R}^n$, thus $(L, \langle, \rangle)$ satisfies the general definition of a lattice. Since the dot product is the only positive definite symmetric bilinear form considered throughout the thesis, we simply use L to denote the lattice.)

We define $\mathbf{B}$ as the $n \times m$ matrix whose columns are $\mathbf{b}_1, \dots, \mathbf{b}_m$, then $\mathbf{B}$ is referred to as a basis of the lattice L . In this case, L is also denoted by $L(\mathbf{B})$. The span of the basis $\mathbf{B}$ is the $\mathbb{R}$ -linear vector space spanned by its column vectors, i.e.,

$\text{span}(\mathbf{B}) := \{\mathbf{B}\mathbf{y} : \mathbf{y} \in \mathbb{R}^m\}$. The set

$$P(\mathbf{B}) := \left\{ \sum_{i=1}^m x_i \mathbf{b}_i : x_i \in \mathbb{R}, 0 \leq x_i < 1 \right\}$$

is called a fundamental region of the lattice L with respect to the basis $\mathbf{B}$.

If two given bases $\mathbf{B}_1, \mathbf{B}_2 \in \mathbb{R}^{n \times m}$ generate the same lattice, i.e., $L(\mathbf{B}_1) = L(\mathbf{B}_2)$, then we say that $\mathbf{B}_1, \mathbf{B}_2$ are equivalent.

Proposition 2.1.2. *Two bases $\mathbf{B}_1, \mathbf{B}_2$ are equivalent if and only if $\mathbf{B}_2 = \mathbf{B}_1 \mathbf{U}$ for some unimodular matrix $\mathbf{U}$ (i.e., a matrix in $\mathbb{Z}^{m \times m}$ with determinant ± 1).*

Any basis $\mathbf{B}$ can be transformed into an orthogonal basis for the vector space $\text{span}(\mathbf{B})$ using the Gram-Schmidt orthogonalization method. Specifically, for $\mathbf{B} = [\mathbf{b}_1 | \cdots | \mathbf{b}_m]$, the column vectors of the orthogonalized basis $\tilde{\mathbf{B}} = [\tilde{\mathbf{b}}_1 | \cdots | \tilde{\mathbf{b}}_m]$ are defined iteratively as $\tilde{\mathbf{b}}_1 = \mathbf{b}_1$, for $i \in \{2, \dots, m\}$: $\tilde{\mathbf{b}}_i = \mathbf{b}_i - \sum_{j < i} \mu_{i,j} \tilde{\mathbf{b}}_j$, where $\mu_{i,j} = \frac{\langle \mathbf{b}_i, \tilde{\mathbf{b}}_j \rangle}{\langle \tilde{\mathbf{b}}_j, \tilde{\mathbf{b}}_j \rangle}$. Geometrically, $\tilde{\mathbf{b}}_i$ is the component of $\mathbf{b}_i$ orthogonal to the space $\text{span}(\mathbf{b}_1, \dots, \mathbf{b}_{i-1})$.

Definition 2.1.3. Let L be a lattice in $\mathbb{R}^n$ with a basis $\mathbf{B} \in \mathbb{R}^{n \times m}$. The determinant of L , denoted by $\det(L)$, is defined as the m -dimensional volume of the fundamental region, i.e.,

$$\det(L) = \text{vol}(P(\mathbf{B})) = \prod_{i=1}^m \|\tilde{\mathbf{b}}_i\|.$$

If L is a full-rank lattice in $\mathbb{R}^n$ and we take a large sphere K centered at the origin with radius r , then the number of lattice points inside the sphere K is asymptotically equivalent to $\text{vol}(K)/\det(L)$ as the radius r grows to infinity.

A natural concern is whether the determinant of L is well-defined, that is, whether the fundamental regions with respect to two equivalent bases $\mathbf{B}_1, \mathbf{B}_2$ have the same volume. We can get the positive answer from the following proposition.

Proposition 2.1.4. *For any basis $\mathbf{B} \in \mathbb{R}^{n \times m}$, we have $\det(L(\mathbf{B})) = \sqrt{\det(\mathbf{B}^\top \mathbf{B})}$. In particular, if $\mathbf{B} \in \mathbb{R}^{n \times n}$ is a square matrix, then $\det(L(\mathbf{B})) = |\det(\mathbf{B})|$.*

To prove the proposition, we observe that the basis $\mathbf{B}$ and its Gram-Schmidt orthogonalized basis $\tilde{\mathbf{B}}$ satisfy the relation $\mathbf{B} = \tilde{\mathbf{B}}\mathbf{T}$, where $\mathbf{T}$ is an upper triangular matrix with 1's on the diagonal, and $\mu_{i,j}$ in the (j, i) -position for all $j < i$. Then

$$\sqrt{\det(\mathbf{B}^\top \mathbf{B})} = \sqrt{\det(\mathbf{T}^\top \tilde{\mathbf{B}}^\top \tilde{\mathbf{B}} \mathbf{T})} = \sqrt{\det(\tilde{\mathbf{B}}^\top \tilde{\mathbf{B}})} = \prod_{i=1}^m \|\tilde{\mathbf{b}}_i\| = \det(L(\mathbf{B})).$$

Consequently, we can deduce the determinant is well-defined. Specifically, if $\mathbf{B}_1, \mathbf{B}_2$ are equivalent, then $\mathbf{B}_2 = \mathbf{B}_1 \mathbf{U}$ for some unimodular matrix $\mathbf{U}$. Thus we have

$$\text{vol}(P(\mathbf{B}_2)) = \sqrt{\det(\mathbf{B}_2^\top \mathbf{B}_2)} = \sqrt{\det(\mathbf{U}^\top \mathbf{B}_1^\top \mathbf{B}_1 \mathbf{U})} = \sqrt{\det(\mathbf{B}_1^\top \mathbf{B}_1)} = \text{vol}(P(\mathbf{B}_1)).$$

2.2 Sphere Packings

Let $\mathcal{P}$ be the set of centers of packed spheres and let $\mathcal{B}_N(R)$ be the set

$$\left\{ \mathbf{v} = (a_1, \dots, a_N) \in \mathbb{R}^N : \|\mathbf{v}\| = \sqrt{a_1^2 + \dots + a_N^2} \leq R \right\}.$$

As a sphere packing construction is uniquely determined by the arrangement of the sphere centers, we just use $\mathcal{P}$ to denote the corresponding packing.

For a packing $\mathcal{P}$, the radius of the equal packed spheres is $r = d_E(\mathcal{P})/2$, where $d_E(\mathcal{P})$ is the minimum Euclidean distance between two distinct points in $\mathcal{P}$. Then the density $\Delta(\mathcal{P})$ of the packing $\mathcal{P}$ is defined as

$$\Delta(\mathcal{P}) = \limsup_{R \rightarrow \infty} \frac{|\mathcal{P} \cap \mathcal{B}_N(R)| \cdot r^N \cdot V_N}{\text{vol}(\mathcal{B}_N(R+r))},$$

2. Preliminaries

where V_N is the volume of the unit sphere in $\mathbb{R}^N$, that is,

$$V_N = \begin{cases} \frac{\pi^{N/2}}{(N/2)!}, & \text{if } N \text{ is even;} \\ \frac{2^N \pi^{(N-1)/2} ((N-1)/2)!}{N!}, & \text{if } N \text{ is odd.} \end{cases}$$

The sphere packing problem is to construct packings obtaining large density $\Delta(\mathcal{P})$. Moreover, the center density $\delta(\mathcal{P})$ and density exponent $\lambda(\mathcal{P})$ are defined respectively as

$$\delta(\mathcal{P}) = \frac{\Delta(\mathcal{P})}{V_N}, \quad \lambda(\mathcal{P}) = \frac{1}{N} \log_2 \Delta(\mathcal{P}).$$

If $\mathcal{P} = L$ forms a lattice, the density of lattice packing L can be simplified as

$$\Delta(L) = \frac{(d_E(L)/2)^N V_N}{\det(L)},$$

where $\det(L)$ is the determinant of L . Note that from Minkowski's Convex Body Theorem (see [79, Theorem 1.4] or [86, Theorem 4]), for any rank N lattice L , the minimum Euclidean distance of L satisfies

$$d_E(L) \leq \sqrt{N} \det(L)^{1/N}. \tag{2.1}$$

When we explore the asymptotic behavior of a packing family $\mathcal{F} = \{\mathcal{P}^{(N)}\}$ as the dimension N of the packing $\mathcal{P}^{(N)}$ tends to ∞ , we consider the asymptotic density exponent of the family

$$\lambda(\mathcal{F}) = \limsup_{N \rightarrow \infty} \frac{1}{N} \log_2 \Delta(\mathcal{P}^{(N)}). \tag{2.2}$$

Note that by Stirling's formula, as $N \rightarrow \infty$, we have

$$\log_2 V_N = -\frac{N}{2} \log_2 \frac{N}{2\pi e} - \frac{1}{2} \log_2(N\pi) - \epsilon, \quad (2.3)$$

where $0 < \epsilon < (\log_2 e)/(6N)$.

2.3 Coding Theory

We recall some notations and results in coding theory.

For a q -ary code C , let $n(C)$, $M(C)$ and $d_H(C)$ denote the length, the size, and the minimum Hamming distance of C , respectively. Such a code is usually referred to as an $(n(C), M(C), d_H(C))$ -code. Moreover, the relative minimum distance $\varrho(C)$ and the rate $R(C)$ are defined respectively as

$$\varrho(C) = \frac{d_H(C)}{n(C)}, \quad R(C) = \frac{\log_q M(C)}{n(C)}.$$

In particular, if a code C forms a linear space over $\mathbb{F}_q$, then the code C is called an $[n(C), k(C), d_H(C)]$ -linear code over $\mathbb{F}_q$, where $k(C) := \log_q M(C)$ is called the dimension of C . In this case, the rate $R(C) = \frac{k(C)}{n(C)}$.

Let U_q be the set of the ordered pair $(\varrho, R) \in \mathbb{R}^2$, for which there exists a family $\{C_i\}_{i=0}^\infty$ of q -ary codes such that $n(C_i)$ increasingly goes to ∞ as i tends to ∞ and

$$\varrho = \lim_{i \rightarrow \infty} \varrho(C_i), \quad \text{and} \quad R = \lim_{i \rightarrow \infty} R(C_i).$$

Here is a result on U_q :

Proposition 2.3.1 ([106, Section 1.3.1] or [109, Proposition 3.1]). *There exists a*

continuous function $R_q(\varrho)$, $\varrho \in [0, 1]$, such that

$$U_q = \{(\varrho, R) \in \mathbb{R}^2 : 0 \leq R \leq R_q(\varrho), 0 \leq \varrho \leq 1\}.$$

Moreover, $R_q(0) = 1$, $R_q(\varrho) = 0$ for $\varrho \in [(q-1)/q, 1]$, and $R_q(\varrho)$ decreases on the interval $[0, (q-1)/q]$.

For $0 < \varrho < 1$, the q -ary entropy function is given as

$$H_q(\varrho) = \varrho \log_q(q-1) - \varrho \log_q \varrho - (1-\varrho) \log_q(1-\varrho).$$

The asymptotic Gilbert-Varshamov (GV) bound indicates that

$$R_q(\varrho) \geq R_{GV}(q, \varrho) := 1 - H_q(\varrho), \quad \text{for all } \varrho \in \left(0, \frac{q-1}{q}\right). \quad (2.4)$$

Moreover, for any given rate R , there exists a family of linear codes which meets the GV bound (see [77, Section 17.7]).

2.4 Some Cryptographic Tools from Lattices

We discuss several cryptographic tools from lattices that will be employed in Chapters 5 and 6.

2.4.1 Notations

The security parameter is denoted by n and its unary representation is denoted by 1^n . A function of n is called negligible, denoted by $\text{negl}(n)$, if it vanishes faster than the reciprocal of any polynomial in n . We say that an event occurs with overwhelming probability if the probability is at least $1 - \epsilon(n)$ for some negligible function ϵ . If D_1

2. Preliminaries

and D_2 are two distributions (more formally, two ensembles of distributions indexed by n) over the same countable support S , then their statistical distance is defined as $\Delta(D_1, D_2) = \frac{1}{2} \sum_{x \in S} |D_1(x) - D_2(x)|$. Two distributions are statistically close if their statistical distance is negligible.

All vectors are assumed to be column vectors. The Hamming weight of a binary vector $\mathbf{x}$ is denoted by $wt(\mathbf{x})$. The concatenation of vectors $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y} \in \mathbb{R}^k$ is denoted by $(\mathbf{x}||\mathbf{y}) \in \mathbb{R}^{m+k}$, while the column concatenation of matrices $\mathbf{A} \in \mathbb{R}^{n \times m}$ and $\mathbf{B} \in \mathbb{R}^{n \times k}$ is denoted by $[\mathbf{A}|\mathbf{B}] \in \mathbb{R}^{n \times (m+k)}$.

The set of all finite binary strings is denoted by $\{0, 1\}^*$. The set of integers $\{1, \dots, k\}$ is denoted by $[k]$, while the set of integers $\{a, a+1, \dots, b-1, b\}$ is denoted by $[a; b]$. The symmetric group of all permutations of k elements is denoted by S_k . If S is a finite set, then $y \xleftarrow{\$} S$ means that y is chosen uniformly at random from S . The bit XOR operation is denoted by $\oplus$. The abbreviation PPT stands for “probabilistic polynomial-time”.

For a matrix $\mathbf{S} = [\mathbf{s}_1 | \dots | \mathbf{s}_k] \in \mathbb{R}^{m \times k}$, the norm of $\mathbf{S}$ is defined as $\|\mathbf{S}\| = \max_{1 \leq i \leq k} \|\mathbf{s}_i\|$, where $\|\mathbf{s}\|$ denotes the Euclidean norm of the column vector $\mathbf{s}$. Moreover, $\tilde{\mathbf{S}} := [\tilde{\mathbf{s}}_1 | \dots | \tilde{\mathbf{s}}_k] \in \mathbb{R}^{m \times k}$ represents the Gram-Schmidt orthogonalization of $\mathbf{S}$ provided the vectors $\mathbf{s}_1, \dots, \mathbf{s}_k$ are $\mathbb{R}$ -linearly independent. Let $\|\tilde{\mathbf{S}}\|$ denote the Gram-Schmidt norm of $\mathbf{S}$.

Throughout the thesis, we write $\log$ for $\log_{10}$, and $\ln$ for $\log_e$.

In lattice-based cryptography, we only consider integer lattices, that is, sublattices of $\mathbb{Z}^m$. As we have used L to represent a general lattice, here we use Λ instead to represent an integer lattice. In particular, the following two integer lattices are widely considered.

Definition 2.4.1. For integer $q \geq 2$, matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and vector $\mathbf{u} \in \mathbb{Z}_q^n$, define

$$\begin{aligned}\Lambda_q^\perp(\mathbf{A}) &= \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{0} \bmod q\}, \\ \Lambda_q^{\mathbf{u}}(\mathbf{A}) &= \{\mathbf{e} \in \mathbb{Z}^m : \mathbf{A}\mathbf{e} = \mathbf{u} \bmod q\}.\end{aligned}$$

Note that if $\mathbf{t} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$ then $\Lambda_q^{\mathbf{u}}(\mathbf{A})$ is a translation of $\Lambda_q^\perp(\mathbf{A})$ by $\mathbf{t}$, i.e., $\Lambda_q^{\mathbf{u}}(\mathbf{A}) = \Lambda_q^\perp(\mathbf{A}) + \mathbf{t}$.

By the fact that for any constant $C > 1$ and any integer $m \geq Cn \log q$, the columns of a uniformly random chosen matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ can $\mathbb{Z}_q$ -linearly generate the whole space $\mathbb{Z}_q^n$ except for some negligible probability (see, e.g., [33, 94]), we assume throughout this thesis that such a uniformly random $\mathbf{A}$ generates $\mathbb{Z}_q^n$.

2.4.2 Discrete Gaussian Distributions

Definition 2.4.2. Let Λ be a lattice in $\mathbb{Z}^m$. For any vector $\mathbf{c} \in \mathbb{R}^m$ and any parameter $\sigma \in \mathbb{R}_{>0}$, define:

$$\rho_{\sigma, \mathbf{c}}(\mathbf{x}) = \exp\left(-\pi \frac{\|\mathbf{x} - \mathbf{c}\|^2}{\sigma^2}\right) \quad \text{and} \quad \rho_{\sigma, \mathbf{c}}(\Lambda) = \sum_{\mathbf{x} \in \Lambda} \rho_{\sigma, \mathbf{c}}(\mathbf{x}).$$

The discrete Gaussian distribution over Λ with center vector $\mathbf{c}$ and parameter σ is

$$\mathcal{D}_{\Lambda, \sigma, \mathbf{c}}(\mathbf{y}) = \frac{\rho_{\sigma, \mathbf{c}}(\mathbf{y})}{\rho_{\sigma, \mathbf{c}}(\Lambda)}, \quad \text{for } \forall \mathbf{y} \in \Lambda.$$

If the center vector $\mathbf{c} = \mathbf{0}$, we simply use the notations ρ_σ and $\mathcal{D}_{\Lambda, \sigma}$.

Micciancio and Regev [81] proved the following lemma that the Euclidean norm of vectors sampled from a discrete Gaussian distribution is small with high probability. Gentry et al. [53] applied it to introduce lattice trapdoors.

Lemma 2.4.3 ([53]). *For any basis $\mathbf{S}$ of an n -dimensional lattice Λ , any vector $\mathbf{c} \in \text{span}(\mathbf{S})$, and $\sigma \geq \|\tilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log n})$,*

$$\Pr [\mathbf{x} \leftarrow \mathcal{D}_{\Lambda, \sigma, \mathbf{c}} : \|\mathbf{x} - \mathbf{c}\| > \sigma \cdot \sqrt{n}] \leq \text{negl}(n).$$

When the l_∞ -norm of vectors is considered, we have that for any $\sigma_0 \geq \omega(\sqrt{\log n})$ and any $c \in \mathbb{R}$,

$$\Pr [x \leftarrow \mathcal{D}_{\mathbb{Z}, \sigma_0, c} : |x - c| > \sigma_0 \cdot \log n] \leq \text{negl}(n).$$

2.4.3 Trapdoor Generation and Sampling Algorithms

Many lattice-based cryptographic constructions rely on two important algorithms: the trapdoor generation algorithm [7, 9, 80] that produces a matrix $\mathbf{A}$ statistically close to uniform over $\mathbb{Z}_q^{n \times m}$ along with a short basis $\mathbf{S}$ of the lattice $\Lambda_q^\perp(\mathbf{A})$; and the preimage sampling algorithm [53] that uses a short basis of $\Lambda_q^\perp(\mathbf{A})$ to generate a short vector in $\Lambda_q^\mathbf{u}(\mathbf{A})$, for any $\mathbf{u}$ in the range of $\mathbf{A}$. We summarize these algorithms in the following lemma.

Lemma 2.4.4. *Let $q \geq 3$ and $m \geq 2n \log q$ be integers. There is a PPT algorithm $\text{TrapGen}(n, m, q)$ that, with overwhelming probability in n , outputs a pair $(\mathbf{A}, \mathbf{S})$ satisfying*

1. $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is statistically close to uniform;
2. $\mathbf{S} \in \mathbb{Z}^{m \times m}$ is a basis of $\Lambda_q^\perp(\mathbf{A})$;
3. $\|\tilde{\mathbf{S}}\| \leq O(\sqrt{n \log q})$ and $\|\mathbf{S}\| \leq O(n \log q)$.

Moreover, for any vector $\mathbf{u} \in \mathbb{Z}_q^n$ and Gaussian parameter $\sigma \geq \|\tilde{\mathbf{S}}\| \cdot \omega(\sqrt{\log n})$, there

is a PPT algorithm $\text{SampleD}(\mathbf{A}, \mathbf{S}, \mathbf{u}, \sigma)$ that returns a vector $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{A})$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{A}), \sigma}$.

In [2, 5], there were another two algorithms proposed to sample short vectors from lattices. In Chapter 6, we will employ these two algorithms, SampleLeft and SampleRight , for our construction and security analysis, respectively, in a black-box manner. We collect the basic information of these two algorithms as follows:

$\text{SampleLeft}(\mathbf{A}, \mathbf{M}, \mathbf{S}_{\mathbf{A}}, \mathbf{u}, \sigma)$:

Input: a rank n matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$; a matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times m_1}$; a short trapdoor basis $\mathbf{S}_{\mathbf{A}}$ of $\Lambda_q^{\perp}(\mathbf{A})$; a vector $\mathbf{u} \in \mathbb{Z}_q^n$; and a Gaussian parameter $\sigma \geq \|\tilde{\mathbf{S}}_{\mathbf{A}}\| \cdot \omega(\sqrt{\log(m + m_1)})$.

Output: a vector $\mathbf{e} \in \mathbb{Z}^{m+m_1}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}_1), \sigma}$, where $\mathbf{F}_1 = [\mathbf{A} \mid \mathbf{M}]$. In particular, $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{F}_1)$.

$\text{SampleRight}(\mathbf{A}, \mathbf{B}, \mathbf{R}, \mathbf{S}_{\mathbf{B}}, \mathbf{u}, \sigma)$:

Input: a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$; a rank n matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$; a uniform random matrix $\mathbf{R} \in \mathbb{Z}^{m \times m}$, let $s_{\mathbf{R}} := \|\mathbf{R}\|_R = \sup_{\|\mathbf{x}\|=1} \|\mathbf{R}\mathbf{x}\|$; a short trapdoor basis $\mathbf{S}_{\mathbf{B}}$ of $\Lambda_q^{\perp}(\mathbf{B})$; a vector $\mathbf{u} \in \mathbb{Z}_q^n$; and a Gaussian parameter $\sigma \geq \|\tilde{\mathbf{S}}_{\mathbf{B}}\| \cdot s_{\mathbf{R}} \cdot \omega(\sqrt{\log m})$.

Output: a vector $\mathbf{e} \in \mathbb{Z}^{2m}$ sampled from a distribution statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{u}}(\mathbf{F}_2), \sigma}$, where $\mathbf{F}_2 = [\mathbf{A} \mid \mathbf{A}\mathbf{R} + \mathbf{B}]$. In particular, $\mathbf{e} \in \Lambda_q^{\mathbf{u}}(\mathbf{F}_2)$.

It is known that the distributions of the outputs given by the two algorithms are statistically close under the appropriate parameters when $m_1 = m$. See [2, 5] for further details of these algorithms.

2.4.4 Randomizing and Extending Control

In the constructions of policy-based signatures and delegatable policy-based signatures in Sections 5.4 and 5.5, respectively, we need two additional algorithms, summarized below, which allow us to perform randomization and delegation of short

lattice bases.

Lemma 2.4.5 ([33]). *Let $\mathbf{A}$ be a matrix in $\mathbb{Z}_q^{n \times m}$ whose columns generate the entire group $\mathbb{Z}_q^n$, and let $\mathbf{S}_\mathbf{A}$ be any basis of the lattice $\Lambda_q^\perp(\mathbf{A})$ and $\sigma \geq \|\tilde{\mathbf{S}}_\mathbf{A}\| \cdot \omega(\sqrt{\log n})$. There is a PPT algorithm $\text{RandBasis}(\mathbf{A}, \mathbf{S}_\mathbf{A}, \sigma)$ that outputs a new basis $\mathbf{S}'_\mathbf{A}$ of $\Lambda_q^\perp(\mathbf{A})$ such that $\|\mathbf{S}'_\mathbf{A}\| \leq \sigma \cdot \sqrt{m}$ with overwhelming probability. Moreover, for any two bases $\mathbf{S}_\mathbf{A}^{(1)}, \mathbf{S}_\mathbf{A}^{(2)}$ of $\Lambda_q^\perp(\mathbf{A})$ and $s \geq \max\{\|\tilde{\mathbf{S}}_\mathbf{A}^{(1)}\|, \|\tilde{\mathbf{S}}_\mathbf{A}^{(2)}\|\} \cdot \omega(\sqrt{\log n})$, the distributions of outputs of $\text{RandBasis}(\mathbf{A}, \mathbf{S}_\mathbf{A}^{(1)}, s)$ and $\text{RandBasis}(\mathbf{A}, \mathbf{S}_\mathbf{A}^{(2)}, s)$ are statistically close.*

Lemma 2.4.6 ([33]). *Let $\mathbf{A}$ be a matrix in $\mathbb{Z}_q^{n \times m}$ whose columns generate the entire group $\mathbb{Z}_q^n$, and let $\mathbf{S}_\mathbf{A}$ be any basis of the lattice $\Lambda_q^\perp(\mathbf{A})$. For any matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times \bar{m}}$, set $\mathbf{A}' = [\mathbf{A} | \bar{\mathbf{A}}]$. Then there is a deterministic polynomial-time algorithm $\text{ExtBasis}(\mathbf{A}, \mathbf{A}', \mathbf{S}_\mathbf{A})$ that returns a basis $\mathbf{S}_{\mathbf{A}'}$ of $\Lambda_q^\perp(\mathbf{A}')$ satisfying $\|\tilde{\mathbf{S}}_{\mathbf{A}'}\| = \|\tilde{\mathbf{S}}_\mathbf{A}\|$. Moreover, the statement holds even if the columns of $\mathbf{A}'$ are permuted. (For simplicity, if $\mathbf{A}'$ is generated by both appending and prepending columns of $\bar{\mathbf{A}}$ to $\mathbf{A}$, we still use $\text{ExtBasis}(\mathbf{A}, \mathbf{A}', \mathbf{S}_\mathbf{A})$ to denote the corresponding algorithm.)*

2.4.5 Random Matrices in $\{-1, 1\}^{m \times k}$

For security analysis of our construction of revocable identity-based encryption in Chapter 6, we will use the following three lemmas about properties of random matrices in $\{-1, 1\}^{m \times k}$. The first lemma is a generalized version of the leftover hash lemma introduced by Dodis et al. [46].

Lemma 2.4.7. ([3, Lemma 13]) *Suppose that $m > (n + 1) \log q + \omega(\log n)$, where $q > 2$ is a prime. Let $\mathbf{A}, \mathbf{B}$ be matrices uniformly chosen from $\mathbb{Z}_q^{n \times m}$, and $\mathbb{Z}_q^{n \times k}$ respectively. Let $\mathbf{R}$ be an $m \times k$ matrix uniformly chosen from $\{-1, 1\}^{m \times k} \bmod q$, where $k = k(n)$ is polynomial in n . Then for any vector $\mathbf{y} \in \mathbb{Z}_q^m$, the distribution of $(\mathbf{A}, \mathbf{A}\mathbf{R}, \mathbf{R}^\top \mathbf{y})$ is statistically close to the distribution of $(\mathbf{A}, \mathbf{B}, \mathbf{R}^\top \mathbf{y})$.*

The following lemma bounds the norm of a random square matrix in $\{-1, 1\}^{m \times m}$.

Lemma 2.4.8. ([3, Lemma 15]) *For a randomly chosen $m \times m$ matrix $\mathbf{R}$ from $\{-1, 1\}^{m \times m}$, there exists a universal constant C such that*

$$\Pr \left[\|\mathbf{R}\|_R = \sup_{\|\mathbf{x}\|=1} \|\mathbf{R}\mathbf{x}\| > C\sqrt{m} \right] \leq e^{-2m}.$$

The next lemma provides a better bound for random matrices in $\{-1, 1\}^{k \times m}$ with $m \gg k$.

Lemma 2.4.9. ([3, Lemma 16]) *For a randomly chosen $k \times m$ matrix $\mathbf{R}$ from $\{-1, 1\}^{k \times m}$, and any vector $\mathbf{y} \in \mathbb{R}^m$, we have*

$$\Pr \left[\|\mathbf{R}\mathbf{y}\| > \|\mathbf{y}\| \sqrt{k} \cdot \omega(\sqrt{\log k}) \right] < \text{negl}(k).$$

2.4.6 The SIS Hardness Assumption

The security of our policy-based signature and delegatable policy-based signature schemes (see Chapter 5) is partially based on the hardness of the Short Integer Solution problem (SIS) in the l_p norm, which is described as follows:

Definition 2.4.10 ($\text{SIS}_{n,m,q,\beta}^p$ problem, [81]). Given a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, the $\text{SIS}_{n,m,q,\beta}^p$ problem asks to find a vector $\mathbf{s} \in \mathbb{Z}^m$ such that $0 < \|\mathbf{s}\|_p \leq \beta$ and $\mathbf{A} \cdot \mathbf{s} = \mathbf{0} \bmod q$.

The hardness of SIS is established by a worst-case to average-case reduction from standard lattice problems, such as the Shortest Independent Vectors Problem (SIVP).

Proposition 2.4.11 ([53]). *For any $m, \beta = \text{poly}(n)$, $q \geq \beta \cdot \omega(n \log n)$, solving a random instance of the $\text{SIS}_{n,m,q,\beta}^2$ problem with non-negligible probability is at least*

as hard as approximating the SIVP_γ^2 problem in the worst case to within a certain factor $\gamma = \beta \cdot \tilde{O}(\sqrt{n})$.

For any vector $\mathbf{x} \in \mathbb{R}^m$, we have the relationship

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \sqrt{m} \cdot \|\mathbf{x}\|_\infty,$$

which implies that solving the $\text{SIS}_{n,m,q,\beta}^\infty$ problem is at least as hard as approximating SIVP_γ^2 (in the l_2 norm) for some approximating factor $\gamma = \beta \cdot \tilde{O}(\sqrt{nm})$.

2.4.7 The LWE Hardness Assumption

The security of our revocable identity-based encryption scheme (see Chapter 6) is based on the hardness of the Learning with Errors (LWE) problem, which is described as follows:

Definition 2.4.12 ($(\mathbb{Z}_q, n, \chi)$ -LWE, [2, 94]). For a positive integer n , a prime $q = q(n)$ and an error distribution χ over $\mathbb{Z}_q$, given access to an unspecified challenge oracle $\mathcal{O}$, the $(\mathbb{Z}_q, n, \chi)$ -LWE problem asks to determine whether $\mathcal{O}$ is a noisy pseudo-random sampler $\mathcal{O}_s$ carrying some constant random secret key $\mathbf{s} \in \mathbb{Z}_q^n$, or a truly random sampler $\mathcal{O}_\S$. The behaviors of the two samplers are as follows:

$\mathcal{O}_s$: outputs samples of the form $(\mathbf{u}_i, v_i) = (\mathbf{u}_i, \mathbf{u}_i^\top \mathbf{s} + x_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where $\mathbf{s} \in \mathbb{Z}_q^n$ is a uniformly distributed vector which remains fixed throughout sampling, $x_i \in \mathbb{Z}_q$ is a fresh sample from χ , and $\mathbf{u}_i$ is uniformly sampled from $\mathbb{Z}_q^n$.

$\mathcal{O}_\S$: outputs samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$ uniformly at random.

We state that an algorithm $\mathcal{B}$ solves the $(\mathbb{Z}_q, n, \chi)$ -LWE problem if the advantage

$$\text{Adv}_{\mathcal{B}}^{(\mathbb{Z}_q, n, \chi)\text{-LWE}} = |\Pr[\mathcal{B}^{\mathcal{O}_s} = 1] - \Pr[\mathcal{B}^{\mathcal{O}_\S} = 1]|$$

is non-negligible for a random secret $\mathbf{s} \in \mathbb{Z}_q^n$.

Note that Regev [94] showed that for a special Gaussian error distribution, denoted by $\bar{\Psi}_\alpha$, the LWE problem is at least as hard as several standard worst-case lattice problems using a quantum reduction. The distribution $\bar{\Psi}_\alpha$ is defined as follows:

Definition 2.4.13. For an $\alpha \in (0, 1)$ and a prime q , $\bar{\Psi}_\alpha$ is the distribution over $\mathbb{Z}_q$ of the random variable $\lfloor qX \rfloor \bmod q$, where the random variable X has normal distribution with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$, and $\lfloor x \rfloor$ represents the closest integer to x (half-integers are rounded to even numbers in order to avoid statistical biasing).

Throughout the thesis, $x \xleftarrow{\bar{\Psi}_\alpha} \mathbb{Z}_q$ (resp., $\mathbf{x} \xleftarrow{\bar{\Psi}_\alpha^m} \mathbb{Z}_q^m$) denotes that x (resp., $\mathbf{x}$) is selected from $\mathbb{Z}_q$ (resp., $\mathbb{Z}_q^m$) according to the distribution $\bar{\Psi}_\alpha$ (resp., $\bar{\Psi}_\alpha^m$).

Proposition 2.4.14 ([94]). *If there exists an efficient, possibly quantum, algorithm that solves the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$ -LWE problem for $\alpha \cdot q > 2\sqrt{n}$, then there exists a $q \cdot \text{poly}(n)$ -time quantum algorithm for approximating the SIVP and GapSVP problems in the l_2 norm, in the worst case, to within $\tilde{O}(n/\alpha)$ factors.*

To the best of our knowledge, the best algorithms for 2^k -approximations of GapSVP and SIVP run in time $2^{\tilde{O}(n/k)}$ [51, 100]. Hence it can be implied that the LWE problem with noise ratio $\alpha = 2^{-n^\epsilon}$ is likely hard for some constant $\epsilon < 1$.

The following lemma, proposed in [3], on the distribution $\bar{\Psi}_\alpha$ will be used in parameter analysis in Subsection 6.4.3. The proof can be deduced from Lemma 8.2 in [53].

Lemma 2.4.15 ([3]). *Let $\mathbf{e} \in \mathbb{Z}^m$ and $\mathbf{y} \xleftarrow{\bar{\Psi}_\alpha^m} \mathbb{Z}_q^m$. Then $|\mathbf{e}^\top \mathbf{y}|$ regarded as an integer*

2. Preliminaries

in $\{0\} \cup [q-1]$ is bounded, with overwhelming probability in m , as

$$|\mathbf{e}^\top \mathbf{y}| \leq \|\mathbf{e}\| q \alpha \omega(\sqrt{\log m}) + \|\mathbf{e}\| \sqrt{m}/2.$$

3. ASYMPTOTICALLY GOOD PACKING FAMILIES DERIVED FROM MULTIPLICATIVE LATTICES

3.1 Concatenation Based on Number Field $K = \mathbb{Q}(\sqrt{-3})$

The concatenation based on $\mathbb{Q}(\sqrt{-3})$, for constructing dense sphere packings, has been explained in [109]. We recall some key properties first.

Let $\omega = (-1 + \sqrt{-3})/2$ and $K = \mathbb{Q}(\sqrt{-3})$. The ring of integers of K is $\mathcal{O}_K = \mathbb{Z}[\omega]$. Via the mapping $\tau_0 : \mathbb{C} \rightarrow \mathbb{R}^2$ as $a + bi \mapsto (a, b)$, we may identify a vector $\mathbf{u} + \omega \mathbf{v} \in \mathbb{R}^n + \omega \mathbb{R}^n$ in $\mathbb{C}^n$ with a vector $(\mathbf{u} - \frac{1}{2}\mathbf{v}, \frac{\sqrt{3}}{2}\mathbf{v})$ in $\mathbb{R}^{2n}$. Hence $\mathcal{O}_K^n$ can be regarded as a subset of $\mathbb{R}^{2n}$. If we define the length $\|\mathbf{c}\|$ of the complex vector $\mathbf{c} = (a_1 + b_1i, \dots, a_n + b_ni)$ ($a_i, b_i \in \mathbb{R}$) as $\sqrt{\sum_{i=1}^n (a_i^2 + b_i^2)}$, then it is obvious that $\|\mathbf{u} + \omega \mathbf{v}\| = \left\| (\mathbf{u} - \frac{1}{2}\mathbf{v}, \frac{\sqrt{3}}{2}\mathbf{v}) \right\|$, where the latter is the Euclidean length of the vector in $\mathbb{R}^{2n}$.

Let $P \subseteq \mathbb{Z}^n$ be a lattice packing in $\mathbb{R}^n$. The minimum Euclidean distance, determinant of $P \subseteq \mathbb{R}^n$ and $\mathcal{P} = P + \omega P \subseteq \mathcal{O}_K^n$ (a lattice packing in $\mathbb{R}^{2n}$) have the following relations.

Lemma 3.1.1 ([109, Proposition 2.2]). *The minimum Euclidean distances of P and $\mathcal{P}$ satisfy*

$$d_E(\mathcal{P}) = d_E(P).$$

Lemma 3.1.2 ([109, Proposition 2.6(i)]). *The determinants of P and $\mathcal{P}$ satisfy*

$$\det(\mathcal{P}) = \left(\frac{\sqrt{3}}{2}\right)^n (\det(P))^2.$$

Here K is a totally complex field and $\mathcal{O}_K$ is a principal ideal domain. Given a non-zero prime ideal $\mathfrak{P} = (t_{\mathfrak{P}})$ with absolute norm $Q := \mathcal{N}(\mathfrak{P}) = |\text{Norm}_{K/\mathbb{Q}}(t_{\mathfrak{P}})|$, we can consider a special packing in $\mathbb{R}^{2n}$:

$$t_{\mathfrak{P}} \cdot \mathcal{P} := \{(t_{\mathfrak{P}}\alpha_1, t_{\mathfrak{P}}\alpha_2, \dots, t_{\mathfrak{P}}\alpha_n) \in \mathcal{O}_K^n : (\alpha_1, \alpha_2, \dots, \alpha_n) \in \mathcal{P}\}.$$

From algebraic number theory (see [84]), we know that the residue class field $\mathbb{F}_{\mathfrak{P}} = \mathcal{O}_K/\mathfrak{P}$ is isomorphic to the finite field $\mathbb{F}_Q$. Let $\beta_1 = 0, \beta_2, \dots, \beta_Q$ be Q elements of $\mathcal{O}_K$ such that

$$\beta_1 \bmod \mathfrak{P}, \dots, \beta_Q \bmod \mathfrak{P}$$

represent the Q distinct elements in $\mathbb{F}_{\mathfrak{P}}$. In the following discussion, we take the alphabet set of Q -ary codes to be $S = \{\beta_1, \dots, \beta_Q\}$. In this way, the codes with length n can be regarded as a finite subset of $\mathcal{O}_K^n$.

We take a family of Q -ary codes $\{C_i\}_{i=0}^{\ell-1}$ where C_i is an $(n, M_i, \geq Q^{\ell-i} d_E^2(\mathcal{P}))$ -code. The following lemma offers the concatenating method of the codes with the packing $t_{\mathfrak{P}}^\ell \cdot \mathcal{P} \subseteq \mathcal{O}_K^n$. Note that the concatenation is just the sumset of the subsets in $\mathcal{O}_K^n$.

Lemma 3.1.3 ([109, Corollary 2.4]). *Given a non-zero prime ideal $\mathfrak{P} = (t_{\mathfrak{P}})$ of $K = \mathbb{Q}(\sqrt{-3})$ such that $Q = \mathcal{N}(\mathfrak{P}) = |\text{Norm}_{K/\mathbb{Q}}(t_{\mathfrak{P}})|$, let*

(i) $\mathcal{P} \subseteq \mathcal{O}_K^n$ be a packing in $\mathbb{R}^{2n}$;

(ii) $\mathcal{C} = \{C_i\}_{i=0}^{\ell-1}$ be a family of Q -ary codes, where C_i is an (n, M_i, d_{C_i}) -code, the alphabet set of C_i is S , and $d_{C_i} \geq Q^{\ell-i} d_E^2(\mathcal{P})$. In addition, for each $0 \leq i \leq \ell - 1$, C_i contains zero codeword.

Then the concatenation $C_0 + t_{\mathfrak{P}} C_1 + \cdots + t_{\mathfrak{P}}^{\ell-1} C_{\ell-1} + t_{\mathfrak{P}}^{\ell} \cdot \mathcal{P}$ is a subset of $\mathcal{O}_K^n$, which is defined as

$$\left\{ \sum_{i=0}^{\ell-1} t_{\mathfrak{P}}^i \mathbf{c}_i + t_{\mathfrak{P}}^{\ell} \mathbf{p} : \mathbf{c}_i \in C_i \text{ for all } 0 \leq i \leq \ell - 1, \mathbf{p} \in \mathcal{P} \right\}.$$

It can be regarded as a packing in $\mathbb{R}^{2n}$ with density at least $\Delta(\mathcal{P}) \cdot \prod_{i=0}^{\ell-1} M_i$. Equivalently, the density exponent satisfies

$$\lambda(C_0 + t_{\mathfrak{P}} C_1 + \cdots + t_{\mathfrak{P}}^{\ell-1} C_{\ell-1} + t_{\mathfrak{P}}^{\ell} \cdot \mathcal{P}) \geq \lambda(\mathcal{P}) + \frac{1}{2n} \sum_{i=0}^{\ell-1} \log_2(M_i).$$

Proof. First consider the case $\ell = 1$, which only concatenates one code C with the packing $t_{\mathfrak{P}} \cdot \mathcal{P}$. Then use induction to get the general result. For the details, the reader may refer to [109, Corollary 2.4]. $\square$

Note that the requirement that each concatenated code contain the zero codeword is necessary for Proposition 2.3 and Corollary 2.4 of [109] as the proof requires that any codeword in C have Hamming weight not less than the minimum Hamming distance of C .

3.2 Remarks on the Asymptotic Properties of Xing's Construction

Based on Lemma 3.1.3 (Xing's generic construction), a direct idea for constructing asymptotically good packing families is to take $\mathcal{P}$ as $\mathcal{O}_K^n$ and let n tend to ∞ . In this section, we first introduce the basic case where $\mathcal{P}$ in Lemma 3.1.3 is set to be

$\mathcal{O}_K^n$. Next, we briefly recall Xing's improvement on the basic case.

3.2.1 Asymptotically Good Packing Family Derived from $\mathcal{O}_K^n$

Based on the GV bound Eq. (2.4), for $0 \leq i \leq \ell - 1$, we can choose $C_i^{(\ell)}$ as a Q -ary

$$\left(n_\ell, Q^{n_\ell R_i^{(\ell)}}, Q^{\ell-i}\right)\text{-code, where } n_\ell = Q^\ell,$$

such that the rate satisfies

$$R_i^{(\ell)} \geq R_{GV}\left(Q, \varrho_i^{(\ell)}\right) = 1 - H_Q\left(\varrho_i^{(\ell)}\right),$$

where the relative minimum distance is

$$\varrho_i^{(\ell)} = \frac{Q^{\ell-i}}{n_\ell} = \frac{1}{Q^i}.$$

Proposition 3.2.1. *Set the packing $\mathcal{P}$ in Lemma 3.1.3 as $\mathcal{O}_K^{n_\ell}$, where $n_\ell = Q^\ell$.*

Then the asymptotic density exponent $\lambda(\mathcal{F})$ of the packing family

$$\mathcal{F} = \left\{ C_0^{(\ell)} + t_{\mathfrak{P}} C_1^{(\ell)} + \cdots + t_{\mathfrak{P}}^{\ell-1} C_{\ell-1}^{(\ell)} + t_{\mathfrak{P}}^\ell \cdot \mathcal{O}_K^{n_\ell} \right\}_{\ell \rightarrow \infty}$$

satisfies

$$\lambda(\mathcal{F}) \geq -1 + \frac{1}{2} \log_2 2\pi e - \frac{1}{4} \log_2 3 - \frac{1}{2} \log_2 Q \cdot \sum_{i=0}^{\infty} H'_Q(1/Q^i), \quad (3.1)$$

where $H'_Q(\varrho) = H_Q(\varrho)$ for $0 < \varrho < \frac{Q-1}{Q}$ and $H'_Q(\varrho) = 1$ for $\frac{Q-1}{Q} \leq \varrho \leq 1$.

Proof. From Lemma 3.1.1 and Lemma 3.1.2, we know for each $n \in \mathbb{Z}_{\geq 1}$,

$$d_E(\mathcal{O}_K^n) = 1, \quad \text{and} \quad \det(\mathcal{O}_K^n) = \left(\frac{\sqrt{3}}{2} \right)^n.$$

Hence for $n = Q^\ell$, the density exponent of $\mathcal{O}_K^n$ satisfies

$$\lambda(\mathcal{O}_K^n) = \frac{1}{2n} \log_2 \frac{(1/2)^{2n} V_{2n}}{(\sqrt{3}/2)^n} = -1 + \frac{1}{2} \log_2 2\pi e - \frac{1}{4} \log_2 3 - \frac{\ell}{2} \log_2 Q.$$

From Lemma 3.1.3, Proposition 2.3.1 and Eq. (2.4), we get Eq. (3.1). $\square$

Remark 3.2.2. There is no clear monotonicity of the lower bound Eq. (3.1). We apply the software **Magma** V2.21-4 [20, 21] to list all prime numbers within 100. Let p run through the list and choose one splitting prime ideal of p as $\mathfrak{P}$. Q is the norm of $\mathfrak{P}$. Set $\ell = 1000$, which is sufficiently large to approximate the limit on the level of **Magma** precision. The best result is

```
[Ring of integers in K]  -1.27196767512213615952191570262
when Q=4 norm of prime ideal lying over 2.
```

On the whole, the result becomes worse when the prime number p increases.

3.2.2 Xing's Asymptotically Good Packing Family

Xing offered one method to improve the asymptotic bound in Remark 3.2.2:

Proposition 3.2.3 (Corollary 3.3 and Theorem 3.4 of [109]). *Given a non-zero prime ideal $\mathfrak{P}$ of $K = \mathbb{Q}(\sqrt{-3})$ such that $Q = \mathcal{N}(\mathfrak{P})$, we can construct a packing family $\mathcal{F}'$ such that for any real number z with $1/Q < z < (Q-1)/Q$, the asymptotic*

density exponent $\lambda(\mathcal{F}')$ of the packing family satisfies

$$\lambda(\mathcal{F}') \geq -1 + \frac{1}{2} \log_2 2\pi e - \frac{1}{4} \log_2 3 + \frac{1}{2} \log_2 z - \frac{1}{2} \log_2 Q \sum_{i=0}^{\infty} H'_Q \left(\frac{z}{Q^i} \right),$$

where $H'_Q(\varrho) = H_Q(\varrho)$ for $0 < \varrho < \frac{Q-1}{Q}$ and $H'_Q(\varrho) = 1$ for $\frac{Q-1}{Q} \leq \varrho \leq 1$.

