Monash University

Practice-Oriented Techniques in Lattice-Based Cryptography

Author
Muhammed Fethullah Esgin

Supervisors
Dr. Ron Steinfeld
Dr. Joseph K. Liu
Dr. Dongxi Liu

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy at

Monash University

in the

Faculty of Information Technology

May 2, 2020

Copyright notice

MONASH UNIVERSITY

Abstract

Faculty of Information Technology

Doctor of Philosophy

Practice-Oriented Techniques in Lattice-Based Cryptography

by Muhammed Fethullah Esgin

In the last decade, lattice-based cryptography, a promising candidate for quantum-safe algorithms, has seen a great interest with many new applications being developed. Although it offers solutions even to problems which long seemed elusive, there is still a gap in some areas where lattice-based cryptographic proposals are not efficient enough for practical use and even fall far behind their classical counterparts in terms of efficiency. This unsatisfactory state of affairs, where practical requirements are not met by lattice-based proposals, forms the fundamental problem tackled in this Ph.D. thesis, where new techniques in lattice-based cryptography are explored with a practice-oriented approach in mind. This is a particularly critical problem to be studied today as today's classical cryptographic algorithms relied on by billions everyday are threatened by the advances in quantum computers and there is an ongoing post-quantum cryptography standardisation process initiated by NIST.

A particular focus of this thesis is on designing efficient zero-knowledge proofs (ZKP), which allow one party to convince another party of the truth of a certain statement without revealing secret information. These proof systems are fundamental tools used in the construction of many privacy-preserving protocols such as anonymous credentials and those used in the blockchain-based applications. The main aim when designing these ZKPs is to develop novel widely applicable techniques that can overcome important challenges in the construction of lattice-based ZKPs in general. More specifically, the target problem here boils down to *efficiently* proving *nonlinear* polynomial relations that can prove more *complex* statements. To this end, the problem is studied in two contexts: *multi-shot* proofs, that does not necessarily reach a convincing soundness level in a single execution, and *one-shot* proofs, that does so. The former allows the problem to be studied in a less constrained setting and build the stepping stones for the latter more practical goal. Then, in the latter setting, fundamental techniques for the construction of efficient lattice-based algebraic proofs are established.

Having demonstrated useful foundational techniques in the consideration of ZKP designs, the attention is turned into proving particular useful relations such as binary proof, range proof, one-out-of-many proof and set membership proof. Then, the constructed ZKPs are used as building blocks for advanced cryptographic tools such as ring signatures, which is a type of *anonymous* signature where the identity of the actual signatory is hidden among a set of identities. Such anonymous signatures have a wide range of applications in areas such as cryptocurrencies and e-voting systems. The evaluation of the proposed ring signatures proves the effectiveness of the foundational techniques, where the proposals in this thesis achieve a dramatic efficiency improvement in comparison to the prior arts.

Proceeding even closer to practice from theory, the final objective of the thesis is set to construct a privacy-aware blockchain application, enabling users to create *confidential transactions (CT)* on blockchain. That is, the goal is to enable a user to create a blockchain transaction while hiding her identity and the transaction amount, and prove in zero-knowledge fashion that the created transaction is valid. In particular, a full-fledged *post-quantum* blockchain CT protocol, named MatRiCT, is designed, accompanied by an efficient implementation. MatRiCT makes use of the foundational tools developed in the context of zero-knowledge proofs and provides the first practical solution to a post-quantum RingCT protocol, the family of CT protocols based on ring signatures. An important feature of MatRiCT is its ability to tune the balance between privacy and accountability. In particular, MatRiCT comes in with an *optional auditability* feature that allows a selected auditor to be able to revoke the anonymity of the users who opt in for auditing by the particular auditor. Such an accountability feature is important for regulatory or financial enterprise applications.

Declaration

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Name: Muhammed Fethullah Esgin

Date: April 30, 2020

List of publications included in the thesis

- Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *Applied Cryptography and Network Security (ACNS)*, volume 11464 of *LNCS*, pages 67–88. Springer, 2019.
(Full version at https://eprint.iacr.org/2018/773)

- Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In *Advances in Cryptology - CRYPTO (I)*, *LNCS*, pages 115–146. Springer, 2019.
(Full version at https://eprint.iacr.org/2019/445)

- Muhammed F. Esgin, Raymond K. Zhao, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. 2019. MatRiCT: Efficient, Scalable and Post-Quantum Blockchain Confidential Transactions Protocol. In *2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19), November 11–15, 2019, London, United Kingdom*. ACM, pages 567–584, New York, NY, USA.
https://doi.org/10.1145/3319535.3354200
(Full version at https://eprint.iacr.org/2019/1287)

Other publications in the Ph.D. course

- Orhun Kara, and Muhammed F. Esgin. On Analysis of Lightweight Stream Ciphers with Keyed Update. In *IEEE Transactions on Computers*, 68(1):99–110, 2019.
https://doi.org/10.1109/TC.2018.2851239

- Tsz Hon Yuen, Shifeng Sun, Joseph K. Liu, Man Ho Au, Muhammed F. Esgin, Qingzhao Zhang, and Dawu Gu. RingCT 3.0 for blockchain confidential transaction: Shorter size and stronger security. *IACR Cryptology ePrint Archive*, 2019:508, 2019.
https://eprint.iacr.org/2019/508
(To appear at Financial Cryptography and Data Security 2020)

Acknowledgements

First and foremost, I would like to convey my deepest gratitude to my main supervisor, Ron Steinfeld. Without his endless support, guidance and teaching, this dissertation would not have existed. He helped me in various aspects even well before the commencement of my Ph.D.

I thank my co-supervisors Joseph K. Liu and Dongxi Liu for their help and support throughout my Ph.D. journey. They did not hesitate to provide their help whenever needed.

I would also like to express my gratitude to Damien Stehlé and Steven Galbraith, who helped me with my Ph.D. choices. I also thank Steven for being a part of my progress review panel.

I owe great thanks to many friends and colleagues at Monash University, including Wilson Abel Alberto Torres, Ahmad Salehi Shahraki, Hagen Lauer and Maxime Buser. Neither my Ph.D. study nor my life in Melbourne would have been as enjoyable as it is without their friendship.

I also would like to thank Vadim Lyubashevsky for hosting me for an internship at IBM Research, Zurich, and everyone there for welcoming me into the group. I am grateful for their friendship.

I am grateful to Data61, CSIRO for providing me a stipend scholarship throughout the course of my Ph.D. study, and supporting me to attend conferences and similar events in multiple occasions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In today's computerised world, billions of people rely heavily on security systems that protect a tremendous amount of sensitive information ranging from personal details, passwords to access various systems, banking information to health records and even government secrets. The algorithms currently used in such systems rely on *classical* cryptographic assumptions, such as integer factorisation and discrete logarithm problem (DLP), that do not provide security against attacks by powerful quantum computers. Although it is hard to predict precisely when a scalable quantum computer powerful enough to break today's classical cryptographic schemes would be developed, NIST's call [NIS17] for proposals for post-quantum cryptography (PQC) standardisation is an important indication that there is an evident need for *post-quantum*[1] alternatives of today's widely used cryptographic tools.[2]

Lattice-based cryptography today stands as one of the most promising candidates for post-quantum cryptography, and it studies the design and analysis of cryptographic constructions whose security is based on computationally hard lattice problems (such as Shortest Vector Problem) that are believed to resist quantum attacks. In terms of basic cryptographic primitives such as encryption schemes and digital signatures, lattice-based cryptography already seems to offer good solutions. For example, there are multiple lattice-based digital signatures such as Dilithium [DLL+18], qTESLA [ABB+19] and Falcon [FHK+18] that moved on to the second round of NIST's PQC standardisation process. Depending on the desired security guarantees, one can pick one of the signatures and its existing computational efficiency would be good enough for many applications. However, post-quantum security does not come for free, of course. The main drawback of such lattice-based schemes over their classical counterparts (especially those based on elliptic curves) is that lattice-based solutions incur higher storage and communication costs for keys and signature. To illustrate, the aforementioned lattice-based signature schemes require in the order of a few KB storage whereas a signature scheme based on Elliptic Curve Discrete Logarithm Problem (ECDLP) requires only a few hundreds of bits storage. Such a gap in storage seems to be inherent in most (if not all) of the existing post-quantum proposals, and the current state of affairs suggests that there is no better alternative than accepting the gap.

On the other hand, if one looks at more advanced cryptographic schemes such

[1] The term "post-quantum" used throughout the thesis means *currently believed to resist efficient attacks by powerful quantum computers.* Just as there is no proof, for example, that AES can never be broken efficiently by classical computers, there is also no proof that any of the existing post-quantum cryptography candidates can never be broken efficiently by quantum computers.

[2] This chapter is partly based on [ESS+19, ESLL19, EZS+19].

as zero-knowledge proofs[3], ring/group signatures[4] and advanced privacy-preserving protocols such as confidential transactions[5], the gap in storage and communication costs grows even much larger. To illustrate, the only logarithmic-sized ring signature from lattices (due to Libert et al. [LLNW16]) prior to this Ph.D. research project results in a signature of length more than 40 MB for only 1000 users. DLP-based analogs (e.g., [GK15, BCC+15]), however, cost only a few KB for even billions of users. This huge gap here indeed stems from the imbalanced costs of the underlying zero-knowledge proofs (ZKP), which are used as building blocks to construct such anonymous signatures. In fact, in a more general sense of proving complicated statements in zero-knowledge fashion, the prior lattice-based ZKPs do not seem to offer practically acceptable solutions. Such a state of affairs is very unfortunate as ZKPs are fundamental building blocks for many privacy-preserving applications ranging from anonymous credentials, secure e-voting to privacy-aware blockchain-based applications (such as anonymous cryptocurrencies, e.g., Zcash and Monero).

This major problem of lack of satisfactory post-quantum ZKP constructions and their higher level protocol applications constitutes the main challenge we address in this Ph.D. research. In particular, in this thesis, new techniques in design and analysis of *algebraic* lattice-based zero-knowledge proofs are explored with a practice-oriented approach in mind. The main reason a particular focus is given on algebraic proofs is that they often lead to very efficient constructions (in practice) by exploiting the algebraic structures in certain polynomial rings. Construction of efficient ZKPs alone, however, does not fully address the practical needs of real-life applications. Indeed, the need to design quantum-secure alternatives of currently deployed algorithms is widely agreed on. An example of this can be seen in Zcash's FAQ page [Tea19], which states that the developers "plan to monitor developments in postquantum-secure components, and if/when they are mature and practical, update the Zcash protocol to use them." In addressing such a practical need, after construction of efficient ZKPs, we turn our attention to the construction of efficient advanced protocols with post-quantum plausibility. A particular application of interest is a blockchain confidential transactions protocol using ring signatures, namely a RingCT protocol [Noe15].

An important goal of the thesis is to establish novel techniques that can be widely applicable in the construction of various lattice-based ZKPs, not just those focused on in the thesis. Since the tools (especially in Chapters 4 and 5) are mostly introduced in a generic protocol setting, they are believed to be of independent interest for future works on efficient lattice-based ZKPs.

Prior works on lattice-based ZKPs focused mostly on proving *linear* relations that are sufficient for basic applications such as (ordinary) signatures. Such a linear relation, for example, can prove knowledge of an opening of a commitment[6] as lattice-based commitments are linear functions of the secrets. The ZKPs in this thesis, on the other hand, are mainly concerned with proving *non-linear* relations. For example,

[3] A zero-knowledge proof is a proof system where one party convinces another party that a certain statement regarding a secret is true without revealing the secret.

[4] Ring and group signatures are a type of *anonymous* signatures where the identity of the real signatory is hidden within a set of identities (i.e., an anonymity set or ring/group). An important efficiency aspect is how the signature length grows with the anonymity set/group/ring size.

[5] A confidential transactions (CT) protocol allows a user to conduct transactions on blockchain while hiding sensitive information such as his/her identity and transaction amount. The family of CT protocols that are focused in the thesis are called RingCT protocols as they rely on ring signatures as a core component.

[6] A commitment is the output of a cryptographic function called commitment function. For now, one may imagine a commitment function to be an encryption scheme, a commitment to be a ciphertext and an opening to be a message. A commitment scheme is formally defined in Chapter 3.

one can use the relation $x \cdot (x-1) = 0$ to prove that x is binary (i.e., a binary proof) and $(x-\alpha_1)\cdots(x-\alpha_k) = 0$ to prove that x is contained in the set $S = \{\alpha_1, \ldots, \alpha_k\}$ (i.e., a set membership proof). Observe that the latter polynomial relation is of degree $k = |S|$. Indeed, one can perform a set membership proof using a polynomial relation of degree $\log |S|$, which is likely to result in a more compact proof than set membership proofs based on linear relations, whose proof length is linear in $|S|$. A similar relation (in particular, a one-out-of-many relation) is one of the polynomial relations of interest to be proven efficiently in the thesis.

As suggested by the above example of set membership proofs, the motivation for studying lattice-based ZKPs with non-linear relations comes from the fact that the ability to prove such non-linear relations has been shown to be very useful in the DL setting (see, e.g., [GK15, BCC+15, BCC+16, BBB+18]) for constructing both asymptotically and practically more compact proofs. However, ZKPs do not run so smoothly in the lattice setting and the tools used in the DL-setting cannot be applied efficiently in a straightforward manner in lattice-based cryptography. Therefore, one requires novel techniques (as those introduced in this thesis) to construct efficient advanced ZKPs based on lattice assumptions. Then, there are further problems to be overcome when one is interested in extending the ZKPs efficiently to higher level protocols such as a confidential transactions protocol.

At a high level, some examples of ZKPs that are of particular interest in this work are as follows.

1. **One-out-of-many proof:** the prover's goal is to prove knowledge of a secret associated to an (undisclosed) public element in a set of public elements.
2. **Binary proof:** the prover's goal is to prove that a commitment opens to a sequence of bits.
3. **Range proof:** the prover's goal is to prove that a commitment opens to a value in a certain range.
4. **Set membership proof:** the prover's goal is to prove that a commitment opens to a value in a certain public set.
5. **Balance proof:** the prover's goal is to prove that the sum of the committed integers in a set of inputs is equal to the sum of the committed integers in a set of outputs.

Let us take a 1-out-of-N proof as an example to give a general idea of how such proofs can be useful in practice. For example, using a 1-out-of-N proof, one can prove knowledge of a secret key corresponding to one of N public keys, i.e., proving membership in a certain set of users. This way, the prover can *anonymously* prove his credibility by showing that he is one of the credible parties. As another simple example, using a range proof, one can prove (without revealing his exact age) that his age is greater than 18. This could serve as an evidence of eligibility for certain practical applications.

For practical purposes, it is very desirable that a protocol achieves a convincing soundness level in one execution. Otherwise, all the operations in the protocol need to be repeated to amplify soundness (i.e., to reduce the chance that a cheating prover succeeds). In turn, protocol repetitions result in multi-fold increase of both the proof length (i.e., the communication size) and the computational cost. We call the protocols that reach a convincing soundness level (i.e., an exponentially small soundness error) to be *one-shot*. The protocols that require repetitions are then called *multi-shot*.

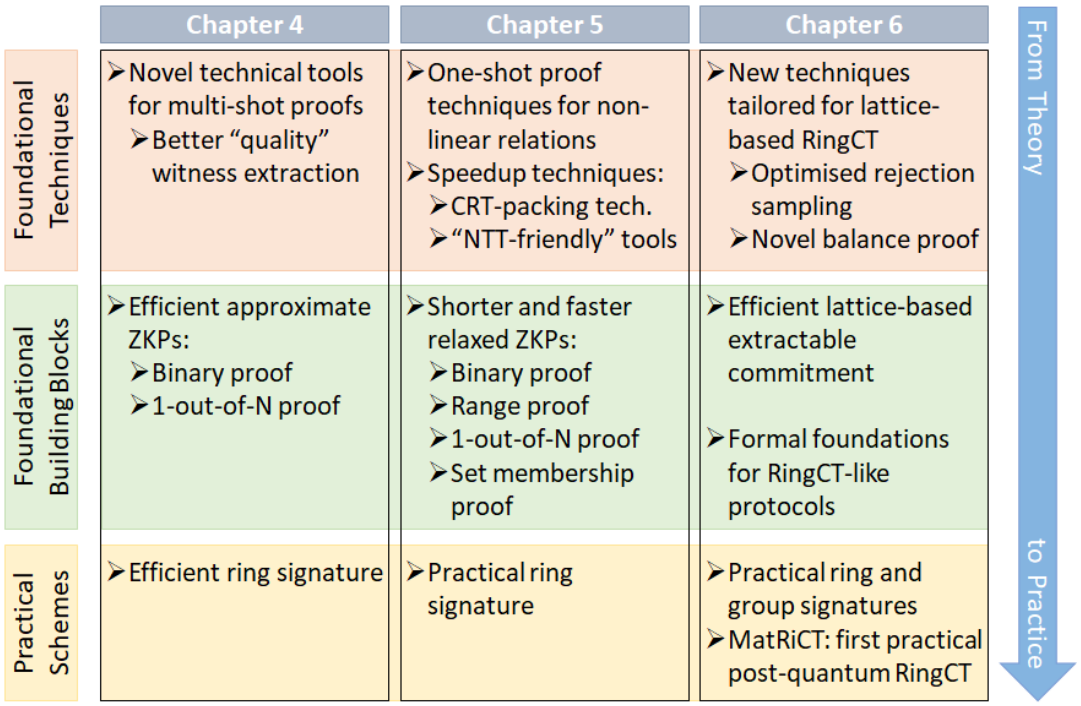


FIGURE 1.1: Overview of contributions.

1.1 Contributions

In general, the main theoretical contribution of the thesis is the introduction of novel technical tools for the design and analysis of algebraic lattice-based zero-knowledge proofs. The tools are applicable in a generalised setting of *many-special sound* protocols (see Definition 3.6), what we also call protocols with "complex" witness extraction. Many existing solutions for the more specific case of *2-special sound* protocols are obtained as a special case of our techniques. Many-special sound protocols are those that can prove non-linear relations whereas 2-special sound protocols in general are restricted to the linear relations.

On a more practical perspective, the new techniques developed are used to construct efficient useful protocols such as binary proofs, range proofs, one-out-of-many proofs, set membership proofs and balance proofs. Further, in terms of advanced cryptographic tools, the common application in all core chapters is to a ring signature. As an indication of the effectiveness of our techniques, we show in Table 1.1 that the ring signatures introduced achieve a dramatic improvement in terms of length in comparison to other scalable "post-quantum" proposals at the same security level. Apart from ring signatures, another application of practical interest is a form of privacy-preserving linkable anonymous credentials.

As a concrete embodiment of various novel techniques, the main *practical* contribution of the thesis is the introduction of MatRiCT, the first practical post-quantum RingCT protocol.

With this general overview of the contributions, let us go over the contributions of each chapter in more detail. To give a better understanding of the "flow" of the contributions, Figure 1.1 provides a high-level overview of the contributions.

TABLE 1.1: Comparison of signature lengths (in KB) of "post-quantum" sub-linear size ring signatures (the challenge space size is 2^{256}). "?" indicates that the signature length for the particular setting cannot be approximated using the results of the respective reference.

Ring Size N :	2	8	64	4096	2^{21}	**Security basis**
[LLNW16]	23000	52000	94000	179000	306000	SIS
[YAZ+19]	?	?	?	$>$ 13000	?	LWE & LWR & SIS
Chapter 4	1000	1200	1600	2400	4100	M-LWE & M-SIS
[DRS18]	236	477	839	1561	2645	LowMC (Symmetric-key)
[KKW18]	?	?	$\sim$ 250	$\sim$ 456	?	LowMC (Symmetric-key)
Chapter 5	36	41	58	103	256	M-LWE & M-SIS
Chapter 6	18	19	31	59	148	M-LWE & M-SIS

1.1.1 Chapter 4

Novel technical tools for the design and analysis of *multi-shot* algebraic lattice-based zero-knowledge proofs.

We study the problem of proving non-linear polynomial relations, which previously did not receive a lot of attention in lattice-based cryptography, and introduce a set of new technical tools for the design and analysis of algebraic proofs. Here, we introduce 3 methods to get better "quality" witness extraction, which allows one to choose less aggressive parameters and thus get more efficient proofs. We can view these new techniques in Chapter 4 as the stepping stones towards more practical goals in the further chapters.

Efficient proofs for useful approximate relations.

We show that our new techniques are useful in constructing efficient ZKPs from lattices. In particular, we construct efficient binary proofs and one-out-of-many proofs, where the proved relations are *approximate*. What this means is that the proved relations are slightly relaxed, but are still stronger than those in Chapter 5. These proofs may serve as better options than those in Chapter 5 when one is interested in extending recursive proofs like Bulletproofs [BBB+18] to the lattice setting.