We retest the asymptotic density exponent of Xing's construction first.

Remark 3.2.4. Let p run through the prime numbers within 50 and choose one splitting prime ideal of p as $\mathfrak{P}$. Q is the norm of $\mathfrak{P}$. Let z run through $1/Q$ to $(Q-1)/Q$ by $1/10000$. In our experiment, the best result of Xing's construction is

[Xing] -1.26532181415209410650824899158

when $z=3049/10000$, $Q=4$ norm of prime ideal lying over 2.

Note that $z \approx 0.3049$ is the computational optimal estimate. Suppose the real optimal is z_0 . We briefly sketch Xing's construction then.

Instead of $\mathcal{O}_K^n$, Xing considered the packing $\mathcal{P}_x \subseteq \mathcal{O}_K^n$ such that $d_E(\mathcal{P}_x)$ is an integer x . Set

$$\mathcal{F}_x = \left\{ C_0^{(x)} + t_{\mathfrak{P}} C_1^{(x)} + \cdots + t_{\mathfrak{P}}^{\ell-1} C_{\ell-1}^{(x)} + t_{\mathfrak{P}}^{\ell} \cdot \mathcal{P}_x \right\}_{n \rightarrow \infty},$$

where $\ell = \lfloor \log_Q(n/x) \rfloor$. One lower bound of its asymptotic density exponent is given in Theorem 3.2 of [109].

If there exists an integer x such that exactly $\frac{x}{Q^{\lceil \log_Q x \rceil}} = z_0$, then the packing family $\mathcal{F}_x$ can obtain the optimal bound of [109, Theorem 3.4]. Otherwise, we can select a sequence of integers $\{x_k\}$ such that $\lim_{k \rightarrow \infty} \frac{x_k}{Q^{\lceil \log_Q x_k \rceil}} = z_0$, and then use the diagonal argument to group a new family $\mathcal{F}'$ from $\{\mathcal{F}_{x_k}\}_{k \rightarrow \infty}$, where the k -th

member of $\mathcal{F}'$ is the k -th member of $\mathcal{F}_{x_k}$. Then the new family $\mathcal{F}'$ can obtain the optimal bound of [109, Theorem 3.4].

3.3 New Method to Construct Asymptotically Good Packing Families

In Xing's construction, the number of codes increases to ∞ as n tends to ∞ . He concatenated these codes to certain families of packings that are not asymptotically good. In this section, we provide a new constructing method where we concatenate finitely many codes to asymptotically good packing families. In particular, our method can obtain some packing families which are derived from, but denser than, the multiplicative lattice packing families. The results will be explicitly shown in the next section.

Suppose we have an asymptotically good lattice packing family $\{L_n \subseteq \mathbb{R}^n\}_{n \rightarrow \infty}$ with $d_E(L_n) \geq c\sqrt{n}$ for some constant $c > 0$.

Let Q be the norm of one non-zero prime ideal $\mathfrak{P} = (t_{\mathfrak{P}})$ in $\mathcal{O}_K$. Set $\ell = \left\lfloor \log_Q \frac{(Q-1)}{c^2 Q} \right\rfloor$. Thus $Q^\ell \cdot c^2 \leq \frac{Q-1}{Q}$. Based on the GV bound Eq. (2.4), for $0 \leq i \leq \ell - 1$, we can choose $C_i^{(n)}$ as a Q -ary

$$\left(n, Q^{nR_i^{(n)}}, \lceil Q^{\ell-i} \cdot c^2 n \rceil\right)\text{-code}$$

such that the rate satisfies

$$R_i^{(n)} \geq R_{GV}\left(Q, \varrho_i^{(n)}\right) = 1 - H_Q\left(\varrho_i^{(n)}\right),$$

where the relative minimum distance is

$$\lim_{n \rightarrow \infty} \varrho_i^{(n)} = Q^{\ell-i} \cdot c^2.$$

Proposition 3.3.1. *We can concatenate $\ell = \left\lfloor \log_Q \frac{Q-1}{c^2 Q} \right\rfloor$ Q -ary codes $\{C_i^{(n)}\}_{i=0}^{\ell-1}$ to $\mathcal{P}_n := L_n + \omega L_n$, where $C_i^{(n)}$ is an $(n, Q^{nR_i^{(n)}}, \lceil Q^{\ell-i} \cdot c^2 n \rceil)$ -code. The asymptotic density exponent of the new packing family*

$$\mathcal{H} = \left\{ C_0^{(n)} + t_{\mathfrak{P}} C_1^{(n)} + \cdots + t_{\mathfrak{P}}^{\ell-1} C_{\ell-1}^{(n)} + t_{\mathfrak{P}}^{\ell} \cdot \mathcal{P}_n \right\}_{n \rightarrow \infty}$$

satisfies

$$\lambda(\mathcal{H}) \geq \underbrace{\frac{1}{2} \log_2 \frac{c^2 \pi e}{2\sqrt{3}} - \frac{1}{n} \log_2 \det(L_n)}_{\lambda_{\text{lattices}}(\mathcal{H})} + \underbrace{\frac{1}{2} \log_2 Q \sum_{i=0}^{\ell-1} (1 - H_Q(Q^{\ell-i} c^2))}_{\lambda_{\text{codes}}(\mathcal{H})}. \quad (3.2)$$

Proof. From the definition of asymptotic density exponent, we have

$$\begin{aligned} \lambda(\mathcal{H}) &\geq \limsup_{n \rightarrow \infty} \frac{1}{2n} \log_2 \frac{(c\sqrt{n})^{2n} V_{2n} \prod_{i=0}^{\ell-1} Q^{nR_i^{(n)}}}{2^{2n} \det(\mathcal{P}_n)} \\ &= \limsup_{n \rightarrow \infty} \log_2 \frac{c}{2} + \frac{1}{2} \log_2 n - \frac{1}{2} \log_2 n + \frac{1}{2} \log_2 \pi e \\ &\quad - \frac{1}{2} \log_2 \frac{\sqrt{3}}{2} - \frac{1}{n} \log_2 \det(L_n) + \frac{1}{2} \log_2 Q \sum_{i=0}^{\ell-1} R_i^{(n)} \\ &\geq \frac{1}{2} \log_2 \frac{c^2 \pi e}{2\sqrt{3}} - \frac{1}{n} \log_2 \det(L_n) + \frac{1}{2} \log_2 Q \sum_{i=0}^{\ell-1} (1 - H_Q(Q^{\ell-i} c^2)). \end{aligned}$$

□

Note that based on Lemma 3.1.3 and the proof above, the bound given in Eq. (3.2) can be split into two parts: the first part, denoted by $\lambda_{\text{lattices}}$, is con-

tributed by the family of lattices; the second part, denoted by λ_{codes} , is contributed by the family of concatenated codes. The notation will be used in later discussions.

3.4 Concatenation with Multiplicative Lattices

Rosenbloom and Tsfasman [96] introduced two kinds of multiplicative lattices in global fields, that is, principal lattices and congruence lattices. In this thesis, we only use the ones in function fields, where both principal and congruence lattices are full rank sublattices of $A_{n-1} = \{\mathbf{x} \in \mathbb{Z}^n \mid \sum x_i = 0\}$. They lead to asymptotically good packing families. In this section, we proceed with our new concatenating method introduced in Section 3.3 to improve the asymptotic density exponent derived from multiplicative lattice packings.

3.4.1 Principal Lattices and Congruence Lattices

We recall the definition of principal lattices from [96] first. Let $k = \mathbb{F}_q$ and $K = k(X)$, where X/k is a smooth proper curve of genus g . Take a nonempty set $S = \{P_1, P_2, \dots, P_n\} \subseteq X(k)$, $n = |S|$, and let

$$U_S = \{f \in K^* \mid f \text{ is a unit outside } S\}.$$

Let $\text{Div}_S(X)$ be the group of divisors supported in S , $\text{Div}_S^0(X) \subseteq \text{Div}_S(X)$ the subgroup of degree zero divisors, $\text{Pr}_S(X)$ the subgroup of principal divisors, and let $J_X = \text{Div}^0(X)/\text{Pr}(X)$ denote the Jacobian of X . The properties of these groups can be found in [105, Chapter 1].

There is a natural map

$$\begin{aligned}\phi : U_S &\rightarrow \text{Div}_S(X) \simeq \mathbb{Z}^n \\ f &\mapsto \text{div}(f),\end{aligned}$$

where $\text{div}(f)$ is the principal divisor of f . The principal lattice is defined as $L_S := \text{Pr}_S(X) = \phi(U_S)$, which is a sublattice of A_{n-1} . The parameters of L_S are

Lemma 3.4.1 ([96, Lemma 1.1]). (i) $\text{rank } L_S = n - 1$;

$$(ii) \det L_S \leq \sqrt{n} \cdot |J_X(k)|;$$

$$(iii) d_E(L_S) \geq \min_{f \in U_S \setminus k^*} \sqrt{2 \cdot \deg f}.$$

Furthermore, let D be a positive divisor on X , $D = \sum a_i Q_i$, $r_i = \deg Q_i$, $N(Q_i) = q^{r_i}$ and $a = \deg D = \sum a_i r_i$. Here we assume $S \cap \text{supp}(D) = \emptyset$. Then the congruence lattice is defined as $L_{S,D} := \phi(U_{S,D})$, where

$$U_{S,D} = \{f \in U_S : f \equiv 1 \pmod{D}\}.$$

The parameters of $L_{S,D}$ are

Lemma 3.4.2 ([96, Lemma 2.2]). (i) $\text{rank } L_{S,D} = n - 1$;

$$(ii) \det L_{S,D} \leq \sqrt{n} \cdot |J_X(k)| \cdot \frac{q^a}{q-1} \cdot \prod (1 - q^{-r_i});$$

$$(iii) d_E(L_{S,D}) \geq \sqrt{2a}.$$

3.4.2 Lattice Dimension Augmentation for Full Rank Sublattices of A_{n-1}

We know the rank of A_{n-1} is $n - 1$. Now we want to apply our concatenating method on certain full rank sublattices of $A_{n-1} \subseteq \mathbb{Z}^n$. First we need to introduce

a dimension augmentation method to make the lattices have rank n without much loss in the parameters.

For any full rank sublattice L of A_{n-1} , the $\mathbb{R}$ -linear span of L is

$$V = \{(x_1, x_2, \dots, x_{n-1}, x_n) \in \mathbb{R}^n : x_1 + x_2 + \dots + x_n = 0\}.$$

We add one extra vector $\mathbf{e}_n = (0, 0, 0, 0, \dots, 0, \chi)$ to a basis of L , where $\chi \in \mathbb{Z} \setminus \{0\}$. The resulting basis generates a rank n lattice in $\mathbb{R}^n$, which is denoted by B and called the augmented lattice of L .

The distance from the point $(0, 0, 0, 0, \dots, 0, \chi)$ to the hyperplane V is $\frac{\chi}{\sqrt{n}}$. Thus the minimum Euclidean distance of B satisfies

$$d_E(B) \geq \min \left\{ d_E(L), \frac{\chi}{\sqrt{n}} \right\}.$$

3.4.3 Concatenation with Principal Lattices

Now set $S = X(k)$, and use the same estimation $\deg f \geq \frac{|X(k)|}{q+1}$ as [96]. Thus the minimum Euclidean distance of $L_S = L_{X(k)}$ satisfies $d_E(L_{X(k)}) \geq \sqrt{\frac{2n}{q+1}}$, where $n = |X(k)|$.

We add the vector $(0, 0, 0, 0, \dots, 0, n)$ to a basis of $L_{X(k)}$ and obtain a rank n lattice $B_{X(k)}$ in $\mathbb{Z}^n$. The parameters of $B_{X(k)}$ are

Proposition 3.4.3. (i) $\text{rank } B_{X(k)} = n$;

$$(ii) \quad \det B_{X(k)} = \frac{n}{\sqrt{n}} \cdot \det(L_{X(k)}) \leq n \cdot |J_X(k)|;$$

$$(iii) \quad d_E(B_{X(k)}) \geq \min \left\{ d_E(L_{X(k)}), \frac{n}{\sqrt{n}} \right\} \geq \sqrt{\frac{2n}{q+1}}.$$

Proof. (i), (iii) follow directly from the dimension augmentation method. For (ii), as the determinant of a lattice is just the volume of the fundamental region of the

3. Asymptotically Good Packing Families Derived from Multiplicative Lattices

lattice, and the distance from the point $(0, \dots, 0, n)$ to the $\mathbb{R}$ -linear span of $L_{X(k)}$ is $\frac{n}{\sqrt{n}}$, we get the determinant of $B_{X(k)}$ as $\frac{n}{\sqrt{n}} \cdot \det(L_{X(k)})$. Following Lemma 3.4.1, we get the result. $\square$

We employ the same families of curves as [96]: For q an even power of a prime, there exist families of curves X/k of growing genus $g(X)$ such that $\lim_{g(X) \rightarrow \infty} \frac{|X(k)|}{g(X)} = \sqrt{q} - 1$. Moreover, such families satisfy

$$|J_X(k)| \sim q^{g(X)} \left(\frac{q}{q-1} \right)^{|X(k)|}.$$

The proof of the estimation can be found in the appendix of [96]. The following lemma shows that the corresponding augmented principal lattices lead to asymptotically good packing families.

Lemma 3.4.4. *A family of curves X/k with $\lim_{g(X) \rightarrow \infty} \frac{|X(k)|}{g(X)} = \sqrt{q} - 1$ yields a family of augmented principal lattices $\mathcal{F}_0 = \left\{ B_{X(k)}^{(N)} \subseteq \mathbb{R}^N \right\}$ with rank $N = |X(k)| \rightarrow \infty$ and*

$$\lambda(\mathcal{F}_0) \geq \log_2 \sqrt{\pi e} - \log_2 \frac{\sqrt{q+1}}{q-1} - \frac{\sqrt{q}}{\sqrt{q}-1} \log_2 q.$$

Proof. Note that $\lim_{N \rightarrow \infty} \frac{1}{N} \log_2 N = 0$. The proof is straightforward from the definition of asymptotic density exponent and Proposition 3.4.3. It is also similar to the proof of [96, Theorem 1.2]. $\square$

Note that the bound in Lemma 3.4.4 is exactly the one for principal lattices [96, Theorem 1.2]. This means the dimension augmentation does not harm the good asymptotic properties of the original lattices. Meanwhile, we put it here as a reference to compare with the following Proposition 3.4.5. The difference is the advantage of our concatenating method.

As $d_E \left(B_{X(k)}^{(N)} \right) \geq \sqrt{\frac{2}{q+1}} \cdot \sqrt{N}$, we can proceed with the method introduced in Section 3.3. We denote $\mathcal{P}_{X(k)}^{(N)} = B_{X(k)}^{(N)} + \omega B_{X(k)}^{(N)}$ and get the following proposition.

Proposition 3.4.5. *A family of curves X/k with $\lim_{g(X) \rightarrow \infty} \frac{|X(k)|}{g(X)} = \sqrt{q} - 1$ and families of Q -ary codes $\left\{ C_i^{(N)} \right\}_{i=0}^{\ell-1}$ with $C_i^{(N)}$ an $\left(N, Q^{NR_i^{(N)}}, \left\lceil Q^{\ell-i} \cdot \frac{2N}{q+1} \right\rceil \right)$ -code, $\ell = \left\lfloor \log_Q \frac{(Q-1)(q+1)}{2Q} \right\rfloor$ and the rate*

$$\lim_{N \rightarrow \infty} R_i^{(N)} \geq R_{GV} \left(Q, Q^{\ell-i} \cdot \frac{2}{q+1} \right) = 1 - H_Q \left(Q^{\ell-i} \cdot \frac{2}{q+1} \right),$$

yield a packing family

$$\mathcal{F}_{Q,q} = \left\{ C_0^{(N)} + t_{\mathfrak{P}} C_1^{(N)} + \cdots + t_{\mathfrak{P}^{\ell-1}} C_{\ell-1}^{(N)} + t_{\mathfrak{P}^\ell} \cdot \mathcal{P}_{X(k)}^{(N)} \subseteq \mathbb{R}^{2N} \right\}_{N \rightarrow \infty}$$

with $N = |X(k)| \rightarrow \infty$ and

$$\begin{aligned} \lambda(\mathcal{F}_{Q,q}) \geq & \underbrace{\log_2 \sqrt{\pi e} - \log_2 \frac{\sqrt{q+1}}{q-1} - \frac{\sqrt{q}}{\sqrt{q}-1} \log_2 q - \frac{1}{4} \log_2 3}_{\lambda_{\text{lattices}}(\mathcal{F}_{Q,q})} \\ & + \underbrace{\frac{1}{2} \log_2 Q \sum_{i=0}^{\ell-1} \left(1 - H_Q \left(\frac{2Q^{\ell-i}}{q+1} \right) \right)}_{\lambda_{\text{codes}}(\mathcal{F}_{Q,q})}. \end{aligned} \quad (3.3)$$

Proof. From Propositions 3.3.1 and 3.4.3. □

Remark 3.4.6. There is no clear monotonicity of the lower bound Eq. (3.3). We apply the software **Magma** V2.21-4 [20, 21] to list all prime numbers within 100. Let p_1 run through the list and choose one splitting prime ideal of p_1 as $\mathfrak{P}$. Q is the norm of $\mathfrak{P}$. Let p_2 run through the list and let r run through the even numbers from 2 to 250. Take $q = p_2^r$. The best output in the experiment is given as

3. Asymptotically Good Packing Families Derived from Multiplicative Lattices

[Improvement on Principal Lattices]

-1.26532182282965944267554218804

when $Q=4$ norm of prime ideal lying over 2; $q=59^{28}$.

Lattice packing contributes: -81.2061477310654255659655563902;

l= 81 Concatenated codes contributes:

79.9408259082357661232900142022.

The above output shows that the optimal result in our experiment is $\lambda \geq -1.26532182283$ when $Q = 4, q = 59^{28}$, which is better than -1.87 from principal lattices. Note that the last two statements show the contribution $\lambda_{\text{lattices}}$ from augmented principal lattices and the contribution λ_{codes} from concatenated codes respectively to the asymptotic density exponent λ . In Section 3.5, we will use $\lambda_{\text{lattices}}$ and λ_{codes} to compare our results on concatenations from principal lattices and congruence lattices.

3.4.4 Concatenation with Congruence Lattices

Similar to the last subsection, we set $S = X(k), n = |X(k)|$, and add the vector $(0, 0, 0, 0, \dots, 0, n)$ to a basis of $L_{S,D} = L_{X(k),D}$ and obtain a rank n lattice $B_{X(k),D}$ in $\mathbb{Z}^n$. The parameters of $B_{X(k),D}$ are

Proposition 3.4.7. (i) $\text{rank } B_{X(k),D} = n$;

$$(ii) \det B_{X(k),D} = \frac{n}{\sqrt{n}} \det(L_{X(k),D}) \leq n \cdot |J_X(k)| \cdot \frac{q^a}{q-1} \cdot \prod (1 - q^{-r_i});$$

$$(iii) d_E(B_{X(k),D}) \geq \min \left\{ d_E(L_{X(k),D}), \frac{n}{\sqrt{n}} \right\}.$$

We consider the same families of curves as principal lattices and further choose divisors in such a way that

$$\lim_{|X(k)| \rightarrow \infty} \frac{\deg D}{|X(k)|} = \frac{y}{2 \ln q}, \text{ where } 0 < y \leq 1.$$

Note that $\lim_{|X(k)| \rightarrow \infty} \frac{\deg D}{|X(k)|} = \frac{1}{2 \ln q}$ is adopted in [96], while here we loosen the requirement for our construction. The following lemma shows that the corresponding augmented congruence lattices lead to asymptotically good packing families.

Lemma 3.4.8. *A family of curves X/k with $\lim_{g(X) \rightarrow \infty} \frac{|X(k)|}{g(X)} = \sqrt{q}-1$ and positive divisors with $\lim_{|X(k)| \rightarrow \infty} \frac{\deg D}{|X(k)|} = \frac{y}{2 \ln q}$ yield a family of augmented congruence lattices $\mathcal{F}'_0 = \left\{ B_{X(k),D}^{(N)} \subseteq \mathbb{R}^N \right\}_{N \rightarrow \infty}$ with rank $N = |X(k)| \rightarrow \infty$ and*

$$\begin{aligned} \lambda(\mathcal{F}'_0) &\geq \log_2 \sqrt{\frac{\pi}{2}} - \frac{1}{2} \log_2 (\ln q) - \frac{\sqrt{q}}{\sqrt{q}-1} \log_2 q \\ &\quad + \log_2 (q-1) + \frac{1}{2} \log_2 y + \frac{1-y}{2} \log_2 e. \end{aligned}$$

Proof. Similar to the proof of Lemma 3.4.4 and [96, Theorem 2.3]. $\square$

As for sufficiently large $N = |X(k)|$, $d_E \left(B_{X(k),D}^{(N)} \right) \geq \sqrt{\frac{y}{\ln q}} \cdot \sqrt{N}$, we can proceed with the method introduced in Section 3.3. We denote $\mathcal{P}_{X(k),D}^{(N)} = B_{X(k),D}^{(N)} + \omega B_{X(k),D}^{(N)}$ and get the following proposition.

Proposition 3.4.9. *A family of curves X/k with $\lim_{g(X) \rightarrow \infty} \frac{|X(k)|}{g(X)} = \sqrt{q}-1$, positive divisors with $\lim_{|X(k)| \rightarrow \infty} \frac{\deg D}{|X(k)|} = \frac{y}{2 \ln q}$ and families of Q -ary codes $\left\{ C_i^{(N)} \right\}_{i=0}^{\ell-1}$ with $C_i^{(N)}$ an $\left(N, Q^{NR_i^{(N)}}, \left\lceil Q^{\ell-i} \cdot \frac{yN}{\ln q} \right\rceil \right)$ -code, $\ell = \left\lfloor \log_Q \frac{(Q-1) \ln q}{yQ} \right\rfloor$ and the rate*

$$\lim_{N \rightarrow \infty} R_i^{(N)} \geq R_{GV} \left(Q, Q^{\ell-i} \cdot \frac{y}{\ln q} \right) = 1 - H_Q \left(Q^{\ell-i} \cdot \frac{y}{\ln q} \right),$$

yield a packing family

$$\mathcal{F}_{Q,q,y} = \left\{ C_0^{(N)} + t_{\mathfrak{P}} C_1^{(N)} + \cdots + t_{\mathfrak{P}}^{\ell-1} C_{\ell-1}^{(N)} + t_{\mathfrak{P}}^{\ell} \cdot \mathcal{P}_{X(k),D}^{(N)} \subseteq \mathbb{R}^{2N} \right\}_{N \rightarrow \infty},$$

3. Asymptotically Good Packing Families Derived from Multiplicative Lattices

with $N = |X(k)| \rightarrow \infty$ and

$$\lambda(\mathcal{F}_{Q,q,y}) \geq \lambda_{\text{lattices}}(\mathcal{F}_{Q,q,y}) + \lambda_{\text{codes}}(\mathcal{F}_{Q,q,y}), \quad (3.4)$$

where

$$\begin{aligned} \lambda_{\text{lattices}}(\mathcal{F}_{Q,q,y}) &= \log_2 \sqrt{\frac{\pi}{2}} - \frac{1}{2} \log_2 (\ln q) - \frac{\sqrt{q}}{\sqrt{q}-1} \log_2 q \\ &\quad + \log_2 (q-1) + \frac{1}{2} \log_2 y + \frac{1-y}{2} \log_2 e - \frac{1}{4} \log_2 3; \\ \lambda_{\text{codes}}(\mathcal{F}_{Q,q,y}) &= \frac{1}{2} \log_2 Q \sum_{i=0}^{\ell-1} \left(1 - H_Q \left(\frac{yQ^{\ell-i}}{\ln q} \right) \right). \end{aligned}$$

Proof. From Propositions 3.3.1 and 3.4.7. □

Remark 3.4.10. There is no clear monotonicity of the lower bound Eq. (3.4). We design the computational experiments in Magma V2.21-4 [20, 21] as follows:

- List all prime numbers within 60. Let p_1 run through the list and choose one splitting prime ideal of p_1 as $\mathfrak{P}$. Q is the norm of $\mathfrak{P}$. Let p_2 run through the list and let r run through the even numbers from 2 to 100. Take $q = p_2^r$.
- Set y from 0.1 to 1 by 0.01. Then we find the good result comes when $y = 0.1$. Next set y from 0.01 to 0.2 by 0.0001. Then repeatedly increase the decimal places to get y for better results.

We cannot run through all prime numbers and all possible values for y . The best output in the experiment is given as

[Improvement on Congruence Lattices]

-1.26532181404273379250349262485

when $Q=4$ norm of prime ideal lying over 2;

`q=11^94; y=1/4000000000.`

`Lattice packing contributes: -19.2060002184860472925950737917;`

`l= 19 Concatenated codes contributes:`

`17.9406784044433135000915811668.`

The above output shows that the optimal result in our experiment is $\lambda \geq -1.26532181404$ when $Q = 4, q = 11^{94}, y = 2.5 \times 10^{-10}$, which is better than -1.39 from congruence lattices.

3.5 Comparison

In Rosenbloom and Tsfasman's construction [96], congruence lattices lead to an asymptotically good family with $\lambda \geq -1.39$, which is much better than $\lambda \geq -1.87$ for the packing family from principal lattices. However, through our concatenating method, the family derived from congruence lattices has bound only slightly better than the one from principal lattices, while both of the bounds on λ are quite similar with Xing's result [109].

In this section, we first compare the two families derived from multiplicative lattices by our concatenating method. We take the case $Q = 4, q = 11^{94}, y = 2.5 \times 10^{-10}$ as an example, which leads to the best result in the experiment in Remark 3.4.10, and compare the concatenations derived from congruence (Proposition 3.4.9) and principal (Proposition 3.4.5) lattices respectively. Let ℓ denote the number of concatenated codes and let c denote the coefficient used in the bound $d_E(B) \geq c\sqrt{n}$, with $c = \sqrt{\frac{2}{q+1}}$ in the principal case and $c = \sqrt{\frac{y}{\ln q}}$ in the congruence case. Let $\lambda_{\text{lattices}}$ and λ_{codes} be the contributions from the families of lattices and concatenated codes respectively to the asymptotic density exponent λ , as shown in Eqs. (3.2), (3.3) and (3.4). The bound on λ is the sum of $\lambda_{\text{lattices}}$ and λ_{codes} . We compare the

3. Asymptotically Good Packing Families Derived from Multiplicative Lattices

contributions and list the numerical results in Table 3.1.

Table 3.1: Contributions from lattices and concatenated codes to asymptotic density exponent of packing families derived from principal and congruence lattices

	from Principal lattices	from Congruence lattices
Lattices	$c = 1.60346245499 \times 10^{-49}$	$c = 1.05315179371 \times 10^{-6}$
	$\lambda_{\text{lattices}} = -161.44243111595$	$\lambda_{\text{lattices}} = -19.20600021848$
Codes	$\ell = 161$	$\ell = 19$
	$\lambda_{\text{codes}} = 160.15877344941$	$\lambda_{\text{codes}} = 17.94067840444$
$(\lambda \geq)$ $\lambda_{\text{lattices}} + \lambda_{\text{codes}}$	-1.28365766654	-1.26532181404

From the table, we can find that for the same q , the density contribution from principal lattices is less than that from congruence lattices, which is consistent with the result in [96]. However, the bound $c\sqrt{n}$ on the minimum Euclidean distance of principal lattices are much smaller than that for congruence lattices, which leads to the benefit that we can concatenate more codes with it. More codes contribute more in the density exponent. As a result, the bounds on λ are similar.

Next we compare our constructions with Xing's construction. As introduced in Section 3.2, we concatenate finitely many codes with asymptotically good packing families, while Xing concatenated approximately infinitely many codes with asymptotically bad packing families. The two constructions are essentially different. Moreover, we also test the sequences $\{\log_Q \lceil c\sqrt{n} \rceil - \lceil \log_Q \lceil c\sqrt{n} \rceil \rceil\}$, where c equals the values shown in the above table. There are only sparse n 's such that the corresponding value is close to 0.3049. Thus our constructions are different from Xing's as they do not satisfy the requirement in Xing's construction.

Based on the numerical results in Remarks 3.2.4, 3.4.6 and 3.4.10, our packing family derived from congruence lattices has slightly better density exponent than the one from principal lattices, and the one from Xing's construction.

4. DENSE PACKINGS FROM ALGEBRAIC NUMBER FIELDS AND CODES

In this chapter, we generalize the concatenation method in Section 3.1 from the special number field $\mathbb{Q}(\sqrt{-3})$ to arbitrary number field.

4.1 *Canonical Embedding of Algebraic Number Fields*

Let K be an algebraic number field of degree m over $\mathbb{Q}$, and let $\mathcal{O}_K$ be its ring of integers. Suppose K has s real embeddings

$$\rho_1, \dots, \rho_s : K \hookrightarrow \mathbb{R},$$

and t pairs of complex conjugate embeddings

$$\sigma_1, \bar{\sigma}_1, \dots, \sigma_t, \bar{\sigma}_t : K \hookrightarrow \mathbb{C}.$$

Thus $m = s + 2t$. We consider the canonical embedding $\tau : K \hookrightarrow \mathbb{R}^{s+2t}$, that is, for any $\alpha \in K$, $\tau(\alpha)$ is the following vector

$$\left(\rho_1(\alpha), \dots, \rho_s(\alpha), \sqrt{2}\operatorname{Re}\sigma_1(\alpha), \sqrt{2}\operatorname{Im}\sigma_1(\alpha), \dots, \sqrt{2}\operatorname{Re}\sigma_t(\alpha), \sqrt{2}\operatorname{Im}\sigma_t(\alpha) \right), \quad (4.1)$$

where Re denotes the real part and Im denotes the imaginary part of a complex number. τ can be directly extended to $K^n \hookrightarrow \mathbb{R}^{mn}$, which is also denoted by τ without causing any confusion.

We remark that in our new construction, packings come from the embedding of subsets of $\mathcal{O}_K^n$. For totally imaginary number fields, we can scale down the embedding image by $\sqrt{2}$ for simplicity as the scalar multiplication of a packing does not change the density. Thus $\tau_0 : \mathbb{Q}(\sqrt{-3}) \hookrightarrow \mathbb{R}^2$ introduced in Section 3.1 is a special case of the canonical embedding.

In the software **Magma** V2.21-4 [20, 21], the embedding τ defined above is called the Minkowski map and it is employed to compute the “minimum norm” (square of minimum Euclidean distance) of a Minkowski lattice. In Subsection 4.2.4, we will use the software to compute the density of some explicit examples from our construction.

Set $\text{Tr} = \text{Trace}_{K/\mathbb{Q}}$ and $N = \text{Norm}_{K/\mathbb{Q}}$. By the Minkowski Theory [84, Section I.5], the embedding τ maps the non-zero ideals of $\mathcal{O}_K$ to some special lattices in $\mathbb{R}^m$. The following lemma characterizes the determinant of such lattices.

Lemma 4.1.1 ([84, Section I.5]). *For any non-zero ideal $I \subseteq \mathcal{O}_K$, $L_I := \tau(I)$ is a lattice of rank m . The determinant of L_I is*

$$\det L_I = \mathcal{N}(I) \sqrt{|D_K|},$$

where D_K is the discriminant of K and $\mathcal{N}(I) = [\mathcal{O}_K : I]$ is the absolute norm of I .

Generally it remains hard to compute the minimum Euclidean distance of lattices. However, we can estimate a lower bound on the Euclidean length of non-zero points in the lattice L_I . The proof is sketched here and a similar discussion on general ideal lattices can be found in [13]. Note that in [13], the distance between

two points is defined using some special quadratic form, while in this thesis, we only focus on the standard Euclidean distance, which refers to the original meaning of packing in the Euclidean space and brings convenience in computation (**Magma** uses standard Euclidean distance as built-in measure).

Lemma 4.1.2. *Let $I \subseteq \mathcal{O}_K$ be a non-zero ideal. For any $0 \neq \alpha \in I$, the Euclidean length of the vector $\tau(\alpha) \in L_I$ satisfies*

$$\|\tau(\alpha)\| \geq \sqrt{m} \cdot |\mathbf{N}(\alpha)|^{1/m} \geq \sqrt{m} \cdot \mathcal{N}(I)^{1/m}.$$

In other words, $d_E(L_I) \geq \sqrt{m} \cdot \mathcal{N}(I)^{1/m}$.

Proof. The Euclidean length of the vector $\tau(\alpha)$ satisfies

$$\begin{aligned} \|\tau(\alpha)\|^2 &= \sum_{i=1}^s \rho_i(\alpha)^2 + 2 \sum_{j=1}^t \sigma_j(\alpha) \bar{\sigma}_j(\alpha) \\ &\geq (s+2t) \left[\prod_{i=1}^s \rho_i(\alpha)^2 \cdot \prod_{j=1}^t \sigma_j(\alpha) \bar{\sigma}_j(\alpha) \cdot \prod_{j=1}^t \sigma_j(\alpha) \bar{\sigma}_j(\alpha) \right]^{\frac{1}{s+2t}} \\ &= m \left[(\mathbf{N}(\alpha))^{2t} \right]^{1/m} \geq m \cdot \mathcal{N}(I)^{2/m}. \end{aligned}$$

The last $\geq$ becomes an equality if and only if the ideal I is a principal ideal generated by α . □

The following lemma is a basic fact in algebraic number theory (see [84, Section I.2]).

Lemma 4.1.3. *For any non-zero element $\beta \in \mathcal{O}_K$, we have $|\mathbf{N}(\beta)| \geq 1$.*

From the above lemmas, the minimum Euclidean length of non-zero elements in $L_{\mathcal{O}_K}$ satisfies $d_E(L_{\mathcal{O}_K}) \geq \sqrt{m}$. Indeed, as the vector $\tau(1)$ has Euclidean length $\sqrt{m}$,

we have

$$d_E(L_{\mathcal{O}_K}) = \sqrt{m}. \quad (4.2)$$

4.2 Concatenation Based on General Algebraic Number Fields

In this section, we introduce our new construction of dense sphere packings. In particular, several explicit constructions and asymptotically good packing families are provided at the end of this section. Our idea is to apply special concatenating methods on L_I defined in Lemma 4.1.1.

Let $[K : \mathbb{Q}] = m$ and let $\mathfrak{p}$ be a prime ideal of $\mathcal{O}_K$. Assume that the residue field $F_{\mathfrak{p}} = \mathcal{O}_K/\mathfrak{p}$ of $\mathfrak{p}$ is isomorphic to the finite field $\mathbb{F}_q$. Let τ be the canonical embedding defined in Eq. (4.1).

For $i \in \mathbb{N}$, let $L_{\mathfrak{p}^i}^n$ denote the Cartesian product of n identical copies of lattice $L_{\mathfrak{p}^i} = \tau(\mathfrak{p}^i)$. Then $L_{\mathfrak{p}^i}^n$ is a lattice of rank mn by Lemma 4.1.1. Moreover, the determinant satisfies

$$\det L_{\mathfrak{p}^i}^n = (\det L_{\mathfrak{p}^i})^n,$$

which is straightforward from the definition of lattice determinant. In addition, by the definition of $L_{\mathfrak{p}^i}^n$, the minimum Euclidean distance satisfies

$$d_E(L_{\mathfrak{p}^i}^n) = d_E(L_{\mathfrak{p}^i}).$$

For simplicity, in this section we write $L_i = L_{\mathfrak{p}^i}$ for short.

As $\mathcal{O}_K$ is a Dedekind domain (see [84, Section I.3] or [11, Chapter 9]), let $\kappa = \mathcal{O}_K/\mathfrak{p}$, then we have $\dim_{\kappa} \mathfrak{p}^i/\mathfrak{p}^{i+1} = 1$ for all $i \in \mathbb{N}$. We know $\kappa \cong \mathbb{F}_q$. Thus for

$i \in \mathbb{N}$, we can choose the sets $S_i := \left\{ \alpha_1^{(i)} = 0, \alpha_2^{(i)}, \dots, \alpha_q^{(i)} \right\} \subseteq \mathfrak{p}^i$ such that

$$\alpha_1^{(i)} \bmod \mathfrak{p}^{i+1}, \dots, \alpha_q^{(i)} \bmod \mathfrak{p}^{i+1} \quad (4.3)$$

represent the q distinct elements in $\mathfrak{p}^i/\mathfrak{p}^{i+1}$.

4.2.1 Concatenation with One Code

We fix one index $i_0 \in \mathbb{N}_{\geq 1}$ in the discussion here and generalize the result to a family of indices in the next subsection. Let $\mathcal{P}$ be a subset of $L_{i_0}^n$, which can be regarded as a packing in $\mathbb{R}^{mn}$.

Given a q -ary code C with length n , in order to perform our concatenation, we take the code alphabet set of C to be $S_{i_0-1} \subseteq \mathfrak{p}^{i_0-1}$ (see Eq. (4.3)). In this way, C can be regarded as a finite subset of $\mathcal{O}_K^n$. The advantage of using S_{i_0-1} as the alphabet set is that it can help us bound the packing radius of our construction. The details are included in the proof of Proposition 4.2.2.

We define $\tau(\mathbf{c})$ for each codeword $\mathbf{c} = (c_1, \dots, c_n) \in C \subseteq \mathcal{O}_K^n$ as

$$\tau(\mathbf{c}) := (\tau(c_1), \dots, \tau(c_n)) \in \mathbb{R}^{mn},$$

and take $\tau(C) := \{\tau(\mathbf{c}) : \mathbf{c} \in C\}$. We consider the concatenation

$$\tau(C) + \mathcal{P} := \{\mathbf{a} + \mathbf{p} \in \mathbb{R}^{mn} : \mathbf{a} \in \tau(C), \mathbf{p} \in \mathcal{P}\},$$

which corresponds to a packing in $\mathbb{R}^{mn}$. We will analyze the density of such a packing in this subsection.

The following lemma characterizes the minimum Euclidean distance of the concatenation $\tau(C) + \mathcal{P}$, which will play an important role in later discussions.

Lemma 4.2.1. *Let $\mathfrak{p}$ be a non-zero prime ideal in $\mathcal{O}_K$ with absolute norm $\mathcal{N}(\mathfrak{p}) = q$ (i.e., the residue field $F_{\mathfrak{p}} \cong \mathbb{F}_q$). For $i_0 \in \mathbb{N}_{\geq 1}$, let*

(i) $\mathcal{P}$ be a subset of $L_{i_0}^n$,

(ii) C be a q -ary (n, M, d_C) -code over the alphabet set S_{i_0-1} . In addition, C contains the zero codeword.

Then the minimum Euclidean distance of the packing $\tau(C) + \mathcal{P}$ satisfies

$$d_E(\tau(C) + \mathcal{P}) \geq \min \left\{ d_E(L_{i_0-1})\sqrt{d_C}, d_E(\mathcal{P}) \right\}.$$

In particular, if $d_E(L_{i_0-1})\sqrt{d_C} \geq d_E(\mathcal{P})$, we have exactly

$$d_E(\tau(C) + \mathcal{P}) = d_E(\mathcal{P}).$$

Proof. Let $\tau(\mathbf{c}) + \mathbf{p}$ be a non-zero element in $\tau(C) + \mathcal{P}$, where $\mathbf{c} \in C$ and $\mathbf{p} \in \mathcal{P}$. If $\mathbf{c} = \mathbf{0}$, then

$$\|\tau(\mathbf{c}) + \mathbf{p}\| = \|\mathbf{p}\| \geq d_E(\mathcal{P}).$$

If $\mathbf{c} \neq \mathbf{0}$, without loss of generality, we assume $\mathbf{c} = (c_1, \dots, c_e, \mathbf{0})$, where e is the Hamming weight of $\mathbf{c}$ and $c_i \neq 0$ for $1 \leq i \leq e$. As $\mathcal{P} \subseteq L_{i_0}^n$, we can further assume $\mathbf{p} = (\tau(\rho_1), \dots, \tau(\rho_n))$, where $\rho_j \in \mathfrak{p}^{i_0}$ for $1 \leq j \leq n$. Thus

$$\tau(\mathbf{c}) + \mathbf{p} = (\tau(c_1 + \rho_1), \dots, \tau(c_e + \rho_e), \tau(\rho_{e+1}), \dots, \tau(\rho_n)).$$

As $c_1, \dots, c_e$ are non-zero elements in S_{i_0-1} , i.e., for $1 \leq i \leq e$, $c_i \in \mathfrak{p}^{i_0-1} \setminus \mathfrak{p}^{i_0}$, we

have that $c_i + \rho_i$ are non-zero elements in $\mathfrak{p}^{i_0-1}$ for $1 \leq i \leq e$. Thus

$$\|\tau(\mathbf{c}) + \mathbf{p}\|^2 \geq \sum_{i=1}^e \|\tau(c_i + \rho_i)\|^2 \geq e \cdot d_E^2(L_{i_0-1}) \geq d_C \cdot d_E^2(L_{i_0-1}).$$

In summary, the minimum Euclidean distance of the packing $\tau(C) + \mathcal{P}$ satisfies

$$d_E(\tau(C) + \mathcal{P}) \geq \min \left\{ d_E(L_{i_0-1})\sqrt{d_C}, d_E(\mathcal{P}) \right\}. \quad (4.4)$$

In particular, if $d_E(L_{i_0-1})\sqrt{d_C} \geq d_E(\mathcal{P})$, then from Eq. (4.4) above, we have the minimum distance $d_E(\tau(C) + \mathcal{P}) \geq d_E(\mathcal{P})$. Meanwhile, as C contains the zero codeword, $\mathcal{P}$ is a subset of $\tau(C) + \mathcal{P}$, thus $d_E(\mathcal{P}) \geq d_E(\tau(C) + \mathcal{P})$. Finally we have exactly

$$d_E(\tau(C) + \mathcal{P}) = d_E(\mathcal{P}).$$

□

The following proposition generalizes Proposition 2.3 of [109], where only the special case $K = \mathbb{Q}(\sqrt{-3})$ is discussed.

Proposition 4.2.2. *Under the same assumption on $\mathcal{P}$ and C as in Lemma 4.2.1,*

(i) *if $d_E(L_{i_0-1})\sqrt{d_C} \geq d_E(\mathcal{P})$, then the density of $\tau(C) + \mathcal{P}$ as a packing in $\mathbb{R}^{mn}$ is at least $M \cdot \Delta(\mathcal{P})$, where $\Delta(\mathcal{P})$ is the density of the packing $\mathcal{P}$ as a packing in $\mathbb{R}^{mn}$, and M is the size of the code C ;*

(ii) *if $\mathcal{P}$ is a lattice and C satisfies that for any $\mathbf{u}, \mathbf{v} \in C$, the sum $\tau(\mathbf{u}) + \tau(\mathbf{v})$ is equal to $\tau(\mathbf{w}) + \mathbf{p}$ for some $\mathbf{w} \in C$ and $\mathbf{p} \in \mathcal{P}$, then $\tau(C) + \mathcal{P}$ is also a lattice.*

Proof. (i) We denote the volume of the unit sphere in $\mathbb{R}^N$ by V_N and the sphere of radius b by $\mathcal{B}_N(b)$. Let $s = \max\{\|\tau(\mathbf{c})\| : \mathbf{c} \in C\}$. For any $\mathbf{c} \in C$ and $\mathbf{p} \in \mathcal{P} \cap \mathcal{B}_{mn}(b)$,

we have $\tau(\mathbf{c}) + \mathbf{p} \in (\tau(\mathbf{c}) + \mathcal{P}) \cap \mathcal{B}_{mn}(b + s)$. This implies that

$$|(\tau(\mathbf{c}) + \mathcal{P}) \cap \mathcal{B}_{mn}(b + s)| \geq |\mathcal{P} \cap \mathcal{B}_{mn}(b)|.$$

Moreover, as the elements in the alphabet set S_{i_0-1} of C represent the q distinct elements in $\mathfrak{p}^{i_0-1}/\mathfrak{p}^{i_0}$ and $\mathcal{P} \subseteq L_{i_0}^n$, we have if $\mathbf{c}_i \neq \mathbf{c}_j$ then

$$(\tau(\mathbf{c}_i) + \mathcal{P}) \cap (\tau(\mathbf{c}_j) + \mathcal{P}) = \emptyset.$$

We write $d = d_E(\tau(C) + \mathcal{P})$ for short and from Lemma 4.2.1 we have $d = d_E(\mathcal{P})$.