Efficient scalable ring signature based on standard lattice assumptions.

As shown in Table 1.1, even our initial ring signature design in Chapter 4 achieves a dramatic improvement in terms of length over the only prior log-sized result from lattices by Libert et al. [LLNW16], where the improvement is almost two orders of magnitude. Even in comparison to a very recent lattice-based proposal [YAZ+19], this scheme is still much shorter.

An important feature of our constructions is that a modulus q of a special form (such as $q \equiv 17 \bmod 32$ as in [dPLNS17]) is not required, which allows the use of fast computation algorithms such as Number Theoretic Transform (NTT).

1.1.2 Chapter 5

One-shot proof techniques for non-linear polynomial relations via adjugate matrices.

We introduce new techniques that provide the first solution to the problem of building efficient *one-shot* lattice-based ZKPs that require a "complex" witness extraction. In

particular, we introduce witness extraction for ZKPs proving that the witness satisfies a non-linear polynomial relation of degree $k \geq 2$ (i.e., "$(k+1)$-special sound protocols", see Definition 3.6) while still having a one-shot proof. Our proofs reach a negligible soundness error in a single run of the protocol. In comparison to relevant multi-shot prior works such as [LLNW16], we improve the asymptotic computation and communication costs by a factor of $\widetilde{O}(\lambda)$ for the security parameter λ (see Table 5.1), and also achieve a dramatic practical efficiency improvement in both costs (see, e.g., Table 1.1). The previous one-shot ideas [Lyu09, Lyu12, BKLP15, BDL+18] are obtained as a special case of our technique (see Section 5.3.2).

Speedup Technique 1: CRT-packing supporting inter-slot operations.

Drawing inspiration from the CRT-packing techniques [SV14, GHS12] used in fully homomorphic encryption, we introduce the first CRT-packing technique in lattice-based ZKPs that supports a *complete* set of "inter-slot" operations. That is, our technique supports operations between messages stored in separate CRT "slots", and gives the ability to commit to/encode multiple messages at once and then "extract" all the messages in a way that permits interoperability among extracted values. In its full potential, it provides an asymptotic improvement of $O(\log q)$ in computation costs of proofs involving $O(\log q)$ messages at no additional cost to the proof length (see Table 5.1).

Speedup Technique 2: "NTT-friendly" tools for fully-splitting rings.

An important obstacle to computational efficiency of lattice-based ZKPs is that one often requires invertibility of short elements in a ring. A common solution to meeting this criterion is to choose a modulus q of a special form (such as $q \equiv 5 \bmod 8$) at the cost of disabling the ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ to fully-split, and thus preventing the (full) use of fast computational algorithms such as NTT. We introduce a new result (Lemma 5.5) that can be used as an alternative to enforcing invertibility, and show how it can be made use of while still supporting the use of NTT-like algorithms. The only requirement of our lemma is for the modulus q to be sufficiently large, without putting any assumptions on its "shape". One can see from, e.g., [LS18, Table 2] that full NTT provides a speedup of a factor between 6-8 in comparison to plain Karatsuba multiplication (with no FFT).

Design of shorter and faster lattice-based protocols.

Our techniques enable the construction of communication and computation efficient lattice-based analogues of DL-based protocols for important applications, where there was previously no efficient lattice-based solutions known. To illustrate this utility of our techniques, we design an efficient range proof that uses speedup technique 1, and an efficient one-out-of-many proof that uses speedup technique 2, where our one-shot proof technique is also applied in both of the proofs.

Application to advanced cryptographic tools.

Despite their relaxed nature, we show that our ZKPs are sufficient for important practical applications. Our one-out-of-many proof is used as a building block for lattice-based ring signatures, and our relaxed aggregated range proof is shown to be sufficient for an application in a form of privacy-preserving linkable anonymous credentials.

As detailed in Section 5.6.1, our ring signature in Chapter 5 achieves a signature length quasi-linear in the security parameter λ, and poly-logarithmic in the ring size N. In practice, the signature length is proportional to $\lambda \log^2 \lambda \log^c N$ for some constant $c \approx 1.67$. This improves on the quadratic dependence on λ in [LLNW16, DRS18, KKW18].[7] In terms of the dependence on $\log N$, our scheme grows slightly faster, however, it still outperforms all these works for N as big as billions and beyond.

We further analyse the computational efficiency of our ring signature in Appendix 5.6.1. The analysis based on reasonable assumptions shows that our construction also greatly improves the practical signing/verification times over the existing ring signature proposals with concrete computational efficiency results. For $N = 1024$, we estimate the signing/verification times of our scheme to be below 30 ms whereas [KKW18] reports 2.8 seconds for both of the running times. Our ring signature as well as its underlying protocols, namely binary proof and one-out-of-many proof, do not require any assumption on the "shape" of the modulus q, and thus permit the use of NTT-like algorithms.

1.1.3 Chapter 6

Improved ring signature.

Our first contribution in this chapter is to introduce the shortest scalable ring signature to date from standard lattice assumptions, namely M-SIS and M-LWE. In particular, we introduce several improvements on the sublinear-sized ring signature in Chapter 5. This new construction, unlike the one in Chapter 5, does not require any (discrete) Gaussian sampling, and therefore it is much easier to protect against side-channel attacks. To get an advantageous use of the uniform distribution, we introduce a new technique for the application of rejection sampling on binary secrets with fixed Hamming weight (see Section 6.2).

MatRiCT: A novel post-quantum RingCT.

Our main contribution in Chapter 6 is the design of a novel RingCT protocol, named MatRiCT, that is *efficient*, *scalable* and *post-quantum*. The main technical novelties of MatRiCT are sketched in Section 6.2. As shown in Table 6.4, in comparison to the only existing post-quantum RingCT protocol supporting multiple inputs and outputs, we achieve a dramatic improvement in transaction size, which is the main metric in determining transaction fees. Our scheme is also very efficient in terms of computational complexity even for an anonymity set as large as 1000 as shown in Table 6.5.

As a bonus feature, we show in Section 6.9 that MatRiCT easily extends to provide *auditability* (i.e., the ability of an authority to trace real spenders) in a way that does not require significant modifications to the system. Auditability is an important feature to prevent illegal use of a cryptocurrency, and is desired, e.g., for regulatory or financial enterprise applications. The auditability feature of MatRiCT allows a selection among different anonymity flavours within the same environment such that each user selects his/her own auditor and can even select to have no auditing.

[7] In [LLNW16], the soundness error goal of $\lambda^{-\omega(1)}$ is used and so the number of protocol repetitions for Stern's framework is taken to be $\omega(\log \lambda)$, which disappears in $\widetilde{O}(\cdot)$ notation. But, we consider a practice-oriented goal for the soundness error of $2^{-\lambda}$, and thus the number of protocol repetitions for Stern-based proofs must be $\Omega(\lambda)$. Also, it is stated in [KKW18] that they have the same asymptotic signature growth with [LLNW16].

Novel extractable commitment.

We introduce a novel *extractable* commitment scheme from lattices, which extends the commonly used Hashed-Message Commitment (HMC) (see Section 3.2.2). An extractable commitment has an additional CExtract function that allows a party to recover the message stored in a commitment using a (secret) *trapdoor*. Without knowledge of the trapdoor, however, the message remains hidden. Therefore, extractable commitments are ideal tools for privacy-preserving applications where accountability, e.g., in case of misbehaviour is desired.

The main advantage of our primitive is that it can be realised with almost the same parameters as HMC, and does not mandate very aggressive parameters. To illustrate, for an $n \times m$ commitment matrix over a ring with modulus q, the GPV trapdoor [GPV08] (see also the improved constructions and Figure 1 in [MP12]) requires $m = O(n \log q)$ whereas, for the same trapdoor norm level, we only require $m = O(n)$ as in standard HMC. The extraction works when the input message space is not too large (i.e., it is feasible to iterate over all messages).

Efficient group signature for moderate-sized groups.

Combination of our ring signature with the extractable commitment results in a group signature (or an accountable ring signature), which shares the same efficiency features as the ring signature. The signature length of our ring/group signature is very short and compared to the state-of-the-art post-quantum proposals in Table 6.6.

New formal definitions for RingCT-like protocols.

Further, we introduce new rigorous security definitions for RingCT-like protocols. Our goal in introducing a new set of definitions is to provide an easy-to-understand model that captures the real-world scenario more closely than the previous attempts [SALY17, YSL+19]. We believe these formal foundations to contribute to the development of future RingCT protocols in general (not only in the lattice setting).

1.2 Thesis Structure

The rationale behind the structure of the thesis is as follows. The topics within the core chapters (i.e., Chapters 4, 5 and 6) start from theoretical foundations and build towards the practical applications. A similar structure is established among these three chapters as foundational techniques, zero-knowledge proofs and cryptographic primitives are constructed in Chapters 4 and 5, which then builds up to a practical system of post-quantum blockchain confidential transactions protocol in Chapter 6.

Chapter 2 covers the related literature in zero-knowledge proofs, ring and group signatures, and RingCT protocols. Then, in Chapter 3, notations used throughout the thesis, cryptographic definitions such as sigma protocols, security assumptions and commitments schemes, and mathematical background are introduced. Chapter 4 discusses new techniques in multi-shot protocols (i.e., protocols that require multiple repetitions) and their applications. The focus in Chapter 5 is on one-shot proof techniques where no protocol repetition is required. Then, Chapter 6 blends the tools introduced so far with novel techniques targeted specifically for confidential transactions protocols to construct a practical RingCT protocol, accompanied by a full implementation. The thesis is concluded in Chapter 7 with some discussions and potential future research directions.