Finally we obtain

$$\begin{aligned} & \Delta(\tau(C) + \mathcal{P}) \\ &= \limsup_{b \rightarrow \infty} \frac{|(\tau(C) + \mathcal{P}) \cap \mathcal{B}_{mn}(b + s)| (d/2)^{mn} V_{mn}}{\text{vol}(\mathcal{B}_{mn}(b + s + d/2))} \\ &= \limsup_{b \rightarrow \infty} \frac{(\sum_{\mathbf{c} \in C} |(\tau(\mathbf{c}) + \mathcal{P}) \cap \mathcal{B}_{mn}(b + s)|) (d/2)^{mn} V_{mn}}{\text{vol}(\mathcal{B}_{mn}(b + s + d/2))} \\ &\geq \limsup_{b \rightarrow \infty} \frac{|C| \cdot |\mathcal{P} \cap \mathcal{B}_{mn}(b)| (d_E(\mathcal{P})/2)^{mn} V_{mn}}{\text{vol}(\mathcal{B}_{mn}(b + s + d/2))} \\ &= |C| \cdot \Delta(\mathcal{P}) = M \cdot \Delta(\mathcal{P}). \end{aligned}$$

(ii) By the definition of a lattice, this part is true. $\square$

4.2.2 Concatenation with a Family of Codes

For a family of q -ary codes $\{C_i\}_{i=0}^{\ell-1}$, where $\ell \in \mathbb{N}_{\geq 1}$, we can take the code alphabet set of C_i to be S_i (see Eq. (4.3)). Note that the choice of the alphabet set of the codes is only used in the proof. For the computation, we only care about the length, size and minimum Hamming distance of the codes.

Similar to Subsection 4.2.1, for any subset $\mathcal{P}$ of L_ℓ^n , and a family of q -ary codes

$\mathcal{C} = \{C_i\}_{i=0}^{\ell-1}$ with length n , we consider the concatenation

$$\tau(\mathcal{C}) + \mathcal{P} := \sum_{i=0}^{\ell-1} \tau(C_i) + \mathcal{P} = \left\{ \sum_{i=0}^{\ell-1} \mathbf{a}_i + \mathbf{p} \in \mathbb{R}^{mn} : \mathbf{a}_i \in \tau(C_i), \mathbf{p} \in \mathcal{P} \right\}. \quad (4.5)$$

Note that for each $0 \leq i \leq \ell - 1$, as the code alphabet set of C_i is S_i , the set $\tau(C_i)$ is a finite subset of L_i^n . Meanwhile $\mathcal{P}$ is a subset of L_ℓ^n . As for any $0 \leq i \leq \ell$, the lattice L_i^n is a sublattice of $L_{\mathcal{O}_K}^n$, the concatenation Eq. (4.5) makes sense within $L_{\mathcal{O}_K}^n$.

Proposition 4.2.3. *Let $\mathfrak{p}$ be a non-zero prime ideal in $\mathcal{O}_K$ with absolute norm $\mathcal{N}(\mathfrak{p}) = q$. For $\ell \in \mathbb{N}_{\geq 1}$, let*

- (i) $\mathcal{P}$ be a subset of L_ℓ^n ,
- (ii) $\mathcal{C} = \{C_i\}_{i=0}^{\ell-1}$ be a family of q -ary codes, where C_i is an (n, M_i, d_{C_i}) -code, the alphabet set of C_i is S_i , and $d_{C_i} \geq \left\lceil \frac{d_E^2(\mathcal{P})}{d_E^2(L_i)} \right\rceil$. In addition, for each $0 \leq i \leq \ell - 1$, C_i contains the zero codeword.

Then the density of $\tau(\mathcal{C}) + \mathcal{P}$ as a packing in $\mathbb{R}^{mn}$ is at least $\Delta(\mathcal{P}) \cdot \prod_{i=0}^{\ell-1} M_i$.

Proof. Let $\mathcal{P}_\ell := \mathcal{P} \subseteq L_\ell^n$. For i from $\ell - 1$ to 0, we can recursively define

$$\mathcal{P}_i := \tau(C_i) + \mathcal{P}_{i+1} \subseteq L_i^n.$$

Note that $\mathcal{P}_0 = \tau(\mathcal{C}) + \mathcal{P}$.

We claim that for $0 \leq i \leq \ell - 1$,

$$d_E(\mathcal{P}_i) = d_E(\mathcal{P}_{i+1}) \quad \text{and} \quad \Delta(\mathcal{P}_i) \geq M_i \cdot \Delta(\mathcal{P}_{i+1}). \quad (4.6)$$

We use induction on $k = \ell - i$ to prove the claim. When $k = 1$, $i = \ell - 1$, as the minimum Hamming distance of $C_{\ell-1}$ satisfies $d_{C_{\ell-1}} d_E^2(L_{\ell-1}) \geq d_E^2(\mathcal{P}) = d_E^2(\mathcal{P}_\ell)$, by

Lemma 4.2.1 and Proposition 4.2.2 (i), we have

$$d_E(\mathcal{P}_{\ell-1}) = d_E(\tau(C_{\ell-1}) + \mathcal{P}_\ell) = d_E(\mathcal{P}_\ell) \quad \text{and} \quad \Delta(\mathcal{P}_{\ell-1}) \geq M_{\ell-1} \cdot \Delta(\mathcal{P}_\ell).$$

Suppose for all k in the range $1 \leq k < k_0 + 1 \leq \ell$, the statement Eq. (4.6) is true, i.e., $i = \ell - k$,

$$d_E(\mathcal{P}_{\ell-k}) = d_E(\mathcal{P}_{\ell-k+1}) \quad \text{and} \quad \Delta(\mathcal{P}_{\ell-k}) \geq M_{\ell-k} \cdot \Delta(\mathcal{P}_{\ell-k+1}) \quad (4.7)$$

is true. By induction, we need to prove

$$d_E(\mathcal{P}_{\ell-k_0-1}) = d_E(\mathcal{P}_{\ell-k_0}) \quad \text{and} \quad \Delta(\mathcal{P}_{\ell-k_0-1}) \geq M_{\ell-k_0-1} \cdot \Delta(\mathcal{P}_{\ell-k_0}).$$

From Eq. (4.7), we have

$$d_E(\mathcal{P}_{\ell-k_0}) = d_E(\mathcal{P}_{\ell-k_0+1}) = \cdots = d_E(\mathcal{P}_{\ell-1}) = d_E(\mathcal{P}_\ell) = d_E(\mathcal{P}).$$

Besides, the minimum Hamming distance of $C_{\ell-k_0-1}$ satisfies

$$d_{C_{\ell-k_0-1}}(L_{\ell-k_0-1}) \geq d_E^2(\mathcal{P}) = d_E^2(\mathcal{P}_{\ell-k_0}).$$

Then by Lemma 4.2.1 and Proposition 4.2.2 (i), as $\mathcal{P}_{\ell-k_0-1} = \tau(C_{\ell-k_0-1}) + \mathcal{P}_{\ell-k_0}$, we get

$$d_E(\mathcal{P}_{\ell-k_0-1}) = d_E(\mathcal{P}_{\ell-k_0}) \quad \text{and} \quad \Delta(\mathcal{P}_{\ell-k_0-1}) \geq M_{\ell-k_0-1} \cdot \Delta(\mathcal{P}_{\ell-k_0}).$$

Thus we have proved the claim Eq. (4.6). From the claim, we easily obtain $\Delta(\mathcal{P}_0) \geq \Delta(\mathcal{P}) \cdot \prod_{i=0}^{\ell-1} M_i$. $\square$

Remark 4.2.4. From the above proposition, given a dense packing $\mathcal{P}$ from the canonical embedding Eq. (4.1) on some prime ideal in a certain algebraic number field, we can concatenate several codes satisfying the requirements in Proposition 4.2.3 to $\mathcal{P}$ to obtain some new denser packings. The density increases by a ratio larger than the product of the sizes of these codes.

Note that generally it is hard to determine the minimum Euclidean distance of $\mathcal{P} \subseteq L_\ell^n \subseteq \mathbb{R}^{mn}$. However, we can consider special choices of $\mathcal{P}$ with some algebraic structure. The following corollary, which plays a crucial role in Subsection 4.2.4 for Examples 4.2.8 - 4.2.11, considers the case $\mathcal{P} = L_\ell^n$. The advantage is that for all $i \in \mathbb{N}$, we have $d_E(L_i^n) = d_E(L_i)$. Instead of searching for the minimum Euclidean distance of L_i^n in $\mathbb{R}^{mn}$, we can search for the minimum Euclidean distance of L_i in $\mathbb{R}^m$. In our examples, m is small such that **Magma** can be used to compute the minimum Euclidean distance.

Corollary 4.2.5. *Let $\mathfrak{p}$ be a non-zero prime ideal in $\mathcal{O}_K$ with absolute norm $\mathcal{N}(\mathfrak{p}) = q$. For $\ell \in \mathbb{N}_{\geq 1}$, let $\mathcal{C} = \{C_i\}_{i=0}^{\ell-1}$ be a family of q -ary codes, where C_i is an (n, M_i, d_{C_i}) -code, the alphabet set of C_i is S_i , and $d_{C_i} \geq \left\lceil \frac{d_E^2(L_\ell)}{d_E^2(L_i)} \right\rceil$. In addition, for each $0 \leq i \leq \ell - 1$, the code C_i contains the zero codeword.*

Then the density of $\tau(\mathcal{C}) + L_\ell^n$ as a packing in $\mathbb{R}^{mn}$ satisfies

$$\Delta(\tau(\mathcal{C}) + L_\ell^n) \geq \frac{(d_E(L_\ell)/2)^{mn} V_{mn}}{\left(q^\ell \sqrt{|D_K|}\right)^n} \cdot \prod_{i=0}^{\ell-1} M_i,$$

where D_K is the discriminant of K and V_{mn} is the volume of the unit sphere in $\mathbb{R}^{mn}$. Moreover, the center density $\delta = \delta(\tau(\mathcal{C}) + L_\ell^n)$ satisfies

$$\log_2 \delta \geq mn \log_2 d_E(L_\ell) - mn - n\ell \log_2 q - \frac{n}{2} \log_2 |D_K| + \sum_{i=0}^{\ell-1} \log_2 M_i.$$

In particular, if for all $0 \leq i \leq \ell - 1$, the code C_i is a linear code of dimension $k_i = \log_q M_i$, then

$$\log_2 \delta \geq mn \log_2 d_E(L_\ell) - mn - n\ell \log_2 q - \frac{n}{2} \log_2 |D_K| + \log_2 q \cdot \sum_{i=0}^{\ell-1} k_i.$$

Proof. Note that $\mathcal{N}(\mathfrak{p}^\ell) = q^\ell$ and then by Lemma 4.1.1

$$\det L_\ell^n = (\det L_\ell)^n = \left(q^\ell \sqrt{|D_K|} \right)^n.$$

□

Remark 4.2.6. For concatenation with finite codes, we care more about the dimension of the packing as we need to compare the new construction with the known good packings in the Euclidean space with the same dimension. Hence we fix the length n of the codes first. As for $0 \leq i \leq \ell - 1$, the lattice L_i is a sublattice of L_0 , we have $d_E(L_i) \geq d_E(L_0)$. Thus the length n satisfies $n \geq \left\lceil \frac{d_E^2(L_\ell)}{d_E^2(L_0)} \right\rceil \geq \lceil q^{2\ell/m} \rceil$ by Eq. (4.2) and Lemma 4.1.2. Therefore, we can concatenate at most $\ell = \left\lfloor \frac{m}{2} \log_q n \right\rfloor$ codes to L_ℓ^n . This bound will be used in the computation of Subsection 4.2.4.

4.2.3 Asymptotic Properties

In this subsection, we show that our construction will lead to asymptotically good packing families, which means we can obtain several constructive bounds on the asymptotic density exponent Eq. (2.2).

We will employ families of linear codes that meet the GV bound. As there exist only exponential time algorithms or randomized polynomial algorithms to construct such families (see [77, Section 17.7]), our corresponding asymptotic density exponent bounds are exponential constructive bounds.

From Eq. (2.1) and Lemma 4.1.2, for $0 \leq i \leq \ell - 1$, we know

$$\frac{d_E^2(L_\ell)}{d_E^2(L_i)} \leq q^{2(\ell-i)/m} |D_K|^{1/m}. \quad (4.8)$$

Different from the finite concatenation in Subsection 4.2.2, for asymptotic results, we first fix ℓ and construct the packing for each $\ell \in \mathbb{N}_{\geq 1}$. The length n_ℓ of the codes is set depending on ℓ .

We uniformly set $n_\ell = \lceil q^{2\ell/m} |D_K|^{1/m} \rceil$. Then by Eq. (4.8), $n_\ell \geq \left\lceil \frac{d_E^2(L_\ell)}{d_E^2(L_i)} \right\rceil$ for all $0 \leq i \leq \ell - 1$. Based on the GV bound Eq. (2.4), for $0 \leq i \leq \ell - 1$, we can choose q -ary

$$\left[n_\ell, k_i^{(n_\ell)}, \left\lceil \frac{d_E^2(L_\ell)}{d_E^2(L_i)} \right\rceil \right]$$

linear code $C_i^{(n_\ell)}$ such that the rate is

$$\frac{k_i^{(n_\ell)}}{n_\ell} \geq R_{GV}(q, \varrho_i^{(n_\ell)}) = 1 - H_q(\varrho_i^{(n_\ell)}),$$

where the relative minimum distance satisfies

$$\varrho_i^{(n_\ell)} = \frac{\left\lceil \frac{d_E^2(L_\ell)}{d_E^2(L_i)} \right\rceil}{n_\ell}.$$

Let $\mathcal{C}^{(n_\ell)} := \left\{ C_i^{(n_\ell)} \right\}_{i=0}^{\ell-1}$. For each n_ℓ , define a packing

$$\mathcal{P}^{(n_\ell)} := \tau(\mathcal{C}^{(n_\ell)}) + L_\ell^{n_\ell}$$

as in Eq. (4.5).

Next we take ℓ increasingly to ∞ . The following proposition describes the

asymptotic density exponent of the packing family $\mathcal{F} = \{\mathcal{P}^{(n_\ell)}\}_{\ell \rightarrow \infty}$, where $n_\ell = \lceil q^{2\ell/m} |D_K|^{1/m} \rceil$ tends to ∞ as ℓ goes to ∞ .

Proposition 4.2.7. *The asymptotic density exponent of the family $\mathcal{F}$ satisfies*

$$\begin{aligned} \lambda(\mathcal{F}) \geq & -1 - \frac{1}{2m} \log_2 |D_K| - \frac{1}{2} \log_2 \frac{m}{2\pi e} \\ & + \limsup_{\ell \rightarrow \infty} \left(\log_2 d_E(L_\ell) - \frac{1}{2} \log_2 n_\ell - \frac{\log_2 q}{m} \sum_{i=0}^{\ell-1} H'_q(\varrho_i^{(n_\ell)}) \right), \end{aligned} \quad (4.9)$$

where $H'_q(\varrho) = H_q(\varrho)$ for $0 < \varrho < \frac{q-1}{q}$ and $H'_q(\varrho) = 1$ for $\frac{q-1}{q} \leq \varrho \leq 1$.

Proof. By Corollary 4.2.5,

$$\begin{aligned} \lambda(\mathcal{F}) & \geq \limsup_{\ell \rightarrow \infty} \frac{1}{mn_\ell} \log_2 \left(\frac{(d_E(L_\ell)/2)^{mn_\ell} V_{mn_\ell}}{(q^\ell \sqrt{|D_K|})^{n_\ell}} \cdot \prod_{i=0}^{\ell-1} M_i^{(n_\ell)} \right) \\ & = -1 - \frac{1}{2m} \log_2 |D_K| + \\ & \quad + \limsup_{\ell \rightarrow \infty} \left(\log_2 d_E(L_\ell) - \frac{\ell}{m} \log_2 q + \frac{1}{mn_\ell} \log_2 V_{mn_\ell} + \sum_{i=0}^{\ell-1} \frac{\log_2 q \cdot k_i^{(n_\ell)}}{mn_\ell} \right). \end{aligned}$$

By Eq. (2.3),

$$\begin{aligned} & \limsup_{\ell \rightarrow \infty} \left(\frac{1}{mn_\ell} \log_2 V_{mn_\ell} + \sum_{i=0}^{\ell-1} \frac{\log_2 q \cdot k_i^{(n_\ell)}}{mn_\ell} \right) \\ & = \limsup_{\ell \rightarrow \infty} \left(-\frac{1}{2} \log_2 \frac{mn_\ell}{2\pi e} - \frac{1}{2mn_\ell} \log_2 mn_\ell \pi + \frac{\log_2 q}{m} \sum_{i=0}^{\ell-1} \frac{k_i^{(n_\ell)}}{n_\ell} \right) \\ & \geq -\frac{1}{2} \log_2 \frac{m}{2\pi e} + \limsup_{\ell \rightarrow \infty} \left(-\frac{1}{2} \log_2 n_\ell + \frac{\log_2 q}{m} \sum_{i=0}^{\ell-1} R_{GV}(q, \varrho_i^{(n_\ell)}) \right) \\ & = -\frac{1}{2} \log_2 \frac{m}{2\pi e} + \limsup_{\ell \rightarrow \infty} \left(-\frac{1}{2} \log_2 n_\ell + \frac{\log_2 q}{m} \sum_{i=0}^{\ell-1} (1 - H'_q(\varrho_i^{(n_\ell)})) \right) \\ & = -\frac{1}{2} \log_2 \frac{m}{2\pi e} + \limsup_{\ell \rightarrow \infty} \left(\frac{\ell}{m} \log_2 q - \frac{1}{2} \log_2 n_\ell - \frac{\log_2 q}{m} \sum_{i=0}^{\ell-1} H'_q(\varrho_i^{(n_\ell)}) \right). \end{aligned}$$

In summary, we get Eq. (4.9). □

4.2.4 Examples

We use some explicit examples to illustrate our new construction. Here **Magma** V2.21-4 [20, 21] will be employed to obtain the numerical results.

For the finite concatenation, we use the linear codes from [58] which have an explicit construction. For the asymptotic result, we set sufficiently large ℓ to approximate the limit.

Example 4.2.8. Consider the number field $K = \mathbb{Q}[\alpha]$, where α is a root of the irreducible polynomial

$$f(x) = x^4 - x^3 - x^2 + x + 1 \in \mathbb{Q}[x].$$

$K/\mathbb{Q}$ is a totally complex number field with extension degree $[K : \mathbb{Q}] = 4$ and absolute discriminant $|D_K| = 117$. The absolute discriminant is the smallest one of all totally complex number fields with degree 4 (see [88]).

- (i) We consider a prime ideal $\mathfrak{p}$ lying over $3 \in \mathbb{Z}$. The **Magma** code of this example is listed in Appendix A, while for other examples, the **Magma** code can be modified from this template.

The key outputs of Appendix A are listed here:

```
The degree of K=Q[x]/(x^4 - x^3 - x^2 + x + 1) is m=4;
The absolute value of the discriminant of K is |d|=117;
P is a Prime Ideal of O
Two element generators:
[3, 0, 0, 0]
```

$[2, 1, 1, 0]$ lying over 3 with norm $q=9$;

The above statement means $\mathfrak{p} = (3, 2 + \alpha + \alpha^2)$.

Finite Concatenation:

We can concatenate at most 3 linear codes to the lattice
constructed by 64 copies of $L_{\{P^3\}}$,
whose Hamming weights are required respectively at least
 $27;9;3$;

Referring to [58], as the norm of $\mathfrak{p}$ is 9, we can use the existing 9-ary linear codes C_0, C_1, C_2 with parameters

$$[64, 25, 27], [64, 49, 9], [64, 61, 3]$$

respectively. The sum of the dimensions is 135. Considering the concatenation in Corollary 4.2.5, we obtain a packing with dimension $4 \times 64 = 256$, whose center density δ satisfies

Our packing is in dimension 256 with $\text{Log}_2(\text{center density})$
at least 208.088204168043224246772217517

Note that our packing is denser than the Barnes-Wall lattice BW_{256} , whose density is listed in the table of dense sphere packings [43, Table 1.3].

Using this ideal to construct a packing family as in Proposition 4.2.7, we get the following result.

Asymptotic result:

The asymptotic density exponent of the packing family is
at least -1.442426720

Note that the above result means our packing family is asymptotically good.

- (ii) For this field, we similarly analyze other prime ideals and list part of our numerical results in Table 4.1, where δ is the center density and λ is the asymptotic density of the corresponding packing family.

Table 4.1: Examples constructed from $\mathbb{Q}[x]/(x^4 - x^3 - x^2 + x + 1)$

$\mathfrak{p}$	q	dimension	$\log_2 \delta$ at least	λ at least
$(3, 2 + \alpha + \alpha^2)$	9	180	108.52	-1.442
		256	208.09 ¹	
		512	590.52	
$(7, 4 + \alpha)$	7	256	190.63	-1.453
		400	410.15	

¹ The packing can be explicitly constructed with $\log_2 \delta$ larger than 192 given by the Barnes-Wall lattice BW_{256} .

Example 4.2.9. Consider the number field $K = \mathbb{Q}[\alpha]$, where α is a root of the irreducible polynomial

$$f(x) = x^3 + x^2 - 2x - 1 \in \mathbb{Q}[x].$$

$K/\mathbb{Q}$ is a totally real number field with extension degree $[K : \mathbb{Q}] = 3$ and absolute discriminant $|D_K| = 49$. The absolute discriminant is the smallest one of all totally real cubic number fields (see [88]). The numerical results on our examples are partially listed in Table 4.2.

Example 4.2.10. Consider the number field $K = \mathbb{Q}[\alpha]$, where α is a root of the

Table 4.2: Examples constructed from $\mathbb{Q}[x]/(x^3 + x^2 - 2x - 1)$

$\mathfrak{p}$	q	dimension	$\log_2 \delta$ at least	λ at least
(2)	8	255	134.46	-1.628
$(7, 5 + \alpha)$	7	192	83.68	-1.585
		255	157.63	

irreducible polynomial

$$f(x) = x^3 + x^2 - 1 \in \mathbb{Q}[x].$$

$K/\mathbb{Q}$ is a number field with extension degree $[K : \mathbb{Q}] = 3$ and absolute discriminant $|D_K| = 23$. The absolute discriminant is the smallest one of all cubic number fields (see [88]). The numerical results on our examples are partially listed in Table 4.3.

Table 4.3: Examples constructed from $\mathbb{Q}[x]/(x^3 + x^2 - 1)$

$\mathfrak{p}$	q	dimension	$\log_2 \delta$ at least	λ at least
(2)	8	96	24.70	-1.429
		192	115.40	
$(5, 2 + \alpha)$	5	150	69.47	-1.445
		180	101.01	
$(7, 11 + \alpha)$	7	255	187.32	-1.430

Example 4.2.11. Consider the number field $K = \mathbb{Q}[\alpha]$, where α is a root of the irreducible polynomial

$$f(x) = x^6 + x^3 + 1 \in \mathbb{Q}[x].$$

$K/\mathbb{Q}$ is a number field with extension degree $[K : \mathbb{Q}] = 6$ and absolute discriminant $|D_K| = 19683$. The polynomial $f(x) = x^6 + x^3 + 1$ is the 9th cyclotomic polynomial over $\mathbb{Q}$. The numerical results on our examples are partially listed in Table 4.4.

Table 4.4: Examples constructed from $\mathbb{Q}[x]/(x^6 + x^3 + 1)$

$\mathfrak{p}$	q	dimension	$\log_2 \delta$ at least	λ at least
$(3, 2 + \alpha)$	3	180	109.71	-1.868
		192	122.72	

5. POLICY-BASED SIGNATURE SCHEMES FROM LATTICES

5.1 Policy-Based Signatures

We first recall the formal definitions and security requirements for policy-based signatures (PBS) [14].

5.1.1 Definitions

Policy. A policy $\mathbf{p}$ is a binary string, i.e., $\mathbf{p} \in \{0, 1\}^*$.

Policy Checker. A policy checker is defined as an **NP**-relation $\text{PC} : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}$, where the first input is a pair $(\mathbf{p}, \mathbf{M})$ consisting of a policy $\mathbf{p}$ and a message $\mathbf{M}$, and the second input is a witness $\mathbf{w}$.

Policy Language. The policy language associated to PC is defined as

$$\mathcal{L}(\text{PC}) = \{(\mathbf{p}, \mathbf{M}) : \exists \mathbf{w} \in \{0, 1\}^* \text{ such that } \text{PC}((\mathbf{p}, \mathbf{M}), \mathbf{w}) = 1\}.$$

If $(\mathbf{p}, \mathbf{M}) \in \mathcal{L}(\text{PC})$ then we say that the message $\mathbf{M}$ satisfies the policy $\mathbf{p}$. The tuple $(\mathbf{p}, \mathbf{M}, \mathbf{w})$ is called **PC-valid** if $\text{PC}((\mathbf{p}, \mathbf{M}), \mathbf{w}) = 1$.

PBS Scheme. A policy-based signature scheme is a tuple of four polynomial-time algorithms $\mathcal{PBS} = (\text{Setup}, \text{KeyGen}, \text{Sign}, \text{Verify})$:

- (1) **Setup:** On input 1^n , this algorithm outputs the public parameters pp and

a master secret key $\mathbf{msk}$.

- (2) **KeyGen**: On input $\mathbf{msk}$ and a policy $\mathbf{p} \in \{0, 1\}^*$, this algorithm outputs a signing key $\mathbf{sk}_{\mathbf{p}}$ for $\mathbf{p}$.
- (3) **Sign**: On input $\mathbf{sk}_{\mathbf{p}}$, $\mathbf{M} \in \{0, 1\}^*$ and $\mathbf{w} \in \{0, 1\}^*$, this algorithm outputs a signature Σ if $\text{PC}((\mathbf{p}, \mathbf{M}), \mathbf{w}) = 1$, or $\perp$ otherwise.
- (4) **Verify**: On input $\mathbf{pp}$, $\mathbf{M}$, Σ , this algorithm returns a bit from $\{0, 1\}$.

The scheme satisfies correctness with respect to a policy checker PC if for all $n \in \mathbb{N}$, all PC -valid tuples $(\mathbf{p}, \mathbf{M}, \mathbf{w})$, all $(\mathbf{pp}, \mathbf{msk}) \in [\text{Setup}(1^n)]$, and all signatures $\Sigma \in [\text{Sign}(\text{KeyGen}(\mathbf{msk}, \mathbf{p}), \mathbf{M}, \mathbf{w})]$, we have $\text{Verify}(\mathbf{pp}, \mathbf{M}, \Sigma) = 1$.

5.1.2 Security Requirements

A secure policy-based signature scheme should satisfy two basic and intuitive requirements: *unforgeability* and *indistinguishability*. Unforgeability requires that it be infeasible to create a valid signature on message $\mathbf{M}$ without possessing a signing key for some policy $\mathbf{p}$ such that $(\mathbf{p}, \mathbf{M}) \in \mathcal{L}(\text{PC})$. Indistinguishability says that the verifier cannot tell under which of the two keys $\mathbf{sk}_{\mathbf{p}_1}$, $\mathbf{sk}_{\mathbf{p}_2}$ a signature on message $\mathbf{M}$ was produced, assuming that $\mathbf{M}$ satisfies both policies $\mathbf{p}_1$ and $\mathbf{p}_2$. As discussed in [14], these two notions do have some weaknesses, and thus, two stronger notions, *simulatability* and *extractability*, were introduced. These strengthened notions capture the spirit of the “sim-ext” security for signatures of knowledge [35], and in particular, they imply indistinguishability and unforgeability.

Simulatability. This property requires the existence of a simulator SimSign that does not have access to any signing key, but that can generate simulated signatures which are indistinguishable from the ones produced by a legitimate signer. The

simulatability experiment $\mathbf{Exp}_{\mathcal{PBS}, \mathcal{A}}^{\text{SIM}}(n)$ between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ is defined as follows:

Initialization: $\mathcal{C}$ runs $(\text{pp}, \text{msk}) \leftarrow \text{Setup}(1^n)$ and returns (pp, msk) to $\mathcal{A}$.

Queries: $\mathcal{A}$ adaptively chooses policies $\mathbf{p}_1, \dots, \mathbf{p}_Q$ and queries the signing key for each $\mathbf{p}_i$, $i \in [Q]$. To answer each query, $\mathcal{C}$ generates $\text{sk}_i \leftarrow \text{KeyGen}(\text{msk}, \mathbf{p}_i)$, records sk_i , then returns sk_i to $\mathcal{A}$.

Challenge: $\mathcal{A}$ sends $i \in [Q]$, $\mathbf{M}^*$ and $\mathbf{w}^*$ to $\mathcal{C}$. If $\text{PC}((\mathbf{p}_i, \mathbf{M}^*), \mathbf{w}^*) = 0$, then return $\perp$. Otherwise, $\mathcal{C}$ picks a uniformly random bit $b \xleftarrow{\$} \{0, 1\}$, and computes $\Sigma_0 \leftarrow \text{SimSign}(\mathbf{M}^*)$ and $\Sigma_1 \leftarrow \text{Sign}(\text{sk}_i, \mathbf{M}^*, \mathbf{w}^*)$. It then sends the challenge signature Σ_b to $\mathcal{A}$.

Finalization: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. The experiment returns **true** if $b' = b$.

We define the advantage of $\mathcal{A}$ in the above experiment as:

$$\mathbf{Adv}_{\mathcal{PBS}, \mathcal{A}}^{\text{SIM}}(n) = \left| \Pr [\mathbf{Exp}_{\mathcal{PBS}, \mathcal{A}}^{\text{SIM}}(n) \Rightarrow \text{true}] - 1/2 \right|.$$

The scheme $\mathcal{PBS}$ is said to satisfy simulatability if $\mathbf{Adv}_{\mathcal{PBS}, \mathcal{A}}^{\text{SIM}}(n)$ is negligible for any adversary $\mathcal{A}$.

Extractability. This property requires the existence of an extractor **Extract** that, on input a message-signature pair $(\mathbf{M}, \Sigma)$, outputs a policy-witness pair $(\mathbf{p}, \mathbf{w})$, such that the advantage of any PPT adversary $\mathcal{A}$ in the following experiment $\mathbf{Exp}_{\mathcal{PBS}, \mathcal{A}}^{\text{EXT}}(n)$ is negligible in n , where the advantage is defined as:

$$\mathbf{Adv}_{\mathcal{PBS}, \mathcal{A}}^{\text{EXT}}(n) = \Pr [\mathbf{Exp}_{\mathcal{PBS}, \mathcal{A}}^{\text{EXT}} \Rightarrow \text{true}].$$

Initialization: $\mathcal{C}$ runs $(\text{pp}, \text{msk}) \leftarrow \text{Setup}(1^n)$ and initializes two recorders $F_K \leftarrow \emptyset$ and $F_S \leftarrow \emptyset$. It then sends pp to $\mathcal{A}$.

Queries: $\mathcal{A}$ adaptively makes the following queries:

- Query for signing key for any policy $\mathbf{p}$: $\mathcal{C}$ generates $\text{sk}_{\mathbf{p}} \leftarrow \text{KeyGen}(\text{msk}, \mathbf{p})$, sets $F_K \leftarrow F_K \cup \{\mathbf{p}\}$, and returns $\text{sk}_{\mathbf{p}}$.
- Query for simulated signature on any message $\mathbf{M}$: $\mathcal{C}$ generates $\Sigma_{\mathbf{M}} \leftarrow \text{SimSign}(\mathbf{M})$, sets $F_S \leftarrow F_S \cup \{(\mathbf{M}, \Sigma_{\mathbf{M}})\}$, and returns $\Sigma_{\mathbf{M}}$.

Finalization: $\mathcal{A}$ outputs a pair $(\mathbf{M}^*, \Sigma^*)$. If $\text{Verify}(\text{pp}, \mathbf{M}^*, \Sigma^*) = 0$ or $(\mathbf{M}^*, \Sigma^*) \in F_S$, then return **false**. Otherwise, $\mathcal{C}$ computes $(\mathbf{p}, \mathbf{w}) \leftarrow \text{Extract}(\mathbf{M}^*, \Sigma^*)$. If $\mathbf{p} \notin F_K$ or $\text{PC}((\mathbf{p}, \mathbf{M}^*), \mathbf{w}) = 0$, then return **true**, while in other cases, return **false**.

We remark that, the definitions of simulatability and extractability described above are the adaptations into the random oracle model of the ones given in [14]. Specifically, we assume that the simulator SimSign has the ability to program the random oracle, and the extractor Extract is capable of replaying the adversary. On the other hand, it is worth noting that the definition of simulatability in [14] only targets polynomial-time adversaries, while ours addresses arbitrary (even computationally unbounded) adversaries.

5.2 Overview of Our Techniques

We will construct policy-based signature schemes under the random oracle model. Our starting point is the Bonsai tree signature scheme proposed by Cash et al. [33] that admits an interactive zero-knowledge argument of knowledge of a valid message-signature pair $(\mathbf{p}, \mathbf{z}) \in \{0, 1\}^\ell \times \mathbb{Z}^{(\ell+1)m}$, given by Langlois et al. [65]. However, we need to solve several questions in developing this protocol to construct policy-based

signatures. The first one is how to define a reasonable policy language that captures the features of real-life scenarios such as:

- (i) The message space should be sufficiently large (e.g., exponential in the security parameter n), while the policy space could be smaller (e.g., polynomial in n).
- (ii) A policy $\mathbf{p}$ could possibly allow many messages.
- (iii) A message $\mathbf{M}$ may simultaneously satisfy many policies.

To address this question, we then come up with the following instantiation. Let $\ell = O(\log n)$ and let d be an integer such that $n - \ell < d$. Let $\mathbf{G}_1 \in \mathbb{Z}_2^{n \times \ell}$ and $\mathbf{G}_2 \in \mathbb{Z}_2^{n \times d}$ be two matrices. Then, a message $\mathbf{M} \in \mathbb{Z}_2^n$ is said to satisfy policy $\mathbf{p} \in \{0, 1\}^\ell$ if there exists a witness vector $\mathbf{w} \in \{0, 1\}^d$ such that:

$$\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{G}_2 \cdot \mathbf{w} = \mathbf{M} \pmod{2}. \quad (5.1)$$

The linear relation given in Eq. (5.1) is in fact reminiscent of the well-known LWE relation $\mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e} = \mathbf{b} \pmod{q}$ in lattice-based cryptography (with matrices $\mathbf{G}_1, \mathbf{G}_2$ instead of the matrix $\mathbf{A}^\top$ and the identity matrix, respectively; and with a different modulus, different parameters n, ℓ, d , and different constraints for the vectors $\mathbf{p}, \mathbf{w}$). However, if we employ an LWE-like relation, then the resulting policy language would not fulfil the practice-driven feature (iii) listed above. Nevertheless, from a technical point of view, our protocol can be adapted to handle an LWE-like relation (e.g., relation among the plaintext, the encryption randomness and the ciphertext in the Dual-Regev encryption scheme [53]), which is potentially useful in the context of other lattice-based cryptographic constructions.

We now will sketch our policy-based signature for the policy language described above. To sign a message $\mathbf{M}$ satisfying policy $\mathbf{p}$, a signer who possesses a pair $(\mathbf{p}, \mathbf{z})$

and a witness $\mathbf{w}$ generates a zero-knowledge argument system to prove that:

Fact 1 : $(\mathbf{p}, \mathbf{z})$ is a valid message-signature pair for the Bonsai tree signature.

Fact 2 : Eq. (5.1) holds.

As discussed earlier, one can use the construction in [65] as a sub-protocol to prove **Fact 1** in zero-knowledge. The remaining question is to prove that the vector $\mathbf{p}$ from **Fact 1** *simultaneously* satisfies **Fact 2**. Towards this goal, we design a sub-protocol to prove **Fact 2**, such that it is compatible with the one from [65], and then combine them together. In the process, we adapt a trick from [70] to make some intriguing extensions to the matrix $\mathbf{G}_1$ and the vector $\mathbf{p}$, so that we can hide $\mathbf{p}$ using a “one-time-pad” as in [65]. The techniques are described in detail in Subsection 5.3.2.

Next, we transform the interactive protocol obtained into a signature scheme in the random oracle model, by repeating it $t = \omega(\log n)$ times to make the soundness error negligibly small, and then making it non-interactive using the Fiat-Shamir heuristic. Roughly speaking, the simulatability of the resulting policy-based signature scheme relies on the statistical zero-knowledge property of the underlying argument system, while its extractability is based on the facts that the Bonsai tree signature scheme is unforgeable, and the underlying argument system is indeed an argument of knowledge. The description of the scheme and the security proofs are presented in Section 5.4.

Finally, we summarize the extension of our PBS scheme to a delegatable one. Recall that in [14], such an extension is based on append-only signatures [63]. Our approach here is more efficient and more straightforward, thanks to the hierarchical structure of the Bonsai tree. Specifically, each policy is associated with a subtree and the signing key is defined to be a short *basis* of the corresponding lattice (instead

of a single short vector as in our basic PBS scheme). We then employ the basis delegation technique in [33] to realize the key delegation mechanism of the scheme. Now, to sign a message, the key holder first samples a Bonsai tree signature-vector, and then follows the same strategy as in the basic scheme, where the underlying argument system is a simple extension of the protocol discussed above. The details are given in Section 5.5.

5.3 The Underlying Zero-Knowledge Argument System

5.3.1 Statistical Zero-Knowledge Argument Systems

We will work with statistical zero-knowledge argument systems, namely, interactive protocols where the zero-knowledge property holds against *any* cheating verifier, while the soundness property only holds for *computationally bounded* cheating provers.

More formally, let $R : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}$ be an NP-relation. A two-party game $\langle P, V \rangle$ is called an interactive argument system for the relation R with soundness error e if the following two conditions hold:

- **Completeness.** If $R(y, w) = 1$ then $\Pr[\langle P(y, w), V(y) \rangle = 1] = 1$.
- **Soundness.** If $R(y, w) = 0$, then for every PPT P^* : $\Pr[\langle P^*(y, w), V(y) \rangle = 1] \leq e$.

An interactive argument system is called statistical zero-knowledge if for any $V^*(y)$, there exists a PPT simulator $\mathcal{Sim}(y)$ producing a simulated transcript that is statistically close to the one of the real interaction between $P(y, w)$ and $V^*(y)$.

A related notion is an *argument of knowledge*, which requires the witness-extended emulation property. For 3-move protocols (i.e., commitment-challenge-response),

the witness-extended emulation is implied by *special soundness* [59], where the latter assumes that there exists a PPT extractor $\mathcal{K}$ which takes as input a set of valid transcripts with respect to all possible values of the ‘challenge’ to the same ‘commitment’, and outputs w' such that $R(y, w') = 1$.

Statistical zero-knowledge arguments of knowledge (ZKAoK) are usually constructed using a statistically hiding and computationally binding string commitment scheme (see, e.g., [85]). In this chapter, we will employ the lattice-based string commitment scheme $\text{COM} : \{0, 1\}^* \times \{0, 1\}^{\bar{m}/2} \rightarrow \mathbb{Z}_q^n$ designed by Kawachi et al. [62]. For appropriate choices of parameters, their scheme is statistically hiding, while the binding property relies on the hardness of SIS. We summarize the property of the commitment scheme from [62] in the following lemma.

Lemma 5.3.1 (Adapted from [62]). *There exists a string commitment scheme*

$$\text{COM} : \{0, 1\}^* \times \{0, 1\}^{\bar{m}/2} \rightarrow \mathbb{Z}_q^n$$

such that:

1. *If $\bar{m} \geq 3n \log q$, then COM is statistically hiding.*
2. *If the $\text{SIS}_{n, \bar{m}, q, 1}^\infty$ problem is hard, then COM is computationally binding.*

5.3.2 A Zero-Knowledge Argument of Knowledge for the Relation R_{PBS}

We now will present an argument of knowledge which will be used as the building block in our policy-based signature scheme in Section 5.4. Our interactive protocol allows the prover $\mathcal{P}$ to convince the verifier $\mathcal{V}$ in zero-knowledge that $\mathcal{P}$ knows a secret triple $(\mathbf{p}, \mathbf{z}, \mathbf{w})$, such that $(\mathbf{p}, \mathbf{z})$ is a valid message-signature pair for the Bonsai tree signature scheme [33] and, at the same time, the pair $(\mathbf{p}, \mathbf{w})$ satisfies an

additional requirement. Specifically, let n, m, ℓ, d, β and $q > \beta$ be positive integers, we will construct a ZKAoK for the NP-relation R_{PBS} defined as follows:

Definition 5.3.2. For $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \dots | \mathbf{A}_\ell^0 | \mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n \times (2\ell+1)m}$, $\mathbf{u} \in \mathbb{Z}_q^n$, $\mathbf{G}_1 \in \mathbb{Z}_2^{n \times \ell}$, $\mathbf{G}_2 \in \mathbb{Z}_2^{n \times d}$ and $\mathbf{M} \in \mathbb{Z}_2^n$, we say that $R_{\text{PBS}}((\mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}), \mathbf{p}, \mathbf{z}, \mathbf{w}) = 1$ if $\mathbf{p} = \mathbf{p}[1] \dots \mathbf{p}[\ell] \in \{0, 1\}^\ell$, $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$, $\mathbf{w} \in \{0, 1\}^d$, and the following facts hold:

- (1) $\|\mathbf{z}\|_\infty \leq \beta$ and $\mathbf{A}_{\mathbf{p}} \cdot \mathbf{z} = \mathbf{u} \bmod q$, where $\mathbf{A}_{\mathbf{p}} = [\mathbf{A}_0 | \mathbf{A}_1^{\mathbf{p}[1]} | \dots | \mathbf{A}_\ell^{\mathbf{p}[\ell]}] \in \mathbb{Z}_q^{n \times (\ell+1)m}$
 - a sub-matrix of $\mathbf{A}$, determined by $\mathbf{p}$.
- (2) $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{G}_2 \cdot \mathbf{w} = \mathbf{M} \bmod 2$.

Our approach is summarized as follows. To prove the possession of a pair $(\mathbf{p}, \mathbf{z})$ satisfying Fact (1), we employ Langlois et al.'s argument system [65] for the Bonsai tree signature scheme. We then develop the protocol so that the prover can additionally prove in zero-knowledge the possession of $\mathbf{w} \in \{0, 1\}^d$ such that, for the *same* $\mathbf{p}$ in Fact (1), the pair $(\mathbf{p}, \mathbf{w})$ satisfies Fact (2). Both Langlois et al.'s protocol and our development rely on the Decomposition-Extension technique, introduced in [70].

The Decomposition-Extension Technique.

In [70], Ling et al. designed a zero-knowledge argument for the Inhomogeneous SIS problem, which allows $\mathcal{P}$ to convince $\mathcal{V}$ that $\mathcal{P}$ knows a vector $\mathbf{x} \in [-\beta; \beta]^m$ such that $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$, for given $(\mathbf{A}, \mathbf{u}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^n$. Towards this goal, they observed that an integer $x \in [0; \beta]$ if and only if there exist $x_1, \dots, x_k \in \{0, 1\}$ such that $x = \sum_{i=1}^k \beta_i x_i$, where $k = \lceil \log \beta \rceil + 1$ and the sequence of integers $\beta_1, \dots, \beta_k$ is determined as follows:

$$\beta_1 = \lceil \beta/2 \rceil; \beta_2 = \lceil (\beta - \beta_1)/2 \rceil; \beta_3 = \lceil (\beta - \beta_1 - \beta_2)/2 \rceil; \dots; \beta_k = 1. \quad (5.2)$$

This observation allows us to efficiently decompose any vector $\mathbf{x} \in [-\beta; \beta]^m$ into $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_k \in \{-1, 0, 1\}^m$ such that $\sum_{i=1}^k \beta_i \tilde{\mathbf{x}}_i = \mathbf{x}$. Each vector $\tilde{\mathbf{x}}_i$ is then extended to $\mathbf{x}_i \in \mathbf{B}_{3m}$, where $\mathbf{B}_{3m}$ is the set of all vectors in $\{-1, 0, 1\}^{3m}$ having exactly m coordinates equal 0, m coordinates equal to 1, and m coordinates equal to -1 . This set has a helpful property: if $\pi \in \mathbf{S}_{3m}$ - the symmetric group of all permutations of $3m$ elements, then $\mathbf{x}_i \in \mathbf{B}_{3m}$ if and only if $\pi(\mathbf{x}_i) \in \mathbf{B}_{3m}$. Thus, to prove that $\mathbf{x} \in [-\beta; \beta]^m$, the prover can instead convince that $\mathbf{x}_i \in \mathbf{B}_{3m}$, for all $i \in [k]$, by using random permutations.

At the same time, let $\hat{\mathbf{A}} \in \mathbb{Z}_q^{n \times 3m}$ be the extended matrix obtained by appending $2m$ “dummy” zero-columns to $\mathbf{A}$, then one has $\mathbf{A} \cdot \mathbf{x} = \hat{\mathbf{A}} \cdot (\sum_{i=1}^k \beta_i \mathbf{x}_i) = \mathbf{u} \bmod q$. Thus, to prove that $\mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q$, the prover instead shows that $\hat{\mathbf{A}} \cdot (\sum_{i=1}^k \beta_i \mathbf{x}_i) = \mathbf{u} \bmod q$.

Langlois et al.’s Construction.