Chapter 2

Literature Review

This chapter summarises the state of the art in the areas relevant to our discussion further on as of the time of writing (July 2019). We start with *zero-knowledge proofs* (ZKP), which form the basis of many constructions described in the thesis. ZKPs enable a *prover* to convince a *verifier* that a certain statement regarding a secret is true with minimal secret information leakage. Then, we discuss two types of *anonymous signatures*, namely *ring signature* and *group signature*. These tools allow a party to generate a signature on behalf of a set of users. Finally, we cover some of the related literate in *RingCT protocols*, where users can create *confidential transactions* on blockchain so that the spender's identity as well as the transaction amount is hidden from the outside world. The discussion in this chapter is kept informal, and relevant formal definitions are given in the following and subsequent chapters.[1]

2.1 Zero-Knowledge Proofs

Zero-knowledge proofs (ZKP) are fundamental building blocks used in many privacy-preserving applications such as anonymous cryptocurrencies and anonymous credentials [Cha85], and the underlying advanced cryptographic primitives such as ring signatures [RST01]. They were introduced by Goldwasser, Micali and Rackoff [GMR89]. In this thesis, we restrict our attention particularly to lattice-based proposals and the relevant classical counterparts. An important feature desired of many protocols (whether they rely on lattice-based or classical assumptions) in practice is *non-interactivity*. This feature enables a prover to create a proof on her own such that the proof can later be verified by outside parties with no interaction required between any parties. Thankfully, there is a generic method that transforms an interactive *sigma protocol*, which is the class of zero-knowledge proofs focused on in this work (see Section 3.2.3), into a non-interactive one. This tool is often called the *Fiat-Shamir transformation* [FS86] and the security of a scheme using this transformation is proven in the *random oracle model*[2] [BR93]. The idea for the Fiat-Shamir transformation is to replace the verifier by a *random oracle* that returns completely random (but consistent) outputs. In practice, the random oracle is often realised by a cryptographic hash function[3].

A core property of ZKPs is *soundness*, that is, a cheating prover should not be able to create a convincing "proof". In the context of *proofs of knowledge* (PoK), this means successful provers know a relevant secret (i.e., a *witness*), and this is usually

[1] This chapter is partly based on [ESS+19, ESLL19, EZS+19].

[2] It is worth noting here that the post-quantum algorithms studied throughout the thesis do not necessarily include a security proof in the *quantum* random oracle model (QROM). There are several exciting recent works, e.g. [LZ19, DFMS19], that study the behaviour of Fiat-Shamir protocols in the quantum setting and provide promising results on the security of Fiat-Shamir protocols in QROM. We refer to these papers for more details.

[3] In this thesis, we always consider a *cryptographic* hash function, even though we may simply write "hash function".

proven by using an *extractor* that efficiently recovers a witness given two accepting protocol transcripts with the same initial message. We call this procedure *"basic" witness extraction* (also known as "2-special soundness", see Definition 3.6). A natural behaviour that is trivially observed in discrete logarithm (DL) based ZKPs is that they achieve a convincing soundness level (i.e., a negligible *soundness error*) in a single protocol run (i.e., they are *one-shot*). However, this natural behaviour turns out to be unexpectedly hard to achieve in lattice-based proofs. There are some works, e.g., [Lyu09, Lyu12, BKLP15, BDL+18, LN17] that address this problem in lattice-based cryptography and provide one-shot proofs in the context of protocols that work with "basic" witness extraction. On the other hand, recent research in the DL setting [GK15, BCC+15, BCC+16, BBB+18] has shown that for proving certain non-linear relations in the secret witness, it is possible to construct more efficient proofs that *require* a *"complex"* witness extraction involving more than two accepting protocol transcripts (and thus more than two challenges) for recovering prover's secret (i.e., the protocols are *many*-special sound). Such proofs rely on higher degree relations to obtain compact results, unlike the 2-special sound proofs that can only check linear (first degree) relations (we refer to the aforementioned works for the motivation behind proving high-degree relations). Again, in the DL setting, these proofs work smoothly and are easily one-shot. However, in the lattice setting, the situation is much more complicated.

Focusing on one-shot proofs, the protocols in [BKLP15] and [BDL+18] are important for our purposes as these protocols explicitly make use of lattice-based commitments similar to the works in the thesis. In fact, the ideas in aforementioned works date back to the works by Lyubashevsky [Lyu09, Lyu12] introducing the "Fiat-Shamir with Aborts" technique in lattice-based cryptography. The advantage of these works is that the (underlying) protocols achieve a negligible soundness error in a single run, which makes them very efficient in practice. However, all these approaches are limited to working with "basic" witness extraction except for a specific multiplicative (second degree) relation in [BKLP15]. The multiplicative argument in [BKLP15] is to prove that the coefficient of a quadratic term is zero and no explicit witness extraction from this non-linear relation is provided (and, indeed, no witness extraction from this second degree relation is needed as witnesses are extracted from the linear relations). Additionally, all these one-shot proofs introduce new complications (more precisely, *relaxations* in the relation being proved) as we discuss in detail in Section 5.3.

Another line of research makes use of *multi-shot* proofs that require multiple protocol repetitions to get a negligible soundness error. Stern-like combinatorial protocols [Ste96] fall into this category, where one needs at least λ protocol repetitions for λ-bit security. Even though these approaches have a wide range of applications such as group and ring signatures as in [LLNW16] with signature length logarithmic in the group/ring size, but quadratic in the security parameter λ, the latter inefficiency in terms of the security parameter makes them seem to fall far behind practical expectations (see Table 1.1 for the concrete results of [LLNW16]). Another approach for proving relations without a relaxation (approximation) factor is to use binary challenges combined with "Fiat-Shamir with Aborts" technique. Again, a single iteration of such a protocol has a soundness error of only $1/2$, and thus requires $O(\lambda)$ repetitions.

In the ring $R = \mathbb{Z}[X]/(X^d + 1)$, it is possible to achieve a soundness error of $1/(2d)$ using the *monomial challenges* from [BCK+14]. Here the challenges are of the form X^i for some $0 \leq i < 2d$ (i.e., there are $2d$ possible challenges in total), and it is shown in [BCK+14] that doubled inverses of challenge differences are short (more precisely, $\left\|2(X^i - X^j)^{-1}\right\| \leq \sqrt{d}$ for $i \neq j$ as recalled in Lemma 3.20). Still proofs

using monomial challenges require at least 10 repetitions for a typical ring dimension $d \leq 2048$. To summarise, for a soundness goal of $2^{-\lambda}$, all the above multi-shot approaches produce proofs of length $\widetilde{O}(\lambda^2)$, as a function of the security parameter λ.

One can also get *asymptotically* efficient lattice-based proofs for arithmetic circuits when the circuit size is large compared to the security parameter λ using the amortisation techniques from [BBC+18]. However, these techniques do not seem to be helpful when the proved relations do not necessarily require a large circuit and still the quadratic increase in the security parameter λ is not avoided.

2.2 Ring Signatures

Ring signatures, introduced by Rivest, Shamir and Tauman-Kalai [RST01], offer a way for anonymous signature generation in that the signer's identity is hidden within a set of identities, called a *ring*. That is, the outside world only knows that one of the *ring* members generated the signature, unable to determine which one exactly. The rigorous security notions of ring signatures were established in the work of Bender, Katz and Morselli [BKM09].

An important aspect of ring signatures in practice is the signature length and its growth with the ring size. One may distinguish in the literature two broad types of ring signatures. The first type, "linear size" ring signatures, has ring signature length linear in the size of the ring and thus does not scale well to very large rings. The second type, "log size" ring signatures, has length that increases only poly-logarithmically with the size of the ring, and thus can be efficient even for very large rings. In this thesis, the focus is mainly on the second type, but the log size ring signature constructions introduced in the thesis reach the same efficiency level of linear size proposals based on comparable security assumptions in terms of length even for very small ring sizes.

Two important log size ring signatures based on classical assumptions are due to Groth and Kohlweiss [GK15] and Bootle et al. [BCC+15], where the main ideas in the latter are borrowed from the former. In [GK15, BCC+15], the authors first describe efficient (in terms of communication complexity) one-out-of-many proofs, which then enable them to design short ring signatures in the DL setting. There are also very recent log size ring signatures proposed in [YSL+19, LRR+19] using Bulletproofs proof system [BBB+18].

On the side of the lattice setting, most of the existing ring signature schemes (e.g., [MBB+13, TSS+18, BLO18]) have linear size. [ZZTA18] attempts to extend Groth-Kohlweiss' scheme [GK15] by replacing Pedersen commitment with a lattice-based commitment scheme. It is claimed that the security requirements for the instantiation with this lattice-based commitment follows from the results of [GK15]. However, as detailed in [ESS+19], this does not hold true without addressing some low-level technical issues, which is also hinted in the works [LLNW16, BLO18] by noting that Groth-Kohlweiss' scheme does not easily extend to the lattice setting.

Prior to the constructions to be described in this thesis, this leaves us with the work of Libert et al. [LLNW16] (and a follow-up by [YAL+17], adding linkability to [LLNW16]) as the only log size ring signature from lattices. In [LLNW16], the authors first design an accumulator through a Merkle tree using SIS-based hash function. Zero-knowledge membership arguments are then built for this accumulator. Having these building blocks, the authors propose ring and group signatures, both of which are log size in the number of users involved. We therefore mainly focus on [LLNW16] for efficiency comparison purposes. For example, even for the smallest ring size of 2, the signature length of [LLNW16] is well above 10 MB.

Very recently, another lattice-based ring signature is proposed in [YAZ+19] using *exact* lattice-based proofs. However, as can be seen from Table 1.1, the practical efficiency of this scheme seems to be far behind practical expectations.

There are other "post-quantum" ring signatures whose security rely on symmetric-key primitives alone. The proposals in [DRS18] and [KKW18] are examples of such log size ring signatures. These proposals are instantiated using LowMC cipher [ARS+15], which is specially designed to reduce the multiplicative complexity of the circuit implementing the algorithm. LowMC is a relatively new symmetric-key cipher and its security seems not as well understood as more established ciphers such as AES. A recent study [dSGDOS19] shows that replacing LowMC with AES in Picnic signature [CDG+19], which is an (ordinary) signature scheme submitted to the NIST PQC process, increases the signature length by a factor of at least 2.5. Therefore, it seems plausible to say that the ring signature lengths in [DRS18] and [KKW18] would increase by a similar factor when instantiated with AES. Even the constructions based on LowMC are significantly longer in comparison to our improved constructions as given in Table 1.1.

2.3 Group Signatures

A group signature is another example of an anonymous signature, where the signatory signs a message on behalf of a group of users. In general, it is similar to a ring signature, but the important difference is that there exists a *group manager*, who can reveal a signatory's identity, e.g., in case of a dispute or a misbehaviour. Group signatures were introduced by Chaum and van Heyst [CvH91] and viable construction were provided in [ACJT00]. Rigorous security notions of group signatures were established in the work of Bellare, Micciancio and Warinschi [BMW03]. The first lattice-based group signature scheme was proposed by Gordon, Katz and Vaikuntanathan [GKV10]. A number of follow-up works appeared in the literature improving upon the efficiency, e.g., [LLLS13, NZZ15, LNW15] or the functionalities provided, e.g., [LLNW14]. All of these schemes make use of a *trapdoor* (usually referred as a GPV trapdoor [GPV08]), which creates a bottleneck in terms of efficiency. On contrary, the work by Benhamouda et al. [BCK+14] does not make use of a trapdoor, but it is not fully lattice-based in that it combines lattice-based assumptions with DLP-related ones. As mentioned before, Libert et al. [LLNW16] introduce a log size lattice-based accumulator, which is then used to build a group signature. This work was followed up by more recent ones such as [LLM+16, LNWX17], where group signatures with additional features are proposed.