In [65], Langlois et al. adapted the Decomposition-Extension technique from [70] to the Bonsai tree setting. Specifically, given a vector $\mathbf{z} \in [-\beta; \beta]^{(\ell+1)m}$ such that $\mathbf{A}_{\mathbf{p}} \cdot \mathbf{z} = \mathbf{u} \bmod q$ for some $\mathbf{p} \in \{0, 1\}^\ell$, they constructed a procedure $\text{WitnessDE}(\mathbf{p}, \mathbf{z})$ that outputs $k = \lfloor \log \beta \rfloor + 1$ vectors $\mathbf{z}_1, \dots, \mathbf{z}_k \in \text{SecretExt}(\mathbf{p})$, such that $\mathbf{A}^* \cdot (\sum_{i=1}^k \beta_i \mathbf{z}_i) = \mathbf{u} \bmod q$, where the extended matrix $\mathbf{A}^* \in \mathbb{Z}_q^{n \times (2\ell+1) \cdot 3m}$ is defined as

$$\mathbf{A}^* = [\mathbf{A}_0 \mid \mathbf{0}^{n \times 2m} \mid \mathbf{A}_1^0 \mid \mathbf{0}^{n \times 2m} \mid \mathbf{A}_1^1 \mid \mathbf{0}^{n \times 2m} \mid \dots \mid \mathbf{A}_\ell^0 \mid \mathbf{0}^{n \times 2m} \mid \mathbf{A}_\ell^1 \mid \mathbf{0}^{n \times 2m}],$$

and $\text{SecretExt}(\mathbf{p})$, for $\mathbf{p} = \mathbf{p}[1] \dots \mathbf{p}[\ell] \in \{0, 1\}^\ell$, is defined as the set of all vectors

$$\mathbf{x} = (\mathbf{x}_0 \parallel \mathbf{x}_1^0 \parallel \mathbf{x}_1^1 \parallel \dots \parallel \mathbf{x}_\ell^0 \parallel \mathbf{x}_\ell^1) \in \{-1, 0, 1\}^{(2\ell+1) \cdot 3m}$$

having $2\ell + 1$ blocks of size $3m$, such that the $\ell + 1$ blocks $\mathbf{x}_0, \mathbf{x}_1^{\mathbf{p}[1]}, \dots, \mathbf{x}_\ell^{\mathbf{p}[\ell]}$ are

elements of $\mathcal{B}_{3m}$, while the remaining ℓ blocks $\mathbf{x}_1^{1-\mathbf{p}[1]}, \dots, \mathbf{x}_\ell^{1-\mathbf{p}[\ell]}$ are zero-blocks $\mathbf{0}^{3m}$. The procedure $\text{WitnessDE}(\mathbf{p}, \mathbf{z})$ works as follows:

1. Write $\mathbf{z}$ as the concatenation of $\ell+1$ blocks of size m , i.e., $\mathbf{z} = (\mathbf{w}_0 || \mathbf{w}_1 || \dots || \mathbf{w}_\ell)$.
2. For $j = 0, 1, \dots, \ell$, apply the Decomposition-Extension technique to the block $\mathbf{w}_j$, to obtain k vectors $\mathbf{w}_{j,1}, \dots, \mathbf{w}_{j,k} \in \mathcal{B}_{3m}$.
3. For each $i = 1, \dots, k$, let $\mathbf{z}_i = \left((\mathbf{x}_0)_i || (\mathbf{x}_1^0)_i || (\mathbf{x}_1^1)_i || \dots || (\mathbf{x}_\ell^0)_i || (\mathbf{x}_\ell^1)_i \right)$, where $(\mathbf{x}_0)_i = \mathbf{w}_{0,i}$, and for each $j = 1, \dots, \ell$: $(\mathbf{x}_j^{\mathbf{p}[j]})_i = \mathbf{w}_{j,i}$ and $(\mathbf{x}_j^{1-\mathbf{p}[j]})_i = \mathbf{0}^{3m}$.
4. Output $\mathbf{z}_1, \dots, \mathbf{z}_k \in \text{SecretExt}(\mathbf{p})$. It can be checked that, by construction, $\mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i \mathbf{z}_i \right) = \mathbf{u} \bmod q$.

The underlying idea of Langlois et al.'s construction is that: One can efficiently show that $\mathbf{z}_i \in \text{SecretExt}(\mathbf{p})$ while keeping *both* $\mathbf{z}_i$ and $\mathbf{p}$ secret, by employing a composition of two random permutations. Specifically, for a vector

$$\mathbf{x} = (\mathbf{x}_0 || \mathbf{x}_1^0 || \mathbf{x}_1^1 || \dots || \mathbf{x}_\ell^0 || \mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1) \cdot 3m}$$

having $2\ell + 1$ blocks of size $3m$, the following two sets of permutations of $\mathbf{x}$ are considered:

- The set $\mathcal{S}$ consists of all permutations that keep the arrangement of the blocks.

If $\pi \in \mathcal{S}$, then:

$$\pi(\mathbf{x}) = (\tau_0(\mathbf{x}_0) || \tau_1^0(\mathbf{x}_1^0) || \tau_1^1(\mathbf{x}_1^1) || \dots || \tau_\ell^0(\mathbf{x}_\ell^0) || \tau_\ell^1(\mathbf{x}_\ell^1)) ,$$

where $\tau_0, \tau_1^0, \tau_1^1, \dots, \tau_\ell^0, \tau_\ell^1$ are certain permutations of $3m$ elements.

- The set $\mathcal{T} = \{T_{\mathbf{e}} : \mathbf{e} \in \{0, 1\}^\ell\}$, where for $\mathbf{e} = \mathbf{e}[1] \cdots \mathbf{e}[\ell] \in \{0, 1\}^\ell$, the permutation $T_{\mathbf{e}} \in \mathcal{T}$ rearranges the blocks of $\mathbf{x}$ as follows:

$$T_{\mathbf{e}}(\mathbf{x}) = \left(\mathbf{x}_0 || \mathbf{x}_1^{\mathbf{e}[1]} || \mathbf{x}_1^{1-\mathbf{e}[1]} || \dots || \mathbf{x}_\ell^{\mathbf{e}[\ell]} || \mathbf{x}_\ell^{1-\mathbf{e}[\ell]} \right).$$

In particular, for $\mathbf{p}, \mathbf{e} \in \{0, 1\}^\ell$, $\pi \in \mathcal{S}$, and $\mathbf{x} \in \mathbb{Z}^{(2\ell+1) \cdot 3m}$, one has:

$$\mathbf{x} \in \text{SecretExt}(\mathbf{p}) \Leftrightarrow \pi(\mathbf{x}) \in \text{SecretExt}(\mathbf{p}) \Leftrightarrow T_{\mathbf{e}} \circ \pi(\mathbf{x}) \in \text{SecretExt}(\mathbf{p} \oplus \mathbf{e}). \quad (5.3)$$

Therefore, if $\mathbf{v}_i = T_{\mathbf{e}} \circ \pi(\mathbf{z}_i)$ and $\mathbf{p}_1 = \mathbf{p} \oplus \mathbf{e}$, where $\mathbf{e}$ is chosen uniformly at random, are sent to the verifier, then, by checking that $\mathbf{v}_i \in \text{SecretExt}(\mathbf{p}_1)$, the verifier can be convinced that $\mathbf{z}_i \in \text{SecretExt}(\mathbf{p})$, but it cannot learn any additional information about $(\mathbf{z}_i, \mathbf{p})$. On the other hand, by using the “random masking” technique for Stern-type protocols [104], the prover can also convince in zero-knowledge that $\mathbf{A}^* \cdot (\sum_{i=1}^k \beta_i \mathbf{z}_i) = \mathbf{u} \bmod q$ by instead showing that:

$$\mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i (\mathbf{z}_i + \mathbf{r}_{\mathbf{z}}^{(i)}) \right) - \mathbf{u} = \mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i \mathbf{r}_{\mathbf{z}}^{(i)} \right) \bmod q,$$

where for each $i \in [k]$, $\mathbf{r}_{\mathbf{z}}^{(i)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1) \cdot 3m}$ is a “mask” that hides $\mathbf{z}_i$.

Our Development.

As discussed above, Langlois et al.’s construction yields a Stern-type ZKAoK of a pair $(\mathbf{p}, \mathbf{z})$ satisfying Fact (1) in Definition 5.3.2. We now develop the protocol so that $\mathcal{P}$ can additionally prove Fact (2), i.e., the knowledge of a vector $\mathbf{w} \in \{0, 1\}^d$ such that the pair $(\mathbf{p}, \mathbf{w})$ satisfies $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{G}_2 \cdot \mathbf{w} = \mathbf{M} \bmod 2$.

First, we observe that, the possession of $\mathbf{w} \in \{0, 1\}^d$ can be proved in zero-knowledge by adapting the Decomposition-Extension technique as follows. The

prover extends $\mathbf{w}$ to a vector $\mathbf{w}^* \in \{0, 1\}^{2d}$ such that $wt(\mathbf{w}^*) = d$ and sends $\mathbf{v}_{\mathbf{w}} = \eta(\mathbf{w}^*)$ to the verifier, where η is a uniformly random permutation of $2d$ elements. The verifier, seeing that $\mathbf{v}_{\mathbf{w}} \in \{0, 1\}^{2d}$ and $wt(\mathbf{v}_{\mathbf{w}}) = d$, will be convinced that the original vector $\mathbf{w}$ belongs to $\{0, 1\}^d$, while learning no additional information about $\mathbf{w}$.

The challenging question is how to prove the possession of $\mathbf{p} \in \{0, 1\}^\ell$ that simultaneously satisfies Fact (1) and Fact (2). To address this question, we map vector $\mathbf{p} = \mathbf{p}[1] \dots \mathbf{p}[\ell]$ to vector $\mathbf{p}^* = \text{PolicyExt}(\mathbf{p})$, where the latter is defined as:

$$\text{PolicyExt}(\mathbf{p}) = (t_0, t_1^0, t_1^1, \dots, t_\ell^0, t_\ell^1) \in \{0, 1\}^{2\ell+1},$$

where $t_0 = 1$, and $\forall i \in [\ell] : t_i^{\mathbf{p}[i]} = 1, t_i^{1-\mathbf{p}[i]} = 0$. It can be checked that the map $\mathbf{p} \mapsto \text{PolicyExt}(\mathbf{p})$ establishes a bijection between the set $\{0, 1\}^\ell$ and the set $\{\text{PolicyExt}(\mathbf{p}) : \mathbf{p} \in \{0, 1\}^\ell\}$. In other words, to prove the possession of $\mathbf{p} \in \{0, 1\}^\ell$ in zero-knowledge, one can instead prove the possession of $\mathbf{p}^* \in \{0, 1\}^{2\ell+1}$ such that $\mathbf{p}^* = \text{PolicyExt}(\mathbf{p})$, for *certain* $\mathbf{p}$. Towards this goal, similar to Langlois et al.'s approach, we randomly permute the vector $\mathbf{p}^*$.

Specifically, for any vector $\mathbf{t} = (t_0, t_1^0, t_1^1, \dots, t_\ell^0, t_\ell^1) \in \{0, 1\}^{2\ell+1}$, we define the set $\mathcal{T}' = \{T'_{\mathbf{e}} : \mathbf{e} \in \{0, 1\}^\ell\}$ of special permutations of $\mathbf{t}$, where for $\mathbf{e} = \mathbf{e}[1] \dots \mathbf{e}[\ell]$, the permutation $T'_{\mathbf{e}} \in \mathcal{T}'$ rearranges the coordinates of $\mathbf{t}$ as follows: $T'_{\mathbf{e}}(\mathbf{t}) = (t_0, t_1^{\mathbf{e}[1]}, t_1^{1-\mathbf{e}[1]}, \dots, t_\ell^{\mathbf{e}[\ell]}, t_\ell^{1-\mathbf{e}[\ell]}).$

We observe that for $\mathbf{p}, \mathbf{e} \in \{0, 1\}^\ell$, and for $\mathbf{t} \in \{0, 1\}^{2\ell+1}$, we have:

$$\mathbf{t} = \text{PolicyExt}(\mathbf{p}) \iff T'_{\mathbf{e}}(\mathbf{t}) = \text{PolicyExt}(\mathbf{p} \oplus \mathbf{e}). \quad (5.4)$$

This observation provides the following method to prove that $\mathbf{p}^* = \text{PolicyExt}(\mathbf{p})$

while keeping $\mathbf{p}$ secret. Namely, the prover $\mathcal{P}$ samples a uniformly random $\mathbf{e} \in \{0, 1\}^\ell$, then sends $\mathbf{p}_1 = \mathbf{p} \oplus \mathbf{e}$ together with $\mathbf{v}_\mathbf{p} = T'_\mathbf{e}(\mathbf{p}^*)$ to $\mathcal{V}$. The verifier, seeing that $\mathbf{v}_\mathbf{p} = \text{PolicyExt}(\mathbf{p}_1)$, will be convinced that $\mathbf{p}^* = \text{PolicyExt}(\mathbf{p})$, but will not be able to learn any additional information, because $\mathbf{p} \oplus \mathbf{e}$ essentially acts as a “one-time pad”. Most importantly, if the *same* randomness $\mathbf{e}$ is used in Eqs. (5.3) and (5.4), then the prover can prove the possession of $\mathbf{p}$ satisfying *both* Fact (1) and Fact (2) in Definition 5.3.2.

Finally, to prove the equation $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{G}_2 \cdot \mathbf{w} = \mathbf{M} \bmod 2$, we perform the following matrix extensions:

- Extend the matrix $\mathbf{G}_1 \in \mathbb{Z}_2^{n \times \ell}$ to the matrix

$$\mathbf{G}_1^* = [\mathbf{0}^{n \times 1} | \mathbf{0}^{n \times 1} | \mathbf{g}_1 | \mathbf{0}^{n \times 1} | \mathbf{g}_2 | \dots | \mathbf{0}^{n \times 1} | \mathbf{g}_\ell] \in \mathbb{Z}_2^{n \times (2\ell+1)},$$

where for all $i \in [\ell]$, $\mathbf{g}_i \in \mathbb{Z}_2^n$ is the i -th column of $\mathbf{G}_1$. Then, for $\mathbf{p}^* = \text{PolicyExt}(\mathbf{p})$, we have $\mathbf{G}_1^* \cdot \mathbf{p}^* = \mathbf{G}_1 \cdot \mathbf{p} \bmod 2$.

- Append d zero-columns to the matrix $\mathbf{G}_2 \in \mathbb{Z}_2^{n \times d}$ to obtain the matrix $\mathbf{G}_2^* \in \mathbb{Z}_2^{n \times 2d}$. Then, it is true that $\mathbf{G}_2^* \cdot \mathbf{w}^* = \mathbf{G}_2 \cdot \mathbf{w} \bmod 2$, where $\mathbf{w}^* \in \{0, 1\}^{2d}$ is an extension of $\mathbf{w}$ such that $wt(\mathbf{w}^*) = d$, as described earlier.

Given the above extensions, the equation to be proved becomes $\mathbf{G}_1^* \cdot \mathbf{p}^* + \mathbf{G}_2^* \cdot \mathbf{w}^* = \mathbf{M} \bmod 2$. In the framework of Stern-type protocols, the validity of this equation can be shown in zero-knowledge by using uniformly random “maskings” for $\mathbf{p}^*$ and $\mathbf{w}^*$.

The Interactive Protocol.

We are now ready to describe our zero-knowledge argument of knowledge for the relation R_{PBS} . Let COM be the statistically hiding and computationally binding string commitment scheme from [62]. (For simplicity, in the protocol description,

we will omit the randomness of **COM**.) Let $k = \lfloor \log \beta \rfloor + 1$ and let the sequence $\beta_1, \dots, \beta_k$ be defined as in Eq. (5.2). The inputs of two parties $\mathcal{P}$ and $\mathcal{V}$ are as follows:

- The common input is $(\mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M})$. Both parties form the extended matrices $\mathbf{A}^*, \mathbf{G}_1^*, \mathbf{G}_2^*$, as previously described.
- The prover's witness is $(\mathbf{p}, \mathbf{z}, \mathbf{w})$. Using the techniques above, the prover maps $\mathbf{p}$ to $\mathbf{p}^* = \text{PolicyExt}(\mathbf{p})$, and forms vectors $\mathbf{z}_1, \dots, \mathbf{z}_k \in \text{SecretExt}(\mathbf{p})$, and $\mathbf{w}^* \in \{0, 1\}^{2d}$ such that $wt(\mathbf{w}^*) = d$. The vectors obtained satisfy:

$$\begin{cases} \mathbf{A}^* \cdot (\sum_{i=1}^k \beta_i \mathbf{z}_i) = \mathbf{u} \bmod q \\ \mathbf{G}_1^* \cdot \mathbf{p}^* + \mathbf{G}_2^* \cdot \mathbf{w}^* = \mathbf{M} \bmod 2. \end{cases}$$

The interaction between $\mathcal{P}$ and $\mathcal{V}$ is described in Figures 5.1 and 5.2.

The following lemmas state that our protocol is a statistical zero-knowledge argument of knowledge for the relation R_{PBS} .

Lemma 5.3.3. *If **COM** is a statistically hiding string commitment scheme, then the protocol in Figures 5.1 and 5.2 is a statistical zero-knowledge argument system. Specifically, there exists an efficient simulator $\mathcal{Sim}$ interacting with any (possibly cheating) verifier $\tilde{V}$, such that: On the common input $(\mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M})$, the simulator produces with probability $2/3$ a successful simulated transcript, the distribution of which is statistically close to that provided by an honest prover in the real interaction.*

The construction of the transcript simulator follows the standard simulation technique for Stern-type protocols [62, 70, 104]. The proof of Lemma 5.3.3 is given in Appendix B.

1. **Commitment.** Prover $\mathcal{P}$ samples

$$\begin{cases} \mathbf{e} \xleftarrow{\$} \{0, 1\}^\ell; \mathbf{r}_\mathbf{p} \xleftarrow{\$} \mathbb{Z}_2^{2\ell+1}; \mathbf{r}_\mathbf{z}^{(1)}, \dots, \mathbf{r}_\mathbf{z}^{(k)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1) \cdot 3m}; \mathbf{r}_\mathbf{w} \xleftarrow{\$} \mathbb{Z}_2^{2d}; \\ \pi_1, \dots, \pi_k \xleftarrow{\$} \mathcal{S}; \eta \xleftarrow{\$} S_{2d}, \end{cases}$$

and sends the commitment $\text{CMT} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ to the verifier $\mathcal{V}$, where

$$\begin{cases} \mathbf{c}_1 = \text{COM}(\mathbf{e}, \{\pi_i\}_{i=1}^k, \eta, \mathbf{A}^* \cdot (\sum_{i=1}^k \beta_i \mathbf{r}_\mathbf{z}^{(i)}) \bmod q, \mathbf{G}_1^* \cdot \mathbf{r}_\mathbf{p} + \mathbf{G}_2^* \cdot \mathbf{r}_\mathbf{w} \bmod 2) \\ \mathbf{c}_2 = \text{COM}(\{T_\mathbf{e} \circ \pi_i(\mathbf{r}_\mathbf{z}^{(i)})\}_{i=1}^k, \eta(\mathbf{r}_\mathbf{w}), T'_\mathbf{e}(\mathbf{r}_\mathbf{p})) \\ \mathbf{c}_3 = \text{COM}(\{T_\mathbf{e} \circ \pi_i(\mathbf{z}_i + \mathbf{r}_\mathbf{z}^{(i)})\}_{i=1}^k, \eta(\mathbf{w}^* \oplus \mathbf{r}_\mathbf{w}), T'_\mathbf{e}(\mathbf{p}^* \oplus \mathbf{r}_\mathbf{p})) \end{cases}$$

2. **Challenge.** The verifier $\mathcal{V}$ sends a challenge $\text{Ch} \xleftarrow{\$} \{1, 2, 3\}$ to $\mathcal{P}$.

3. **Response.** The prover $\mathcal{P}$ computes the response with respect to the challenge Ch :

- a. If $\text{Ch} = 1$, then reveal $\mathbf{c}_2$ and $\mathbf{c}_3$. Let $\mathbf{p}_1 = \mathbf{p} \oplus \mathbf{e}$, $\mathbf{v}_\mathbf{p} = T'_\mathbf{e}(\mathbf{p}^*)$, $\mathbf{y}_\mathbf{p} = T'_\mathbf{e}(\mathbf{r}_\mathbf{p})$, $\mathbf{v}_\mathbf{w} = \eta(\mathbf{w}^*)$, $\mathbf{y}_\mathbf{w} = \eta(\mathbf{r}_\mathbf{w})$. For each $i \in [k]$, let $\mathbf{v}_\mathbf{z}^{(i)} = T_\mathbf{e} \circ \pi_i(\mathbf{z}_i)$, $\mathbf{y}_\mathbf{z}^{(i)} = T_\mathbf{e} \circ \pi_i(\mathbf{r}_\mathbf{z}^{(i)})$. The prover then sends:

$$\text{RSP} = (\mathbf{p}_1, \mathbf{v}_\mathbf{p}, \mathbf{y}_\mathbf{p}, \{\mathbf{v}_\mathbf{z}^{(i)}\}_{i=1}^k, \{\mathbf{y}_\mathbf{z}^{(i)}\}_{i=1}^k, \mathbf{v}_\mathbf{w}, \mathbf{y}_\mathbf{w}). \quad (5.5)$$

- b. If $\text{Ch} = 2$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_3$. Let $\mathbf{p}_2 = \mathbf{e}$, $\rho = \eta$, $\mathbf{t}_\mathbf{p} = \mathbf{p}^* \oplus \mathbf{r}_\mathbf{p}$, $\mathbf{t}_\mathbf{w} = \mathbf{w}^* \oplus \mathbf{r}_\mathbf{w}$. For each $i \in [k]$, let $\phi_i = \pi_i$, $\mathbf{t}_\mathbf{z}^{(i)} = \mathbf{z}_i + \mathbf{r}_\mathbf{z}^{(i)} \bmod q$. The prover then sends:

$$\text{RSP} = (\mathbf{p}_2, \{\phi_i\}_{i=1}^k, \rho, \mathbf{t}_\mathbf{p}, \{\mathbf{t}_\mathbf{z}^{(i)}\}_{i=1}^k, \mathbf{t}_\mathbf{w}). \quad (5.6)$$

- c. If $\text{Ch} = 3$, then reveal $\mathbf{c}_1$ and $\mathbf{c}_2$. Let $\mathbf{p}_3 = \mathbf{e}$, $\theta = \eta$, $\mathbf{h}_\mathbf{p} = \mathbf{r}_\mathbf{p}$, $\mathbf{h}_\mathbf{w} = \mathbf{r}_\mathbf{w}$. For each $i \in [k]$, let $\psi_i = \pi_i$, $\mathbf{h}_\mathbf{z}^{(i)} = \mathbf{r}_\mathbf{z}^{(i)}$. The prover then sends:

$$\text{RSP} = (\mathbf{p}_3, \{\psi_i\}_{i=1}^k, \theta, \mathbf{h}_\mathbf{p}, \{\mathbf{h}_\mathbf{z}^{(i)}\}_{i=1}^k, \mathbf{h}_\mathbf{w}). \quad (5.7)$$

Figure 5.1: A zero-knowledge argument of knowledge for the relation R_{PBS}

4. **Verification.** Receiving the response RSP, the verifier $\mathcal{V}$ performs the following checks:

a. If Ch = 1: Parse RSP as in Eq. (5.5). Check that:

$$\begin{cases} \mathbf{v}_p = \text{PolicyExt}(\mathbf{p}_1); \mathbf{v}_w \in \{0, 1\}^{2d}; wt(\mathbf{v}_w) = d; \forall i \in [k] : \mathbf{v}_z^{(i)} \in \text{SecretExt}(\mathbf{p}_1); \\ \mathbf{c}_2 = \text{COM}(\{\mathbf{y}_z^{(i)}\}_{i=1}^k, \mathbf{y}_w, \mathbf{y}_p); \\ \mathbf{c}_3 = \text{COM}(\{\mathbf{v}_z^{(i)} + \mathbf{y}_z^{(i)}\}_{i=1}^k, \mathbf{v}_w \oplus \mathbf{y}_w, \mathbf{v}_p \oplus \mathbf{y}_p). \end{cases}$$

b. If Ch = 2: Parse RSP as in Eq. (5.6). Check that:

$$\begin{cases} \mathbf{c}_1 = \text{COM}(\mathbf{p}_2, \{\phi_i\}_{i=1}^k, \rho, \mathbf{A}^* \cdot (\sum_{i=1}^k \beta_i \mathbf{t}_z^{(i)}) - \mathbf{u} \bmod q, \\ \quad \mathbf{G}_1^* \cdot \mathbf{t}_p + \mathbf{G}_2^* \cdot \mathbf{t}_w - \mathbf{M} \bmod 2); \\ \mathbf{c}_3 = \text{COM}(\{T_{\mathbf{p}_2} \circ \phi_i(\mathbf{t}_z^{(i)})\}_{i=1}^k, \rho(\mathbf{t}_w), T'_{\mathbf{p}_2}(\mathbf{t}_p)). \end{cases}$$

c. If Ch = 3: Parse RSP as in Eq. (5.7). Check that:

$$\begin{cases} \mathbf{c}_1 = \text{COM}(\mathbf{p}_3, \{\psi_i\}_{i=1}^k, \theta, \mathbf{A}^* \cdot (\sum_{i=1}^k \beta_i \mathbf{h}_z^{(i)}) \bmod q, \\ \quad \mathbf{G}_1^* \cdot \mathbf{h}_p + \mathbf{G}_2^* \cdot \mathbf{h}_w \bmod 2); \\ \mathbf{c}_2 = \text{COM}(\{T_{\mathbf{p}_3} \circ \psi_i(\mathbf{h}_z^{(i)})\}_{i=1}^k, \theta(\mathbf{h}_w), T'_{\mathbf{p}_3}(\mathbf{h}_p)). \end{cases}$$

The verifier $\mathcal{V}$ outputs 1 (Valid) if and only if all the conditions hold. Otherwise it outputs 0 (Invalid).

Figure 5.2: A zero-knowledge argument of knowledge for the relation R_{PBS} (cont.)

Lemma 5.3.4. *If COM is a computationally binding string commitment scheme, then the protocol in Figures 5.1 and 5.2 is an argument of knowledge for the relation R_{PBS} . In particular, the protocol satisfies the special soundness property. Namely, assume that for a given commitment CMT with respect to the common input $(\mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M})$, there exist 3 valid responses $\text{RSP}^{(1)}$, $\text{RSP}^{(2)}$, and $\text{RSP}^{(3)}$ corresponding to all 3 possible values of the challenge Ch , then, there exists an efficient extractor $\mathcal{K}$ that outputs $(\mathbf{p}', \mathbf{z}', \mathbf{w}')$ such that:*

$$R_{PBS}((\mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}), \mathbf{p}', \mathbf{z}', \mathbf{w}') = 1.$$

The proof of Lemma 5.3.4 is presented in Appendix C.

5.4 A Policy-Based Signature Scheme from Lattices

5.4.1 Description of the Scheme

We first specify the parameters and the underlying policy language of our scheme.

Parameters. Let n be the security parameter. The scheme works with policies of bit-length $\ell = O(\log n)$, and witnesses of bit-length d such that $n - \ell < d$. We choose a sufficiently large modulus $q = \tilde{O}(n^2)$, a lattice dimension $m \geq 2n \log q$, a Gaussian parameter $\sigma = O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log n})$, and an integer bound $\beta = \lceil \sigma \cdot \log n \rceil$. Let $t = \omega(\log n)$ be the number of protocol repetitions and let $\mathcal{H} : \{0, 1\}^* \rightarrow \{1, 2, 3\}^t$ be a collision-resistant hash function, to be modeled as a random oracle.

Policy Language. Given two matrices $\mathbf{G}_1 \in \mathbb{Z}_2^{n \times \ell}$ and $\mathbf{G}_2 \in \mathbb{Z}_2^{n \times d}$, we define the policy checker $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2} : (\{0, 1\}^\ell \times \mathbb{Z}_2^n) \times \{0, 1\}^d \rightarrow \{0, 1\}$, where:

$$\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\mathbf{p}, \mathbf{M}), \mathbf{w}) = 1 \iff \mathbf{G}_1 \cdot \mathbf{p} + \mathbf{G}_2 \cdot \mathbf{w} = \mathbf{M} \bmod 2.$$

The associated policy language $\mathcal{L}(\text{PC}_{\mathbf{G}_1, \mathbf{G}_2})$ is defined as the set of all $(\mathbf{p}, \mathbf{M}) \in \{0, 1\}^\ell \times \mathbb{Z}_2^n$ such that there exists $\mathbf{w} \in \{0, 1\}^d$ satisfying $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{G}_2 \cdot \mathbf{w} = \mathbf{M} \pmod{2}$.

Remark 5.4.1. The structure of our policy language can be found in many lattice-based cryptographic constructions. Here is a small example, in which we employ a similar language to obtain a lattice-based group signature scheme.

We consider the ℓ -bit version of the Dual-Regev encryption scheme proposed by Gentry et al. [53]. Given the public key $(\mathbf{A} \in \mathbb{Z}_q^{n \times m}, \mathbf{B} \in \mathbb{Z}_q^{n \times \ell})$, a ciphertext of the plaintext $\mathbf{p} \in \{0, 1\}^\ell$ is a pair $(\mathbf{c}_1 \in \mathbb{Z}_q^m, \mathbf{c}_2 \in \mathbb{Z}_q^\ell)$ of the form

$$\begin{cases} \mathbf{c}_1 = \mathbf{A}^\top \mathbf{s} + \mathbf{e}_1, \\ \mathbf{c}_2 = \mathbf{B}^\top \mathbf{s} + \mathbf{e}_2 + \lfloor q/2 \rfloor \mathbf{p}, \end{cases} \quad (5.8)$$

where the encryption randomness $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e}_1 \in \mathbb{Z}_q^m$, $\mathbf{e}_2 \in \mathbb{Z}_q^\ell$ are vectors having infinity norm bounded by some small β . (Note that we use the ‘‘Hermite Normal Form’’ of LWE here.)

Now observe that, if we let $\mathbf{M} = (\mathbf{c}_1 || \mathbf{c}_2) \in \mathbb{Z}_q^{m+\ell}$, $\mathbf{G}_1 = \begin{bmatrix} \mathbf{0}^{m \times \ell} \\ \lfloor q/2 \rfloor \mathbf{I}_{\ell \times \ell} \end{bmatrix} \in \mathbb{Z}_q^{(m+\ell) \times \ell}$, $\mathbf{G}_2 = \begin{bmatrix} \mathbf{A}^\top \\ \mathbf{B}^\top \end{bmatrix} \mathbf{I}_{m+\ell} \in \mathbb{Z}_q^{(m+\ell) \times (n+m+\ell)}$, $\mathbf{w} = (\mathbf{s} || \mathbf{e}_1 || \mathbf{e}_2) \in \mathbb{Z}_q^{n+m+\ell}$, then we will have

$$\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{G}_2 \cdot \mathbf{w} = \mathbf{M} \pmod{q}. \quad (5.9)$$

We now construct a group signature scheme based on our PBS scheme (the details will be introduced later in this subsection), as follows. Let $N = 2^\ell$ be the number of group users, where each user is identified by some $\mathbf{p} \in \{0, 1\}^\ell$. The secret signing key of user $\mathbf{p}$ is $\mathbf{sk}_{\mathbf{p}} \leftarrow \text{PBS.Keygen}(\text{msk}, \mathbf{p})$. To sign an arbitrary message μ , the user encrypts $\mathbf{p}$ as Eq. (5.8), and generates a zero-knowledge argument to prove that it has a valid signing key $\mathbf{sk}_{\mathbf{p}}$ and that Eq. (5.9) holds. The protocol is

basically the same as the one in our PBS construction. Here we work with modulus $q = \text{poly}(n)$ instead of 2 and we handle the additional condition $\|\mathbf{w}\|_\infty \leq \beta$ by the Decomposition-Extension technique. The interactive argument is then converted into a signature using the Fiat-Shamir heuristic (the message μ is included in the hashing). To open a signature, the tracing authority simply decrypts $(\mathbf{c}_1 || \mathbf{c}_2)$ to recover $\mathbf{p}$.

In the random oracle model, the anonymity of the group signature scheme obtained is based on the security of the Dual-Regev encryption and the zero-knowledge property of the underlying argument system; while its traceability relies on the hardness of the SIS problem.

In fact, the core idea of our above transformation into group signature is that the relation among the plaintext, the encryption randomness and the ciphertext in the Dual-Regev encryption scheme can be described by Eq. (5.9) which has the same structure as the one defined by our policy language.

PBS Scheme. For given matrices $\mathbf{G}_1 \in \mathbb{Z}_2^{n \times \ell}$ and $\mathbf{G}_2 \in \mathbb{Z}_2^{n \times d}$, our policy-based signature scheme with respect to $\mathcal{L}(\text{PC}_{\mathbf{G}_1, \mathbf{G}_2})$ is described as follows.

Setup(1^n): This algorithm performs the following steps:

1. Generate a verification key $(\mathbf{A}, \mathbf{u})$ and a signing key $\mathbf{S}$ for the Bonsai tree signature scheme as follows:
 - (a) Run $\text{TrapGen}(n, m, q)$ from Lemma 2.4.4 to obtain a matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ together with a short basis $\mathbf{S} \in \mathbb{Z}^{m \times m}$ of $\Lambda_q^\perp(\mathbf{A}_0)$.
 - (b) Sample a vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$ and for each $(i, b) \in [\ell] \times \{0, 1\}$, sample $\mathbf{A}_i^b \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.
 - (c) Define $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \dots | \mathbf{A}_\ell^0 | \mathbf{A}_\ell^1] \in \mathbb{Z}_q^{n \times (2\ell+1)m}$.

2. Output the public parameters $\mathbf{pp} = (\mathbf{A}, \mathbf{u})$ and the master secret key $\mathbf{msk} = \mathbf{S}$.

KeyGen($\mathbf{S}, \mathbf{p}$): On input the master secret key $\mathbf{S}$, and a policy $\mathbf{p} = \mathbf{p}[1]\mathbf{p}[2] \cdots \mathbf{p}[\ell] \in \{0, 1\}^\ell$, this algorithm outputs a signing key $\mathbf{sk}_{\mathbf{p}} = (\mathbf{p}, \mathbf{z})$ for $\mathbf{p}$, where $\mathbf{z}$ is a Bonsai tree signature on the “message” $\mathbf{p}$, computed as follows:

1. Define $\mathbf{A}_{\mathbf{p}} = [\mathbf{A}_0 | \mathbf{A}_1^{\mathbf{p}[1]} | \cdots | \mathbf{A}_\ell^{\mathbf{p}[\ell]}] \in \mathbb{Z}_q^{n \times (\ell+1)m}$.
2. Run **ExtBasis**($\mathbf{A}_0, \mathbf{A}_{\mathbf{p}}, \mathbf{S}$) to obtain a short basis $\mathbf{S}_{\mathbf{p}}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathbf{p}})$ (see Lemma 2.4.6).
3. Run **SampleD**($\mathbf{A}_{\mathbf{p}}, \mathbf{S}_{\mathbf{p}}, \mathbf{u}, \sigma$) to sample a short vector $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$ (see Lemma 2.4.4).

Sign($\mathbf{sk}_{\mathbf{p}}, \mathbf{M}, \mathbf{w}$): On input a signing key $\mathbf{sk}_{\mathbf{p}} = (\mathbf{p}, \mathbf{z})$, a message $\mathbf{M} \in \mathbb{Z}_2^n$ and $\mathbf{w} \in \{0, 1\}^d$, this algorithm proceeds as follows:

1. If $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\mathbf{p}, \mathbf{M}), \mathbf{w}) \neq 1$, then return $\perp$.
2. If $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\mathbf{p}, \mathbf{M}), \mathbf{w}) = 1$, i.e., $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{G}_2 \cdot \mathbf{w} = \mathbf{M} \bmod 2$, then generate a non-interactive zero-knowledge argument of knowledge Π to show the possession of $(\mathbf{p}, \mathbf{z}, \mathbf{w})$ such that:
 - $(\mathbf{p}, \mathbf{z})$ is a valid message-signature pair for the Bonsai tree signature scheme, i.e., $\|\mathbf{z}\|_\infty \leq \beta$ and $\mathbf{A}_{\mathbf{p}} \cdot \mathbf{z} = \mathbf{u} \bmod q$.
 - $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\mathbf{p}, \mathbf{M}), \mathbf{w}) = 1$.

This is done by employing the argument system in Subsection 5.3.2 with common input $(\mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M})$ and the prover’s input $(\mathbf{p}, \mathbf{z}, \mathbf{w})$. The interactive protocol is repeated $t = \omega(\log n)$ times to achieve negligible

soundness error, and then made non-interactive using the Fiat-Shamir heuristic as a triple $\Pi = (\{\text{CMT}^{(j)}\}_{j=1}^t, \text{CH}, \{\text{RSP}^{(j)}\}_{j=1}^t)$, where

$$\text{CH} = (\{\text{Ch}^{(j)}\}_{j=1}^t) = \mathcal{H}(\{\text{CMT}^{(j)}\}_{j=1}^t, \mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}) \in \{1, 2, 3\}^t.$$

3. Output $\Sigma = (\mathbf{M}, \Pi)$.

Verify(pp, Σ): On input the public parameters $\text{pp} = (\mathbf{A}, \mathbf{u})$ and a signature $\Sigma = (\mathbf{M}, \Pi)$, this algorithm proceeds as follows:

1. Parse Π as $(\{\text{CMT}^{(j)}\}_{j=1}^t, \{\text{Ch}^{(j)}\}_{j=1}^t, \{\text{RSP}^{(j)}\}_{j=1}^t)$.
2. For each $j \in [t]$, run the verification of the protocol in Subsection 5.3.2 on common input $(\mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M})$, to check the validity of $\text{RSP}^{(j)}$ with respect to $\text{CMT}^{(j)}$ and $\text{Ch}^{(j)}$. If any of the verification conditions does not hold, then output 0 (*Invalid*) and terminate.
3. Output 1 (*Valid*).

5.4.2 Analysis of the Scheme

We now prove that the policy-based signature scheme given in Subsection 5.4.1 satisfies the three requirements: correctness, simulatability and extractability.

Correctness. We demonstrate that, for the chosen parameters, the scheme is correct with overwhelming probability.

- We set $m \geq 2n \log q$, so that by Lemma 2.4.4, $\text{TrapGen}(n, m, q)$ works properly and generates a basis $\mathbf{S} \in \mathbb{Z}^{m \times m}$ satisfying $\|\tilde{\mathbf{S}}\| \leq O(\sqrt{n \log q})$ with overwhelming probability. Hence, for any $\mathbf{p} \in \{0, 1\}^\ell$, the Gram-Schmidt norm

of the extended basis $\mathbf{S}_{\mathbf{p}} \in \mathbb{Z}^{(\ell+1)m \times (\ell+1)m}$, outputted by $\text{ExtBasis}(\mathbf{A}_0, \mathbf{A}_{\mathbf{p}}, \mathbf{S})$, satisfies the same upper bound (by Lemma 2.4.6).

- We set the Gaussian parameter $\sigma = O(\sqrt{n \log q}) \cdot \omega(\sqrt{\log n})$, so that the algorithm $\text{SampleD}(\mathbf{A}_{\mathbf{p}}, \mathbf{S}_{\mathbf{p}}, \mathbf{u}, \sigma)$ works properly and outputs a vector $\mathbf{z} \in \mathbb{Z}^{(\ell+1)m}$ such that $\mathbf{A}_{\mathbf{p}} \cdot \mathbf{z} = \mathbf{u} \bmod q$, and that $\|\mathbf{z}\|_{\infty} \leq \beta = \lceil \sigma \cdot \log n \rceil$ except for some negligible probability (by Lemma 2.4.4 and Lemma 2.4.3).

As a result, for any $(\mathbf{p}, \mathbf{M}) \in \mathcal{L}(\text{PC}_{\mathbf{G}_1, \mathbf{G}_2})$, with overwhelming probability an honest signer can obtain a triple $(\mathbf{p}, \mathbf{z}, \mathbf{w})$ satisfying the conditions specified in Step 2 of the signing algorithm Sign . Finally, we observe that, the argument Π , generated by an honest signer possessing a valid witness $(\mathbf{p}, \mathbf{z}, \mathbf{w})$, should always be accepted by the verification algorithm, thanks to the perfect completeness of the underlying argument system.

Simulatability. We prove that the scheme satisfies the simulatability requirement based on the statistical zero-knowledge property of the underlying argument system.

Theorem 5.4.2. *The policy-based signature scheme described in Subsection 5.4.1 satisfies the simulatability requirement (as defined in Subsection 5.1.2) in the random oracle model.*

Proof. We first construct a PPT algorithm SimSign , such that, on input the public parameters $\text{pp} = (\mathbf{A}, \mathbf{u})$, and any message $\mathbf{M} \in \mathbb{Z}_2^n$, it outputs a simulated signature Σ^* , whose distribution is statistically close to the one produced by the real signing algorithm $\text{Sign}(\text{sk}_{\mathbf{p}}, \mathbf{M}, \mathbf{w})$, where $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\mathbf{p}, \mathbf{M}), \mathbf{w}) = 1$. Our construction employs the simulator Sim (see Lemma 5.3.3) of the underlying interactive argument system, and relies on the programmability of the random oracle $\mathcal{H}$. In particular, algorithm SimSign works as follows:

1. For each $j \in [t]$, pick a uniformly random “fake challenge” $\overline{\text{Ch}}^{(j)} \in \{1, 2, 3\}$, and use the simulator $\mathcal{Sim}$ to compute the commitment $\text{CMT}^{(j)}$ depending on the value of $\overline{\text{Ch}}^{(j)}$. Then pick a uniformly random “real challenge” $\text{Ch}^{(j)} \in \{1, 2, 3\} \setminus \{\overline{\text{Ch}}^{(j)}\}$.
2. Let $\text{CH} = (\{\text{Ch}^{(j)}\}_{j=1}^t)$. Program the random oracle:

$$\mathcal{H}(\{\text{CMT}^{(j)}\}_{j=1}^t, \mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}) = \text{CH}.$$

3. Use $\mathcal{Sim}$ to prepare the responses $\text{RSP}^{(1)}, \dots, \text{RSP}^{(t)}$ with respect to the challenges $\text{Ch}^{(1)}, \dots, \text{Ch}^{(t)}$.
4. Output $\Sigma^* = (\mathbf{M}, \Pi^*)$, where $\Pi^* = (\{\text{CMT}^{(j)}\}_{j=1}^t, \text{CH}, \{\text{RSP}^{(j)}\}_{j=1}^t)$.

By construction, for each $j \in [t]$, the “real challenge” $\text{Ch}^{(j)}$ is uniformly distributed in $\{1, 2, 3\}$, which meets the requirement on the output of the random oracle $\mathcal{H}$, and $\text{RSP}^{(j)}$ is a valid response to $\text{CMT}^{(j)}$ and $\text{Ch}^{(j)}$ (see the proof of Lemma 5.3.3). Moreover, by the statistical zero-knowledge property of the underlying argument system, the distribution of Π^* is statistically close to that of a real argument generated by an honest prover. As a result, the simulated signature $\Sigma^* = (\mathbf{M}, \Pi^*)$ is a valid one, and is statistically close to a legitimate signature produced by the real signing algorithm.

Given the algorithm **SimSign** constructed above, the simulatability of our scheme in the random oracle model can be proved in a straightforward manner. In the simulatability experiment, the challenger runs **Setup**(1^n) to obtain $\mathbf{pp} = (\mathbf{A}, \mathbf{u})$ and $\mathbf{msk} = \mathbf{S}$, and sends $(\mathbf{pp}, \mathbf{msk})$ to the adversary $\mathcal{A}$. The adversary then can adaptively choose policies $\mathbf{p}_1, \dots, \mathbf{p}_Q$, and query the signing key for each of the $\mathbf{p}_i$ ’s. To answer each of the queries, the challenger generates $\mathbf{sk}_i \leftarrow \text{KeyGen}(\mathbf{S}, \mathbf{p}_i)$, records $\mathbf{sk}_i$, and

then sends it to $\mathcal{A}$. In the challenge phase, $\mathcal{A}$ outputs $i \in [Q]$, a message $\mathbf{M} \in \mathbb{Z}_2^n$ and $\mathbf{w} \in \{0, 1\}^d$. If $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\mathbf{p}_i, \mathbf{M}), \mathbf{w}) \neq 1$, then the challenger returns $\perp$. Otherwise, it picks a uniformly random bit $b \in \{0, 1\}$, computes $\Sigma_0 \leftarrow \text{SimSign}(\mathbf{M})$ and $\Sigma_1 \leftarrow \text{Sign}(\mathbf{sk}_i, \mathbf{M}, \mathbf{w})$, and returns the challenge signature Σ_b to $\mathcal{A}$.

As discussed above, the distributions of the simulated signature Σ_0 and the real signature Σ_1 are statistically close. Therefore, the advantage of any (even computationally unbounded) adversary in distinguishing Σ_0 and Σ_1 is negligible in the security parameter n . This concludes the proof. $\square$

Extractability. We prove that the given policy-based signature scheme satisfies the extractability requirement in the random oracle model, assuming that the $\text{SIS}_{n, m', q, \beta}^\infty$ problem is hard, where $m' = (\ell + 1)m + 1$.

Theorem 5.4.3. *If there exists an adversary $\mathcal{A}$ attacking the extractability of the scheme described in Subsection 5.4.1 with advantage ϵ and running time T , then, there exists an algorithm $\mathcal{B}$ solving the $\text{SIS}_{n, (\ell+1)m+1, q, \beta}^\infty$ problem with success probability $\epsilon' > \frac{1 - (7/9)^t}{2^{\ell+1}}$, and running time $T' = 32 \cdot T \cdot q_{\mathcal{H}} / (\epsilon - 3^{-t}) + \text{poly}(n, 2^\ell)$, where $q_{\mathcal{H}}$ is the number of queries to the random oracle $\mathcal{H} : \{0, 1\}^* \rightarrow \{1, 2, 3\}^t$.*

Proof. First, without loss of generality, we assume that the commitment scheme **COM** used in the underlying argument system is computationally binding, because an adversary breaking its computational binding property can be used to solve the $\text{SIS}_{n, \bar{m}, q, \beta}^\infty$ problem, for $\bar{m} = 3n \log q$ (by Lemma 5.3.1). Such an adversary is capable of solving the $\text{SIS}_{n, (\ell+1)m+1, q, \beta}^\infty$ problem (by appending $(\ell + 1)m + 1 - 3n \log q > 0$ coordinates 0 to its $\text{SIS}_{n, \bar{m}, q, \beta}^\infty$ solution).