In a recent work, del Pino, Lyubashevsky and Seiler [dPLS18] introduce new techniques in building sub-linear size group signatures from lattices. Their construction relies on zero-knowledge proofs of automorphism stability, which allows proving knowledge of secrets that remain stable under certain automorphisms. The group signature length remains constant even up to a huge group size of 2^{80}, but the concrete signature length for typical security levels is still relatively large. In particular, it is around 581 KB for a group of size at most 2^{80} (for "CPA-anonymous" version).

Another recent "post-quantum" group signature with logarithmic length is due to Katz, Kolesnikov and Wang [KKW18]. The construction is based on symmetric-key primitives alone, and is in particular instantiated with LowMC cipher [ARS+15]. As discussed earlier, the use of LowMC cipher helps reduce the signature length significantly by minimising the multiplicative complexity of the cipher used. We compare our group signature proposal to these state-of-the-art works in Table 6.6, which shows

that our proposal produces significantly shorter group signatures for groups of size millions and beyond.

2.4 RingCT Protocols

In a blockchain environment, the two main entities for the purposes of this thesis are *spenders*, who create a transaction and its proof of validity, and *verifiers*, who check the validity of a transaction and its proof. There are also *recipients* of the transactions, but they do not play a central role in RingCT system.

A RingCT [Noe15] protocol allows users to create *confidential transactions* on blockchain so that the spender's identity as well as the transaction amount is hidden from the outside world. Additionally, the protocol must guarantee that the transaction is a *valid* spending. This is mainly captured by a fundamental property, *balance*, which requires that the total amount being spent by the spender is exactly equal to the total amount received by the recipients where no double-spending or negative amount spending occurs.

Monero cryptocurrency is currently the most prominent application that heavily relies on RingCT protocol to provide privacy-preserving solutions. It also uses *stealth addresses* to allow the users to hide the recipient's identity. *Auditability* (i.e., the ability of an authority to trace real spenders) is another important feature required to prevent illegal use of a cryptocurrency, and is desired, e.g., for regulatory or financial enterprise applications.

Previous RingCT protocols, e.g. [Noe15, SALY17], make use of three core ingredients: 1) a homomorphic commitment scheme, allowing one to hide some secret with the ability to later reveal it while ensuring that the secret committed in the first place and the opened one are the same, 2) a linkable ring signature [RST01, LWW04] (or one-out-of-many proof), allowing one to prove knowledge of a secret key corresponding to an (undisclosed) element in a set of public keys, and 3) a range proof, showing that a secret committed value falls within a certain range. In RingCT protocol, the transaction amount is hidden via the use of the commitment scheme, and the spender's identity is hidden via the use of the ring signature. The main purpose of the range proof is for guaranteeing the validity of a transaction by proving that the real transaction amount hidden in a commitment is in a valid (positive) range.

The first RingCT protocol was introduced in [Noe15] (called RingCT 1.0, hereafter), and more formal definitions were then provided in [SALY17] (called RingCT 2.0, hereafter). Both of these solutions are in the DL setting and the latter requires a trusted setup, which undermines the idea of a blockchain environment where there is no particular trusted authority. Very recently, another DL-based RingCT is proposed in [YSL+19] (called RingCT 3.0, hereafter), where the security model is also improved over RingCT 2.0.

The RingCT correctness and security models defined in RingCT 2.0 and 3.0 have some unsatisfactory aspects. In particular, they seem complicated to understand, and do not capture the inherent stateful nature of a blockchain system. Therefore, there are gaps in some definitions. For example, the balance definition in RingCT 3.0, which requires *all* input accounts to be uncorrupted, seems to include a very strong assumption that leaves a real-life attack out of the scope of the security model. In this thesis, we introduce a new set of formal definitions for RingCT-like blockchain protocols in Section 6.3, where a more detailed comparative discussion between our model and the prior ones is provided.

TABLE 2.1: Overview of RingCT proposals. The scalability referred here is in terms of anonymity set size.

	No trusted setup	Post-quantum	Scalable	Multiple in/outputs	Efficient Implement.
RingCT 1.0 [Noe15]	✓	✗	✗	✓	✗
RingCT 2.0 [SALY17]	✗	✗	✓	✓	✗
RingCT 3.0 [YSL+19]	✓	✗	✓	✓	✗
Omniring [LRR+19]	✓	✗	✓	✓	✓
LRCT v1.0 [TSS+18]	✓	✓	✗	✗	✗
LRCT v2.0 [TKS+19]	✓	✓	✗	✓	✗
MatRiCT, Chapter 6	✓	✓	✓	✓	✓

In a concurrent and independent work [LRR+19], another set of formal definitions for RingCT is provided together with a construction in the DL-setting. As in [YSL+19], this construction also makes use of advanced DL-based ZKPs, namely Bulletproofs [BBB+18].

In the post-quantum world, an initial attempt to design a lattice-based RingCT was done by Torres et al. in [TSS+18] (called LRCT v1.0, hereafter). This protocol is restricted to the single-input single-output wallets, i.e., the user spends a single account to a single output address, and therefore the balance property is easy to satisfy. Moreover, it does not involve a range proof to make sure that the amount being spent is positive. Very recently, LRCT v1.0 was extended to support multiple-input and multiple-output wallets in [TKS+19] (called LRCT v2.0, hereafter), where a range proof is used to prove balance. However, the concrete efficiency of this scheme is far behind practical expectations (see Table 6.4 in Chapter 6).

In Chapter 6, we introduce an efficient post-quantum RingCT protocol, named MatRiCT. Two crucial advantages of MatRiCT over LRCT v2.0 are that 1) the underlying ZKPs of MatRiCT reach a convincing soundness level in a single execution, and thus no protocol repetition is required, and 2) MatRiCT does not require a range proof on a 64-bit range even though 64-bit amounts are allowed (note that a single 64-bit range proof from lattices alone costs at least about 100 KB currently). An overview of existing RingCT proposals is given in Table 2.1.

Another project to design a post-quantum privacy-preserving cryptocurrency has been initiated in [Fou18], where lattice-based techniques are to be used. Though there is currently no concrete scheme available, the authors mention that they aim to design a ring signature of size less than 400 KB for rings of 2^{15} users and that the range proof is expected to cost a few hundred KB. Our results are far ahead of these goals (see Tables 1.1 and 6.4).

Chapter 3

Preliminaries

This chapter covers the preliminaries made use of in the next chapters. First, commonly used notations are introduced. Then, we cover the cryptographic definitions, namely sigma protocols, our security assumptions, commitment schemes and ring signatures. Finally, relevant mathematical background is covered, including some basics from Linear Algebra, discrete Gaussian distribution and rejection sampling as used in our protocols later on.[1]

3.1 Notations

$\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ denotes the ring of integers modulo q represented by the range $\left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$ where q is an odd positive integer. $[a, b] = \{a, \ldots, b\}$, and $[a, b) = \{a, \ldots, b-1\}$ for $a < b \in \mathbb{Z}$. Logarithms are base 2 unless explicitly specified otherwise.

Matrices and vectors.

Throughout the thesis, bold-face lower-case letters such as $\boldsymbol{x}$ are used to denote column vectors and bold-face capital letters such as $\boldsymbol{A}$ to denote matrices with $\boldsymbol{I}_n$ being the n-dimensional identity matrix. $(\boldsymbol{x}, \boldsymbol{y})$ denotes appending the vector $\boldsymbol{y}$ to the vector $\boldsymbol{x}$ to form a single longer vector.

For a vector $\boldsymbol{v} = (v_0, \ldots, v_{n-1})$, the Euclidean, infinity and ℓ_1 norms are defined as

$$\begin{aligned} \|\boldsymbol{v}\| &= \sqrt{\sum_{i=0}^{n-1} v_i^2}, \\ \|\boldsymbol{v}\|_\infty &= \max_{0 \le i \le n-1} |v_i|, \quad \text{and} \\ \|\boldsymbol{v}\|_1 &= \sum_{i=0}^{n-1} |v_i|. \end{aligned}$$

Algorithmic notations.

We denote the main security parameter by λ and say that a function $\nu(\lambda)$ is negligible (denoted by $\nu = \mathrm{negl}(\lambda)$) if $\nu(\lambda) < 1/2^{c\lambda}$, for a constant $c > 0$.[2] $a \leftarrow \mathcal{Z}$ means a is chosen uniformly from a set $\mathcal{Z}$. If $\mathcal{Z}$ is a distribution, we use the same notation to sample a from a distribution $\mathcal{Z}$. In the case that $\mathcal{Z}$ is an algorithm, the same notation

[1] This chapter is partly based on [ESS+19, ESLL19, EZS+19].

[2] Note that we use a more practically-oriented definition of being negligible as opposed to more theoretical works, which use $\nu(\lambda) < 1/\lambda^c$ for any $c > 0$ and all sufficiently large λ.

is used to denote that the algorithm outputs a. $\mathcal{U}(S)$ denotes the uniform distribution on a set S.

Polynomials and polynomial rings.

For a polynomial p, the corresponding norms are defined analogously on the coefficient vector of p. For a vector $\boldsymbol{p} = (p_0, \ldots, p_{s-1})$ of polynomials, $\|\boldsymbol{p}\| = \sqrt{\sum_{i=0}^{s-1} \|p_i\|^2}$, $\|\boldsymbol{p}\|_1 = \sum_{i=0}^{s-1} \|p_i\|_1$, $\|\boldsymbol{p}\|_\infty = \max_{0 \le i \le s-1} \|p_i\|_\infty$, and $\mathsf{HW}(\boldsymbol{p})$ denotes the Hamming weight of the coefficient vector of $\boldsymbol{p}$.

We define the rings $R = \mathbb{Z}[X]/(X^d + 1)$ and $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ where $d > 1$ is a power of 2. $\mathfrak{U}_c$ defines the set of all polynomials in R with infinity norm at most $c \in \mathbb{Z}^+$. When sampling polynomials in R, we write $\boldsymbol{p} \leftarrow S^{md}$ to indicate that $\boldsymbol{p} \in R^m$ is a vector of m polynomials where each coefficient is sampled uniformly at random from a set S (i.e., md coefficients are sampled in total).