Now, using the adversary $\mathcal{A}$ that attacks the extractability of the PBS scheme with respect to $\mathcal{L}(\text{PC}_{\mathbf{G}_1, \mathbf{G}_2})$, we construct a $\text{SIS}_{n, (\ell+1)m+1, q, \beta}^\infty$ solver $\mathcal{B}$. At the beginning, $\mathcal{B}$ is given a uniformly random challenge matrix $\widehat{\mathbf{C}} \in \mathbb{Z}_q^{n \times [(\ell+1)m+1]}$, and

its goal is to find a non-zero vector $\mathbf{v} \in \mathbb{Z}^{(\ell+1)m+1}$ such that $\|\mathbf{v}\|_\infty \leq \beta$ and $\widehat{\mathbf{C}} \cdot \mathbf{v} = \mathbf{0} \bmod q$. The algorithm $\mathcal{B}$ parses the matrix $\widehat{\mathbf{C}}$ as $[\mathbf{C}|\mathbf{u}]$, where the matrix $\mathbf{C} = [\mathbf{C}_0|\mathbf{C}_1|\dots|\mathbf{C}_\ell] \in \mathbb{Z}_q^{n \times (\ell+1)m}$ and the vector $\mathbf{u} \in \mathbb{Z}_q^n$. Then $\mathcal{B}$ proceeds as follows.

Initialization. $\mathcal{B}$ simulates the **Setup** algorithm as follows:

1. Pick a uniformly random target policy $\mathbf{p}^* = \mathbf{p}^*[1] \dots \mathbf{p}^*[\ell] \xleftarrow{\$} \{0, 1\}^\ell$.
2. Set $\mathbf{A}_0 = \mathbf{C}_0 \in \mathbb{Z}_q^{n \times m}$, and for each $i \in [\ell]$, set $\mathbf{A}_i^{\mathbf{p}^*[i]} = \mathbf{C}_i \in \mathbb{Z}_q^{n \times m}$.
3. Run $\text{TrapGen}(n, m, q)$ (see Lemma 2.4.4) repeatedly to obtain ℓ pairs $(\mathbf{F}_i, \mathbf{T}_i)$ for $i \in [\ell]$, where $\mathbf{F}_i \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{T}_i \in \mathbb{Z}^{m \times m}$ is a short basis of $\Lambda_q^\perp(\mathbf{F}_i)$. Then for each $i \in [\ell]$, set $\mathbf{A}_i^{1-\mathbf{p}^*[i]} = \mathbf{F}_i$.
4. Define $\mathbf{A} = [\mathbf{A}_0|\mathbf{A}_1^0|\mathbf{A}_1^1|\dots|\mathbf{A}_\ell^0|\mathbf{A}_\ell^1]$. (By Lemma 2.4.4, the distribution of $\mathbf{A}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n \times (2\ell+1)m}$.)
For each $\mathbf{p} \in \{0, 1\}^\ell$, the sub-matrix $\mathbf{A}_{\mathbf{p}} \in \mathbb{Z}_q^{n \times (\ell+1)m}$ of $\mathbf{A}$ is defined as usual, i.e., $\mathbf{A}_{\mathbf{p}} = [\mathbf{A}_0|\mathbf{A}_1^{\mathbf{p}[1]}|\dots|\mathbf{A}_\ell^{\mathbf{p}[\ell]}]$. By construction, $\mathbf{A}_{\mathbf{p}^*} = \mathbf{C}$.
5. Send $\text{pp}^* = (\mathbf{A}, \mathbf{u})$ to the adversary $\mathcal{A}$.

Note that the distribution of pp^* is statistically close to that of the real pp produced by the legitimate **Setup** algorithm. The adversary $\mathcal{A}$ then can query signing keys and simulated signatures, to which the algorithm $\mathcal{B}$ responds as follows:

Key Queries. Suppose that $\mathcal{A}$ queries a signing key for a certain policy $\mathbf{p} \in \{0, 1\}^\ell$. If $\mathbf{p} = \mathbf{p}^*$, for which $\mathcal{B}$ does not have a signing key, then $\mathcal{B}$ aborts. Otherwise, it performs the following steps:

1. Let $i_0 = \min\{i \in [\ell] : \mathbf{p}[i] \neq \mathbf{p}^*[i]\}$. (Since $\mathbf{p} \neq \mathbf{p}^*$, such an index i_0 must exist.) Note that one has:

$$\mathbf{A}_{i_0}^{\mathbf{p}[i_0]} = \mathbf{F}_{i_0} = \mathbf{A}_{i_0}^{1-\mathbf{p}^*[i_0]}.$$

2. Sample ℓ vectors $\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{i_0-1}, \mathbf{y}_{i_0+1}, \dots, \mathbf{y}_\ell \xleftarrow{\$} \mathcal{D}_{\mathbb{Z}^m, \sigma}$, and compute

$$\mathbf{t} = \mathbf{A}_0 \cdot \mathbf{y}_0 + \sum_{i \neq i_0} \mathbf{A}_i^{\mathbf{p}[i]} \cdot \mathbf{y}_i \in \mathbb{Z}_q^n.$$

3. Run $\text{SampleD}(\mathbf{F}_{i_0}, \mathbf{T}_{i_0}, \mathbf{u} - \mathbf{t}, \sigma)$ to sample a vector $\mathbf{y}_{i_0} \in \mathbb{Z}^m$ (see Lemma 2.4.4).
4. Set $\mathbf{z}_{\mathbf{p}} = (\mathbf{y}_0 \| \mathbf{y}_1 \| \dots \| \mathbf{y}_\ell) \in \mathbb{Z}^{(\ell+1)m}$.
5. Return $\mathbf{sk}_{\mathbf{p}} = (\mathbf{p}, \mathbf{z}_{\mathbf{p}})$ to the adversary $\mathcal{A}$.

Note that, by construction, $\mathbf{A}_{\mathbf{p}} \cdot \mathbf{z}_{\mathbf{p}} = \mathbf{u} \bmod q$, and $\|\mathbf{z}_{\mathbf{p}}\|_\infty \leq \beta$ with overwhelming probability; moreover, the distribution of $\mathbf{sk}_{\mathbf{p}}$ is statistically close to that of the real signing key (by Lemma 2.4.3 and Lemma 2.4.4).

Signature Queries. To answer a query for simulated signature on an arbitrary message $\mathbf{M} \in \mathbb{Z}_2^n$, the algorithm $\mathcal{B}$ generates $\Sigma_{\mathbf{M}} \leftarrow \text{SimSign}(\mathbf{M})$ (see the proof of Theorem 5.4.2 for the construction of algorithm SimSign), and returns $\Sigma_{\mathbf{M}}$ to $\mathcal{A}$.

We remark that, at this point, if the algorithm $\mathcal{B}$ has not aborted, then the choice of the target policy $\mathbf{p}^*$ is completely hidden from the adversary's view.

Forgery and Extraction. Eventually, $\mathcal{A}$ outputs a message $\mathbf{M}^*$ and a signature

$$\Sigma^* = (\mathbf{M}^*, \{\text{CMT}^{(j)}\}_{j=1}^t, \{\text{Ch}^{(j)}\}_{j=1}^t, \{\text{RSP}^{(j)}\}_{j=1}^t),$$

such that $\text{Verify}(\text{pp}, \Sigma^*, \mathbf{M}^*) = 1$ and the pair $(\mathbf{M}^*, \Sigma^*)$ is not a query/response pair

in the above query phases.

We now construct an algorithm **Extract** that, on input $(\mathbf{M}^*, \Sigma^*)$, outputs a policy-witness pair $(\bar{\mathbf{p}}, \bar{\mathbf{w}})$ together with a vector $\bar{\mathbf{z}}$ such that:

$$R_{\text{PBS}}((\mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}^*), \bar{\mathbf{p}}, \bar{\mathbf{z}}, \bar{\mathbf{w}}) = 1, \quad (5.10)$$

where the relation R_{PBS} is formalized in Definition 5.3.2. First, observe that $\mathcal{A}$ must have queried the random oracle $\mathcal{H}$ on input $(\{\text{CMT}^{(j)}\}_{j=1}^t, \mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}^*)$, since otherwise, the probability that

$$(\text{Ch}^{(1)}, \dots, \text{Ch}^{(t)}) = \mathcal{H}(\{\text{CMT}^{(j)}\}_{j=1}^t, \mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}^*)$$

is at most 3^{-t} . Therefore, with probability at least $\epsilon - 3^{-t}$, there exists certain κ such that $\kappa \leq q_{\mathcal{H}}$ and the κ -th oracle query involves as input the tuple $(\{\text{CMT}^{(j)}\}_{j=1}^t, \mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}^*)$.

Now the algorithm **Extract** 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 $r_1, \dots, r_{\kappa-1}$ as in the original run, while from the κ -th query onwards, the algorithm **Extract** replies with fresh random values $r'_\kappa, \dots, r'_{q_{\mathcal{H}}} \xleftarrow{\$} \{1, 2, 3\}^t$. According to the Improved Forking Lemma by Pointcheval and Vaudenay ([93], see also Appendix D), with probability larger than $1/2$, the algorithm **Extract** can obtain a 3-fork involving the tuple $(\{\text{CMT}^{(j)}\}_{j=1}^t, \mathbf{A}, \mathbf{u}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{M}^*)$ after at most $32 \cdot T \cdot q_{\mathcal{H}} / (\epsilon - 3^{-t})$ executions of the algorithm $\mathcal{A}$. Let

$$r_{\kappa,1} = (\text{Ch}_1^{(1)}, \dots, \text{Ch}_1^{(t)}); r_{\kappa,2} = (\text{Ch}_2^{(1)}, \dots, \text{Ch}_2^{(t)}); r_{\kappa,3} = (\text{Ch}_3^{(1)}, \dots, \text{Ch}_3^{(t)})$$

be the replies to the random oracle queries by **Extract** with respect to the 3-fork branches. One can observe that

$$\Pr \left[\exists j \in [t] : \{ \text{Ch}_1^{(j)}, \text{Ch}_2^{(j)}, \text{Ch}_3^{(j)} \} = \{1, 2, 3\} \right] = 1 - (7/9)^t.$$

Conditioned on the existence of such an index j , one can parse the 3 forgeries corresponding to the fork branches to obtain $(\text{RSP}_1^{(j)}, \text{RSP}_2^{(j)}, \text{RSP}_3^{(j)})$. They are indeed 3 valid responses with respect to all 3 possible challenges for the same commitment $\text{CMT}^{(j)}$. Since **COM** is assumed to be computationally binding, the algorithm **Extract** can run the knowledge extractor $\mathcal{K}$ in the proof of Lemma 5.3.4 to extract $(\bar{\mathbf{p}}, \bar{\mathbf{z}}, \bar{\mathbf{w}})$ such that Eq. (5.10) holds.

At this point, the **SIS** solver $\mathcal{B}$ exploits the output of the algorithm **Extract** as follows. Observe that the target policy $\mathbf{p}^*$ was chosen uniformly at random from $\{0, 1\}^\ell$, and $\mathcal{A}$'s behavior is independent of $\mathbf{p}^*$. Hence the event that

$$(\mathcal{B} \text{ did not abort}) \wedge (\bar{\mathbf{p}} = \mathbf{p}^*)$$

occurs with probability $1/2^\ell$. Conditioned on this event, then the vector $\bar{\mathbf{z}} \in \mathbb{Z}^{(\ell+1)m}$ produced by **Extract** satisfies $\|\bar{\mathbf{z}}\|_\infty \leq \beta$ and $\mathbf{A}_{\mathbf{p}^*} \cdot \bar{\mathbf{z}} = \mathbf{C} \cdot \bar{\mathbf{z}} = \mathbf{u} \bmod q$.

Now let $\mathbf{v} \in \mathbb{Z}^{(\ell+1)m+1}$ be the vector obtained by appending a coordinate -1 to the vector $\bar{\mathbf{z}}$, then one has

$$\mathbf{v} \neq \mathbf{0} \text{ and } \|\mathbf{v}\|_\infty \leq \beta \text{ and } \hat{\mathbf{C}} \cdot \mathbf{v} = \mathbf{C} \cdot \bar{\mathbf{z}} - \mathbf{u} = \mathbf{0} \bmod q.$$

The algorithm $\mathcal{B}$ finally outputs the vector $\mathbf{v}$, which is a valid solution to the given $\text{SIS}_{n, (\ell+1)m+1, q, \beta}^\infty$ challenge. It follows from the above discussions that $\mathcal{B}$ succeeds with probability $\epsilon' > \frac{1 - (7/9)^t}{2^{\ell+1}}$, and running time $T' = 32 \cdot T \cdot q_{\mathcal{H}} / (\epsilon - 3^{-t}) + \text{poly}(n, 2^\ell)$.

(Recall that we have set $\ell = O(\log n)$ so that $2^\ell \in \text{poly}(n)$.) Thus, if the adversary $\mathcal{A}$ runs in polynomial time and succeeds with non-negligible probability, then so does $\mathcal{B}$. This concludes the proof. $\square$

5.5 An Extended Scheme Supporting Delegation

In practice, policies may be set up hierarchically with respect to the organizational structure. For example, a president may issue high-level policies to the vice presidents and produce signing keys to them respectively. Each of the vice presidents extends the policy with sub-policies which are set for the managers working below them, and then issues the signing keys to them accordingly. In [14], Bellare and Fuchsbaauer defined delegatable policy-based signatures (DPBS) to address these real-life scenarios, and extended their pairing-based PBS construction to a DPBS scheme. In this section, we show that our lattice-based PBS construction can also be extended to support delegation.

5.5.1 Delegatable Policy-Based Signatures

In a DPBS scheme, a user possessing a signing key for some policy can derive a new key to a delegated user, who needs to obey a sub-policy for signing.

Definitions

We now recall the definition of DPBS from [14]. In a DPBS scheme, each signing key is associated to a vector of policies, and the key can only sign the messages which are allowed under all policies in the vector.

Policy Vector. Each coordinate of a policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$ is a policy in $\{0, 1\}^*$.

Policy Checker. A DPBS policy checker is an extension of the PBS policy checker.

Specifically, it is defined as an **NP**-relation $\text{PC} : \{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}$, where the first input is a pair $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M})$ consisting of a policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$ and a message $\mathbf{M}$, while the second input is a witness vector $(\mathbf{w}_1, \dots, \mathbf{w}_r)$. It satisfies:

$$\text{PC}(((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}), (\mathbf{w}_1, \dots, \mathbf{w}_r)) = 1 \iff \forall i \in [r] : \text{PC}((\mathbf{p}_i, \mathbf{M}), \mathbf{w}_i) = 1.$$

Policy Language. The policy language $\mathcal{L}(\text{PC})$ associated to PC is defined as the set consisting of all the pairs $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M})$ such that there exists a witness vector $(\mathbf{w}_1, \dots, \mathbf{w}_r) \in \{0, 1\}^*$ satisfying

$$\text{PC}(((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}), (\mathbf{w}_1, \dots, \mathbf{w}_r)) = 1.$$

If $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}) \in \mathcal{L}(\text{PC})$ then we say that the message $\mathbf{M}$ satisfies the policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$.

DPBS Scheme. A policy-based signature scheme is a tuple of five polynomial-time algorithms $\mathcal{DPBS} = (\text{Setup}, \text{Delegate}, \text{ChkKey}, \text{Sign}, \text{Verify})$:

- (1) **Setup:** On input 1^n , this algorithm outputs the public parameters pp and a master secret key msk .
- (2) **Delegate:** This algorithm takes as input pp , a signing key $\text{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$ for policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$, and a policy $\mathbf{p}_{r+1}$, then outputs a signing key $\text{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r) \parallel \mathbf{p}_{r+1}}$ which allows one to sign the messages satisfying all policies $\mathbf{p}_1, \dots, \mathbf{p}_r$ and $\mathbf{p}_{r+1}$. When the authority issues a signing key, **Delegate** takes as input pp , msk and $\mathbf{p}$, then outputs a signing key $\text{sk}_{\mathbf{p}}$ for $\mathbf{p}$.

- (3) **ChkKey**: On input pp , a signing key sk , and a policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$, this algorithm outputs 1 if sk is a valid signing key for the policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$; or outputs 0 otherwise.
- (4) **Sign**: On input $\text{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$, $\mathbf{M}$ and $(\mathbf{w}_1, \dots, \mathbf{w}_r)$, this algorithm outputs a signature Σ if $\text{PC}((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}), (\mathbf{w}_1, \dots, \mathbf{w}_r)) = 1$, or $\perp$ otherwise.
- (5) **Verify**: On input pp , a message $\mathbf{M}$, and a signature Σ , this algorithm returns a bit from $\{0, 1\}$.

The delegatable policy-based signature scheme $\mathcal{DPBS}$ satisfies correctness with respect to a policy checker PC if the following holds: for all $n \in \mathbb{N}$, all PC -valid tuples $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}, (\mathbf{w}_1, \dots, \mathbf{w}_r))$, all $(\text{pp}, \text{msk}) \in [\text{Setup}(1^n)]$, all

$$\text{sk} \in [\text{Delegate}(\text{pp}, \text{Delegate}(\text{pp}, \dots, \text{Delegate}(\text{pp}, \text{msk}, \mathbf{p}_1), \dots, \mathbf{p}_{r-1}), \mathbf{p}_r)],$$

and all signatures $\Sigma \in [\text{Sign}(\text{sk}, \mathbf{M}, (\mathbf{w}_1, \dots, \mathbf{w}_r))]$, we have

$$\text{ChkKey}(\text{pp}, \text{sk}, (\mathbf{p}_1, \dots, \mathbf{p}_r)) = 1 \quad \text{and} \quad \text{Verify}(\text{pp}, \mathbf{M}, \Sigma) = 1.$$

Security Requirements

Simulatability and extractability for $\mathcal{DPBS}$ are defined similarly to those for $\mathcal{PBS}$, but the power of the adversary is strengthened. We recall the details from [14].

Simulatability. In comparison with the simulatability experiment for $\mathcal{PBS}$, the adversary $\mathcal{A}$ in the simulatability experiment for $\mathcal{DPBS}$ is allowed to generate a signing key sk from msk and to provide it to the challenger in the query phase. If the key can pass **ChkKey**, then $\mathcal{A}$ will receive either a signature signed under its key or a simulated signature, on which the simulatability requires that $\mathcal{A}$ can hardly tell

the difference. More formally, we consider the experiment $\mathbf{Exp}_{\mathcal{DPBS}, \mathcal{A}}^{\text{SIM}}(n)$ defined as follows:

Initialization: The challenger runs $(\text{pp}, \text{msk}) \leftarrow \text{Setup}(1^n)$ and returns (pp, msk) to the adversary $\mathcal{A}$.

Queries and Challenge: $\mathcal{A}$ provides a signing key sk , a challenge message $\mathbf{M}$, a policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$ and a vector $(\mathbf{w}_1, \dots, \mathbf{w}_r)$. The challenger first runs $\text{ChkKey}(\text{pp}, \text{sk}, (\mathbf{p}_1, \dots, \mathbf{p}_r))$. It returns $\perp$ if the output of ChkKey is 0, otherwise it checks whether $\text{PC}(((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}), (\mathbf{w}_1, \dots, \mathbf{w}_r)) = 1$. If the policy checker PC is not passed, the challenger returns $\perp$. Otherwise, the challenger picks a uniformly random bit $b \xleftarrow{\$} \{0, 1\}$, sets $\Sigma_0 \leftarrow \text{SimSign}(\mathbf{M})$ and $\Sigma_1 \leftarrow \text{Sign}(\text{sk}, \mathbf{M}, (\mathbf{w}_1, \dots, \mathbf{w}_r))$, and returns the challenge signature Σ_b to $\mathcal{A}$.

Finalization: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. The experiment returns **true** if $b' = b$.

We define the advantage of $\mathcal{A}$ in the above experiment as

$$\mathbf{Adv}_{\mathcal{DPBS}, \mathcal{A}}^{\text{SIM}}(n) = |\Pr [\mathbf{Exp}_{\mathcal{DPBS}, \mathcal{A}}^{\text{SIM}}(n) \Rightarrow \text{true}] - 1/2|.$$

The scheme $\mathcal{DPBS}$ is said to satisfy simulatability if $\mathbf{Adv}_{\mathcal{DPBS}, \mathcal{A}}^{\text{SIM}}(n)$ is negligible for any adversary $\mathcal{A}$.

Extractability. In the extractability experiment for $\mathcal{DPBS}$, the adversary $\mathcal{A}$ is allowed to receive not only keys for policy vectors of its choice, but also keys derived from other keys not revealed to $\mathcal{A}$. More formally, we consider the experiment $\mathbf{Adv}_{\mathcal{DPBS}, \mathcal{A}}^{\text{EXT}}(n)$ defined as follows:

Initialization: The challenger generates $(\mathbf{pp}, \mathbf{msk}) \leftarrow \text{Setup}(1^n)$, sets $j \leftarrow 0$, and initializes a matrix recorder F such that $F[0] \leftarrow (\varepsilon, \mathbf{msk})$, together with two set recorders $F_P \leftarrow \emptyset$ and $F_S \leftarrow \emptyset$. It then sends $\mathbf{pp}$ to $\mathcal{A}$.

Queries: $\mathcal{A}$ can adaptively make the following queries:

- Query for key delegation on input $(i, \mathbf{p}')$. The challenger proceeds as follows:
 - If $i \notin \{0, \dots, j\}$ or $F[i][2] = \perp$, then return $\perp$.
 - Otherwise, compute $j \leftarrow j + 1$, $F[j][1] \leftarrow F[i][1] \parallel \mathbf{p}'$, and $F[j][2] \leftarrow \text{Delegate}(\mathbf{pp}, F[i][2], \mathbf{p}')$.
- Query for key revelation on input an index i . The challenger proceeds as follows:
 - If $i \notin [j]$, then return $\perp$.
 - Otherwise, compute $\mathbf{sk} \leftarrow F[i][2]$, $F_P \leftarrow F_P \cup \{F[i][1]\}$ and return $\mathbf{sk}$.
- Query for simulated signature on input message $\mathbf{M}$. The challenger generates $\Sigma \leftarrow \text{SimSign}(\mathbf{M})$, sets $F_S \leftarrow F_S \cup \{(\mathbf{M}, \Sigma)\}$, and returns Σ .

Finalization: $\mathcal{A}$ outputs a pair $(\mathbf{M}^*, \Sigma^*)$. If $\text{Verify}(\mathbf{pp}, \mathbf{M}^*, \Sigma^*) = 0$ or $(\mathbf{M}^*, \Sigma^*) \in F_S$, then return **false**. Otherwise, the challenger runs $\text{Extract}(\mathbf{M}^*, \Sigma^*)$ to obtain $(\mathbb{p}, \mathbb{w})$, where $\mathbb{p}$ is a policy vector and $\mathbb{w}$ is a witness vector. If there does not exist a policy vector in F_P which is a prefix of $\mathbb{p}$, or if $\text{PC}((\mathbb{p}, \mathbf{M}^*), \mathbb{w}) = 0$, then return **true**, while in other cases, return **false**.

We define the advantage of $\mathcal{A}$ in the above experiment as

$$\text{Adv}_{\mathcal{DPBS}, \mathcal{A}}^{\text{EXT}}(n) = \Pr [\text{Exp}_{\mathcal{DPBS}, \mathcal{A}}^{\text{EXT}}(n) \Rightarrow \text{true}] .$$

The scheme $\mathcal{DPBS}$ is said to satisfy extractability if $\mathbf{Adv}_{\mathcal{DPBS}, \mathcal{A}}^{\text{EXT}}(n)$ is negligible for any polynomial-time adversary $\mathcal{A}$.

5.5.2 A Delegatable Policy-Based Signature Scheme from Lattices

Description of the Scheme

We first specify the parameters and the underlying policy language of our scheme.

Parameters. Let n be the security parameter. Each policy has bit-length $\ell = O(\log n)$. Let $D = O(1)$ be the maximal number of policies in the policy vectors. Thus the maximal bit-length of a policy vector is $D\ell$. We choose parameters (m, d, σ, β, q) such that:

- $m \geq 2n \log q$ and $n - \ell < d$.
- $\sigma = O(\sqrt{n \log q}) \sqrt{(\ell + 1)(2\ell + 1) \cdots (d\ell + 1) \cdot m^D} \cdot \omega(\sqrt{\log n})^{D+1}$.
- $\beta = \lceil \sigma \cdot \log n \rceil$.
- $q \geq \beta \cdot \omega(n \log n)$.

Let $t = \omega(\log n)$ be the number of protocol repetitions and let $\mathcal{H} : \{0, 1\}^* \rightarrow \{1, 2, 3\}^t$ be a collision-resistant hash function, to be modeled as a random oracle.

Policy Language. We consider policy vectors of the form $(\mathbf{p}_1, \dots, \mathbf{p}_r)$, for some $r \in [D]$, where for each $i \in [r]$, $\mathbf{p}_i \in \{0, 1\}^\ell$. Similarly, witness vectors are of the form $(\mathbf{w}_1, \dots, \mathbf{w}_r)$, where for each $i \in [r]$, $\mathbf{w}_i \in \{0, 1\}^d$. Given two matrices $\mathbf{G}_1 \in \mathbb{Z}_2^{n \times \ell}$ and $\mathbf{G}_2 \in \mathbb{Z}_2^{n \times d}$, we extend the policy checker $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}$ in our PBS scheme. Let $\mathbf{M} \in \mathbb{Z}_2^n$ be a message. We define:

$$\begin{aligned} & \text{PC}_{\mathbf{G}_1, \mathbf{G}_2}(((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}), (\mathbf{w}_1, \dots, \mathbf{w}_r)) = 1 \\ \iff & \forall i \in [r] : \text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\mathbf{p}_i, \mathbf{M}), \mathbf{w}_i) = 1. \end{aligned}$$

The associated policy language $\mathcal{L}(\text{PC}_{\mathbf{G}_1, \mathbf{G}_2})$ is defined as the set consisting of all tuples $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}) \in \{0, 1\}^{r\ell} \times \mathbb{Z}_2^n$ for all $r \in [D]$, such that for each $i \in [r]$, there exists a vector $\mathbf{w}_i \in \{0, 1\}^d$ satisfying the equation $\mathbf{G}_1 \cdot \mathbf{p}_i + \mathbf{G}_2 \cdot \mathbf{w}_i = \mathbf{M} \bmod 2$.

We now summarize our approach in extending the PBS scheme in Section 5.4 to support delegation. We consider a Bonsai tree defined by a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times (2D\ell+1)m}$, where each policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$ is associated with a subtree $\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} \in \mathbb{Z}_q^{n \times (r\ell+1)m}$. However, a signing key for $(\mathbf{p}_1, \dots, \mathbf{p}_r)$ is no longer defined as a short vector-signature: It is now a short *basis* $\mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$ of the lattice $\Lambda_q^\perp(\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)})$. This modification enables a key delegation mechanism in a similar way as in the hierarchical identity-based encryption scheme from [33]: to derive a key for $(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})$, the holder of $\mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$ can employ algorithms `ExtBasis` and `RandBasis` to produce a randomized delegated basis of $\Lambda_q^\perp(\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r) \parallel \mathbf{p}_{r+1}})$. To sign a message $\mathbf{M}$ satisfying the policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$, the signer first uses $\mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$ to sample a signature-vector $\mathbf{z} \in \mathbb{Z}^{(r\ell+1)m}$ on the “message” $(\mathbf{p}_1, \dots, \mathbf{p}_r)$, using the algorithm `SampleD`. Next, he uses $(\mathbf{p}_1, \dots, \mathbf{p}_r)$, $\mathbf{z}$, and the witness vector $(\mathbf{w}_1, \dots, \mathbf{w}_r)$ to generate a non-interactive zero-knowledge argument of knowledge Π , based on a straightforward extension of the underlying argument system considered in the PBS scheme. The final signature is of the form $(\mathbf{M}, \Pi, r)^1$.

Our delegatable policy-based signature scheme with respect to $\mathcal{L}(\text{PC}_{\mathbf{G}_1, \mathbf{G}_2})$ is described as follows.

Setup(1^n): This algorithm performs the following steps:

1. Generate a verification key $(\mathbf{A}, \mathbf{u})$ and a signing key $\mathbf{S}_0$ for the Bonsai tree signature scheme as follows:

¹ As noted in [14], in PBS and DPBS schemes, the signatures cannot hide the length of the underlying policies. Here, for simplicity, we explicitly include r as a part of the signature.

- (a) Run $\text{TrapGen}(n, m, q)$ from Lemma 2.4.4 to obtain a matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ together with a basis $\mathbf{S}_0 \in \mathbb{Z}^{m \times m}$ of the lattice $\Lambda_q^\perp(\mathbf{A}_0)$.
 - (b) Sample a vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$ and for each $(i, b) \in [D\ell] \times \{0, 1\}$, sample $\mathbf{A}_i^b \xleftarrow{\$} \mathbb{Z}_q^{n \times m}$.
 - (c) Define $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \dots | \mathbf{A}_{D\ell}^0 | \mathbf{A}_{D\ell}^1] \in \mathbb{Z}_q^{n \times (2D\ell+1)m}$.
2. Output the public parameters $\mathbf{pp} = (\mathbf{A}, \mathbf{u})$ and the master secret key $\mathbf{msk} = \mathbf{S}_0$.

$\text{Delegate}(\mathbf{S}_0, \mathbf{p}_1)$: On input $\mathbf{S}_0$, and a policy $\mathbf{p}_1 = \mathbf{p}_1[1]\mathbf{p}_1[2] \dots \mathbf{p}_1[\ell] \in \{0, 1\}^\ell$, this algorithm outputs $\mathbf{sk}_{\mathbf{p}_1} = (\mathbf{p}_1, \mathbf{S}_{\mathbf{p}_1})$, where $\mathbf{S}_{\mathbf{p}_1}$ is computed as follows:

1. Define $\mathbf{A}_{\mathbf{p}_1} = [\mathbf{A}_0 | \mathbf{A}_1^{\mathbf{p}_1[1]} | \dots | \mathbf{A}_\ell^{\mathbf{p}_1[\ell]}] \in \mathbb{Z}_q^{n \times (\ell+1)m}$.
2. Run $\text{ExtBasis}(\mathbf{A}_0, \mathbf{A}_{\mathbf{p}_1}, \mathbf{S}_0)$ to obtain a short basis $\bar{\mathbf{S}}_{\mathbf{p}_1}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathbf{p}_1})$ (see Lemma 2.4.6).
3. Run $\text{RandBasis}(\mathbf{A}_{\mathbf{p}_1}, \bar{\mathbf{S}}_{\mathbf{p}_1}, \sigma)$ to obtain another basis $\mathbf{S}_{\mathbf{p}_1}$ of $\Lambda_q^\perp(\mathbf{A}_{\mathbf{p}_1})$ (see Lemma 2.4.5).

$\text{Delegate}(\mathbf{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, \mathbf{p}_{r+1})$: On input a signing key $\mathbf{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$ for a policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$, and a policy $\mathbf{p}_{r+1} = \mathbf{p}_{r+1}[1]\mathbf{p}_{r+1}[2] \dots \mathbf{p}_{r+1}[\ell] \in \{0, 1\}^\ell$, this algorithm outputs

$$\mathbf{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})} = ((\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1}), \mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})}),$$

where $\mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})}$ is computed as follows:

1. Parse $\mathbf{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$ as $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)})$.

2. Define $\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})}$ as

$$\left[\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} \mid \mathbf{A}_{r\ell+1}^{\mathbf{p}_{r+1}[1]} \mid \dots \mid \mathbf{A}_{(r+1)\ell}^{\mathbf{p}_{r+1}[\ell]} \right] \in \mathbb{Z}_q^{n \times ((r+1)\ell+1)m}.$$

3. Run $\text{ExtBasis}(\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, \mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})}, \mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)})$ to obtain a short basis $\bar{\mathbf{S}}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})}$ of $\Lambda_q^\perp(\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})})$ (see Lemma 2.4.6).

4. Run $\text{RandBasis}(\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})}, \bar{\mathbf{S}}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})}, \sigma)$ to obtain another basis $\mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})}$ of $\Lambda_q^\perp(\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r, \mathbf{p}_{r+1})})$ (see Lemma 2.4.5).

$\text{ChkKey}((\mathbf{A}, \mathbf{u}), \text{sk}, (\mathbf{p}_1, \dots, \mathbf{p}_r))$: On input the public parameters $(\mathbf{A}, \mathbf{u})$, a signing key sk , and a policy vector $(\mathbf{p}_1, \dots, \mathbf{p}_r)$, the algorithm proceeds as follow:

1. Parse sk as $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{T})$.
2. Derive $\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$ from $\mathbf{A}$ and $(\mathbf{p}_1, \dots, \mathbf{p}_r)$.
3. Output 1 if all the following items are correct:
 - $\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} \cdot \mathbf{T} = \mathbf{0} \bmod q$;
 - $\mathbf{T}$ is a full rank matrix in $\mathbb{Z}^{(r\ell+1)m \times (r\ell+1)m}$;
 - $\|\mathbf{T}\| \leq O(\sqrt{n \log q}) \cdot \sqrt{(\ell+1)(2\ell+1) \cdots (r\ell+1) \cdot m^r} \cdot \omega(\sqrt{\log n})^r$.

Otherwise, output 0.

$\text{Sign}(\text{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, \mathbf{M}, (\mathbf{w}_1, \dots, \mathbf{w}_r))$: On input a signing key $\text{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$, a message $\mathbf{M} \in \mathbb{Z}_2^n$ and a vector $(\mathbf{w}_1, \dots, \mathbf{w}_r) \in \{0, 1\}^{rd}$, the algorithm proceeds as follows:

1. Parse $\text{sk}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}$ as $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)})$.
2. If $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}(((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}), (\mathbf{w}_1, \dots, \mathbf{w}_r)) \neq 1$, then return $\perp$.
3. If $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}(((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}), (\mathbf{w}_1, \dots, \mathbf{w}_r)) = 1$, namely, $\mathbf{G}_1 \cdot \mathbf{p}_i + \mathbf{G}_2 \cdot \mathbf{w}_i = \mathbf{M} \bmod 2$ for all $i \in [r]$, then perform the following steps:

- (a) Run $\text{SampleD}(\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, \mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, \mathbf{u}, \sigma)$ to sample a short vector $\mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} \in \mathbb{Z}^{(r\ell+1)m}$ (see Lemma 2.4.4).
- (b) Generate a non-interactive zero-knowledge argument of knowledge Π to show the possession of $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, (\mathbf{w}_1, \dots, \mathbf{w}_r))$ such that
- $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)})$ is a valid message-signature pair for the Bonsai tree signature scheme, i.e.,

$$\|\mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}\|_\infty \leq \beta; \quad \mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} \cdot \mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} = \mathbf{u} \bmod q. \quad (5.11)$$

- $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}(((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}), (\mathbf{w}_1, \dots, \mathbf{w}_r)) = 1$, namely,

$$\forall i \in [r] : \mathbf{G}_1 \cdot \mathbf{p}_i + \mathbf{G}_2 \cdot \mathbf{w}_i = \mathbf{M} \bmod 2. \quad (5.12)$$

This can be done similarly as in the signing algorithm of our PBS scheme. In fact, if we let

$$\begin{aligned} \mathbf{G}_1^{(r)} &= \begin{pmatrix} \mathbf{G}_1 & & \\ & \ddots & \\ & & \mathbf{G}_1 \end{pmatrix} \in \mathbb{Z}_2^{rn \times r\ell}, \quad \mathbf{G}_2^{(r)} = \begin{pmatrix} \mathbf{G}_2 & & \\ & \ddots & \\ & & \mathbf{G}_2 \end{pmatrix} \in \mathbb{Z}_2^{rn \times rd}, \\ \mathbf{w}_r &= \begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_r \end{pmatrix} \in \{0, 1\}^{rd}, \quad \mathbf{M}_r = \begin{pmatrix} \mathbf{M} \\ \vdots \\ \mathbf{M} \end{pmatrix} \in \mathbb{Z}_2^{rn}, \quad \mathbf{P}_r = \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_r \end{pmatrix} \in \{0, 1\}^{r\ell}, \end{aligned}$$

then Eq. (5.12) can be simply written as

$$\mathbf{G}_1^{(r)} \cdot \mathbf{P}_r + \mathbf{G}_2^{(r)} \cdot \mathbf{w}_r = \mathbf{M}_r \bmod 2. \quad (5.13)$$

Hence it is easy to see that the zero-knowledge argument system in Subsection 5.3.2 can be extended in dimensions in a straightforward manner to handle the relation defined by Eqs. (5.11) and (5.13).

4. Output $\Sigma = (\mathbf{M}, \Pi, r)$.

Verify(pp, Σ): On input $\mathbf{pp} = (\mathbf{A}, \mathbf{u})$ and $\Sigma = (\mathbf{M}, \Pi, r)$, this algorithm proceeds similarly as the verification algorithm in PBS. More precisely, it works as follows:

1. Parse $\mathbf{A}$ as $[\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \cdots | \mathbf{A}_{D\ell}^0 | \mathbf{A}_{D\ell}^1] \in \mathbb{Z}_q^{n \times (2D\ell+1)m}$.
2. Apply the verification algorithm of the basic PBS with respect to the language $\mathcal{L}(\text{PC}_{\mathbb{G}_1^{(r)}, \mathbb{G}_2^{(r)}})$ to return

$$\text{Verify}_{\text{PBS}} \left(([\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \cdots | \mathbf{A}_{r\ell}^0 | \mathbf{A}_{r\ell}^1], \mathbf{u}), \mathbf{M}^{(r)}, \Pi \right).$$

Analysis of the Scheme

We now demonstrate that the delegatable policy-based signature scheme described above satisfies the three requirements: correctness, simulatability and extractability.

Correctness. We claim that, for the chosen parameters, the scheme is correct with overwhelming probability.

- We set $m \geq 2n \log q$, so that by Lemma 2.4.4, $\text{TrapGen}(n, m, q)$ works properly and generates a basis $\mathbf{S}_0 \in \mathbb{Z}^{m \times m}$ satisfying $\|\tilde{\mathbf{S}}_0\| \leq O(\sqrt{n \log q})$ with overwhelming probability. Hence, the Gram-Schmidt norm of the delegated basis $\bar{\mathbf{S}}_{\mathbf{p}_1}$, outputted by $\text{ExtBasis}(\mathbf{A}_0, \mathbf{A}_{\mathbf{p}_1}, \mathbf{S}_0)$, satisfies the same upper bound (by Lemma 2.4.6). Thus the norm of the randomized basis $\mathbf{S}_{\mathbf{p}_1}$, outputted by $\text{RandBasis}(\mathbf{A}_{\mathbf{p}_1}, \bar{\mathbf{S}}_{\mathbf{p}_1}, \sigma)$ satisfies $\|\mathbf{S}_{\mathbf{p}_1}\| \leq O(\sqrt{n \log q}) \cdot \sqrt{(\ell+1)m} \cdot \omega(\sqrt{\log n})$

(by Lemma 2.4.5). Since $\|\tilde{\mathbf{S}}_{\mathbf{p}_1}\| \leq \|\mathbf{S}_{\mathbf{p}_1}\|$, we know $\|\tilde{\mathbf{S}}_{\mathbf{p}_1}\| \leq O(\sqrt{n \log q}) \cdot \sqrt{(\ell+1)m} \cdot \omega(\sqrt{\log n})$. Recursively, we have

$$\|\mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}\| \leq O(\sqrt{n \log q}) \cdot \sqrt{(\ell+1)(2\ell+1) \cdots (r\ell+1) \cdot m^r} \cdot \omega(\sqrt{\log n})^r.$$

Hence the key generated by `Delegate` can pass `ChkKey` with overwhelming probability.

- Since for all $r \in [D]$, the Gaussian parameter $\sigma \geq \|\tilde{\mathbf{S}}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}\| \cdot \omega(\log n)$, we have that the algorithm `SampleD`($\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, \mathbf{S}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, \mathbf{u}, \sigma$) works properly and outputs $\mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} \in \mathbb{Z}^{(r\ell+1)m}$ such that $\mathbf{A}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} \cdot \mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)} = \mathbf{u} \bmod q$ and $\|\mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}\|_\infty \leq \beta = \lceil \sigma \cdot \log n \rceil$ except for some negligible probability (by Lemma 2.4.3 and Lemma 2.4.4).

As a result, for any $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{M}) \in \mathcal{L}(\text{PC}_{\mathbf{G}_1, \mathbf{G}_2})$, with overwhelming probability an honest signer can obtain a tuple $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, (\mathbf{w}_1, \dots, \mathbf{w}_r))$ satisfying the conditions specified by Eqs. (5.11) and (5.12). Finally, as in our PBS scheme, the argument Π generated by an honest signer possessing a valid witness $((\mathbf{p}_1, \dots, \mathbf{p}_r), \mathbf{z}_{(\mathbf{p}_1, \dots, \mathbf{p}_r)}, (\mathbf{w}_1, \dots, \mathbf{w}_r))$, should always be accepted by the verification algorithm, thanks to the perfect completeness of the underlying argument system.

Simulatability. We can similarly construct an algorithm `SimSign` based on the simulator *Sim* (see Lemma 5.3.3) of the underlying interactive argument system and the programmability of the random oracle $\mathcal{H}$, as in the proof of Theorem 5.4.2.

In the simulatability experiment, the challenger runs `Setup`(1^n) to obtain $\text{pp} = (\mathbf{A}, \mathbf{u})$ and $\text{msk} = \mathbf{S}_0$, and sends (pp, msk) to the adversary $\mathcal{A}$. The adversary then can adaptively choose a policy vector $\mathbf{p} = (\mathbf{p}_1, \dots, \mathbf{p}_r)$ and produce a signing key sk on its own. In the challenge phase, $\mathcal{A}$ outputs sk , a message $\mathbf{M}$, the policy vector $\mathbf{p}$

and a vector $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_r)$. If $\text{ChkKey}(\text{pp}, \text{sk}, \mathbb{P}) = 0$ or $\text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\mathbb{P}, \mathbf{M}), \mathbf{w}) \neq 1$, then the challenger returns $\perp$. Otherwise, it picks a uniformly random bit $b \in \{0, 1\}$, computes $\Sigma_0 \leftarrow \text{SimSign}(\mathbf{M})$ and $\Sigma_1 \leftarrow \text{Sign}(\text{sk}, \mathbf{M}, \mathbf{w})$, and returns the challenge signature Σ_b to $\mathcal{A}$.

As discussed in the proof of Theorem 5.4.2, the distributions of the simulated signature Σ_0 and the real signature Σ_1 are statistically close. Therefore, the advantage of any (even computationally unbounded) adversary in distinguishing Σ_0 and Σ_1 is negligible in the security parameter n .

Extractability. To prove the extractability of the given DPBS scheme under the SIS assumption, we follow a similar strategy as in the proof of Theorem 5.4.3. However, since a signing key is now a basis instead of a vector, we need to modify the **Initialization** and **Key Queries** phases from the proof of Theorem 5.4.3. The details are described below.

Let $\mathcal{A}$ be an adversary attacking the extractability of the DPBS scheme with respect to $\mathcal{L}(\text{PC}_{\mathbf{G}_1, \mathbf{G}_2})$. We use $\mathcal{A}$ to construct a $\text{SIS}_{n, (D\ell+1)m+1, q, \beta}^\infty$ solver $\mathcal{B}$ as follows. At the beginning, the algorithm $\mathcal{B}$ is given a uniformly random challenge matrix $\hat{\mathbf{C}} \in \mathbb{Z}_q^{n \times [(D\ell+1)m+1]}$, and its goal is to find a non-zero vector $\mathbf{v} \in \mathbb{Z}^{(D\ell+1)m+1}$ such that $\|\mathbf{v}\|_\infty \leq \beta$ and $\hat{\mathbf{C}} \cdot \mathbf{v} = \mathbf{0} \pmod{q}$. The algorithm $\mathcal{B}$ parses the matrix $\hat{\mathbf{C}}$ as $[\mathbf{C}|\mathbf{u}]$, where the matrix $\mathbf{C} = [\mathbf{C}_0|\mathbf{C}_1|\dots|\mathbf{C}_\ell|\dots|\mathbf{C}_{D\ell}] \in \mathbb{Z}_q^{n \times (D\ell+1)m}$ and the vector $\mathbf{u} \in \mathbb{Z}_q^n$. Then $\mathcal{B}$ proceeds as follows.

Initialization. $\mathcal{B}$ simulates the **Setup** algorithm as follows:

1. Pick a uniformly random target policy vector $\mathbb{p}^* = (\mathbf{p}_1^*, \dots, \mathbf{p}_D^*) \xleftarrow{\$} \{0, 1\}^{D\ell}$.
2. Set $\mathbf{A}_0 = \mathbf{C}_0 \in \mathbb{Z}_q^{n \times m}$, and for each $i \in [D\ell]$, set $\mathbf{A}_i^{\mathbb{p}^*[i]} = \mathbf{C}_i \in \mathbb{Z}_q^{n \times m}$.
3. Run $\text{TrapGen}(n, m, q)$ (see Lemma 2.4.4) repeatedly to obtain $D\ell$ pairs $(\mathbf{F}_i, \mathbf{T}_i)$

for $i \in [D\ell]$, where $\mathbf{F}_i \in \mathbb{Z}_q^{n \times m}$, and $\mathbf{T}_i \in \mathbb{Z}^{m \times m}$ is a short basis of $\Lambda_q^\perp(\mathbf{F}_i)$. Then for each $i \in [D\ell]$, set $\mathbf{A}_i^{1-\mathbb{P}^*[i]} = \mathbf{F}_i$.

4. Define $\mathbf{A} = [\mathbf{A}_0 | \mathbf{A}_1^0 | \mathbf{A}_1^1 | \dots | \mathbf{A}_{D\ell}^0 | \mathbf{A}_{D\ell}^1]$. (By Lemma 2.4.4, the distribution of $\mathbf{A}$ is statistically close to the uniform distribution over $\mathbb{Z}_q^{n \times (2D\ell+1)m}$.)

For each $\mathbb{p} \in \{0, 1\}^{r\ell}$ with $r \in [D]$, the sub-matrix $\mathbf{A}_{\mathbb{p}} \in \mathbb{Z}_q^{n \times (r\ell+1)m}$ of $\mathbf{A}$ is defined as usual, i.e., $\mathbf{A}_{\mathbb{p}} = [\mathbf{A}_0 | \mathbf{A}_1^{\mathbb{p}[1]} | \dots | \mathbf{A}_{r\ell}^{\mathbb{p}[r\ell]}]$. By construction, $\mathbf{A}_{\mathbb{p}^*} = \mathbf{C}$.

5. Set $j \leftarrow 0$ and initialize a matrix recorder F such that $F[0] \leftarrow (\varepsilon, \varepsilon)$, together with two set recorders $F_P \leftarrow \emptyset$ and $F_S \leftarrow \emptyset$.
6. Send $\mathbf{pp}^* = (\mathbf{A}, \mathbf{u})$ to the adversary $\mathcal{A}$.

Note that the distribution of $\mathbf{pp}^*$ is statistically close to that of the real $\mathbf{pp}$ produced by the legitimate **Setup** algorithm.

Queries. Suppose that $\mathcal{A}$ queries key delegation on input $(i, \mathbf{p}')$. $\mathcal{B}$ proceeds as follows:

- i. If $i \notin \{0, \dots, j\}$, then return $\perp$.
- ii. Else if $i = 0$, then
 - if $\mathbf{p}' = \mathbf{p}_1^*$, for which $\mathcal{B}$ does not have a signing key, then $\mathcal{B}$ aborts;
 - otherwise,
 - let $k_0 = \min\{k \in [\ell] : \mathbf{p}'[k] \neq \mathbf{p}_1^*[k]\}$. (Since $\mathbf{p}' \neq \mathbf{p}_1^*$, such an index k_0 must exist.) Note that one has:

$$\mathbf{A}_{k_0}^{\mathbf{p}'[k_0]} = \mathbf{F}_{k_0} = \mathbf{A}_{k_0}^{1-\mathbf{p}_1^*[k_0]}.$$

- Run $\text{ExtBasis}(\mathbf{F}_{k_0}, \mathbf{A}_{\mathbf{p}'}, \mathbf{T}_{k_0})$ to obtain $\bar{\mathbf{S}}_{\mathbf{p}'}$ (see Lemma 2.4.6).
 - Run $\text{RandBasis}(\mathbf{A}_{\mathbf{p}'}, \bar{\mathbf{S}}_{\mathbf{p}'}, \sigma)$ to obtain $\mathbf{S}_{\mathbf{p}'}$ (see Lemma 2.4.5).
 - Set $j \leftarrow j + 1$, $F[j][1] \leftarrow \mathbf{p}'$, and $F[j][2] \leftarrow \mathbf{S}_{\mathbf{p}'}$.
- iii. Else if $0 < i \leq j$, then compute $j \leftarrow j + 1$, $F[j][1] \leftarrow F[i][1] \parallel \mathbf{p}'$, and $F[j][2] \leftarrow \text{Delegate}(\text{pp}^*, F[i][2], \mathbf{p}')$.

Suppose that $\mathcal{A}$ queries key revelation on input an index i . $\mathcal{B}$ proceeds as follows:

- i. If $i \notin [j]$, then return $\perp$.
- ii. Otherwise, compute $\mathbf{sk} \leftarrow F[i][2]$, $F_P \leftarrow F_P \cup \{F[i][1]\}$ and return $\mathbf{sk}$.

Suppose that $\mathcal{A}$ queries simulated signature on input message $\mathbf{M}$. $\mathcal{B}$ generates $\Sigma \leftarrow \text{SimSign}(\mathbf{M})$, sets $F_S \leftarrow F_S \cup \{(\mathbf{M}, \Sigma)\}$, and returns Σ .

We remark that, at this point, if the algorithm $\mathcal{B}$ has not aborted, then the choice of the target policy vector $\mathbb{p}^*$ is completely hidden from $\mathcal{A}$'s view.

After $\mathcal{A}$ eventually outputs a forged signature $\Sigma^* = (\mathbf{M}^*, \Pi^*, r)$ satisfying the property: $\text{Verify}(\text{pp}, \Sigma^*) = 1$ and $(\mathbf{M}^*, \Sigma^*) \notin F_S$, then $\mathcal{B}$ starts to exploit the signature for solving the given $\text{SIS}_{n, (D\ell+1)m+1, q, \beta}^\infty$ challenge. Similar to the proof of Theorem 5.4.3, $\mathcal{B}$ can construct an algorithm **Extract** that, on input the forged signature of $\mathcal{A}$, with non-negligible probability and within polynomial time, outputs a tuple $(\bar{\mathbb{p}}, \bar{\mathbb{z}}, \bar{\mathbb{w}})$ such that $\bar{\mathbb{p}}$ is a prefix of $\mathbb{p}^*$ and:

$$\begin{cases} \text{PC}_{\mathbf{G}_1, \mathbf{G}_2}((\bar{\mathbb{p}}, \mathbf{M}^*), \bar{\mathbb{w}}) = 1; \\ \mathbf{A}_{\bar{\mathbb{p}}} \cdot \bar{\mathbb{z}} = \mathbf{u} \bmod q; \quad \|\bar{\mathbb{z}}\|_\infty \leq \beta. \end{cases}$$

The algorithm $\mathcal{B}$ then appends 0's to $\bar{\mathbb{z}}$ to obtain a vector $\mathbf{z} \in \{0, 1\}^{(D\ell+1)m}$. Now let $\mathbf{v} \in \mathbb{Z}^{(D\ell+1)m+1}$ be the vector obtained by appending a coordinate -1 to

the vector $\mathbf{z}$, then one has

$$\mathbf{v} \neq \mathbf{0} \text{ and } \|\mathbf{v}\|_\infty \leq \beta \text{ and } \widehat{\mathbf{C}} \cdot \mathbf{v} = \mathbf{C} \cdot \mathbf{z} - \mathbf{u} = \mathbf{0} \bmod q.$$

Finally the algorithm $\mathcal{B}$ outputs the vector $\mathbf{v}$, which is a valid solution to the given $\text{SIS}_{n,(D\ell+1)m+1,q,\beta}^\infty$ challenge.

6. REVOCABLE IDENTITY-BASED ENCRYPTION FROM LATTICES VIA SUBSET DIFFERENCE METHOD

6.1 *Revocable Identity-Based Encryption*

We first recall the definition and security requirements for revocable identity-based encryption (RIBE) as defined in [15, 37, 101].

6.1.1 *Definitions*

An identity-based encryption with efficient revocation or simply an RIBE scheme has seven probabilistic polynomial-time (PPT) algorithms, **Setup**, **PriKeyGen**, **UpdateKey**, **DecKeyGen**, **Encrypt**, **Decrypt**, and **Revoke** with associated message space $\mathcal{M}$, ciphertext space $\mathcal{C}$, identity space $\mathcal{I}$, and time space $\mathcal{T}$. The size of $\mathcal{T}$ is polynomial in the security parameter. The time is treated as discrete. There are three kinds of parties in this scheme – key authority, sender and receiver. Each algorithm is run by one of them. The key authority maintains a revocation list RL and a state st . An algorithm is called stateful if it updates RL or st . The RIBE scheme proceeds as follows:

Setup($1^\lambda, 1^N$). On input a security parameter λ and a maximal number of users N , this algorithm outputs public parameters pp , a master key msk , a revocation list RL (initially empty), and a state st . (This is stateful and run by the key authority.)

PriKeyGen($\text{pp}, \text{msk}, \text{id}, \text{st}$). On input the public parameters pp , the master key msk , an identity $\text{id} \in \mathcal{I}$, and the state st , this algorithm outputs a private key sk_{id} and an updated state st . (This is stateful and run by the key authority.)

UpdateKey($\text{pp}, \text{msk}, \text{t}, \text{RL}, \text{st}$). On input the public parameters pp , the master key msk , a key update time $\text{t} \in \mathcal{T}$, the revocation list RL at time t , and the state st , this algorithm outputs an update key uk_{t} . (This is run by the key authority.)

DecKeyGen($\text{pp}, \text{sk}_{\text{id}}, \text{uk}_{\text{t}}$). On input the public parameters pp , a private key sk_{id} , and an update key uk_{t} , this algorithm outputs a decryption key $\text{dk}_{\text{id}, \text{t}}$, or $\perp$ if id has been revoked before t . (This is deterministic and run by the receiver.)

Encrypt($\text{pp}, \text{id}, \text{t}, b$). On input the public parameters pp , an identity $\text{id} \in \mathcal{I}$, a time $\text{t} \in \mathcal{T}$, and a message $b \in \mathcal{M}$, this algorithm outputs a ciphertext $\text{ct}_{\text{id}, \text{t}}$. (This is run by the sender.)

Decrypt($\text{pp}, \text{dk}_{\text{id}, \text{t}}, \text{ct}_{\text{id}, \text{t}}$). On input the public parameters pp , a decryption key $\text{dk}_{\text{id}, \text{t}}$, and a ciphertext $\text{ct}_{\text{id}, \text{t}}$, this algorithm outputs a message $b \in \mathcal{M}$ or $\perp$. (This is deterministic and run by the receiver.)

Revoke($\text{id}, \text{t}, \text{RL}, \text{st}$). On input an identity $\text{id} \in \mathcal{I}$ to be revoked, a revocation time $\text{t} \in \mathcal{T}$, the current revocation list RL and a state st , this algorithm outputs an updated revocation list RL . (This is stateful and run by the key authority.)

The correctness condition requires that for any $\lambda \in \mathbb{N}$ and N , any $(\text{pp}, \text{msk}) \leftarrow \text{Setup}(1^\lambda, 1^N)$, any $b \in \mathcal{M}$, all possible state st , and a revocation list RL , if $\text{id} \in \mathcal{I}$

is not revoked on a time $t \in \mathcal{T}$, then for

$$\begin{aligned} (sk_{id}, st) &\leftarrow \mathbf{PriKeyGen}(\mathbf{pp}, \mathbf{msk}, id, st), \\ uk_t &\leftarrow \mathbf{UpdatKey}(\mathbf{pp}, \mathbf{msk}, t, \mathbf{RL}, st), \\ dk_{id,t} &\leftarrow \mathbf{DecKeyGen}(\mathbf{pp}, sk_{id}, uk_t), \end{aligned}$$

the following always holds except for a negligible probability:

$$\mathbf{Decrypt}(\mathbf{pp}, dk_{id,t}, \mathbf{Encrypt}(\mathbf{pp}, id, t, b)) = b.$$

6.1.2 Security Requirements

The security model of RIBE was introduced by Boldyreva et al. [15], where a selectively secure RIBE was given. Later Libert and Vergnaud [68] proposed an RIBE scheme satisfying the adaptive security model. Recently, Seo and Emura [101] advanced the security model by considering the decryption key exposure attack. In our construction, we follow Libert and Vergnaud's security model.

The slight difference is that we formalize the RIBE adaptive security via a game that expresses a strong privacy property called *indistinguishability from random*. Agrawal et al. [2] defined this kind of game for IBE. They also referred that indistinguishability from random implies both the recipient anonymity and semantic security, and the ciphertext does not reveal the public parameters ($\mathbf{pp}$) used to create it. For RIBE, the game is designed as follows.

Setup: The challenger $\mathcal{B}$ runs $\mathbf{Setup}(1^\lambda, 1^N)$ to generate public parameters $\mathbf{pp}$, a master key $\mathbf{msk}$, a revocation list $\mathbf{RL}$ (initially empty), and a state $\mathbf{st}$. Then $\mathcal{B}$ gives $\mathbf{pp}$ to the adversary $\mathcal{A}$.

Query: $\mathcal{A}$ is allowed to adaptively make a polynomial number of queries of the following oracles (the oracles share the state) with some restrictions.

1. **PriKeyGen**($\cdot$): On input an identity id , return a private key sk_{id} by running the algorithm $\text{PriKeyGen}(\text{pp}, \text{msk}, \text{id}, \text{st}) \rightarrow \text{sk}_{\text{id}}$.
2. **UpdateKey**($\cdot$): On input a time t , return an update key uk_{t} by running the algorithm $\text{UpdateKey}(\text{pp}, \text{msk}, \text{t}, \text{RL}, \text{st}) \rightarrow \text{uk}_{\text{t}}$.
3. **Revoke**($\cdot, \cdot$): On input an identity id and a time t , update RL by running the algorithm $\text{Revoke}(\text{id}, \text{t}, \text{RL}, \text{st})$.

Challenge: $\mathcal{A}$ outputs an identity id^* , a time t^* , and a plaintext $b^* \in \mathcal{M}$, on which it chooses to be challenged. $\mathcal{B}$ randomly picks a bit r from $\{0, 1\}$ and a ciphertext C from $\mathcal{C}$. If $r = 1$, it sets the challenge ciphertext to $C^* := C$. If $r = 0$, it sets the challenge ciphertext to $C^* := \text{Encrypt}(\text{pp}, \text{id}^*, \text{t}^*, b^*)$. It sends C^* as the challenge ciphertext to the adversary $\mathcal{A}$.

Guess: $\mathcal{A}$ may further make a polynomial number of queries of the oracles as in the query phase, then it outputs a bit r' . We say $\mathcal{A}$ succeeds if $r' = r$.

In the above game, the following restrictions must hold:

1. **UpdateKey**($\cdot$) and **Revoke**($\cdot, \cdot$) can only be queried on time that is no less than the time of all previous queries.
2. **Revoke**($\cdot, \cdot$) cannot be queried on time t if **UpdateKey**($\cdot$) has already been queried on time t .
3. If **PriKeyGen**($\cdot$) was queried on id^* , then **Revoke**($\cdot, \cdot$) must be subsequently queried on the pair (id^*, t) for some $\text{t} \leq \text{t}^*$.

The above adversary $\mathcal{A}$ is called an IND-RID-CPA PPT adversary. The advantage of $\mathcal{A}$ attacking an RIBE scheme $\mathcal{RIBE}$ is defined as

$$\text{Adv}_{\mathcal{RIBE}, \mathcal{A}}^{\text{IND-RID-CPA}}(\lambda) = \left| \Pr[r' = r] - \frac{1}{2} \right|.$$

We say that an RIBE scheme $\mathcal{RIBE}$ is indistinguishable from random under adaptive identity-time attacks, if for any IND-RID-CPA PPT adversary $\mathcal{A}$, we have that its advantage $\text{Adv}_{\mathcal{RIBE}, \mathcal{A}}^{\text{IND-RID-CPA}}(\lambda)$ is a negligible function in λ .

6.2 Overview of Our Techniques

Our new construction of revocable identity-based encryption (RIBE) from lattices consists of the following building blocks:

- (i) adaptive-ID secure identity-based encryption (IBE) proposed by Agrawal et al. [2];
- (ii) the subset difference (SD) method proposed by Naor et al. [83];
- (iii) Shamir's secret sharing scheme utilized in fuzzy IBE proposed by Agrawal et al. [5].

Agrawal et al.'s adaptive-ID secure IBE scheme is a lattice analogue of Waters' fully secure IBE scheme [108]. Note that as the time space in RIBE is polynomial size in the public parameters, we may just use Agrawal et al.'s selective-ID secure IBE scheme for the time part. However, for simplicity, we apply two instances of Agrawal et al.'s adaptive-ID secure IBE scheme with regards to the users' identities and times, respectively. Our scheme does not just roughly combine the two adaptive-ID secure IBE instances to achieve the adaptive-ID security of RIBE. In contrast to

single adaptive-ID secure IBE schemes, the adversary in our RIBE scheme may have already queried the private key for the challenge identity ID^* and the update key for the challenge time T^* before the challenge phase. Hence we cannot easily adopt the non-abort probability as in [2]. Instead, to achieve the adaptive-ID security, we further investigate the abort-resistant hash functions used in [2, 107] and provide more general analysis of the abort-resistant properties of Waters' hash families [107] with respect to our requirements. Then we carefully design the abort events in the series of security challenging games and prove that they are indistinguishable under the properties of the abort-resistant hash functions (see Appendix E.1) and the learning with errors (LWE) assumption (see Definition 2.4.12).

For revocation, we adopt the SD scheme, an alternative subset cover framework to the well-known complete subtree (CS) scheme. Both instantiations are binary tree based and introduced by Naor et al. [83]. However, the main advantage of the SD method is that for r revoked leaf nodes out of a total of N leaf nodes in one binary tree, the size of the covering set is at most $2r - 1$ (in the worst case, or $1.25r$ in the average case) and independent of N , whereas the covering set in the CS method has size $r \log(N/r)$, logarithmic complexity in N . The tradeoff is that the size of the private sets is slightly increased from $O(\log N)$ to $O(\log^2 N)$. While the private key is a long-term key transferred only once via a secure channel, the update key is a short-term key that will be broadcast periodically via public channels, therefore, using the SD method can efficiently reduce the overheads of the key authority.

We require a random n -vector $\mathbf{u}$ to be part of the public parameters. It acts as the secret for sharing and the subsets in one binary tree, formed by the same ancestor and same depth descendants, act as sharing parties. Different from the CS framework used in prior works [37, 101], the SD method requires a pair of certain subsets, instead of one common node, to determine the non-revocation and generate

the decryption key. Thus we cannot apply Chen et al.’s methods that directly and randomly split the vector $\mathbf{u}$ inside one node, which indeed is a 2-out-of-2 secret sharing scheme. Based on the SD framework, we have to share $\mathbf{u}$ inside families of subsets formed by the same ancestor and same depth descendants. The size of one such family is a power of 2. Thus it is necessary to consider 2-out-of- n secret sharing schemes. One natural candidate to solve this problem is Shamir’s secret sharing scheme. However, to ensure the correctness of the dual-Regev type encryption scheme in our scheme, we have to bound the size of the vectors in the decryption key. Then we apply the technique to “clear the denominators” of the Lagrangian coefficients from [5] to revise the encryption and set proper parameters to ensure the correctness.

Chen et al.’s RIBE scheme [37] is a pioneering work on constructing RIBE schemes from lattices. In Table 6.1, we elucidate the pros and cons of our scheme in comparison with Chen et al.’s scheme.

Table 6.1: Comparison of our RIBE scheme with Chen et al.’s

	Chen et al.’s [37]	Ours
public key size	$\tilde{O}(n^{2+\delta})$	$\tilde{O}(n^{2+\delta+\epsilon})$
private key size	$O(\log N) \cdot \tilde{O}(n^{1+\delta})$	$O(\log^2 N) \cdot \tilde{O}(n^{1+\delta})$
update key size	$r \log \frac{N}{r} \cdot \tilde{O}(n^{1+\delta})$	$(2r - 1) \cdot \tilde{O}(n^{1+\delta})$
ciphertext size	$\tilde{O}(n^{1+\delta})$	$\tilde{O}(n^{1+\delta+\epsilon})$
Security	Selective-ID	Adaptive-ID

In Table 6.1, δ is a small constant such that $n^\delta > O(\log n)$, and ϵ is a small constant such that $\epsilon < 1/2$. r is the number of revoked users out of N users. When r is small compared with N , our scheme has shorter update keys. Both schemes are secure under the LWE assumption in the standard model. However, our scheme is based on the LWE assumption with sub-exponential factors, in contrast to polynomial factors in Chen et al.’s scheme.

6.3 Subset Difference Method

Naor et al. [83] gave two subset-cover revocation schemes with different performances: the complete subtree (CS) method and the subset difference (SD) method. Just as for the CS method, the SD method is also r -flexible, namely it works with any number of revocations. However, any leaf node in the SD method belongs to more subsets than in the CS method ($O(N)$ instead of $O(\log N)$). This characteristic of the SD method leads to the reduction in the size of the covering set.

Let BT be a full binary tree with $N_{\max}$ leaf nodes. Label the nodes in BT by 1 to $2N_{\max} - 1$ in the way that the root is labeled 1, and if the parent is labeled i , then the left child is labeled $2i$ and the right child is labeled $2i + 1$. One example of the labeling is in Figure 6.1. Let v_i denote the node labeled i for $i \in [2N_{\max} - 1]$. The depth of a node v_i is the length of the path from the root to v_i . For any $v_i \in \text{BT}$, T_i is defined as a subtree that is rooted at v_i . For any $v_i, v_j \in \text{BT}$, $T_{i,j}$ is defined as a subtree $T_i - T_j$. S_i , $S_{i,j}$ are the sets of leaf nodes of T_i , $T_{i,j}$, respectively.

For a binary tree BT and a subset R of nodes, $ST(R)$ is the Steiner tree induced by R and the root node, that is, the minimal subtree of BT that connects all the leaf nodes in R and the root node.

The SD scheme is summarized as follows (an example is shown in Figure 6.1):

SD.Setup($N_{\max}$): This algorithm takes as input the maximum number of users $N_{\max}$ (for simplicity, $N_{\max} = 2^d$). It sets a full binary tree BT of depth d . Every user is assigned to a different leaf node in BT . The collection S is the set of all subsets $S_{i,j}$ with $v_i, v_j \in \text{BT}$ where v_i is an ancestor of v_j . It outputs BT .

SD.Assign(BT, ν): This algorithm takes as input the full binary tree BT and a user ν . Let v_ν be the leaf node assigned to ν and let $\text{Path}(v_\nu) = (v_{k_0}, v_{k_1}, \dots, v_{k_n})$

be the path from the root node v_{k_0} to the leaf node $v_{k_n} = v_\nu$. For all $i, j \in \{k_0, k_1, \dots, k_n\}$ such that v_j is a descendant of v_i , it adds $S_{i,j}$ defined by two nodes v_i and v_j in the path into a private set PV_ν . Finally, it outputs PV_ν .

SD.Cover(BT, R): This algorithm takes as input the binary tree **BT** and a revoked set R of users. Let T be the Steiner tree $ST(R)$, then it computes CV_R iteratively by removing nodes from T until T only has a single node as follows:

1. Find two leaves v_i and v_j in T such that their least-common-ancestor (lca) v does not have any other descendants being leaf node in T except v_i and v_j . Let v_l and v_k be the two children of v such that v_l is an ancestor of v_i , while v_k is an ancestor of v_j . (If there is only one leaf node, set $v_i = v_j$ as the leaf, v as the root of T and $v_l = v_k = v$.)
2. If $v_l \neq v_i$, $CV_R = CV_R \cup \{S_{l,i}\}$; if $v_k \neq v_j$, $CV_R = CV_R \cup \{S_{k,j}\}$.
3. Remove all the descendants of v from T and make v a leaf of T .

SD.Match(CV_R, PV_ν): This algorithm takes as input a covering set $CV_R = \{S_{i,j}\}$ and a private set $PV_\nu = \{S'_{i',j'}\}$. If it finds two subsets $S_{i,j} \in CV_R$ and $S'_{i',j'} \in PV_\nu$ with $(i = i') \wedge (d_j = d_{j'}) \wedge (j \neq j')$ where d_j is the depth of v_j , then it outputs $(S_{i,j}, S'_{i',j'})$. Otherwise, it outputs $\perp$.

Figure 6.1 serves as an example for $R = \{\text{id}_4, \text{id}_5, \text{id}_6, \text{id}_7\}$. First we compute CV_R :

1. Choose v_6 as lca of v_{12} and v_{13} . As v_{12} and v_{13} are exactly the children of v_6 , there is no subset chosen for CV_R .
2. Get v_6 as a leaf; then v_3 is lca of v_{14} and v_6 . As v_7 is a child of v_3 and an ancestor of v_{14} , we have $CV_R = \{S_{7,14}\}$;

3. Get v_3 as a leaf; then the root v_1 is lca of v_{11} and v_3 . As v_2 is a child of v_1 and an ancestor of v_{11} , we have $CV_R = \{S_{7,14}, S_{2,11}\}$.
4. Get the root as a leaf.

Meanwhile, $PV_{id_2} = \{S_{1,2}, S_{1,4}, S_{1,9}, S_{2,4}, S_{2,9}, S_{4,9}\}$ and $SD.Match(CV_R, PV_{id_2})$ outputs $(S_{2,11}, S_{2,9})$. Similarly, $SD.Match(CV_R, PV_{id_8})$ outputs $(S_{7,14}, S_{7,15})$. For $\nu \in R$, $SD.Match(CV_R, PV_\nu)$ outputs $\perp$.

Figure 6.1: Example of SD scheme for $R = \{id_4, id_5, id_6, id_7\}$

The correctness of the SD scheme is defined as follows: For all $N = 2^d$, $BT \leftarrow SD.Setup(N)$, $PV_\nu \leftarrow SD.Assign(BT, \nu)$, R a set of leaf nodes in BT , let $CV_R \leftarrow SD.Cover(BT, R)$, then

1. If $u \notin R$, then $SD.Match(CV_R, PV_\nu) = (S_{i,j}, S'_{i',j'})$;
2. If $u \in R$, then $SD.Match(CV_R, PV_\nu) = \perp$.

Naor et al. [83] characterized the size of a private set and a covering set in the SD method in the following lemma:

Lemma 6.3.1 ([83]). *Let $N_{\max}$ be the number of leaf nodes and let r be the size of a revoked set. The size of a private set is $O(\log^2 N_{\max})$ and the size of a covering set is at most $2r - 1$ in the SD scheme.*

6.4 *RIBE from Lattices via Subset Difference Method*

In this section, we provide our RIBE scheme from lattices via the SD method. Then, we look into the proper parameter sizes for our construction. Since the security analysis of our scheme is derived from and similar to the analysis of Agrawal et al.'s adaptive-ID secure IBE scheme given in [3], we place it in Appendix E.

6.4.1 *Description of the Scheme*

We begin by demonstrating some requirement about an identity and a time in our scheme. We treat an identity id and a time $\mathbf{t}$ each as a sequence of length ℓ in $\{-1, 1\}^\ell$, with possible redundancy, for some $\ell \in \mathbb{Z}$. The requirement is that for any tuple $(\text{id}, \mathbf{t}) \in \mathcal{I} \times \mathcal{T}$, the vectors id and $\mathbf{t}$ are $\mathbb{Z}_q$ -linearly independent. One possible technique to realize it is that if a sequence ends with 1, we put it in $\mathcal{T}$, otherwise in $\mathcal{I}$. Then for each sequence in $\mathcal{I} \cup \mathcal{T}$, we add an extra 1 to the end. Thus, any $\text{id} \in \mathcal{I}$ ends with 01, while any $\mathbf{t} \in \mathcal{T}$ ends with 11. As a result, the two vectors id and $\mathbf{t}$ are $\mathbb{Z}_q$ -linearly independent.

Now, we describe our construction.

Setup($1^\lambda, 1^N$): On input a security parameter λ and the maximal number of users

N , set the parameters $n, m, q, \sigma, \ell, \alpha$, as specified below in Subsection 6.4.3.

Then, it performs as follows:

1. Use the algorithm **TrapGen**(n, m, q) from Lemma 2.4.4 to generate a uniformly random matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with a short basis $\mathbf{T}_\mathbf{A}$ for $\Lambda_q^\perp(\mathbf{A})$.
2. Select $2\ell + 2$ uniformly random matrices $\mathbf{B}_1, \dots, \mathbf{B}_\ell, \mathbf{C}; \mathbf{D}_1, \dots, \mathbf{D}_\ell, \mathbf{G} \in \mathbb{Z}_q^{n \times m}$ and select a uniformly random vector $\mathbf{u} \in \mathbb{Z}_q^n$.

3. Let $\text{UL}, \text{RL}, \text{FL}$ be three initially empty sets, which will be used to record the user list $\{(\text{id}, \nu)\}$, revoked list $\{(\text{id}, \mathbf{t})\}$, and function list $\{(\text{GL}_{d_j}^i, \text{F}_{d_j}^i)\}$ respectively, where the domain of the parameters $\nu, \text{GL}_{d_j}^i, \text{F}_{d_j}^i$ will be specified in later steps.
4. Obtain a binary tree BT by running $\text{SD.Setup}(N)$. Let $\mathcal{S}$ be the collection of all subsets $S_{i,j}$ of BT and let $\text{GL}_{d_j}^i$ be the subset of $\mathcal{S}$ consisting of the $S_{i,j}$ such that the depth of node ν_j is d_j . Save $(\text{GL}_{d_j}^i, \perp)$ to FL . Set the state to $\text{st} = (\text{BT}, \text{UL})$.
5. Output the public parameters PP and master key MK given by

$$\begin{aligned} \text{PP} &= (\mathbf{A}, \{\mathbf{B}_1, \dots, \mathbf{B}_\ell, \mathbf{C}\}, \{\mathbf{D}_1, \dots, \mathbf{D}_\ell, \mathbf{G}\}, \mathbf{u}), \\ \text{MK} &= (\mathbf{T}_\mathbf{A}, \text{FL}). \end{aligned}$$

PriKeyGen($\text{PP}, \text{MK}, \text{id}, \text{st}$): On input the public parameters PP , the master key MK , an identity $\text{id} = (b_1, \dots, b_\ell) \in \{-1, 1\}^\ell$, and the state st , it works as follows:

1. Randomly choose an unassigned leaf node ν in BT and assign it to the identity id . Save (id, ν) to UL . Run $\text{SD.Assign}(\text{BT}, \nu)$ to obtain PV_ν .
2. Let $\mathbf{A}_{\text{id}} = \mathbf{C} + \sum_{i=1}^\ell b_i \mathbf{B}_i \in \mathbb{Z}_q^{n \times m}$.
3. For each $S_{i,j} \in PV_\nu$, perform the following:
 - Retrieve the record $(\text{GL}_{d_j}^i, *)$ from FL . If the second coordinate $*$ is not $\perp$, go to the next step. Else, randomly select $\text{F}_{d_j}^i(x) \in (\mathbb{Z}_q[x])^n$ such that each coordinate of $\text{F}_{d_j}^i$ is a polynomial of degree 1 and $\text{F}_{d_j}^i(0) = \mathbf{u}$. Update $(\text{GL}_{d_j}^i, \text{F}_{d_j}^i)$ to FL .

- Sample $\mathbf{e}_{i,j}^{\text{id}}$ as

$$\mathbf{e}_{i,j}^{\text{id}} \leftarrow \text{SampleLeft} \left(\mathbf{A}, \mathbf{A}_{\text{id}}, \mathbf{T}_{\mathbf{A}}, \mathbf{F}_{d_j}^i(j), \sigma \right).$$

Note that for two sets $S_{i,j}, S_{i,j'} \in \text{GL}_{d_j}^i$ with $j \neq j'$, we have $d_j = d_{j'}$. For given i and d_j , let $\mathbf{F} = \mathbf{F}_{d_j}^i$. We can compute the fractional Lagrangian coefficients

$$L_j = j' \cdot (j' - j)^{-1} \quad \text{and} \quad L_{j'} = j \cdot (j - j')^{-1}$$

such that $\mathbf{u} = \mathbf{F}(0) = L_j \mathbf{F}(j) + L_{j'} \mathbf{F}(j') \bmod q$. That is, as a fraction of integers, L_j can also be evaluated in $\mathbb{Z}_q$, then L_j can be regarded as an integer in $\{0\} \cup [q-1]$.

4. Output the updated state $\mathbf{st}$ and the private key $\mathbf{sk}_{\text{id}}$ for id as

$$(PV_\nu, \{\mathbf{e}_{i,j}^{\text{id}}\}_{S_{i,j} \in PV_\nu}).$$

UpdateKey(PP, MK, $\mathbf{t}$, RL, $\mathbf{st}$): On input the public parameters PP, the master key MK, a time $\mathbf{t} = (t_1, \dots, t_\ell) \in \{-1, 1\}^\ell$, the revoked list RL, and the state $\mathbf{st}$, it works as follows:

1. Take R as a set consisting of such id 's that for some $\mathbf{t}' \leq \mathbf{t}$, $(\text{id}, \mathbf{t}') \in \text{RL}$.
By using UL, define RI as the set of index of leaf nodes corresponding the id 's in R . That is, for any $\text{id} \in R$, if $(\text{id}, \nu) \in \text{UL}$, then put ν in RI .
2. Run $\text{SD.Cover}(\text{BT}, RI)$ to obtain CV_{RI} .
3. Let $\mathbf{A}_{\mathbf{t}} = \mathbf{G} + \sum_{i=1}^{\ell} t_i \mathbf{D}_i \in \mathbb{Z}_q^{n \times m}$.
4. For each $S_{i,j} \in CV_{\text{RI}}$, perform the following:

- Retrieve the record $(\text{GL}_{d_j}^i, *)$ from FL. If the second coordinate $*$ is not $\perp$, go to the next step. Else, randomly select $F_{d_j}^i(x) \in (\mathbb{Z}_q[x])^n$ such that each coordinate of $F_{d_j}^i$ is a polynomial of degree 1 and $F_{d_j}^i(0) = \mathbf{u}$. Update $(\text{GL}_{d_j}^i, F_{d_j}^i)$ to FL.
- Sample $\mathbf{e}_{i,j}^t$ as

$$\mathbf{e}_{i,j}^t \leftarrow \text{SampleLeft} \left(\mathbf{A}, \mathbf{A}_t, \mathbf{T}_A, F_{d_j}^i(j), \sigma \right).$$

5. Output the update key uk_t at time t as

$$(CV_{\text{RI}}, \{\mathbf{e}_{i,j}^t\}_{S_{i,j} \in CV_{\text{RI}}}).$$

Encrypt(PP, id, t, b): On input the public parameters PP, an identity id, a current time t, and a message $b \in \{0, 1\}$, it performs as follows:

1. Let $D = (N - 1)!$.
2. Choose a uniformly random vector $\mathbf{s} \in \mathbb{Z}_q^n$.
3. Choose 2ℓ uniformly random matrixes $\mathbf{R}_i \in \{-1, 1\}^{m \times m}$ for $i = 1, \dots, 2\ell$ and define $\mathbf{R}_{\text{id}} = \sum_{i=1}^{\ell} b_i \mathbf{R}_i$, $\mathbf{R}_t = \sum_{i=1}^{\ell} t_i \mathbf{R}_{\ell+i} \in \{-\ell, \dots, \ell\}^{m \times m}$.
4. Choose a noise value $x \xleftarrow{\bar{\Psi}_\alpha} \mathbb{Z}_q$ and a noise vector $\mathbf{y} \xleftarrow{\bar{\Psi}_\alpha^m} \mathbb{Z}_q^m$. Set

$$\mathbf{x}_{\text{id}} = \begin{pmatrix} \mathbf{y} \\ \mathbf{R}_{\text{id}}^\top \mathbf{y} \end{pmatrix} \quad \text{and} \quad \mathbf{x}_t = \begin{pmatrix} \mathbf{y} \\ \mathbf{R}_t^\top \mathbf{y} \end{pmatrix}.$$

5. Compute

$$\begin{aligned} c_0 &\leftarrow \mathbf{u}^\top \mathbf{s} + Dx + b \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q, \\ \mathbf{c}_1 &\leftarrow (\mathbf{A} \parallel \mathbf{A}_{\text{id}})^\top \mathbf{s} + D\mathbf{x}_{\text{id}} \in \mathbb{Z}_q^{2m}, \\ \mathbf{c}_2 &\leftarrow (\mathbf{A} \parallel \mathbf{A}_{\text{t}})^\top \mathbf{s} + D\mathbf{x}_{\text{t}} \in \mathbb{Z}_q^{2m}. \end{aligned}$$

and output the ciphertext $\text{ct}_{\text{id},\text{t}} = (c_0, \mathbf{c}_1, \mathbf{c}_2)$.

DecKeyGen(PP, sk_{id} , uk_{t}): On input the public parameters PP, a private key sk_{id} and an update key uk_{t} given by

$$\begin{aligned} \text{sk}_{\text{id}} &= (PV_\nu, \{\mathbf{e}_{i,j}^{\text{id}}\}_{S_{i,j} \in PV_\nu}) \text{ and} \\ \text{uk}_{\text{t}} &= (CV_{\text{RI}}, \{\mathbf{e}_{i,j}^{\text{t}}\}_{S_{i,j} \in CV_{\text{RI}}}), \end{aligned}$$

it performs as follows:

1. If $\text{id} \notin R$, run $\text{SD.Match}(CV_{\text{RI}}, PV_\nu)$ to obtain a set pair $(S_{i,j}, S_{i',j'})$ such that $S_{i,j} \in CV_{\text{RI}}$, $S_{i',j'} \in PV_\nu$, and $(i = i') \wedge (d_j = d_{j'}) \wedge (j \neq j')$. Else, output $\perp$.
2. Retrieve $\mathbf{e}_{i',j'}^{\text{id}}$ from sk_{id} and $\mathbf{e}_{i,j}^{\text{t}}$ from uk_{t} .
3. Compute the fractional Lagrangian coefficients L_j and $L_{j'}$ and evaluate them in $\mathbb{Z}_q$, that is, they can be interpreted as elements in $\mathbb{Z}_q$, i.e., integers in $\{0\} \cup [q-1]$.
4. Output the decryption key $\text{dk}_{\text{id},\text{t}} = (L_{j'} \mathbf{e}_{i',j'}^{\text{id}}, L_j \mathbf{e}_{i,j}^{\text{t}})$.

Decrypt(PP, $\text{ct}_{\text{id},\text{t}}$, $\text{dk}_{\text{id},\text{t}}$): On input the public parameters PP, a ciphertext $\text{ct}_{\text{id},\text{t}}$, and a decryption key $\text{dk}_{\text{id},\text{t}}$, it performs as follows:

1. Evaluate $\mathbf{dk}_{\text{id},t}$ in $\mathbb{Z}_q^{2m}$ as $(\mathbf{d}_1, \mathbf{d}_2)$, where $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{Z}_q^m$.
2. Compute $w \leftarrow c_0 - \mathbf{d}_1^\top \mathbf{c}_1 - \mathbf{d}_2^\top \mathbf{c}_2 \bmod q$.
3. Compare w and $\lfloor \frac{q}{2} \rfloor$ regarding them as integers in $\mathbb{Z}$. If $|w - \lfloor \frac{q}{2} \rfloor| < \lfloor \frac{q}{4} \rfloor$ output 1. Else, output 0.

Revoke(id, t, RL, st): On input an identity id, a revocation time t, the revocation list RL, and the state $\mathbf{st} = (\mathbf{BT}, \mathbf{UL})$: If $(\text{id}, *) \notin \mathbf{UL}$, then output $\perp$ as the private key for id was not generated. Else, add (id, t) to RL. Then output the updated revocation list RL.

6.4.2 Correctness

Now, we demonstrate the correctness of our construction. Let $\mathbf{F} = \mathbf{F}_{d_j}^i$. If the ciphertext is generated by operating as described above, we can compute the error term in the decryption algorithm from the following relation:

$$\begin{aligned}
 w &= c_0 - L_{j'} (\mathbf{e}_{i',j'}^{\text{id}})^\top \mathbf{c}_1 - L_j (\mathbf{e}_{i,j}^{\text{t}})^\top \mathbf{c}_2 \\
 &= \mathbf{u}^\top \mathbf{s} - [L_{j'} \mathbf{F}(j') + L_j \mathbf{F}(j)]^\top \mathbf{s} + Dx + b \left\lfloor \frac{q}{2} \right\rfloor \\
 &\quad - DL_{j'} (\mathbf{e}_{i',j'}^{\text{id}})^\top \mathbf{x}_{\text{id}} - DL_j (\mathbf{e}_{i,j}^{\text{t}})^\top \mathbf{x}_{\text{t}} \\
 &= b \left\lfloor \frac{q}{2} \right\rfloor + \underbrace{Dx - DL_{j'} (\mathbf{e}_{i',j'}^{\text{id}})^\top \mathbf{x}_{\text{id}} - DL_j (\mathbf{e}_{i,j}^{\text{t}})^\top \mathbf{x}_{\text{t}}}_{\text{error term}},
 \end{aligned}$$

where all the equalities are in $\mathbb{Z}_q$.

Hence, for the correctness of our construction, it suffices to set the parameters such that the norm of the error term satisfies

$$\begin{aligned}
 &|Dx - DL_{j'} (\mathbf{e}_{i',j'}^{\text{id}})^\top \mathbf{x}_{\text{id}} - DL_j (\mathbf{e}_{i,j}^{\text{t}})^\top \mathbf{x}_{\text{t}}| \\
 &\leq (N!) \left(|x| + 2 \left| (\mathbf{e}_{i',j'}^{\text{id}})^\top \mathbf{x}_{\text{id}} \right| + 2 \left| (\mathbf{e}_{i,j}^{\text{t}})^\top \mathbf{x}_{\text{t}} \right| \right) \leq q/5,
 \end{aligned} \tag{6.1}$$

except for a negligible probability. For the first inequality, as j and j' have the same depth, we know $|j - j'| \leq N - 1$, and hence $j - j'$ divides D . Moreover, j and j' lie in the set $[2N - 1]$, so $DL_j, DL_{j'}$ are integers not exceeding $(N - 1)! \times (2N - 1) < 2(N!)$. Therefore, if we set the parameters such that q is sufficiently larger than $N!$, then our construction satisfies the correctness.

6.4.3 Parameters

To bound the error term in the decryption algorithm of our construction, we first introduce the following lemma. The proof of this lemma is straightforward from Lemma 2.4.8 and Lemma 2.4.15, and a similar case has been analyzed in Lemma 24 in [3]. We omit the details of the proof here.

Lemma 6.4.1. *The norm of the error term in Subsection 6.4.2 is bounded by*

$$4q\sigma\ell m\alpha \cdot \omega(\sqrt{\log m}) \cdot (N!) + 4O(\sigma m^{1.5}) \cdot (N!)$$

except for a negligible probability.

Now we set the parameters to guarantee that the decryption is correct and the security reduction is meaningful. The parameters are set under the following requirements:

1. For the lattice trapdoor generation algorithm in Proposition 2.4.4, the parameters should satisfy $m \geq 2n \log q$. Under this selection of m , the output basis of **TrapGen** has Gram-Schmidt norm at most $O(\sqrt{m})$. The private key $\mathbf{sk}_{\text{id}}$ and the update key $\mathbf{uk}_{\text{t}}$ are generated from the algorithm **SampleLeft** in the real scheme and **SampleRight** in the simulated games. The two algorithms are presented in Subsection 2.4.3. By Lemma 2.4.3, if we set the Gaussian parameter $\sigma \geq$

$\ell m \log m$, then the vectors inside the two keys have length at most $\sigma\sqrt{2m} \leq 2\ell m^{1.5} \log m$ with high probability.

2. The noise distribution is set as $\chi = \overline{\Psi}_\alpha^m$, where $\alpha \geq 2\sqrt{m}/q$ in order to apply Regev's reduction (see Proposition 2.4.14). Any vector $\mathbf{y}$ sampled from this distribution has length satisfying $O(\alpha q \sqrt{m}) \leq 2m$ with high probability.
3. The norm of the error term in Subsection 6.4.2 is bounded by

$$\begin{aligned}
& (N!) \left(|x| + 2 \left| (\mathbf{e}_{i',j'}^{\text{id}})^\top \mathbf{x}_{\text{id}} \right| + 2 \left| (\mathbf{e}_{i,j}^{\text{t}})^\top \mathbf{x}_{\text{t}} \right| \right) \\
& \leq 4q\sigma\ell m\alpha \cdot \omega(\sqrt{\log m}) \cdot (N!) + 4O(\sigma m^{1.5}) \cdot (N!) \\
& \leq 16\ell^2(N!) \cdot m^{2.5} \cdot \log m \cdot \omega(\sqrt{\log m}) \\
& \quad + 4(N!) \cdot O(\ell m^{2.5} \log m) \\
& \leq 16\ell^2(N!) \cdot m^{2.5} (\log m)^{1.5} \\
& \leq 2^{N \log N} \ell^2 \cdot m^{2.5} \cdot (\log m)^{1.5}
\end{aligned}$$

where we used the fact that $N! \leq 2^{N \log N}$. The modulus q satisfying $q \geq 2^{N \log N} \ell^2 \cdot m^{2.5} \cdot (\log m)^{1.5}$ ensures the correctness of our construction.

4. The modulus q should satisfy $q \geq 2(Q^{\text{id}} + |\mathcal{T}|)$, where Q^{id} is the number of queries on the identity for the private key from the adversary, and $|\mathcal{T}|$ is the size of time space. This requirement ensures that our security reduction applies. The details are given in Appendix E.
5. The identity space $\mathcal{I}$ is sufficient for N users, so $2^\ell \geq N$.

To satisfy these requirements, given a small constant $\epsilon \in (0, 1)$, we may set the parameters as follows:

$$n = (N \log N)^{1/\epsilon}, \quad \ell = \log N, \quad (6.2)$$

$$m = 2n^{1+\delta}, \quad q \geq \max \{ 2(Q^{\text{id}} + |\mathcal{T}|), 2^{N \log N} \ell^2 \cdot n^5 \}, \quad (6.3)$$

$$\sigma = \ell n^2 \log n, \quad \alpha = [2^{N \log N} \ell^2 \cdot n^4]^{-1}, \quad (6.4)$$

where q is the nearest prime larger than the bound, and δ is selected such that $1 > \delta > \epsilon$.