3.2 Cryptographic Definitions

3.2.1 Security assumptions: Module-SIS and Module-LWE

The constructions studied in this thesis rely mainly on two well-studied assumptions in lattice-based cryptography, namely Module Short Integer Solution (M-SIS) and Module Learning With Errors (M-LWE) [LS15]. They can be seen as generalisations of SIS [Ajt96] and LWE [Reg09] problems. Module variants of these two problems are defined over a ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ for a positive integer "modulus" q and a power-of-two ring dimension d.

Definition 3.1 ($\text{M-SIS}_{n,m,q,\beta_{\text{SIS}}}$)**.** *Let $R_q = \mathbb{Z}_q[X]/(X^d + 1)$. Given $\boldsymbol{A} \leftarrow \mathcal{U}\left(R_q^{n \times m}\right)$, find $\boldsymbol{z} \in R_q^m$ such that $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{0} \bmod q$ and $0 < \|\boldsymbol{z}\| \le \beta_{\text{SIS}}$.*

As in [BDL+18], one can define M-SIS in "Hermite normal form" such that $\boldsymbol{A} = [\,\boldsymbol{I}_n \,\|\, \boldsymbol{A'}\,] \in R_q^{n \times m}$ and $\boldsymbol{A'} \leftarrow \mathcal{U}\left(R_q^{n \times (m-n)}\right)$. This standard variant is known to be as hard as the given M-SIS definition above. When we want to be more explicit, we call the definition in "Hermite normal form" as *M-SIS in HNF*.

For simplicity, we consider a special case of M-LWE problem where each error and secret key coefficient is sampled uniformly from $\{-\mathcal{B}, \ldots, \mathcal{B}\}$ for some $\mathcal{B} \in \mathbb{Z}^+$. A more special case of $\mathcal{B} = 1$ is commonly practised in recent lattice-based proposals such as [BDL+18, LN17, dPLS18]. The secret key coefficients can equivalently be sampled uniformly from $\mathbb{Z}_q$.

Definition 3.2 ($\text{M-LWE}_{n,m,q,\mathcal{B}}$)**.** *Let $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ and $\boldsymbol{s} \leftarrow \mathfrak{U}_{\mathcal{B}}^n$ be a secret key. Define $\mathtt{LWE}_{q,\boldsymbol{s}}$ as the distribution obtained by sampling $e \leftarrow \mathfrak{U}_{\mathcal{B}}$, $\boldsymbol{a} \leftarrow R_q^n$ and returning $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e)$. Given m samples from either $\mathtt{LWE}_{q,\boldsymbol{s}}$ or $\mathcal{U}(R_q^n, R_q)$, the problem asks to distinguish which is the case.*

When R_q is set to be $\mathbb{Z}_q$ in M-SIS and M-LWE definitions, one obtains the plain SIS and LWE definitions, respectively. Therefore, module variants can be seen as a generalisation of the original forms.

The hardness of M-LWE against currently known attacks seems not to be affected significantly by the number of samples m unless it is cubic in the overall dimension parameter. In particular, with respect to the Definition 3.2, $m \approx (n \cdot d)^3$ samples are needed for the attack in [AG11]. The number of samples in our constructions is

always significantly smaller than this value, and thus m does not play a crucial role in parameter settings.

It is also clear from the respective definitions that M-SIS problem gets harder as β_{SIS} gets smaller, and M-LWE problem gets harder when the error is sampled from a wider distribution (i.e., as $\mathcal{B}$ increases). Therefore, when estimating practical security of a construction against known attacks, it is enough to investigate the easiest cases, i.e., when $\mathcal{B}$ or the standard deviation of the error distribution (in case the error is sampled from a Gaussian distribution) is the smallest for M-LWE, and when β_{SIS} is the largest for M-SIS.

3.2.2 Commitment schemes

Commitment schemes are very powerful tools used in many cryptographic protocols. They consist of three algorithms as below.

- $\mathsf{CKeygen}$ is a PPT algorithm that, on input security parameter 1^λ, outputs the public parameters pp together with the specifications of a message space $\mathcal{S}_M$, a randomness space $\mathcal{S}_R$ and a commitment space $\mathcal{S}_C$.
- Commit is a PPT algorithm that, on input public parameters pp and a message $M \in \mathcal{S}_M$, outputs a commitment $C \in \mathcal{S}_C$.
- COpen is a deterministic polynomial-time algorithm that, on input public parameters pp, a tuple $(M, r; C) \in \mathcal{S}_M \times \mathcal{S}_R \times \mathcal{S}_C$, outputs a bit b, indicating 'accept' when $b = 1$, and 'reject' otherwise.

Some lattice-based commitment schemes, as those used in this thesis, have an additional input y, called *the relaxation factor*, to the COpen algorithm, which is also parameterised by a norm bound γ_{com}. These are crucial differences of lattice-based commitment schemes and they will become clear when the concrete instantiations are defined further below. We also provide an example of a proof relation at the end of Section 3.2.3, where the role of the relaxation factor is shown.

Two important properties of commitment schemes are *hiding* and *binding*, and in this thesis, we are interested in *computational* variants. *Computational hiding* property is satisfied if the following holds for all PPT algorithms $\mathcal{A}$.

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{CKeygen}(1^\lambda); (M_0, M_1) \leftarrow \mathcal{A}^{\mathsf{CKeygen}(pp)} \\ b \leftarrow \{0,1\};\ C \leftarrow \mathsf{Commit}_{pp}(M_b) \end{array} : \mathcal{A}(C) = b\right] \leq 1/2 + \text{negl}(\lambda).$$

Computational strong γ_{com}*-binding* property is satisfied if the following holds for all PPT algorithms $\mathcal{A}$.

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{CKeygen}(1^\lambda); \\ (C, \boldsymbol{t}_0, \boldsymbol{t}_1) \leftarrow \mathcal{A}(pp) \end{array} : \begin{array}{c} (M_0, r_0) \neq (M_1, r_1) \wedge \\ \mathsf{COpen}_{pp}(C, \boldsymbol{t}_0) = \mathsf{COpen}_{pp}(C, \boldsymbol{t}_1) = 1 \end{array}\right] \leq \text{negl}(\lambda),$$

where $\boldsymbol{t}_i = (y_i, M_i, r_i)$ for $i = 0, 1$ and the norm bound parameter in COpen is γ_{com}. In the case of *computational* γ_{com}*-binding*, the requirement $(M_0, r_0) \neq (M_1, r_1)$ in strong binding is replaced with $M_0 \neq M_1$.

Algebraic lattice-based protocols mainly make use of two commitment schemes: *Unbounded-Message Commitment* (UMC) [BKLP15, BDL+18] and *Hashed-Message Commitment* (HMC) (derived from Ajtai's SIS hash function [Ajt96]). A statistically-hiding variant of HMC over $\mathbb{Z}_q$ is also provided in [KTX08], and HMC is already used prior works such as [BKLP15, BDL+18]. We define the commitment schemes over module lattices and rely on computational hiding and binding properties as in [BDL+18].

These commitment schemes offer different tradeoffs. For example, UMC allows one to commit to messages of unbounded length but the commitment vector dimension increases linearly with the message vector dimension in a commitment. For HMC, on the other hand, one can only commit to messages of bounded length (when binding is based on M-SIS) but the height of the commitment vector is independent of the message vector dimension. Thus, one can commit to long message vectors without significantly increasing the commitment vector size using HMC.

Let $n, m, \mathcal{B}, q$ be positive integers, and assume that we commit to v-dimensional vectors over R_q for $v \geq 1$. As mentioned, the opening algorithm COpen is *relaxed* in the sense that there is an additional input $y \in R_q$, called *relaxation factor*, to COpen algorithm along with a message-randomness pair $(\boldsymbol{m}', \boldsymbol{r}')$ such that COpen checks if $y \cdot C = \mathrm{Com}_{ck}(\boldsymbol{m}'; \boldsymbol{r}')$.

Instantiation of HMC

The message space of HMC consist of elements of R_q with small norm. How small a message needs to be depends on the application, but in any case we must have $\|\boldsymbol{m}\|_\infty < q$ as indicated by COpen below. The instantiation of HMC with $m > n$ is as follows.

- $\mathsf{CKeygen}(1^\lambda)$: Pick $\boldsymbol{G}'_r \leftarrow R_q^{n\times(m-n)}$ and $\boldsymbol{G}_m \leftarrow R_q^{n\times v}$. Output $ck = \boldsymbol{G} = [\,\boldsymbol{G}_r \,\|\, \boldsymbol{G}_m\,] \in R_q^{n\times(m+v)}$ where $\boldsymbol{G}_r = [\,\boldsymbol{I}_n \,\|\, \boldsymbol{G}'_r\,]$. We assume that Commit and COpen takes ck as an input implicitly.
- $\mathsf{Commit}(\boldsymbol{m})$: Pick $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$. Output

$$\mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r}) = \boldsymbol{G} \cdot \begin{pmatrix} \boldsymbol{r} \\ \boldsymbol{m} \end{pmatrix} = \boldsymbol{G}_r \cdot \boldsymbol{r} + \boldsymbol{G}_m \cdot \boldsymbol{m}.$$

- $\mathsf{COpen}(C, (y, \boldsymbol{m}', \boldsymbol{r}'))$: If $\mathrm{Com}_{ck}(\boldsymbol{m}'; \boldsymbol{r}') = yC$ and $\|(\boldsymbol{r}', \boldsymbol{m}')\| \leq \gamma_{\mathrm{com}}$, return 1. Otherwise, return 0.

It is easy to see from the above definition that the commitment key $\boldsymbol{G}$ is in Hermite normal form. In this case, it is easy to show the following lemma, whose variant also appears in [BDL+18].

Lemma 3.3. *HMC with a commitment key $\boldsymbol{G}$ in HNF as above is*

- *computationally hiding if M-LWE$_{m-n,n,q,\mathcal{B}}$ problem is hard, and*
- *computationally strong γ_{com}-binding with respect to the same relaxation factor y if M-SIS$_{n,m+v,q,2\gamma_{\mathrm{com}}}$ in HNF is hard.*

Furthermore, if M-LWE$_{m-n,n,q,\mathcal{B}}$ problem is hard, commitment to any message is computationally indistinguishable from a uniformly random element in R_q^n.