Note that from Regev's reduction (see Proposition 2.4.14), the security of our scheme is based on the LWE problem, which is finally based on the hardness of the $2^{O(n^\epsilon)}$ -approximating **GapSVP** or **SIVP** on n -dimensional lattices using algorithms that run in time $q \cdot \text{poly}(n) = 2^{O(n^\epsilon)}$. The same assumption has been used to construct fuzzy IBE by Agrawal et al. in [5]. The security holds for small $\epsilon < 1/2$. Actually we may enhance the security by setting $2^{N \log N} = \text{poly}(n)$, but this will reduce the maximal number of users. In other words, there is a tradeoff between the security and the practical requirement in the number of users. That is one disadvantage of our construction. We include our RIBE scheme here to show the potential of lattices in constructing alternative RIBE schemes.

7. CONCLUSIONS

In this thesis, we have provided some instances, especially in sphere packing and cryptography, to demonstrate the potential of lattices in multi-disciplinary research. It is desirable to uncover more hidden treasure from the structure of lattices.

In Chapter 3, we explicitly constructed two asymptotically good packing families. The main technique is to concatenate families of codes attaining the GV-bound with multiplicative lattices via the canonical embedding $\tau_0 : \mathbb{Q}(\sqrt{-3}) \hookrightarrow \mathbb{R}^2$. Our constructions improve the bounds on the asymptotic density exponent of packing families derived from multiplicative lattices. Moreover, the concatenation method serves as a link that connects packing construction with different fields, such as curves over finite fields and coding theory, which were used in our construction in Chapter 3.

Furthermore, we generalized the concatenation based on the special number field $\mathbb{Q}(\sqrt{-3})$ to one based on an arbitrary number field. This was introduced in Chapter 4, where we proposed a new method to construct dense packings using prime ideals of general number fields, and special codes over the residue fields of the prime ideals. Via the canonical embedding τ , prime ideals can be embedded as lattices, while the codes can be embedded as finite discrete subsets in Euclidean spaces. Then we can apply the concatenation techniques. With the help of **Magma** V2.21-4, several explicit constructions were provided in Examples 4.2.8 - 4.2.11. Especially, in $\mathbb{R}^{256}$, a packing denser than the Barnes-Wall lattice BW_{256} was obtained. More-

over, this method can be utilized to construct asymptotically good packing families. For each number field and prime ideal in Tables 4.1 - 4.4, a lower bound on the asymptotic density exponent of the corresponding packing family was provided.

As sphere packings are continuous analogues of error-correcting codes, the development of one will potentially influence that of the other. We introduced two methods that generalize ideas from concatenated codes to dense sphere packing constructions. In the future, we may consider to further explore the connections and similarities to extend both topics. In addition, we propose to apply the concatenating method to more general packing problems, for example, superball packing [50] and unequal sphere packing [45].

In Chapter 5, we introduced a construction of a (delegatable) policy-based signature scheme from lattices. The contribution is twofold. On the one hand, we enrich the scope of policy-based signatures by providing the first quantum-resistant instantiation. On the other hand, our technical approach can potentially be applied to design a wide variety of privacy-enhanced lattice-based cryptographic constructions.

In Chapter 6, we provided an adaptive-ID secure revocable identity-based encryption scheme from lattices in the standard model under the LWE assumption. To realize identity revocation, our scheme employs the subset difference method.

However, our construction in Chapter 6 cannot thwart the decryption key exposure attack, proposed by Seo and Emura [101], which considers not only the exposure of a long-term private key and a short-term update key, but also the exposure of a short-term decryption key. To directly apply Seo and Emura's approach which randomizes a short-term decryption key, we need a proper one-way function to randomize the short-term decryption key in our scheme, but we could not find such a one-way function and we leave it as a problem for future investigations. In our scheme, we just put $\mathbf{sk}_{\text{id}}$ and $\mathbf{uk}_t$ together to generate the decryption key. If we have

a proper one-way function mapping the pair of vectors to another pair which also satisfies the dual-Regev encryption scheme, while it is hard to regain $\mathbf{sk}_{\text{id}}$ and $\mathbf{uk}_t$ from the new pair, then our scheme can be modified to achieve the security against the decryption key exposure attack.

Lattice-based cryptography has grown in popularity due to the potential quantum resistance feature and new functionalities such as fully-homomorphic encryption. In this thesis, we have provided detailed realizations of two cryptographic functionalities from lattices. Our technical approaches can potentially be generalized to construct other relevant cryptographic functionalities, e.g. functional signatures [24], anonymous credential [31], revocable identity-based signatures [101], revocable attribute-based encryption [98], etc.

BIBLIOGRAPHY

- [1] Abe, M., Fuchsbauer, G., Groth, J., Haralambiev, K., Ohkubo, M.: Structure-Preserving Signatures and Commitments to Group Elements. In: Rabin, T. (ed.) CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 209–236. Springer Berlin Heidelberg (2010)
- [2] Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. In: Gilbert, H. (ed.) EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 553–572. Springer Berlin Heidelberg (2010)
- [3] Agrawal, S., Boneh, D., Boyen, X.: Efficient Lattice (H)IBE in the Standard Model. Full version of [2]. (2010), <http://crypto.stanford.edu/~dabo/pubs/papers/latticebb.pdf>
- [4] Agrawal, S., Boneh, D., Boyen, X.: Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 98–115. Springer Berlin Heidelberg (2010)
- [5] Agrawal, S., Boyen, X., Vaikuntanathan, V., Voulgaris, P., Wee, H.: Functional Encryption for Threshold Functions (or Fuzzy IBE) from Lattices. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. Lecture Notes in Computer Science, vol. 7293, pp. 280–297. Springer Berlin Heidelberg (2012)
- [6] Ajtai, M.: Generating Hard Instances of Lattice Problems (Extended Ab-

- tract). In: STOC 1996. pp. 99–108. ACM, New York (1996)
- [7] Ajtai, M.: Generating Hard Instances of the Short Basis Problem. In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. Lecture Notes in Computer Science, vol. 1644, pp. 1–9. Springer Berlin Heidelberg (1999)
- [8] Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous Hardcore Bits and Cryptography against Memory Attacks. In: Reingold, O. (ed.) TCC 2009, Lecture Notes in Computer Science, vol. 5444, pp. 474–495. Springer Berlin Heidelberg (2009)
- [9] Alwen, J., Peikert, C.: Generating Shorter Bases for Hard Random Lattices. In: Albers, S., Marion, J. (eds.) STACS 2009. LIPIcs, vol. 3, pp. 75–86. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2009)
- [10] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. In: Halevi, S. (ed.) CRYPTO 2009, Lecture Notes in Computer Science, vol. 5677, pp. 595–618. Springer Berlin Heidelberg (2009)
- [11] Atiyah, M., MacDonald, I.: Introduction to Commutative Algebra. Addison-Wesley Series in Mathematics, Westview Press (1994)
- [12] Bachoc, C.: Applications of Coding Theory to the Construction of Modular Lattices. *Journal of Combinatorial Theory, Series A* 78(1), 92 – 119 (1997)
- [13] Bayer-Fluckiger, E.: Ideal Lattices. In: Wüstholtz, G. (ed.) *A Panorama of Number Theory Or The View from Baker’s Garden*. pp. 168–184. Cambridge University Press (2002)
- [14] Bellare, M., Fuchsbauer, G.: Policy-Based Signatures. In: Krawczyk, H. (ed.) PKC 2014. Lecture Notes in Computer Science, vol. 8383, pp. 520–537.

- Springer Berlin Heidelberg (2014), full version available at <http://eprint.iacr.org/2013/413>
- [15] Boldyreva, A., Goyal, V., Kumar, V.: Identity-Based Encryption with Efficient Revocation. In: Ning, P., Syverson, P.F., Jha, S. (eds.) CCS 2008. pp. 417–426. ACM, New York (2008)
- [16] Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. Lecture Notes in Computer Science, vol. 3027, pp. 223–238. Springer Berlin Heidelberg (2004)
- [17] Boneh, D., Franklin, M.K.: Identity-Based Encryption from the Weil Pairing. In: Kilian, J. (ed.) CRYPTO 2001. Lecture Notes in Computer Science, vol. 2139, pp. 213–229. Springer Berlin Heidelberg (2001)
- [18] Boneh, D., Gentry, C., Hamburg, M.: Space-Efficient Identity Based Encryption Without Pairings. In: FOCS 2007. pp. 647–657. IEEE (2007)
- [19] Borevich, Z.I., Shafarevich, I.R.: Number Theory, vol. 20. Academic Press, Florida (1966)
- [20] Bosma, W., Cannon, J., Fieker, C., Steel, A.: Handbook of Magma Functions. Available online <http://magma.maths.usyd.edu.au/magma/handbook/>, v2.21 edn. (2015)
- [21] Bosma, W., Cannon, J., Playoust, C.: The Magma Algebra System. I. The User Language. Journal of Symbolic Computation 24(3-4), 235–265 (1997)
- [22] Boyen, X.: Mesh Signatures. In: Naor, M. (ed.) EUROCRYPT 2007. Lecture Notes in Computer Science, vol. 4515, pp. 210–227. Springer Berlin Heidelberg (2007)

- [23] Boyen, X.: Lattice Mixing and Vanishing Trapdoors: A Framework for Fully Secure Short Signatures and More. In: Nguyen, P., Pointcheval, D. (eds.) PKC 2010. Lecture Notes in Computer Science, vol. 6056, pp. 499–517. Springer Berlin Heidelberg (2010)
- [24] Boyle, E., Goldwasser, S., Ivan, I.: Functional Signatures and Pseudorandom Functions. In: Krawczyk, H. (ed.) PKC 2014, Lecture Notes in Computer Science, vol. 8383, pp. 501–519. Springer Berlin Heidelberg (2014)
- [25] Brakerski, Z.: Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012, Lecture Notes in Computer Science, vol. 7417, pp. 868–886. Springer Berlin Heidelberg (2012)
- [26] Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) Fully Homomorphic Encryption Without Bootstrapping. In: ITCS 2012. pp. 309–325. ACM, New York (2012)
- [27] Brakerski, Z., Kalai, Y.T.: A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model. Cryptology ePrint Archive, Report 2010/086 (2010), <http://eprint.iacr.org/2010/086>
- [28] Brakerski, Z., Langlois, A., Peikert, C., Regev, O., Stehlé, D.: Classical Hardness of Learning with Errors. In: STOC 2013. pp. 575–584. ACM, New York (2013)
- [29] Brakerski, Z., Vaikuntanathan, V.: Efficient Fully Homomorphic Encryption from (Standard) LWE. In: FOCS 2011. pp. 97–106. IEEE (2011)
- [30] Camenisch, J., Hohenberger, S., Lysyanskaya, A.: Compact E-Cash. In: Cramer, R. (ed.) EUROCRYPT 2005. Lecture Notes in Computer Science,

- vol. 3494, pp. 302–321. Springer Berlin Heidelberg (2005)
- [31] Camenisch, J., Lysyanskaya, A.: An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. Lecture Notes in Computer Science, vol. 2045, pp. 93–118. Springer Berlin Heidelberg (2001)
- [32] Camenisch, J., Neven, G., Rückert, M.: Fully Anonymous Attribute Tokens from Lattices. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. Lecture Notes in Computer Science, vol. 7485, pp. 57–75. Springer Berlin Heidelberg (2012)
- [33] Cash, D., Hofheinz, D., Kiltz, E., Peikert, C.: Bonsai Trees, or How to Delegate a Lattice Basis. In: Gilbert, H. (ed.) EUROCRYPT 2010. Lecture Notes in Computer Science, vol. 6110, pp. 523–552. Springer Berlin Heidelberg (2010)
- [34] Cassels, J.: An Introduction to the Geometry of Numbers. Springer Berlin Heidelberg (1997)
- [35] Chase, M., Lysyanskaya, A.: On Signatures of Knowledge. In: Dwork, C. (ed.) CRYPTO 2006. Lecture Notes in Computer Science, vol. 4117, pp. 78–96. Springer Berlin Heidelberg (2006)
- [36] Chaum, D., van Heyst, E.: Group Signatures. In: Davies, D. (ed.) EUROCRYPT 1991. Lecture Notes in Computer Science, vol. 547, pp. 257–265. Springer Berlin Heidelberg (1991)
- [37] Chen, J., Lim, H.W., Ling, S., Wang, H., Nguyen, K.: Revocable Identity-Based Encryption from Lattices. In: Susilo, W., Mu, Y., Seberry, J. (eds.) ACISP 2012. Lecture Notes in Computer Science, vol. 7372, pp. 390–403. Springer Berlin Heidelberg (2012)
- [38] Cheng, S.: Dense Packings from Algebraic Number Fields and Codes.

- arXiv:1506.00419 (2015), <http://arxiv.org/abs/1506.00419>
- [39] Cheng, S.: Improvement on Asymptotic Density of Packing Families Derived from Multiplicative Lattices. *Finite Fields and Their Applications* 36, 133–150 (2015)
- [40] Cheng, S., Nguyen, K., Wang, H.: Policy-Based Signature Scheme from Lattices. *Designs, Codes and Cryptography* (2015), in press, <http://dx.doi.org/10.1007/s10623-015-0126-y>
- [41] Cheng, S., Zhang, J.: Adaptive-ID Secure Revocable Identity-Based Encryption from Lattices via Subset Difference Method. In: Lopez, J., Wu, Y. (eds.) *ISPEC 2015, Lecture Notes in Computer Science*, vol. 9065, pp. 283–297. Springer International Publishing (2015)
- [42] Cocks, C.: An Identity Based Encryption Scheme Based on Quadratic Residues. In: Honary, B. (ed.) *Cryptography and Coding 2001. Lecture Notes in Computer Science*, vol. 2260, pp. 360–363. Springer Berlin Heidelberg (2001)
- [43] Conway, J., Sloane, N.: *Sphere Packings, Lattices and Groups*. Springer New York, 3rd edn. (1999)
- [44] Coppersmith, D.: Small Solutions to Polynomial Equations, and Low Exponent RSA Vulnerabilities. *Journal of Cryptology* 10(4), 233–260 (1997)
- [45] De Laat, D., De Oliveira Filho, F.M., Vallentin, F.: Upper Bounds for Packings of Spheres of Several Radii. *Forum of Mathematics, Sigma* 2, e23 (2014)
- [46] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. *SIAM Journal on Computing* 38(1), 97–139 (2008)
- [47] Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice Signatures and

- Bimodal Gaussians. In: Canetti, R., Garay, J. (eds.) CRYPTO 2013. Lecture Notes in Computer Science, vol. 8042, pp. 40–56. Springer Berlin Heidelberg (2013)
- [48] Ducas, L., Micciancio, D.: Improved Short Lattice Signatures in the Standard Model. In: Garay, J., Gennaro, R. (eds.) CRYPTO 2014. Lecture Notes in Computer Science, vol. 8616, pp. 335–352. Springer Berlin Heidelberg (2014)
- [49] Elkies, N.D.: Lattices, Linear Codes, and Invariants, Part II. Notices of the AMS 47(11), 1382–1391 (2000)
- [50] Elkies, N.D., Odlyzko, A.M., Rush, J.A.: On the Packing Densities of Superballs and Other Bodies. *Inventiones mathematicae* 105(1), 613–639 (1991)
- [51] Gama, N., Nguyen, P.Q.: Finding Short Lattice Vectors within Mordell’s Inequality. In: STOC 2008. pp. 207–216. ACM, New York (2008)
- [52] Gentry, C.: Fully Homomorphic Encryption Using Ideal Lattices. In: STOC 2009. pp. 169–178. ACM, New York (2009)
- [53] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for Hard Lattices and New Cryptographic Constructions. In: STOC 2008. pp. 197–206. ACM, New York (2008)
- [54] Goldman, J.: The Queen of Mathematics: A Historically Motivated Guide to Number Theory. A K Peters, Massachusetts (1997)
- [55] Goldreich, O., Goldwasser, S., Halevi, S.: Collision-Free Hashing from Lattice Problems. In: Goldreich, O. (ed.) Studies in Complexity and Cryptography: Miscellanea on the Interplay between Randomness and Computation, Lecture Notes in Computer Science, vol. 6650, pp. 30–39. Springer Berlin Heidelberg (2011)

- [56] Goldwasser, S., Kalai, Y.T., Peikert, C., Vaikuntanathan, V.: Robustness of the Learning with Errors Assumption. In: ICS 2010. pp. 230–240 (2010), <http://conference.itcs.tsinghua.edu.cn/ICS2010/content/papers/19.html>
- [57] Gordon, S.D., Katz, J., Vaikuntanathan, V.: A Group Signature Scheme from Lattice Assumptions. In: Abe, M. (ed.) ASIACRYPT 2010. Lecture Notes in Computer Science, vol. 6477, pp. 395–412. Springer Berlin Heidelberg (2010)
- [58] Grassl, M.: Bounds on the Minimum Distance of Linear Codes and Quantum Codes. Online available at <http://www.codetables.de> (2007), accessed on 2015-7-5
- [59] Groth, J.: Evaluating Security of Voting Schemes in the Universal Composability Framework. In: Jakobsson, M., Yung, M., Zhou, J. (eds.) ACNS 2004. Lecture Notes in Computer Science, vol. 3089, pp. 46–60. Springer Berlin Heidelberg (2004)
- [60] Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N. (ed.) EUROCRYPT 2008. Lecture Notes in Computer Science, vol. 4965, pp. 415–432. Springer Berlin Heidelberg (2008)
- [61] Kabatiansky, G., Rush, J.: Sphere Packing and Coding Theory. In: Handbook of Discrete and Computational Geometry. pp. 1355–1376. Chapman & Hall/CRC (2004)
- [62] Kawachi, A., Tanaka, K., Xagawa, K.: Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. Lecture Notes in Computer Science, vol. 5350, pp. 372–389. Springer Berlin Heidelberg (2008)
- [63] Kiltz, E., Mityagin, A., Panjwani, S., Raghavan, B.: Append-Only Signatures.

- In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. Lecture Notes in Computer Science, vol. 3580, pp. 434–445. Springer Berlin Heidelberg (2005)
- [64] Laguillaumie, F., Langlois, A., Libert, B., Stehlé, D.: Lattice-Based Group Signatures with Logarithmic Signature Size. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013. Lecture Notes in Computer Science, vol. 8270, pp. 41–61. Springer Berlin Heidelberg (2013)
- [65] Langlois, A., Ling, S., Nguyen, K., Wang, H.: Lattice-Based Group Signature Scheme with Verifier-Local Revocation. In: Krawczyk, H. (ed.) PKC 2014. Lecture Notes in Computer Science, vol. 8383, pp. 345–361. Springer Berlin Heidelberg (2014)
- [66] Lee, K., Lee, D.H., Park, J.H.: Efficient Revocable Identity-Based Encryption via Subset Difference Methods. Cryptology ePrint Archive, Report 2014/132 (2014), <http://eprint.iacr.org/2014/132>
- [67] Lenstra, A., Lenstra, H.W., Jr., Lovász, L.: Factoring Polynomials with Rational Coefficients. *Mathematische Annalen* 261(4), 515–534 (1982)
- [68] Libert, B., Vergnaud, D.: Adaptive-ID Secure Revocable Identity-Based Encryption. In: Fischlin, M. (ed.) CT-RSA 2009. Lecture Notes in Computer Science, vol. 5473, pp. 1–15. Springer Berlin Heidelberg (2009)
- [69] Lindner, R., Peikert, C.: Better Key Sizes (and Attacks) for LWE-Based Encryption. In: Kiayias, A. (ed.) CT-RSA 2011, Lecture Notes in Computer Science, vol. 6558, pp. 319–339. Springer Berlin Heidelberg (2011)
- [70] Ling, S., Nguyen, K., Stehlé, D., Wang, H.: Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications. In: Kurosawa, K.,

- Hanaoka, G. (eds.) PKC 2013. Lecture Notes in Computer Science, vol. 7778, pp. 107–124. Springer Berlin Heidelberg (2013)
- [71] Ling, S., Nguyen, K., Wang, H.: Group Signatures from Lattices: Simpler, Tighter, Shorter, Ring-Based. In: Katz, J. (ed.) PKC 2015. Lecture Notes in Computer Science, vol. 9020, pp. 427–449. Springer Berlin Heidelberg (2015)
- [72] Lyubashevsky, V.: Lattice-Based Identification Schemes Secure Under Active Attacks. In: Cramer, R. (ed.) PKC 2008, Lecture Notes in Computer Science, vol. 4939, pp. 162–179. Springer Berlin Heidelberg (2008)
- [73] Lyubashevsky, V.: Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. Lecture Notes in Computer Science, vol. 5912, pp. 598–616. Springer Berlin Heidelberg (2009)
- [74] Lyubashevsky, V.: Lattice Signatures without Trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 738–755. Springer Berlin Heidelberg (2012)
- [75] Lyubashevsky, V., Micciancio, D.: Asymptotically Efficient Lattice-Based Digital Signatures. In: Canetti, R. (ed.) TCC 2008. Lecture Notes in Computer Science, vol. 4948, pp. 37–54. Springer Berlin Heidelberg (2008)
- [76] Lyubashevsky, V., Micciancio, D.: On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem. In: Halevi, S. (ed.) CRYPTO 2009, Lecture Notes in Computer Science, vol. 5677, pp. 577–594. Springer Berlin Heidelberg (2009)
- [77] MacWilliams, F., Sloane, N.: The Theory of Error-correcting Codes. North-Holland Mathematical Library, North-Holland Publishing Company (1977)

- [78] Maji, H.K., Prabhakaran, M., Rosulek, M.: Attribute-Based Signatures. In: Kiayias, A. (ed.) CT-RSA 2011. Lecture Notes in Computer Science, vol. 6558, pp. 376–392. Springer Berlin Heidelberg (2011)
- [79] Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: A Cryptographic Perspective, The Kluwer International Series in Engineering and Computer Science, vol. 671. Kluwer Academic Publishers, Boston, Massachusetts (2002)
- [80] Micciancio, D., Peikert, C.: Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. Lecture Notes in Computer Science, vol. 7237, pp. 700–718. Springer Berlin Heidelberg (2012)
- [81] Micciancio, D., Regev, O.: Worst-Case to Average-Case Reductions Based on Gaussian Measures. SIAM Journal on Computing 37(1), 267–302 (2007)
- [82] Micciancio, D., Vadhan, S.P.: Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In: Boneh, D. (ed.) CRYPTO 2003, Lecture Notes in Computer Science, vol. 2729, pp. 282–298. Springer Berlin Heidelberg (2003)
- [83] Naor, D., Naor, M., Lotspiech, J.: Revocation and Tracing Schemes for Stateless Receivers. In: Kilian, J. (ed.) CRYPTO 2001. Lecture Notes in Computer Science, vol. 2139, pp. 41–62. Springer Berlin Heidelberg (2001)
- [84] Neukirch, J.: Algebraic Number Theory, Grundlehren der mathematischen Wissenschaften : a series of comprehensive studies in mathematics, vol. 322. Springer-Verlag Berlin Heidelberg (1999)
- [85] Nguyen, M.H., Ong, S.J., Vadhan, S.P.: Statistical Zero-Knowledge Argu-

- ments for NP from Any One-Way Function. In: FOCS 2006. pp. 3–14. IEEE (2006)
- [86] Nguyen, P.Q.: Hermite’s Constant and Lattice Algorithms. In: Nguyen, P.Q., Vallée, B. (eds.) *The LLL Algorithm - Survey and Applications*, pp. 19–69. *Information Security and Cryptography*, Springer Berlin Heidelberg (2010)
- [87] Nguyen, P.Q., Zhang, J., Zhang, Z.: Simpler Efficient Group Signatures from Lattices. In: Katz, J. (ed.) *PKC 2015. Lecture Notes in Computer Science*, vol. 9020, pp. 401–426. Springer Berlin Heidelberg (2015)
- [88] Odlyzko, A.M.: Bounds for Discriminants and Related Estimates for Class Numbers, Regulators and Zeros of Zeta Functions : A Survey of Recent Results. *Journal de théorie des nombres de Bordeaux* 2(1), 119–141 (1990)
- [89] Odlyzko, A.M.: The Rise and Fall of Knapsack Cryptosystems. In: Pomerance, C. (ed.) *Cryptology and Computational Number Theory. Proceedings of Symposia in Applied Mathematics*, vol. 42, pp. 75–88. American Mathematical Society (1990)
- [90] Peikert, C.: Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem: Extended Abstract. In: *STOC 2009*. pp. 333–342. ACM, New York (2009)
- [91] Peikert, C., Vaikuntanathan, V., Waters, B.: A Framework for Efficient and Composable Oblivious Transfer. In: Wagner, D. (ed.) *CRYPTO 2008, Lecture Notes in Computer Science*, vol. 5157, pp. 554–571. Springer Berlin Heidelberg (2008)
- [92] Peikert, C., Waters, B.: Lossy Trapdoor Functions and Their Applications. In: *STOC 2008*. pp. 187–196. ACM, New York (2008)

- [93] Pointcheval, D., Vaudenay, S.: On Provable Security for Digital Signature Algorithms. Technical Report LIENS-96-17, Laboratoire d'Informatique de l'Ecole Normale Supérieure (1996)
- [94] Regev, O.: On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. In: STOC 2005. pp. 84–93. ACM, New York (2005)
- [95] Rivest, R.L., Shamir, A., Tauman, Y.: How to Leak a Secret. In: Boyd, C. (ed.) ASIACRYPT 2001. Lecture Notes in Computer Science, vol. 2248, pp. 552–565. Springer Berlin Heidelberg (2001)
- [96] Rosenbloom, M.Y., Tsfasman, M.A.: Multiplicative Lattices in Global Fields. *Inventiones mathematicae* 101, 687–696 (1990)
- [97] Rückert, M.: Lattice-Based Blind Signatures. In: Abe, M. (ed.) ASIACRYPT 2010. Lecture Notes in Computer Science, vol. 6477, pp. 413–430. Springer Berlin Heidelberg (2010)
- [98] Sahai, A., Seyalioglu, H., Waters, B.: Dynamic Credentials and Ciphertext Delegation for Attribute-Based Encryption. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012, Lecture Notes in Computer Science, vol. 7417, pp. 199–217. Springer Berlin Heidelberg (2012)
- [99] Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 457–473. Springer Berlin Heidelberg (2005)
- [100] Schnorr, C.: A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Theoretical Computer Science* 53, 201–224 (1987)
- [101] Seo, J.H., Emura, K.: Revocable Identity-Based Cryptosystem Revisited: Security Models and Constructions. *IEEE Transactions on Information Forensics*

- and Security 9(7), 1193–1205 (2014)
- [102] Shamir, A.: Identity-Based Cryptosystems and Signature Schemes. In: Blakeley, G.R., Chaum, D. (eds.) CRYPTO 1984. Lecture Notes in Computer Science, vol. 196, pp. 47–53. Springer Berlin Heidelberg (1985)
- [103] Shoup, V.: A Computational Introduction to Number Theory and Algebra. Cambridge University Press (2008)
- [104] Stern, J.: A New Paradigm for Public Key Identification. IEEE Transactions on Information Theory 42(6), 1757–1768 (1996)
- [105] Stichtenoth, H.: Algebraic Function Fields and Codes, Graduate Texts in Mathematics, vol. 254. Springer Berlin Heidelberg (2009)
- [106] Tsfasman, M., Vlăduț, S.: Algebraic-Geometric Codes. Kluwer Academic Publishers, Norwell (1991)
- [107] Waters, B.: Efficient Identity-Based Encryption Without Random Oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 114–127. Springer Berlin Heidelberg (2005)
- [108] Waters, B.: Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In: Halevi, S. (ed.) CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 619–636. Springer Berlin Heidelberg (2009)
- [109] Xing, C.: Dense Packings from Quadratic Fields and Codes. Journal of Combinatorial Theory, Series A 115(6), 1021–1035 (2008)
- [110] Zong, C.: Sphere Packings. Universitext, Springer New York (1999)

APPENDIX

A. Magma CODE FOR EXAMPLE 4.2.8

```
R:=RealField(10); Q:=RationalField();
W<x>:=PolynomialRing(Q); f:=x^4-x^3-x^2+x+1;

      /*Input a polynomial over rational number field Q*/
if IsIrreducible(f) then /* Test whether f is irreducible*/
    K<a>:=NumberField([f]); O:=MaximalOrder(K); m:=AbsoluteDegree(K);
    printf"The degree of K=Q[x]/(%o) is m=%o;\n",f,m;
    d:=AbsoluteDiscriminant(K);
    printf"The absolute value of the discriminant of K \
    is |d|=%o;\n",d;

p:=3; /*Choose p=3 as a base prime number*/
J:=Decomposition(O,p); P:=J[1,1]; q:=Norm(P);
printf"P is a %o lying over %o with norm q=%o;\n",P,p,q;

printf"\nFinite Concatenation:\n";
n:=64; l:=Floor(m/2*Log(n)/Log(q));

      /*Set the length of the codes*/
L:=MinkowskiLattice(P^l); b:=Minimum(L);
printf"We can concatenate at most %o linear codes to \
the lattice constructed by %o copies of L_{P^%o},\n \
```

```

whose Hamming weights are required respectively at least \
\n",l,n,l;

for t in [0..l-1] do
    Z:=MinkowskiLattice(P^t); h:=Ceiling(Minimum(L)/Minimum(Z));
    printf"%o;",h;
end for;
printf"\n";

printf"Refer to the Codetable.de to get the largest dimension \
of the corresponding linear codes with length %o; \n",n;
fromtable:=135;
printf"Here the sum of dimensions T=%o;\n",fromtable;
    /*For this example, the sum of the dimensions is 135*/
c:=(m*n*Log(b^0.5)-n*l*Log(q)-0.5*n*Log(d))/Log(2)-m*n;
printf"Our packing is in dimension %o with \
Log_2(center density) at least\n %o\n",n*m,\
c+Log(q)/Log(2)*fromtable;

printf"\nAsymptotic result:\n";
l:=1000; n:=Ceiling(q^(2*l/m)*d^(1/m));
    /*Set l sufficiently large to approximate the limit*/
L:=MinkowskiLattice(P^l);b:=Minimum(L);
Sum:=0;
for t in [l-1..0 by -1] do
    Z:=MinkowskiLattice(P^t);

```

```

g:=Ceiling(Minimum(L)/Minimum(Z));
varrho:=g/n;
if varrho le (q-1)/q then
    Sum:=Sum+(varrho*Log(q-1)-varrho*Log(varrho)\
    -(1-varrho)*Log(1-varrho))/Log(q);
else Sum:=Sum+1;
end if;
end for;

Lambda:=-1-1/(2*m)*Log(d)/Log(2)-0.5*Log(m/(2*Pi(R)*Exp(1))) \
/Log(2)+Log(b^0.5)/Log(2)-0.5*Log(n)/Log(2) \
-1/m*Log(q)/Log(2)*Sum;

printf"The asymptotic density exponent of the packing family is \
\n at least %o",Lambda;

else
printf"The polynomial %o is not irreducible over Q.\n",f;
end if;

```

B. PROOF OF LEMMA 5.3.3

The simulator $\mathcal{Sim}$ begins by selecting a random $\overline{\mathbf{Ch}} \in \{1, 2, 3\}$, and a random tape r' of $\widetilde{V}$.

Case $\overline{\mathbf{Ch}} = 1$: The simulator works as follows:

1. Compute $\mathbf{z}'_1, \dots, \mathbf{z}'_k \in \mathbb{Z}_q^{(2\ell+1) \cdot 3m}$ such that $\mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i \mathbf{z}'_i \right) = \mathbf{u} \bmod q$.
2. Compute $\overline{\mathbf{p}} \in \{0, 1\}^\ell$ and $\overline{\mathbf{w}}^* \in \mathbb{Z}_2^{2d}$ such that $\mathbf{G}_1 \cdot \overline{\mathbf{p}} + \mathbf{G}_2^* \cdot \overline{\mathbf{w}}^* = \mathbf{M} \bmod 2$.

As we set the parameter d to satisfy $d + \ell > n$, it is feasible to compute $\overline{\mathbf{p}}$ and $\overline{\mathbf{w}}^*$. Then set $\overline{\mathbf{p}}^* = \text{PolicyExt}(\overline{\mathbf{p}})$.

3. Sample

$$\begin{cases} \rho'_1, \rho'_2, \rho'_3 \xleftarrow{\$} \{0, 1\}^n; \mathbf{e} \xleftarrow{\$} \{0, 1\}^\ell; \mathbf{r}_{\mathbf{p}} \xleftarrow{\$} \mathbb{Z}_2^{2\ell+1}; \\ \mathbf{r}_{\mathbf{z}}^{(1)}, \dots, \mathbf{r}_{\mathbf{z}}^{(k)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1) \cdot 3m}; \mathbf{r}_{\mathbf{w}} \xleftarrow{\$} \mathbb{Z}_2^{2d}; \\ \pi_1, \dots, \pi_k \xleftarrow{\$} \mathcal{S}; \eta \xleftarrow{\$} \mathbb{S}_{2d}, \end{cases}$$

and send the commitment $\mathbf{CMT} = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to the verifier, where

$$\begin{cases} \mathbf{c}'_1 = \text{COM}\left(\mathbf{e}, \{\pi_i\}_{i=1}^k, \eta, \mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i \mathbf{r}_{\mathbf{z}}^{(i)} \right) \bmod q, \right. \\ \quad \left. \mathbf{G}_1^* \cdot \mathbf{r}_{\mathbf{p}} + \mathbf{G}_2^* \cdot \mathbf{r}_{\mathbf{w}} \bmod 2; \rho'_1 \right) \\ \mathbf{c}'_2 = \text{COM}\left(\left\{T_{\mathbf{e}} \circ \pi_i(\mathbf{r}_{\mathbf{z}}^{(i)})\right\}_{i=1}^k, \eta(\mathbf{r}_{\mathbf{w}}), T'_{\mathbf{e}}(\mathbf{r}_{\mathbf{p}}); \rho'_2\right) \\ \mathbf{c}'_3 = \text{COM}\left(\left\{T_{\mathbf{e}} \circ \pi_i(\mathbf{z}'_i + \mathbf{r}_{\mathbf{z}}^{(i)})\right\}_{i=1}^k, \eta(\overline{\mathbf{w}}^* \oplus \mathbf{r}_{\mathbf{w}}), T'_{\mathbf{e}}(\overline{\mathbf{p}}^* \oplus \mathbf{r}_{\mathbf{p}}); \rho'_3\right) \end{cases}$$

Receiving a challenge from $\tilde{V}$:

- If $\text{Ch} = 1$: Output $\perp$ and abort.
- If $\text{Ch} = 2$: Set

$$\text{RSP} = \left(\mathbf{e}, \{\pi_i\}_{i=1}^k, \eta, \bar{\mathbf{p}}^* \oplus \mathbf{r}_{\mathbf{p}}, \{\mathbf{z}'_i + \mathbf{r}_{\mathbf{z}}^{(i)} \bmod q\}_{i=1}^k, \bar{\mathbf{w}}^* \oplus \mathbf{r}_{\mathbf{w}} \right),$$

and output $(r'; \text{CMT}; 2; \text{RSP}; \rho'_1, \rho'_3)$.

- If $\text{Ch} = 3$: Set

$$\text{RSP} = \left(\mathbf{e}, \{\pi_i\}_{i=1}^k, \eta, \mathbf{r}_{\mathbf{p}}, \{\mathbf{r}_{\mathbf{z}}^{(i)}\}_{i=1}^k, \mathbf{r}_{\mathbf{w}} \right),$$

and output $(r'; \text{CMT}; 3; \text{RSP}; \rho'_1, \rho'_2)$.

Case $\overline{\text{Ch}} = 2$: The simulator samples

$$\begin{cases} \rho'_1, \rho'_2, \rho'_3 \xleftarrow{\$} \{0, 1\}^n; \bar{\mathbf{p}}, \mathbf{e} \xleftarrow{\$} \{0, 1\}^\ell; \mathbf{r}_{\mathbf{p}} \xleftarrow{\$} \mathbb{Z}_q^{2\ell+1}; \\ \mathbf{z}'_1, \dots, \mathbf{z}'_k \xleftarrow{\$} \text{SecretExt}(\bar{\mathbf{p}}); \mathbf{r}_{\mathbf{z}}^{(1)}, \dots, \mathbf{r}_{\mathbf{z}}^{(k)} \xleftarrow{\$} \mathbb{Z}_q^{(2\ell+1) \cdot 3m}; \bar{\mathbf{w}}^*, \mathbf{r}_{\mathbf{w}} \xleftarrow{\$} \mathbb{Z}_2^{2d}; \\ \pi_1, \dots, \pi_k \xleftarrow{\$} \mathcal{S}; \eta \xleftarrow{\$} S_{2d}. \end{cases}$$

Set $\bar{\mathbf{p}}^* = \text{PolicyExt}(\bar{\mathbf{p}})$ and send the commitment $\text{CMT} = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to the verifier, where

$$\begin{cases} \mathbf{c}'_1 = \text{COM}\left(\mathbf{e}, \{\pi_i\}_{i=1}^k, \eta, \mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i \mathbf{r}_{\mathbf{z}}^{(i)}\right) \bmod q, \right. \\ \quad \left. \mathbf{G}_1^* \cdot \mathbf{r}_{\mathbf{p}} + \mathbf{G}_2^* \cdot \mathbf{r}_{\mathbf{w}} \bmod 2; \rho'_1\right) \\ \mathbf{c}'_2 = \text{COM}\left(\left\{T_{\mathbf{e}} \circ \pi_i(\mathbf{r}_{\mathbf{z}}^{(i)})\right\}_{i=1}^k, \eta(\mathbf{r}_{\mathbf{w}}), T'_{\mathbf{e}}(\mathbf{r}_{\mathbf{p}}); \rho'_2\right) \\ \mathbf{c}'_3 = \text{COM}\left(\left\{T_{\mathbf{e}} \circ \pi_i(\mathbf{z}'_i + \mathbf{r}_{\mathbf{z}}^{(i)})\right\}_{i=1}^k, \eta(\bar{\mathbf{w}}^* \oplus \mathbf{r}_{\mathbf{w}}), T'_{\mathbf{e}}(\bar{\mathbf{p}}^* \oplus \mathbf{r}_{\mathbf{p}}); \rho'_3\right) \end{cases}$$

Receiving a challenge from $\tilde{V}$:

- If Ch = 1: Set RSP as

$$\left(\bar{\mathbf{p}} \oplus \mathbf{e}, T'_e(\bar{\mathbf{p}}^*), T'_e(\mathbf{r}_p), \{T_e \circ \pi_i(\mathbf{z}'_i)\}_{i=1}^k, \{T_e \circ \pi_i(\mathbf{r}_z^{(i)})\}_{i=1}^k, \eta(\bar{\mathbf{w}}^*), \eta(\mathbf{r}_w) \right),$$

and output $(r'; \text{CMT}; 1; \text{RSP}; \rho'_2, \rho'_3)$.

- If Ch = 2: Output $\perp$ and abort.
- If Ch = 3: Set

$$\text{RSP} = \left(\mathbf{e}, \{\pi_i\}_{i=1}^k, \eta, \mathbf{r}_p, \{\mathbf{r}_z^{(i)}\}_{i=1}^k, \mathbf{r}_w \right),$$

and output $(r'; \text{CMT}; 3; \text{RSP}; \rho'_1, \rho'_2)$.

Case $\overline{\text{Ch}} = 3$: The simulator proceeds with the sampling as in the case $\overline{\text{Ch}} = 2$ above.

Then send the commitment $\text{CMT} = (\mathbf{c}'_1, \mathbf{c}'_2, \mathbf{c}'_3)$ to the verifier, where

$$\begin{cases} \mathbf{c}'_1 = \text{COM} \left(\mathbf{e}, \{\pi_i\}_{i=1}^k, \eta, \mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i (\mathbf{z}'_i + \mathbf{r}_z^{(i)}) \right) - \mathbf{u} \bmod q, \right. \\ \quad \left. \mathbf{G}_1^* \cdot (\bar{\mathbf{p}}^* \oplus \mathbf{r}_p) + \mathbf{G}_2^* \cdot (\bar{\mathbf{w}}^* \oplus \mathbf{r}_w) - \mathbf{M} \bmod 2; \rho'_1 \right) \\ \mathbf{c}'_2 = \text{COM} \left(\left\{ T_e \circ \pi_i(\mathbf{r}_z^{(i)}) \right\}_{i=1}^k, \eta(\mathbf{r}_w), T'_e(\mathbf{r}_p); \rho'_2 \right) \\ \mathbf{c}'_3 = \text{COM} \left(\left\{ T_e \circ \pi_i(\mathbf{z}'_i + \mathbf{r}_z^{(i)}) \right\}_{i=1}^k, \eta(\bar{\mathbf{w}}^* \oplus \mathbf{r}_w), T'_e(\bar{\mathbf{p}}^* \oplus \mathbf{r}_p); \rho'_3 \right) \end{cases}$$

Receiving a challenge from $\tilde{V}$:

- If Ch = 1: Set RSP as

$$\left(\bar{\mathbf{p}} \oplus \mathbf{e}, T'_e(\bar{\mathbf{p}}^*), T'_e(\mathbf{r}_p), \{T_e \circ \pi_i(\mathbf{z}'_i)\}_{i=1}^k, \{T_e \circ \pi_i(\mathbf{r}_z^{(i)})\}_{i=1}^k, \eta(\bar{\mathbf{w}}^*), \eta(\mathbf{r}_w) \right),$$

and output $(r'; \text{CMT}; 1; \text{RSP}; \rho'_2, \rho'_3)$.

- If $\text{Ch} = 2$: Set

$$\text{RSP} = \left(\mathbf{e}, \{\pi_i\}_{i=1}^k, \eta, \bar{\mathbf{p}}^* \oplus \mathbf{r}_{\mathbf{p}}, \{\mathbf{z}'_i + \mathbf{r}_{\mathbf{z}}^{(i)} \bmod q\}_{i=1}^k, \bar{\mathbf{w}}^* \oplus \mathbf{r}_{\mathbf{w}} \right),$$

and output $(r'; \text{CMT}; 2; \text{RSP}; \rho'_1, \rho'_3)$.

- If $\text{Ch} = 3$: Output $\perp$ and abort.

Since **COM** is statistically hiding, the distribution of the commitment **CMT** and the challenge **Ch** from $\tilde{V}$ are statistically close to those in the real interaction. Hence the simulator outputs $\perp$ with probability negligibly close to $1/3$. Moreover, if it does not halt, it is easy to prove that the simulator produces a successful transcript, and the distribution of the transcript is statistically close to that of the prover in the real interaction.