Proof. Let $(y, \boldsymbol{m}, \boldsymbol{r})$ and $(y, \boldsymbol{m}', \boldsymbol{r}')$ with $(\boldsymbol{r}, \boldsymbol{m}) \neq (\boldsymbol{r}', \boldsymbol{m}')$ be two valid openings of a commitment C. That is,

$$yC = \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r}) = \mathrm{Com}_{ck}(\boldsymbol{m}'; \boldsymbol{r}') \quad \text{and} \quad \|(\boldsymbol{r}, \boldsymbol{m})\|, \|(\boldsymbol{r}', \boldsymbol{m}')\| \leq \gamma_{\mathrm{com}}.$$

Therefore, we have $yC = \boldsymbol{G}\cdot(\boldsymbol{r}, \boldsymbol{m}) = \boldsymbol{G}\cdot(\boldsymbol{r}', \boldsymbol{m}')$, which implies $\boldsymbol{G}\cdot(\boldsymbol{r}-\boldsymbol{r}', \boldsymbol{m}-\boldsymbol{m}') = 0$. Hence, $(\boldsymbol{r}-\boldsymbol{r}', \boldsymbol{m}-\boldsymbol{m}')$ is a solution to M-SIS$_{n,m+v,q,2\gamma_{\mathrm{com}}}$ problem (in Hermite normal form). This proves the computational strong binding with respect to the same relaxation factor y.

For the hiding property, we can write $\mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r}) = \boldsymbol{G}_r \cdot \boldsymbol{r} + \boldsymbol{G}_m \cdot \boldsymbol{m} = \boldsymbol{r}_0 + \boldsymbol{G}'_r \cdot \boldsymbol{r}_1 + \boldsymbol{G}_m \cdot \boldsymbol{m}$ where $\boldsymbol{r} = (\boldsymbol{r}_0, \boldsymbol{r}_1)$ since $\boldsymbol{G}_r = [\,\boldsymbol{I}_n \,\|\, \boldsymbol{G}'_r\,]$. The result of the computation

$\boldsymbol{r}_0 + \boldsymbol{G}'_r \cdot \boldsymbol{r}_1$ gives n M-LWE samples with $\boldsymbol{r}_1 \in \mathfrak{U}_{\mathcal{B}}^{m-n}$ as the secret key. Therefore, if M-LWE$_{m-n,n,q,\mathcal{B}}$ is hard, $\boldsymbol{r}_0 + \boldsymbol{G}'_r \cdot \boldsymbol{r}_1$ looks uniformly random in R_q^n and so does commitments to any message. □

If we replace the commitment key $\boldsymbol{G}$ by a completely random matrix, then still a very similar result as below holds.

Lemma 3.4. *Let s be the number of fields $\mathbb{F}_{p_1}, \ldots, \mathbb{F}_{p_s}$ the ring R_q (for some $q \in \mathbb{Z}^+$) splits into, and $p = \min\{p_1, \ldots, p_s\}$. Assume that $\frac{n \cdot s}{p^{m-n+1}}$ is negligible. Then, HMC with a completely random commitment key $\boldsymbol{G}$ is*

- *computationally hiding if M-LWE$_{m-n,m,q,\mathcal{B}}$ problem is hard, and*
- *computationally strong γ_{com}-binding with respect to the same relaxation factor y if M-SIS$_{n,m+v,q,2\gamma_{\text{com}}}$ is hard.*

Furthermore, if M-LWE$_{m-n,m,q,\mathcal{B}}$ problem is hard, commitment to any message is computationally indistinguishable from a uniformly random element in R_q^n.

Proof Sketch. The proof of the binding property is analogous to the binding proof of Lemma 3.3.

To prove the hiding property, we draw attention to the "duality" between the knapsack problems and LWE, which has already been noticed in cryptography (see, e.g., [MM11, BDL+18]). Given $(\boldsymbol{A}, \boldsymbol{u})$ for $\boldsymbol{A} \leftarrow R_q^{n \times m}$, the goal in the knapsack problem considered in this work is to distinguish between the cases: 1) $\boldsymbol{u} \leftarrow R_q^n$ and 2) $\boldsymbol{u} = \boldsymbol{A}\boldsymbol{x}$ for $\boldsymbol{x} \leftarrow \mathfrak{U}_{\mathcal{B}}$ for some $\mathcal{B} \in \mathbb{Z}^+$. In Lemma 4.8 and Lemma 4.9 of [MM11], it is shown that this knapsack problem is as hard as LWE$_{m-n,m,q,\mathcal{B}}$ when one works over $\mathbb{Z}_q$. The result extends to the ring case provided that the probability of $\boldsymbol{A}$ being singular is negligible. We can argue this as follows.

$\boldsymbol{A}$ is non-singular if it is non-singular over all the fields the ring R_q (for some $q \in \mathbb{Z}^+$) splits into. The probability that a random $n \times m$ matrix $\boldsymbol{A}$ is full-rank over $\mathbb{F}_p$ is at least $\left(1 - 1/p^{m-n+1}\right)^n$ (see, e.g., [FG15] and the references therein). Then, the probability that $\boldsymbol{A}$ is full-rank over all the s fields is at least $\left(1 - 1/p^{m-n+1}\right)^{n \cdot s} \geq 1 - \frac{ns}{p^{m-n+1}} = 1 - \text{negl}(\lambda)$ by assumption. This concludes the argument. □

For the setting where we use this lemma, the parameters satisfy the following. 1) $m \geq 2n$, 2) $p \gg 2^{20}$ and 3) the modulus q has at most 2 prime factors and thus s is at most $2d$ (note that the polynomial $X^d + 1$ can split into at most d factors). From here, we can easily conclude that the probability of $\boldsymbol{A}$ being singular over R_q is negligibly small.

Since M-SIS in HNF is equivalent to original M-SIS and the number of samples in M-LWE problem does not play a crucial role in our estimations of practical security, both of the instantiations of HMC are in equal positions in terms of security.

Instantiation of UMC

Unlike HMC, UMC allows arbitrarily long messages as inputs. The instantiation of UMC is similar to that of HMC and defined as below for $m > n + v$.

- $\mathsf{CKeygen}(1^\lambda)$: Pick $\boldsymbol{G}'_1 \leftarrow R_q^{n \times (m-n)}$ and $\boldsymbol{G}'_2 \leftarrow R_q^{v \times (m-n-v)}$. Set $\boldsymbol{G}_1 = [\,\boldsymbol{I}_n \,\|\, \boldsymbol{G}'_1\,]$ and $\boldsymbol{G}_2 = [\,\boldsymbol{0}^{v \times n} \,\|\, \boldsymbol{I}_v \,\|\, \boldsymbol{G}'_2\,]$. Output $ck = \boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{G}_2 \end{bmatrix} \in R_q^{(n+v) \times m}$. We assume that Commit and COpen takes ck as an input implicitly.
- $\mathsf{Commit}(\boldsymbol{m})$: Pick $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$. Output

$$\text{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r}) = \boldsymbol{G} \cdot \boldsymbol{r} + (\boldsymbol{0}^n, \boldsymbol{m}).$$

- $\mathsf{COpen}(C, (y, \boldsymbol{m}', \boldsymbol{r}'))$: If $\mathrm{Com}_{ck}(\boldsymbol{m}'; \boldsymbol{r}') = yC$ and $\|\boldsymbol{r}'\| \leq \gamma_{\mathrm{com}}$, return 1. Otherwise, return 0.

Observe from the above definition that only the norm of $\boldsymbol{r}'$ is checked in the COpen algorithm of UMC whereas that of $(\boldsymbol{m}', \boldsymbol{r}')$ is checked in HMC. Also, our definition of COpen for UMC is slightly different than that in [BDL+18] because we do not multiply the relaxation factor with the message as the invertibility of the relaxation factor y is not assumed in our case.

Lemma 3.5 ([BDL+18]). *UMC defined above is*

- *computationally hiding if M-LWE$_{m-n-v,n+v,q,\mathcal{B}}$ problem is hard, and*
- *computationally γ_{com}-binding with respect to the same relaxation factor y if M-SIS$_{n,m,q,2\gamma_{\mathrm{com}}}$ is hard.*

Furthermore, if M-LWE$_{m-n-v,n+v,q,\mathcal{B}}$ problem is hard, commitment to any message is computationally indistinguishable from a uniformly random element in R_q^{n+v}.

We use the same notation for both of the commitment schemes and will clarify in the relevant sections which specific instantiation is used. We say that $(y, \boldsymbol{m}', \boldsymbol{r}')$ is a *valid* opening of C if $\mathsf{COpen}(C, (y, \boldsymbol{m}', \boldsymbol{r}')) = 1$. A valid opening $(y, \boldsymbol{m}', \boldsymbol{r}')$ with $y = 1$ is called an *exact valid* opening. We call the message part $\boldsymbol{m}'$ of an opening as *message opening*, and if $(y, \boldsymbol{m}', \boldsymbol{r}')$ is a valid opening such that $yC = \mathrm{Com}_{ck}(y\boldsymbol{m}'; \boldsymbol{r}')$, then we call $\boldsymbol{m}'$ a *relaxed message opening* with relaxation factor y. It is also straightforward that both UMC and HMC satisfy the following homomorphic properties: $\mathrm{Com}_{ck}(\boldsymbol{m}_0; \boldsymbol{r}_0) + \mathrm{Com}_{ck}(\boldsymbol{m}_1; \boldsymbol{r}_1) = \mathrm{Com}_{ck}(\boldsymbol{m}_0 + \boldsymbol{m}_1; \boldsymbol{r}_0 + \boldsymbol{r}_1)$ and $c \cdot \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r}) = \mathrm{Com}_{ck}(c \cdot \boldsymbol{m}; c \cdot \boldsymbol{r})$ for *short* $c \in R_q$.

For HMC, we see that M-SIS security increases with n whereas M-LWE security increases with $m - n$. On the other hand, for UMC, M-SIS security increases with n while M-LWE security increases with $m - n - v$. In the constructions to be described, these two security aspects are balanced when setting the concrete parameters. Moreover, for a scheme using UMC, the parameter setting is done using the results of Lemma 3.5. Similarly, for a scheme using HMC, the parameter setting is done using the results of Lemma 3.3 or Lemma 3.4.