C. PROOF OF LEMMA 5.3.4

Let $\text{CMT} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3) \in (\mathbb{Z}_q^n)^3$ and let $\text{RSP}^{(1)}$, $\text{RSP}^{(2)}$, and $\text{RSP}^{(3)}$ be as in Eqs. (5.5), (5.6) and (5.7). Since all 3 responses satisfy the verification conditions, the following is true:

$$\left\{ \begin{array}{l} \mathbf{v}_p = \text{PolicyExt}(\mathbf{p}_1); \mathbf{v}_w \in \{0, 1\}^{2d}; \text{wt}(\mathbf{v}_w) = d; \forall i \in [k] : \mathbf{v}_z^{(i)} \in \text{SecretExt}(\mathbf{p}_1) \\ \mathbf{c}_1 = \text{COM}(\mathbf{p}_2, \{\phi_i\}_{i=1}^k, \rho, \mathbf{A}^*(\sum_{i=1}^k \beta_i \mathbf{t}_z^{(i)}) - \mathbf{u} \bmod q, \mathbf{G}_1^* \cdot \mathbf{t}_p + \mathbf{G}_2^* \cdot \mathbf{t}_w - \mathbf{M} \bmod 2) \\ \quad = \text{COM}(\mathbf{p}_3, \{\psi_i\}_{i=1}^k, \theta, \mathbf{A}^*(\sum_{i=1}^k \beta_i \mathbf{h}_z^{(i)}) \bmod q, \mathbf{G}_1^* \cdot \mathbf{h}_p + \mathbf{G}_2^* \cdot \mathbf{h}_w \bmod 2) \\ \mathbf{c}_2 = \text{COM}(\{\mathbf{y}_z^{(i)}\}_{i=1}^k, \mathbf{y}_w, \mathbf{y}_p) = \text{COM}(\{T_{\mathbf{p}_3} \circ \psi_i(\mathbf{h}_z^{(i)})\}_{i=1}^k, \theta(\mathbf{h}_w), T'_{\mathbf{p}_3}(\mathbf{h}_p)) \\ \mathbf{c}_3 = \text{COM}(\{\mathbf{v}_z^{(i)} + \mathbf{y}_z^{(i)}\}_{i=1}^k, \mathbf{v}_w \oplus \mathbf{y}_w, \mathbf{v}_p \oplus \mathbf{y}_p) \\ \quad = \text{COM}(\{T_{\mathbf{p}_2} \circ \phi_i(\mathbf{t}_z^{(i)})\}_{i=1}^k, \rho(\mathbf{t}_w), T'_{\mathbf{p}_2}(\mathbf{t}_p)) \end{array} \right.$$

Since the commitment scheme COM is computationally binding, one can deduce that $\mathbf{p}_2 = \mathbf{p}_3$, $\rho = \theta$, $\phi_i = \psi_i$ for all $i \in [k]$, and that

$$\left\{ \begin{array}{l} \mathbf{y}_{\mathbf{p}} = T'_{\mathbf{p}_2}(\mathbf{h}_{\mathbf{p}}); \mathbf{y}_{\mathbf{p}} \oplus \mathbf{v}_{\mathbf{p}} = T'_{\mathbf{p}_2}(\mathbf{t}_{\mathbf{p}}); \\ \mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i (\mathbf{t}_{\mathbf{z}}^{(i)} - \mathbf{h}_{\mathbf{z}}^{(i)}) \right) = \mathbf{u} \bmod q; \\ \mathbf{G}_1^* \cdot (\mathbf{t}_{\mathbf{p}} \oplus \mathbf{h}_{\mathbf{p}}) + \mathbf{G}_2^* \cdot (\mathbf{t}_{\mathbf{w}} \oplus \mathbf{h}_{\mathbf{w}}) = \mathbf{M} \bmod 2; \\ \mathbf{y}_{\mathbf{w}} = \rho(\mathbf{h}_{\mathbf{w}}); \mathbf{v}_{\mathbf{w}} \oplus \mathbf{y}_{\mathbf{w}} = \rho(\mathbf{t}_{\mathbf{w}}); \\ \forall i \in [k] : \mathbf{y}_{\mathbf{z}}^{(i)} = T_{\mathbf{p}_2} \circ \phi_i(\mathbf{h}_{\mathbf{z}}^{(i)}); \\ \mathbf{y}_{\mathbf{z}}^{(i)} + \mathbf{v}_{\mathbf{z}}^{(i)} = T_{\mathbf{p}_2} \circ \phi_i(\mathbf{t}_{\mathbf{z}}^{(i)}). \end{array} \right.$$

The extractor $\mathcal{K}$ proceeds as follows:

- Extract $\mathbf{p}^* = \mathbf{t}_{\mathbf{p}} \oplus \mathbf{h}_{\mathbf{p}}$. From $\text{PolicyExt}(\mathbf{p}_1) = \mathbf{v}_{\mathbf{p}} = T'_{\mathbf{p}_2}(\mathbf{t}_{\mathbf{p}} \oplus \mathbf{h}_{\mathbf{p}}) = T'_{\mathbf{p}_2}(\mathbf{p}^*)$ and the equivalence Eq. (5.4), get $\mathbf{p}^* = \text{PolicyExt}(\mathbf{p}_1 \oplus \mathbf{p}_2)$. As $\mathbf{d} \mapsto \text{PolicyExt}(\mathbf{d})$ is a bijection between the set $\{0, 1\}^\ell$ and the set $\{\text{PolicyExt}(\mathbf{d}) : \mathbf{d} \in \{0, 1\}^\ell\}$, extract uniquely $\mathbf{p}' = \mathbf{p}_1 \oplus \mathbf{p}_2$ from $\mathbf{p}^*$. Note that the vectors $\mathbf{p}', \mathbf{p}^*$ clearly satisfy $\mathbf{p}^* = \text{PolicyExt}(\mathbf{p}')$.
- Extract $\mathbf{w}^* = \mathbf{t}_{\mathbf{w}} \oplus \mathbf{h}_{\mathbf{w}}$. From $\mathbf{v}_{\mathbf{w}} \in \{0, 1\}^{2d}$, $wt(\mathbf{v}_{\mathbf{w}}) = d$, $\mathbf{v}_{\mathbf{w}} = \rho(\mathbf{t}_{\mathbf{w}} \oplus \mathbf{h}_{\mathbf{w}})$, get $\mathbf{w}^* \in \{0, 1\}^{2d}$, $wt(\mathbf{w}^*) = d$. Parse $\mathbf{w}^*$ as $(\mathbf{w}', \mathbf{w}'')$, where $\mathbf{w}', \mathbf{w}'' \in \{0, 1\}^d$. Note that from

$$\mathbf{G}_1^* \cdot (\mathbf{t}_{\mathbf{p}} \oplus \mathbf{h}_{\mathbf{p}}) + \mathbf{G}_2^* \cdot (\mathbf{t}_{\mathbf{w}} \oplus \mathbf{h}_{\mathbf{w}}) = \mathbf{M} \bmod 2$$

and the structure of $\mathbf{G}_1^*, \mathbf{G}_2^*$, the vector $\mathbf{w}'$ satisfies $\mathbf{G}_1 \cdot \mathbf{p}' + \mathbf{G}_2 \cdot \mathbf{w}' = \mathbf{M} \bmod 2$.

- From

$$\mathbf{A}^* \cdot \left(\sum_{i=1}^k \beta_i (\mathbf{t}_{\mathbf{z}}^{(i)} - \mathbf{h}_{\mathbf{z}}^{(i)}) \right) = \mathbf{u} \bmod q,$$

extract $\mathbf{z}^* = \sum_{i=1}^k \beta_i \tilde{\mathbf{w}}_i$, where for each $i \in [k]$, $\tilde{\mathbf{w}}_i$ is obtained as follows:

- Write the vector $\hat{\mathbf{w}}_i = \mathbf{t}_z^{(i)} - \mathbf{h}_z^{(i)} \in \{-1, 0, 1\}^{(2\ell+1) \cdot 3m}$ as $2\ell + 1$ blocks of length $3m$.
- Delete the last $2m$ coordinates in each block of $\hat{\mathbf{w}}_i$.

Note that from $\mathbf{A}^* = \text{MatrixExt}(\mathbf{A})$, the vector $\mathbf{z}^*$ satisfies $\mathbf{A} \cdot \mathbf{z}^* = \mathbf{u} \bmod q$. Moreover, from $\text{SecretExt}(\mathbf{p}_1) \ni \mathbf{v}_z^{(i)} = T_{\mathbf{p}_2} \circ \phi_i \left(\mathbf{t}_z^{(i)} - \mathbf{h}_z^{(i)} \right) = T_{\mathbf{p}_2} \circ \phi_i \left(\hat{\mathbf{w}}_i \right)$ and $\phi_i \in \mathcal{S}$, by the equivalence Eq. (5.3), the vector $\hat{\mathbf{w}}_i$ satisfies $\hat{\mathbf{w}}_i \in \text{SecretExt}(\mathbf{p}_1 \oplus \mathbf{p}_2) = \text{SecretExt}(\mathbf{p}')$.

Furthermore, $\|\mathbf{z}^*\|_\infty \leq \sum_{i=1}^k \beta_i \cdot \|\tilde{\mathbf{w}}_i\|_\infty \leq \sum_{i=1}^k \beta_i = \beta$, thus $\mathbf{z}^* \in \text{Secret}_\beta(\mathbf{p}')$, where $\text{Secret}_\beta(\mathbf{p})$ is the set of all vectors

$$\mathbf{x} = (\mathbf{x}_0 \| \mathbf{x}_1^0 \| \mathbf{x}_1^1 \| \cdots \| \mathbf{x}_\ell^0 \| \mathbf{x}_\ell^1) \in \mathbb{Z}^{(2\ell+1) \cdot m}$$

having $2\ell + 1$ blocks of size m , such that $\|\mathbf{x}\|_\infty \leq \beta$, and the following ℓ blocks are zero-blocks $\mathbf{0}^m$: $\mathbf{x}_1^{1-\mathbf{p}[1]}, \dots, \mathbf{x}_\ell^{1-\mathbf{p}[\ell]}$. Deleting the zero blocks, extract $\mathbf{z}' \in \mathbb{Z}^{(\ell+1)m}$ from $\mathbf{z}^* \in \mathbb{Z}^{(2\ell+1)m}$. Then the vector $\mathbf{z}'$ satisfies $\|\mathbf{z}'\|_\infty \leq \beta$ and $\mathbf{A}_{\mathbf{p}'} \cdot \mathbf{z}' = \mathbf{u} \bmod q$.

From the discussion above, the output $(\mathbf{p}', \mathbf{z}', \mathbf{w}')$ of the extractor $\mathcal{K}$ satisfies the requirement in R_{PBS} (see Definition 5.3.2).

D. AN IMPROVED FORKING LEMMA

In [93], Pointcheval and Vaudenay introduced an improved forking lemma for multiple forks. We state an adaptation of their result to the 3-fork case as follows.

Lemma D.0.2. *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 parameters, and which can ask $q_{\mathcal{H}}$ queries to the random oracle, where $q_{\mathcal{H}} > 0$. Assume that, within the time bound T , the algorithm $\mathcal{A}$ produces, with probability ϵ , a valid signature $(\{CMT^{(i)}\}_{i=1}^t, \{Ch^{(i)}\}_{i=1}^t, \{RSP^{(i)}\}_{i=1}^t)$ of the message $\mathbf{M}$. Then, within time $32 \cdot T \cdot q_{\mathcal{H}} / \epsilon$, and with probability $\epsilon' > 1/2$, a replay of $\mathcal{A}$ can output 3 valid signatures of the message $\mathbf{M}$:*

$$\left(\{CMT^{(i)}\}_{i=1}^t, \{Ch_1^{(i)}\}_{i=1}^t, \{RSP_1^{(i)}\}_{i=1}^t \right); \left(\{CMT^{(i)}\}_{i=1}^t, \{Ch_2^{(i)}\}_{i=1}^t, \{RSP_2^{(i)}\}_{i=1}^t \right);$$

$$\left(\{CMT^{(i)}\}_{i=1}^t, \{Ch_3^{(i)}\}_{i=1}^t, \{RSP_3^{(i)}\}_{i=1}^t \right),$$

such that the vectors $(Ch_1^{(1)}, \dots, Ch_1^{(t)})$, $(Ch_2^{(1)}, \dots, Ch_2^{(t)})$, $(Ch_3^{(1)}, \dots, Ch_3^{(t)})$ are pairwise distinct.

E. SECURITY ANALYSIS OF OUR RIBE SCHEME

Our RIBE scheme defined in Subsection 6.4.1 is indistinguishable from random under adaptive identity-time attacks provided that the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$ -LWE assumption holds. In particular, we have the following theorem.

Theorem E.0.3. *Under the condition that the number Q^{id} of private key queries and the size $|\mathcal{T}|$ of time space satisfy $2(Q^{id} + |\mathcal{T}|) \leq q$, and both of them are polynomial in n , if there exists a PPT adversary $\mathcal{A}$ that wins the IND_r-RID-CPA game with advantage ϵ , then there exists a PPT algorithm $\mathcal{B}$ that can solve the $(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)$ -LWE problem with advantage $\epsilon' \geq \epsilon / (8q^2 Q^{id} \cdot |\mathcal{T}|)$, in approximately the same time as $\mathcal{A}$.*

E.1 Abort-Resistant Hash Functions

Similar as in [2], the proof of Theorem E.0.3 requires *abort-resistant hash functions* to implement abort events in indistinguishable games. We adaptively modify the definition and propositions of abort-resistant hash functions from [3].

Definition E.1.1. Let $\mathcal{H} := \{H : X \rightarrow Y\}$ be a family of hash functions from X to Y where $0 \in Y$. For a tuple of Q inputs $\bar{x} = (x_1, \dots, x_Q) \in X^Q$, the non-abort probability in terms of $\mathfrak{I}$ -place of $\bar{x}$ is defined as

$$\alpha_{\mathfrak{I}}(\bar{x}) := \Pr [H \in \mathcal{H} : E_{\mathfrak{I}}(H)],$$

where $E_{\mathfrak{J}}(H)$ is the event that given $\mathfrak{J} \subseteq \{1, \dots, Q\}$,

$$(H(x_i) = 0, \forall i \in \mathfrak{J}) \wedge (H(x_k) \neq 0, \forall k \in [Q] \setminus \mathfrak{J}), \quad (\text{E.1})$$

and the probability is computed as H is randomly chosen from $\mathcal{H}$. We say that $\mathcal{H}$ is $(Q, \alpha_{\mathfrak{J}, \min}, \alpha_{\mathfrak{J}, \max})$ *abort-resistant* in terms of $\mathfrak{J}$ -place if for all $\bar{x} \in X^Q$ with the condition $\{x_i : i \in \mathfrak{J}\} \cap \{x_k : k \in [Q] \setminus \mathfrak{J}\} = \emptyset$, we have $\alpha_{\mathfrak{J}}(\bar{x}) \in [\alpha_{\mathfrak{J}, \min}, \alpha_{\mathfrak{J}, \max}]$.

Inspired by [2, 108], the following abort-resistant hash family will be employed. Let $(\mathbb{Z}_q^\ell)^* := \mathbb{Z}_q^\ell \setminus \{0^\ell\}$ for prime q , and define the family $\mathcal{H}_{\text{Wat}} := \{H_h : (\mathbb{Z}_q^\ell)^* \rightarrow \mathbb{Z}_q\}_{h \in \mathbb{Z}_q^\ell}$ as

$$H_h(\mathbf{e}) := 1 + \sum_{i=1}^{\ell} h_i e_i \in \mathbb{Z}_q \quad (\text{E.2})$$

where $\mathbf{e} = (e_1, \dots, e_\ell) \in (\mathbb{Z}_q^\ell)^*$ and $h = (h_1, \dots, h_\ell) \in \mathbb{Z}_q^\ell$.

In our proof, the inputs of these hash functions will be only in $\{-1, 1\}^\ell$, a smaller domain than the general domain where abort-resistance holds. Furthermore, in our reduction, we need the non-abort probability in terms of $\mathfrak{J}$ -place with $|\mathfrak{J}| = 1$ or 2.

Lemma E.1.2. *Let q be a prime, $0 < Q < q$ and let $\mathfrak{J}$ be a fixed subset of $[Q]$. Then the hash family $\mathcal{H}_{\text{Wat}}$ defined in Eq. (E.2) satisfies*

1. *if $\mathfrak{J} = \{i\}$, then $\mathcal{H}_{\text{Wat}}$ is $\left(Q, \frac{1}{q} \left(1 - \frac{Q}{q}\right), \frac{1}{q}\right)$ abort-resistant in terms of $\{i\}$ -place.*
2. *if $\mathfrak{J} = \{i, j\}$, and $\mathbf{e}_i, \mathbf{e}_j$ are $\mathbb{Z}_q$ -linearly independent vectors, then $\mathcal{H}_{\text{Wat}}$ is $\left(Q, \frac{1}{q^2} \left(1 - \frac{Q}{q}\right), \frac{1}{q^2}\right)$ abort-resistant in terms of $\{i, j\}$ -place.*

Proof. The first part for $|\mathfrak{J}| = 1$ has been proved in [2]. For $\mathfrak{J} = \{i, j\}$, we consider a set of inputs $\mathbf{e}_1, \dots, \mathbf{e}_Q \in (\mathbb{Z}_q^\ell)^*$. For any $k \in [Q]$, let S_k be the set of functions in

$\mathcal{H}_{\text{Wat}}$ such that $H(\mathbf{e}_k) = 0$. As $\mathbf{e}_i, \mathbf{e}_j$ are $\mathbb{Z}_q$ -linearly independent, $|S_i \cap S_j| = q^{\ell-2}$. Moreover, for $k \notin \{i, j\}$, as $\mathbf{e}_k \neq \mathbf{e}_i$ or $\mathbf{e}_j$ and all coordinates of $\mathbf{e}_k$ are -1 or 1 , if $\mathbf{e}_k = s\mathbf{e}_i + t\mathbf{e}_j$ for $s, t \in \mathbb{Z}_q$, then one of s, t must be zero. This implies $\mathbf{e}_k = -\mathbf{e}_i$ or $-\mathbf{e}_j$, or $\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k$ are $\mathbb{Z}_q$ -linearly independent. Hence $|S_i \cap S_j \cap S_k| \leq q^{\ell-3}$. The set of functions in $\mathcal{H}_{\text{Wat}}$, such that $E_3(H)$ defined in Eq. (E.1), is exactly $S := (S_i \cap S_j) \setminus \cup_{k \neq i, j} S_k$. By the inclusion-exclusion principle, we have $|S| \geq q^{\ell-2} - Qq^{\ell-3}$. Therefore the non-abort probability in terms of (i, j) -place, which equals $|S|/q^\ell$, is at least $\left(1 - \frac{Q}{q}\right)/q^2$, and at most $|S_i \cap S_j|/q^\ell \leq q^{\ell-2}/q^\ell = 1/q^2$. $\square$

E.2 Proof of Theorem E.0.3

In this subsection we prove the security of our scheme against an adaptive adversary. In Game i , let W_i denote the event that the adversary's guess on the challenge bit is correct, i.e., $r' = r$ at the end of the guess phase. The advantage of adversary in Game i is denoted by $|\Pr[W_i] - \frac{1}{2}|$.

Game 0. This is the original IND_r-RID-CPA game from Subsection 6.1.1.

Game 1. In Game 1, the way that the challenger produces the matrices $\mathbf{B}_i, \mathbf{D}_i$, $i \in [\ell]$, is changed. The Game 1 challenger selects $\mathbf{R}_i^*$, $i \in [2\ell]$ at the setup phase and also picks 2ℓ random scalars $h_i \in \mathbb{Z}_q$ for $i \in [2\ell]$. Next it generates $\mathbf{A}, \mathbf{C}, \mathbf{G}$ as in Game 0 and generated the matrices $\mathbf{B}_i, \mathbf{D}_i$ for $i \in [\ell]$ as

$$\mathbf{B}_i \leftarrow \mathbf{A} \cdot \mathbf{R}_i^* + h_i \mathbf{C} \text{ and } \mathbf{D}_i \leftarrow \mathbf{A} \cdot \mathbf{R}_{\ell+i}^* + h_{\ell+i} \mathbf{G}. \quad (\text{E.3})$$

The remainder of the game is unchanged. Note that the knowledge of the challenge identity id^* and time $\mathbf{t}^*$ is not required for such setup.

We prove $\Pr[W_0] = \Pr[W_1]$. The $\mathbf{R}_i^*$, $i \in [2\ell]$ are only used in the generation of the matrices $\mathbf{B}_i, \mathbf{D}_i$ and the challenge ciphertext where $\mathbf{z}_{\text{id}} \leftarrow (\mathbf{R}_{\text{id}}^*)^\top \mathbf{y}$, $\mathbf{z}_{\mathbf{t}} \leftarrow (\mathbf{R}_{\mathbf{t}}^*)^\top \mathbf{y}$,

and $\mathbf{R}_{\text{id}^*}^* = \sum_{i=1}^{\ell} b_i^* \mathbf{R}_i^*$, $\mathbf{R}_{\text{t}^*}^* = \sum_{i=1}^{\ell} t_i^* \mathbf{R}_{\ell+i}^*$. Let $\mathbf{R}^* = (\mathbf{R}_1^* | \cdots | \mathbf{R}_{\ell}^*) \in \mathbb{Z}_q^{m \times \ell m}$, then by Lemma 2.4.7, the distributions

$$\begin{aligned} & \left(\mathbf{A}, \mathbf{A} \cdot \mathbf{R}^*, (\mathbf{R}^*)^\top \mathbf{y} \right) \quad \text{and} \\ & \left(\mathbf{A}, (\mathbf{B}'_1 | \cdots | \mathbf{B}'_{\ell} | \mathbf{D}'_1 | \cdots | \mathbf{D}'_{\ell}), (\mathbf{R}^*)^\top \mathbf{y} \right) \end{aligned}$$

are statistically close, where $\mathbf{B}'_i, \mathbf{D}'_i \ i \in [\ell]$ are uniformly independent matrices in $\mathbb{Z}_q^{n \times m}$. Then provided $\mathbf{z} = (\mathbf{z}_{\text{id}} || \mathbf{z}_{\text{t}})$, the distributions

$$\begin{aligned} & \left(\mathbf{A}, \mathbf{A}\mathbf{R}_1^*, \dots, \mathbf{A}\mathbf{R}_{\ell}^*, \mathbf{A}\mathbf{R}_{\ell+1}^*, \dots, \mathbf{A}\mathbf{R}_{2\ell}^*, \mathbf{z} \right) \quad \text{and} \\ & \left(\mathbf{A}, \mathbf{B}'_1, \dots, \mathbf{B}'_{\ell}, \mathbf{D}'_1, \dots, \mathbf{D}'_{\ell}, \mathbf{z} \right) \end{aligned}$$

are statistically close. Thus from the adversary's view, the matrices $\mathbf{A}\mathbf{R}_i^*$ are statistically close to uniform and independent of $\mathbf{z}$. Hence the $\mathbf{B}_i, \mathbf{D}_i$ defined in Eq. (E.3) are close to uniform, thus

$$\Pr[W_0] = \Pr[W_1]. \tag{E.4}$$

Game 2. Game 2 is the same as Game 1 except an abort event is added, which is hidden from the adversary's view. Let Q^{id} be the maximal number of private key queries, and let $|\mathcal{T}|$ be the size of time space, which are polynomial in n . Let $q \geq 2(|\mathcal{T}| + Q^{\text{id}})$. Without loss of generality, since the time space $\mathcal{T}$ has size polynomial in λ , we assume the adversary $\mathcal{A}$ queries the update keys for all time $\text{t} \in \mathcal{T}$. The challenger can return the same response to repeated queries from local records, so we may assume $(\text{t}_1, \dots, \text{t}_{|\mathcal{T}|})$ is the time queries tuple, which is listed in an ascending order since the update key cannot be queried in a descending order.

The challenger of Game 2 proceeds as follows:

1. The setup phase is the same as Game 1 but additionally the challenger picks a random hash function $\tilde{H} \in \mathcal{H}_{\text{Wat}}$ and records it to itself.
2. The challenger, exactly as in Game 1, responds to the update key queries and issues the challenge ciphertext.
3. The Game 2 challenger randomly guesses $i^* \in [|\mathcal{T}|]$ such that $\mathcal{A}$'s i^* -th update key query is for time $\mathbf{t}^*$. The challenger guesses it correctly with probability $1/|\mathcal{T}|$.
4. The Game 2 challenger guesses an adversary type from the two following types:
 - i. Type-1 adversary: $\mathcal{A}$ does not query sk_{id^*} .
 - ii. Type-2 adversary: $\mathcal{A}$ makes private key query for the challenge identity id^* , so id^* should be revoked before time $\mathbf{t}^*$.

The challenger guesses it correctly with probability $1/2$.

If the guess is Type-2, the Game 2 challenger randomly guesses $j' \in [Q^{\text{id}}]$ such that $\mathcal{A}$'s j' -th private key query is for id^* . The challenger guesses it correctly with probability $1/Q^{\text{id}}$. Let $j^* = |\mathcal{T}| + j'$.

5. Set $Q = Q^{\text{id}} + |\mathcal{T}|$. In the final guess phase, the adversary outputs a guess $r' \in \{0, 1\}$. The challenger does the following:
 - a. Set $\mathbf{e} = (\mathbf{e}_1, \dots, \mathbf{e}_Q) = (\mathbf{t}_1, \dots, \mathbf{t}_{|\mathcal{T}|}, \text{id}_1, \dots, \text{id}_{Q^{\text{id}}})$, where $(\mathbf{t}_1, \dots, \mathbf{t}_{|\mathcal{T}|})$ and $(\text{id}_1, \dots, \text{id}_{Q^{\text{id}}})$ are the queried time tuple and the identity tuple respectively.

Set

$$\mathfrak{J} = \begin{cases} \{i^*\}, & \text{if } \mathcal{A} \text{ is guessed Type-1;} \\ \{i^*, j^*\}, & \text{if } \mathcal{A} \text{ is guessed Type-2.} \end{cases}$$

- b. **Abort check:** on input $\mathfrak{e}$, the challenger checks if $\tilde{H}$ satisfies $E_{\mathcal{J}}(\tilde{H})$ defined in Eq. (E.1). If not, it picks a fresh random bit from $\{0, 1\}$ to overwrite r' and we say the game is aborted by the challenger. Note that $\tilde{H}$ is hidden from the adversary's view, so the adversary has no idea if the game ends with an abort.
- c. **Artificial abort:** the challenger samples a bit $\Gamma \in \{0, 1\}$ satisfying $\Pr[\Gamma = 1] = \gamma(\mathfrak{e})$ where the analysis of such artificial abort and the definition of the function $\gamma(\cdot)$ are given in [108]. If $\Gamma = 1$, the challenger picks a fresh random bit from $\{0, 1\}$ to overwrite r' and we say the game is aborted by the challenger because of an artificial abort. Later we will briefly discuss the reason for this step, while for the details one may refer to [2, 108].

For a queries tuple $\mathfrak{e}$, let $\epsilon(\mathfrak{e})$ be the probability of the event that an abort (either real or artificial) does not occur. Let $\epsilon_{\max}^{(b)}$ and $\epsilon_{\min}^{(b)}$ be bounds such that $\epsilon(\mathfrak{e}) \in [\epsilon_{\min}^{(b)}, \epsilon_{\max}^{(b)}]$ for the case $\mathcal{A}$ is guessed Type- b , $b = 1, 2$.

Lemma E.2.1. ([3, Lemma 28]) *For a Type- b adversary, we have that*

$$\left| \Pr[W_2] - \frac{1}{2} \right| \geq \epsilon_{\min}^{(b)} \left| \Pr[W_1] - \frac{1}{2} \right| - \frac{1}{2}(\epsilon_{\max}^{(b)} - \epsilon_{\min}^{(b)}).$$

In our case, $\epsilon(\mathfrak{e})$ is the probability that Game 2 does not abort. Set

$$\Delta_b = \begin{cases} \frac{1}{2|\mathcal{T}|}, & \text{if } b = 1; \\ \frac{1}{2|\mathcal{T}| \cdot Q^{\text{id}}}, & \text{if } b = 2, \end{cases}$$

i.e., the probability that the challenger guesses the type of adversary, $\mathcal{J}$ -place correctly.

If there is no artificial abort step, i.e., $\gamma(\cdot)$ is the zero function, then $\epsilon_{\min}^{(b)} =$

$\Delta_b \cdot \alpha_{\mathcal{J}_b, \min}, \epsilon_{\max}^{(b)} = \Delta_b \cdot \alpha_{\mathcal{J}_b, \max}$, where $|\mathcal{J}_b| = b$. By Lemma E.1.2

$$\epsilon_{\max}^{(b)} - \epsilon_{\min}^{(b)} = \begin{cases} \Delta_1 \cdot Q/q^2, & \text{if } \mathcal{A} \text{ is guessed Type-1,} \\ \Delta_2 \cdot Q/q^3, & \text{if } \mathcal{A} \text{ is guessed Type-2,} \end{cases}$$

which is non-negligible and fails to lead to a good lower bound on $|\Pr[W_2] - \frac{1}{2}|$. Thus we apply Waters' approach [108] to add an artificial abort. With the strategy, $(\epsilon_{\max}^{(b)} - \epsilon_{\min}^{(b)})$ is less than $\epsilon_{\min}^{(b)} \cdot |\Pr[W_1] - \frac{1}{2}|$ and therefore $|\Pr[W_2] - \frac{1}{2}|$ is not less than

$$\frac{1}{2} \cdot \epsilon_{\min}^{(b)} \left| \Pr[W_1] - \frac{1}{2} \right| \geq \begin{cases} \frac{|\Pr[W_1] - \frac{1}{2}|}{8q|\mathcal{T}|}, & \text{if } b = 1; \\ \frac{|\Pr[W_1] - \frac{1}{2}|}{8q^2Q^{\text{id}} \cdot |\mathcal{T}|}, & \text{if } b = 2. \end{cases}$$

In summary, the advantages of $\mathcal{A}$ in Game 1 and Game 2 (no matter which type $\mathcal{A}$ is) satisfy

$$\left| \Pr[W_2] - \frac{1}{2} \right| \geq \frac{|\Pr[W_1] - \frac{1}{2}|}{8q^2Q^{\text{id}} \cdot |\mathcal{T}|}. \quad (\text{E.5})$$

Game 3. We change the way $\mathbf{A}$ and $\mathbf{C}, \mathbf{G}$ in Game 2 are chosen. The Game 3 challenger picks $\mathbf{A}$ randomly from $\mathbb{Z}_q^{m \times n}$, but generates $\mathbf{C}, \mathbf{G}$ using the algorithm `TrapGen` so that $\mathbf{C}, \mathbf{G}$ are random matrices in $\mathbb{Z}_q^{m \times n}$ and the Game 3 challenger owns trapdoors $\mathbf{T}_{\mathbf{C}}, \mathbf{T}_{\mathbf{G}}$ for $\Lambda_q^\perp(\mathbf{C}), \Lambda_q^\perp(\mathbf{G})$, respectively. The construction of $\mathbf{B}_i, \mathbf{D}_i$ for $i \in [\ell]$ remains the same as in Game 2, i.e., by Eq. (E.3). The challenger responds to the queries as follows:

`PriKeyGen`($\cdot$): The Game 3 challenger does the following to respond to a private key query for $\text{id} = (b_1, \dots, b_\ell) \in \{-1, 1\}^\ell$ using the trapdoor $\mathbf{T}_{\mathbf{C}}$:

1. Randomly choose an unassigned leaf node ν in BT and assign it to the identity id . Save (id, ν) to UL . Run $\text{SD.Assign}(\text{BT}, \nu)$ to obtain $PV_\nu = \{S_{i,j}\}$.
2. Construct $\mathbf{R}_{\text{id}}^* \leftarrow \sum_{i=1}^{\ell} b_i \mathbf{R}_i^* \in \mathbb{Z}_q^{m \times m}$ and $h_{\text{id}} \leftarrow 1 + \sum_{i=1}^{\ell} b_i h_i$ from Eq. (E.3), then $\mathbf{A}_{\text{id}} = \mathbf{C} + \sum_{i=1}^{\ell} b_i \mathbf{B}_i = \mathbf{A} \mathbf{R}_{\text{id}}^* + h_{\text{id}} \mathbf{C} \in \mathbb{Z}_q^{n \times m}$. Note that $h_{\text{id}} = H_{\text{ID}}(\text{id})$ where H_{ID} is the hash function in $\mathcal{H}_{\text{Wat}}$ defined by $(h_1, \dots, h_\ell)$.
3. If $h_{\text{id}} = 0$, abort the game and pick a fresh random bit in $\{0, 1\}$ to overwrite r' , as in Game 2. Else, go to the next step.
4. For each $S_{i,j} \in PV_\nu$, perform as follows:
 - Retrieve the record $(\text{GL}_{d_j}^i, *)$ from FL . If the second coordinate $*$ is not $\perp$, go to the next step. Else, randomly select $\text{F}_{d_j}^i(x) \in (\mathbb{Z}_q[x])^n$ such that each coordinate of $\text{F}_{d_j}^i$ is a polynomial of degree 1 and $\text{F}_{d_j}^i(0) = \mathbf{u}$. Update $(\text{GL}_{d_j}^i, \text{F}_{d_j}^i)$ to FL .
 - Sample $\mathbf{e}_{i,j}^{\text{id}}$ as

$$\mathbf{e}_{i,j}^{\text{id}} \leftarrow \text{SampleRight} \left(\mathbf{A}, h_{\text{id}} \mathbf{C}, \mathbf{R}_{\text{id}}^*, \mathbf{T}_{\mathbf{C}}, \text{F}_{d_j}^i(j), \sigma \right).$$

5. Output the updated state st and the private key sk_{id} for id as

$$(PV_\nu, \{\mathbf{e}_{i,j}^{\text{id}}\}_{S_{i,j} \in PV_\nu}).$$

UpdateKey($\cdot$): The Game 3 challenger does the following to respond to an update key query for $\mathbf{t} = (t_1, \dots, t_\ell) \in \{-1, 1\}^\ell$ using the trapdoor $\mathbf{T}_{\mathbf{G}}$:

1. Take R as a set consisting of such id 's that for some $\mathbf{t}' \leq \mathbf{t}$, $(\text{id}, \mathbf{t}') \in \text{RL}$. By using UL , define RI as the set of indices of leaf nodes corresponding to the id 's in R .

2. Run $\text{SD.Cover}(\text{BT}, RI)$ to obtain $CV_{\text{RI}} = \{S_{i,j}\}$.
3. Construct $\mathbf{R}_{\mathbf{t}}^* \leftarrow \sum_{i=1}^{\ell} t_i \mathbf{R}_{\ell+i}^* \in \mathbb{Z}_q^{m \times m}$ and $h_{\mathbf{t}} \leftarrow 1 + \sum_{i=1}^{\ell} t_i h_{\ell+i}$ from Eq. (E.3), then $\mathbf{A}_{\mathbf{t}} = \mathbf{G} + \sum_{i=1}^{\ell} t_i \mathbf{D}_i = \mathbf{A} \mathbf{R}_{\mathbf{t}}^* + h_{\mathbf{t}} \mathbf{G} \in \mathbb{Z}_q^{n \times m}$. Note that $h_{\mathbf{t}} = H_{\text{T}}(\mathbf{t})$ where H_{T} is the hash function in $\mathcal{H}_{\text{Wat}}$ defined by $(h_{\ell+1}, \dots, h_{2\ell})$.
4. If $h_{\mathbf{t}} = 0$, abort the game and pick a fresh random bit in $\{0, 1\}$ to overwrite r' , as in Game 2. Else, go to the next step.
5. For each $S_{i,j} \in CV_{\text{RI}}$, perform the following:
 - Retrieve the record $(\text{GL}_{d_j}^i, *)$ from FL. If the second coordinate $*$ is not $\perp$, go to the next step. Else, randomly select $\mathbf{F}_{d_j}^i(x) \in (\mathbb{Z}_q[x])^n$ such that each coordinate of $\mathbf{F}_{d_j}^i$ is a polynomial of degree 1 and $\mathbf{F}_{d_j}^i(0) = \mathbf{u}$. Update $(\text{GL}_{d_j}^i, \mathbf{F}_{d_j}^i)$ to FL.
 - Sample $\mathbf{e}_{i,j}^{\mathbf{t}}$ as

$$\mathbf{e}_{i,j}^{\mathbf{t}} \leftarrow \text{SampleRight} \left(\mathbf{A}, h_{\mathbf{t}} \mathbf{G}, \mathbf{R}_{\mathbf{t}}^*, \mathbf{T}_{\mathbf{G}}, \mathbf{F}_{d_j}^i(j), \sigma \right).$$

6. Output the update key $\text{uk}_{\mathbf{t}}$ at time $\mathbf{t}$ as

$$(CV_{\text{RI}}, \{\mathbf{e}_{i,j}^{\mathbf{t}}\}_{S_{i,j} \in CV_{\text{RI}}}).$$

For other phases, Game 3 is identical to Game 2. Specially, in the challenge phase the challenger checks which type the adversary $\mathcal{A}$ is.

- a. If $\mathcal{A}$ is Type-1, the challenger checks if the challenge time $\mathbf{t}^* = (t_1^*, \dots, t_{\ell}^*) \in \{-1, 1\}^{\ell}$ satisfies $h_{\mathbf{t}^*} := 1 + \sum_{i=1}^{\ell} t_i^* h_{\ell+i} = 0$. If not, the challenger aborts the game as in Game 2. Similarly, an artificial abort is implemented in Game 3.

- b. If $\mathcal{A}$ is Type-2, the challenger checks if the challenger time $\mathbf{t}^* = (t_1^*, \dots, t_\ell^*) \in \{-1, 1\}^\ell$ satisfies $h_{\mathbf{t}^*} := 1 + \sum_{i=1}^\ell t_i^* h_{\ell+i} = 0$ and the challenge identity $\mathbf{id}^* = (b_1^*, \dots, b_\ell^*) \in \{-1, 1\}^\ell$ satisfies $h_{\mathbf{id}^*} := 1 + \sum_{i=1}^\ell b_i^* h_i = 0$. If not, the challenger aborts the game as in Game 2. Similarly, an artificial abort is implemented in Game 3.

Now we prove the outputs from **SampleRight** we used in Game 3 here are indistinguishable from the corresponding outputs from **SampleLeft** in Game 2. We consider the responses to private key queries first. Since $h_{\mathbf{id}}$ in Step 4 of **PriKeyGen** is not zero, the matrix $\mathbf{T}_{\mathbf{C}}$ is a trapdoor for $h_{\mathbf{id}}\mathbf{C}$. Furthermore, $h_{\mathbf{id}}\mathbf{C}$ is a matrix with rank n with high probability as $\mathbf{C}$ is. By the introduction in Subsection 2.4.3, when $\sigma \geq \left\| \widetilde{\mathbf{T}_{\mathbf{C}}} \right\| \cdot s_{\mathbf{R}_{\mathbf{id}}}^* \cdot \omega(\sqrt{\log m})$, where $s_{\mathbf{R}_{\mathbf{id}}}^* = \|\mathbf{R}_{\mathbf{id}}^*\|_R \leq \sum_{i=1}^\ell \|\mathbf{R}_i^*\|_R = O(\ell\sqrt{m})$ from Lemma 2.4.8, the $\mathbf{e}_{i,j}^{\mathbf{id}} \in \mathbb{Z}^{2m}$ generated is distributed statistically close to $\mathcal{D}_{\Lambda_q^{\mathbf{f}}(\mathbf{F}_{\mathbf{id}}), \sigma}$, where $\mathbf{f} = \mathbf{F}_{d_j}^i(j)$ and $\mathbf{F}_{\mathbf{id}} = (\mathbf{A}|\mathbf{A}\mathbf{R}_{\mathbf{id}}^* + h_{\mathbf{id}}\mathbf{C}) = (\mathbf{A}|\mathbf{C} + \sum_{i=1}^\ell b_i\mathbf{B}_i)$, as Game 2. Recall that $\left\| \widetilde{\mathbf{T}_{\mathbf{C}}} \right\| \leq O(\sqrt{m})$ from Lemma 2.4.4. Therefore, the σ we select in Eq. (6.4) is sufficiently large to meet our sampling requirement. The situation for update key queries is similar.

From the adversary's view, it is hard to distinguish Game 2 and Game 3 since the distributions of the public parameters, responses to private key and update key queries, challenge ciphertext and abort conditions, are statistically close respectively. Therefore, the advantages of the adversary in both games are identical, i.e.,

$$\Pr[W_2] = \Pr[W_3]. \quad (\text{E.6})$$

We remark that, in Game 2, we have a uniform random hash function $\tilde{H} \in \mathcal{H}_{\text{Wat}}$ in common. In Game 3, we have two separate uniform random hash functions H_{ID} and H_{T} . But in terms of the non-abort probability, it is easy to show that there is

no difference. We adopt one hash instead of two in Game 2 to make the description concise and precise.

Game 4. Game 4 is the same as Game 3 except that the challenge ciphertext (c_0^*, c_1^*, c_2^*) is randomly chosen from the ciphertext space $\mathbb{Z}_q \times \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{2m}$, which directly implies the advantage of the adversary is zero.

Next we give a reduction from the LWE problem to show Game 3 and Game 4 are computationally indistinguishable.

Reduction from LWE. Suppose the adversary $\mathcal{A}$ can distinguish Game 3 and Game 4. We use $\mathcal{A}$ to construct an LWE algorithm $\mathcal{B}$. From Definition 2.4.12, we know an LWE instance grants access to a sampling oracle $\mathcal{O}$ which could be either truly random $\mathcal{O}_s$ or noisy pseudo-random $\mathcal{O}_s$ carrying some secret key $s \in \mathbb{Z}_q^n$. The simulator $\mathcal{B}$ distinguishes between the two by utilizing the adversary $\mathcal{A}$ as follows:

Instance. $\mathcal{B}$ queries from $\mathcal{O}$ and receives a fresh pair $(\mathbf{u}_i, v_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ for each $i = 0, \dots, m$.

Setup. $\mathcal{B}$ prepares a simulated attack environment for $\mathcal{A}$ as follows:

1. Assemble the random matrix $\mathbf{A}_0 \in \mathbb{Z}_q^{n \times m}$ from m of the given LWE instances, that is, let $\mathbf{A}_0 = [\mathbf{u}_1 | \dots | \mathbf{u}_m]$.
2. Generate the other components of the public parameters, namely, $\mathbf{B}_i, \mathbf{D}_i$ for $i \in [\ell]$ and $\mathbf{C}, \mathbf{G}$ as in Game 3 by means of random h_i and $\mathbf{R}_i^*$ for $i \in [2\ell]$.
3. Send $\text{PP} = (\mathbf{A}_0, \{\mathbf{B}_1, \dots, \mathbf{B}_\ell, \mathbf{C}\}, \{\mathbf{D}_1, \dots, \mathbf{D}_\ell, \mathbf{G}\}, \mathbf{u}_0)$ to $\mathcal{A}$, where $\mathbf{u}_0$ is the zeroth LWE sample.

Queries. $\mathcal{B}$ responds to private key queries and update key queries as in Game 3, involving aborting the game if required.

Challenge. $\mathcal{A}$ sends a challenge identity $\text{id}^* = (b_1^*, \dots, b_\ell^*) \in \{-1, 1\}^\ell$, a challenge

time $\mathbf{t}^* = (t_1^*, \dots, t_\ell^*) \in \{-1, 1\}^\ell$ and a message bit $b^* \in \{0, 1\}$. $\mathcal{B}$ prepares the challenge ciphertext as follows:

1. Retrieve $v_0, \dots, v_m$ from the LWE instance and set $\mathbf{v}^* = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^m$.
2. Blind the message bit by setting $c_0^* = Dv_0 + b^* \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$.
3. Set $\mathbf{R}_{\text{id}^*}^* \leftarrow \sum_{i=1}^\ell b_i^* \mathbf{R}_i^* \in \mathbb{Z}_q^{m \times m}$ and $\mathbf{R}_{\mathbf{t}^*}^* \leftarrow \sum_{i=1}^\ell t_i^* \mathbf{R}_{\ell+i}^* \in \mathbb{Z}_q^{m \times m}$ where for $i \in [2\ell]$, $\mathbf{R}_i^*$ were selected at the setup phase.
4. Set $\mathbf{c}_1^* = D \cdot \begin{bmatrix} \mathbf{v}^* \\ (\mathbf{R}_{\text{id}^*}^*)^\top \mathbf{v}^* \end{bmatrix}$, $\mathbf{c}_2^* = D \cdot \begin{bmatrix} \mathbf{v}^* \\ (\mathbf{R}_{\mathbf{t}^*}^*)^\top \mathbf{v}^* \end{bmatrix} \in \mathbb{Z}_q^{2m}$.
5. Send $\text{ct}^* = (c_0^*, \mathbf{c}_1^*, \mathbf{c}_2^*) \in \mathbb{Z}_q \times \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{2m}$ to the adversary.

We claim that when $\mathcal{O}$ is a pseudo-random LWE oracle $\mathcal{O}_s$ then ct^* is distributed exactly as in Game 3. By the definition of $\mathcal{O}_s$, $\mathbf{v}^* = \mathbf{A}_0^\top \mathbf{s} + \mathbf{y}$ for some random noise vector $\mathbf{y}$ sampled from $\overline{\Psi}_\alpha^m$. Note that $\mathbf{A}_{\text{id}^*} = \mathbf{A}_0 \mathbf{R}_{\text{id}^*}^*$. Therefore we have

$$\begin{aligned} \mathbf{c}_1^* &= D \cdot \begin{bmatrix} \mathbf{A}_0^\top \mathbf{s} + \mathbf{y} \\ (\mathbf{R}_{\text{id}^*}^*)^\top \mathbf{A}_0^\top \mathbf{s} + (\mathbf{R}_{\text{id}^*}^*)^\top \mathbf{y} \end{bmatrix} \\ &= (\mathbf{A}_0 | \mathbf{A}_{\text{id}^*})^\top \cdot (D\mathbf{s}) + D \cdot \begin{bmatrix} \mathbf{y} \\ (\mathbf{R}_{\text{id}^*}^*)^\top \mathbf{y} \end{bmatrix} \end{aligned}$$

and similarly

$$\mathbf{c}_2^* = (\mathbf{A}_0 | \mathbf{A}_{\mathbf{t}^*})^\top \cdot (D\mathbf{s}) + D \cdot \begin{bmatrix} \mathbf{y} \\ (\mathbf{R}_{\mathbf{t}^*}^*)^\top \mathbf{y} \end{bmatrix},$$

which implies $\mathbf{c}_1^*$ and $\mathbf{c}_2^*$ are precisely $\mathbf{c}_1, \mathbf{c}_2$ parts of a legitimate challenge ciphertext in Game 3. We also have $v_0 = \mathbf{u}_0^\top \mathbf{s} + x$, where x has distribution $\bar{\Psi}_\alpha$. Therefore $c_0^* = \mathbf{u}_0^\top \cdot (D\mathbf{s}) + Dx + b^* \left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$, which is precisely the c_0 part of a legitimate challenge ciphertext in Game 3.

When $\mathcal{O}$ is a random oracle $\mathcal{O}_\S$, then v_0 is uniform in $\mathbb{Z}_q$ and $\mathbf{v}^*$ is uniform in $\mathbb{Z}_q^m$. Therefore $(c_0^*, \mathbf{c}_1^*, \mathbf{c}_2^*)$ is uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{2m}$ by the standard leftover hash lemma (e.g Theorem 8.38 of [103]). Hence the challenge ciphertext is always uniform in $\mathbb{Z}_q \times \mathbb{Z}_q^{2m} \times \mathbb{Z}_q^{2m}$, as in Game 4.

Guess. $\mathcal{A}$ is able to make more queries, then guesses if the interaction is conducted with a Game 3 or Game 4 challenger. Next the simulator employs the artificial abort from Games 3 and 4, then returns the final guess to the **LWE** challenger.

As discussed above, if $\mathcal{O} = \mathcal{O}_\S$, the adversary's view is the same as in Game 3; if $\mathcal{O} = \mathcal{O}_\$, the adversary's view is the same as in Game 4. Thus $\mathcal{B}$ has the same advantage in solving the **LWE** as $\mathcal{A}$ in distinguishing Game 3 and Game 4, as expected. Since $\Pr[W_4] = 1/2$, we obtain$

$$\left| \Pr[W_3] - \frac{1}{2} \right| = |\Pr[W_3] - \Pr[W_4]| \leq \text{Adv}_{\mathcal{B}}^{(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)\text{-LWE}}. \quad (\text{E.7})$$

By combining Eqs. (E.4) - (E.7), we obtain the following result as stated in Theorem E.0.3:

$$\left| \Pr[W_0] - \frac{1}{2} \right| \leq 8q^2 Q^{\text{id}} \cdot |\mathcal{T}| \cdot \text{Adv}_{\mathcal{B}}^{(\mathbb{Z}_q, n, \bar{\Psi}_\alpha)\text{-LWE}}.$$