For both of the commitment schemes, practical security estimations are done similarly. To estimate practical M-SIS hardness, we use the methodology in [MR09]. In particular, we say that the commitment scheme is γ-*binding* if the following is satisfied

$$\min\left\{q, 2^{2\sqrt{nd\log q \log \delta}}\right\} > 2\gamma, \tag{3.1}$$

where δ is the so-called "root Hermite factor", and n, d and q are the parameters in the definition of HMC/UMC. On the other hand, practical security of M-LWE against known attacks is estimated using the well-known "LWE estimator" due to Albrecht et al. [APS15]. We run the LWE estimator under both enumeration and sieving techniques, and ensure that both of them require a complexity of at least about 2^{λ}. We often adapt $\delta \approx 1.0045$ for $\lambda = 128$-bit post-quantum security.

3.2.3 Sigma protocols

Σ-protocols are a type of zero-knowledge proofs between two parties: the prover and the verifier. A language $\mathcal{L} \subseteq \{0,1\}^*$ is said to have a witness relationship $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ provided $v \in \mathcal{L}$ if and only if there exists $w \in \{0,1\}^*$ such that $(v, w) \in \mathcal{R}$. The quantity w is referred to as a witness for v. The definition of

Σ-protocols from [BCK+14] generalises the well-known notion of Σ-protocols. We further extend it to allow $(k+1)$-special soundness as in [GK15, BCC+15]. Our work focuses on *special honest-verifier zero-knowledge* since it suffices for non-interactive proof applications via Fiat-Shamir heuristic [FS86].

Definition 3.6 (Extension of Definition 2.5 in [BCK+14])**.** *Let $(\mathcal{P}, \mathcal{V})$ be a two-party protocol where $\mathcal{V}$ is a PPT algorithm, and $\mathcal{L}, \mathcal{L}'$ be languages with witness relations $\mathcal{R}, \mathcal{R}'$ such that $\mathcal{R} \subseteq \mathcal{R}'$. Then, $(\mathcal{P}, \mathcal{V})$ is called a Σ-protocol for $\mathcal{R}, \mathcal{R}'$ with completeness error α, a challenge set $\mathcal{C}$, public input v and private input w, if it satisfies the following conditions:*

- ***Three-move form:*** *The protocol has the following form. On input (v, w), $\mathcal{P}$ computes initial commitment t and sends it to $\mathcal{V}$. On input v, $\mathcal{V}$ draws a challenge $x \leftarrow \mathcal{C}$ and sends it to $\mathcal{P}$. The prover sends a response s to $\mathcal{V}$. The verifier accepts or rejects depending on the protocol transcript (t, x, s). The transcript (t, x, s) is called* accepting *if the verifier accepts the protocol run.*
- ***Completeness:*** *Whenever $(v, w) \in \mathcal{R}$, the honest verifier accepts with probability at least $1 - \alpha$ when interacting with an honest prover.*
- $(k+1)$***-special soundness:*** *There exists a PPT algorithm $\mathcal{E}$ (called* the extractor*) which takes $(k+1)$ accepting transcripts $(t, x_0, s_0), \ldots, (t, x_k, s_k)$ with pairwise distinct $x_i \in \mathcal{C}$ $(0 \leq i \leq k)$ as inputs, and outputs w' satisfying $(v, w') \in \mathcal{R}'$. We call this procedure* witness extraction*, and say that the protocol has a soundness error $\frac{k}{|\mathcal{C}|}$.* [3]
- ***Special honest-verifier zero-knowledge (SHVZK):*** *There exists a PPT algorithm $\mathcal{S}$ (called* the simulator*) that takes $v \in \mathcal{L}$ and $x \in \mathcal{C}$ as inputs, and outputs (t, s) such that (t, x, s) is (computationally) indistinguishable from an accepting protocol transcript generated by a real protocol run.*

Let us give an example pair of relations $(\mathcal{R}, \mathcal{R}')$ that is of interest in many lattice-based proof systems. The pair of relations below corresponds to a *relaxed* proof of knowledge (RPoK) of an opening of a commitment C under a commitment key ck.

$$\mathcal{R}_{\text{RPoK}}(\mathcal{T}) = \left\{ \; ((ck, C), (\boldsymbol{m}, \boldsymbol{r})) \; : \; \|(\boldsymbol{m}, \boldsymbol{r})\| \leq \mathcal{T} \; \wedge \; C = \text{Com}_{ck}\left(\boldsymbol{m}; \, \boldsymbol{r}\right) \; \right\}.$$
$$\mathcal{R}'_{\text{RPoK}}(\hat{\mathcal{T}}) = \left\{ \; ((ck, C), (y, \boldsymbol{m}', \boldsymbol{r}')) \; : \; \|(\boldsymbol{m}', \boldsymbol{r}')\| \leq \hat{\mathcal{T}} \; \wedge \; yC = \text{Com}_{ck}\left(\boldsymbol{m}'; \, \boldsymbol{r}'\right) \; \right\}.$$

Typically, we have $\mathcal{T} \leq \hat{\mathcal{T}}$ for some publicly known real numbers $\mathcal{T}, \hat{\mathcal{T}}$, and the relaxation factor y, which is part of the prover's witness, is a non-zero element of the set of challenge differences.

3.2.4 An easy-to-use method of setting parameters for lattice schemes

In lattice-based cryptography, concrete parameter setting may often get quite complicated due to a large set of parameters requiring careful adjustments. Here, we present an easy-to-use algorithm for setting parameters in practice, based on known attacks on LWE and SIS, which we believe could be useful for other lattice schemes. This is indeed an algorithmic summary of what has been explained about practical security estimations at the end of Section 3.2.2. In particular, this method uses the LWE estimator by Albrecht et al. [APS15] for LWE estimations and the methodology by Micciancio and Regev [MR09] for SIS estimations. The LWE estimator is run under both "sieving" and "enumeration", and the largest root Hermite factor returned is taken to be the final root Hermite factor. We emphasise that this method should

[3] Further discussion on soundness error can be found in Section 2.2 of [BKLP15].

Algorithm 3.1 GetParams(DSET, QSET, $F_{\beta_{\text{SIS}}}$)

INPUT: DSET : a set of potential d values; QSET : a set of potential $\log q$ values; $F_{\beta_{\text{SIS}}}$: a function computing the SIS solution norm given $(n, \ell, d, \log q)$.
OUTPUT: a list L of suitable $(d, \log q, n, \ell)$ tuples
ASSUME: $\delta \approx 1.0045, \lambda = 128, \mathcal{B} = 1, \log q \geq 8$

1: $L = \emptyset$ ▷ Initialise output list
2: $\delta = 1.0045$ ▷ Set root Hermite factor
3: BaseDim $= 342$ ▷ Set LWE dim. param. required for $(q, \mathcal{B}) = (2^8, 1)$
4: StepSize $= 39$ ▷ Set gap between LWE dim.'s of $\log q$ and $\log q + 1$
5: **for** each $d \in$ DSET **do**
6: **for** each $\log q \in$ QSET **do**
7: $\ell = \lceil (\text{BaseDim} + \text{StepSize} \cdot (\log q - 8))/d \rceil$ ▷ Set LWE module rank
8: $q = 2^{\log q}$
9: $n = 1$ ▷ Initialise/Reset SIS module rank
10: **while** $nd \leq 8192$ **do** ▷ Overall SIS dim. param. rarely exceeds 8192
11: $\beta_{\text{SIS}} = F_{\beta_{\text{SIS}}}(n, \ell, d, \log q)$ ▷ Set SIS solution length
12: **if** $\min\left\{q, 2^{2\sqrt{nd \log q \log \delta}}\right\} > \beta_{\text{SIS}}$ **then** ▷ Check SIS condition
13: **while** $\min\left\{q, 2^{2\sqrt{nd \log q \log \delta}}\right\} > \beta_{\text{SIS}}$ **do** ▷ Loop while SIS holds
14: $n = n - 1$ ▷ Decrease n one-by-one
15: $\beta_{\text{SIS}} = F_{\beta_{\text{SIS}}}(n, \ell, d, \log q)$ ▷ Compute β_{SIS} with new n
16: **end while**
17: $L = L \cup \{(d, \log q, n + 1, \ell)\}$ ▷ $n + 1$ is the smallest rank possible
18: End 'while' loop starting at Step 10
19: **end if**
20: $n = n + 10$ ▷ Increase of n can be adjusted
21: **end while**
22: **end for**
23: **end for**
24: **return** L

not be seen as a way to guarantee the security of a scheme as there are many other concerns, which cannot be included in a simple algorithm, affecting the security of a cryptographic algorithm.

In the algorithm, we adapt $\delta \approx 1.0045$ for $\lambda = 128$-bit post-quantum security, and $\mathcal{B} = 1$ for M-LWE. We denote the module rank of M-SIS by n and module rank of M-LWE by ℓ. The procedure is presented in Algorithm 3.1.

Given sets of potential d and $\log q$ values, Algorithm 3.1 simply iterates over all the pairs and tries to find possible SIS and LWE module ranks (n, ℓ) satisfying the security requirements for the given $\delta = 1.0045$. The idea for setting the LWE module rank turns out to be quite simple as given at Step 7, which uses a linear extrapolation (in $\log q$) formula to estimate the LWE rank needed for M-LWE$_{\ell,*,q,\mathcal{B}}$ security against lattice reduction attacks with Hermite factor δ and $\dim(R_q) = d$. We tested the LWE estimator [APS15] on a wide range of parameter sets and found that for a fixed $(\delta, \mathcal{B})$ pair, LWE dimension parameter (i.e., ℓd) increments by fixed-size steps for increasing $\log q$. For example, for fixed $(\delta, \mathcal{B}) = (1.0045, 1)$, $\ell d = 342$ is sufficient for $q \approx 2^8$ and $\ell d = 381$ for $q \approx 2^9$. If we keep doubling q one-by-one, we see that ℓd increments by almost 39 for each doubling. Therefore, we accordingly set StepSize and BaseDim parameters, which are then used in the formula at Step 7.

A similar behaviour is observed for different $(\delta, \mathcal{B})$ pairs as well. For example,

for $(\delta, \mathcal{B}) = (1.0045, 5)$, StepSize $= 38.5$ and BaseDim $= 202$ were observed to give accurate results. Therefore, our algorithm can be easily adjusted to different δ and $\mathcal{B}$ values by testing first the dimension required for $q = 2^8$ (which allows to set BaseDim) and then figuring out StepSize by observing the gap between the required dimensions for $q = 2^8$ and a large modulus of, say, $q = 2^{80}$. In particular, StepSize $= \frac{\text{DIM}_{80} - \text{BaseDim}}{(80-8)}$ would need to be set, where DIM_{80} is the dimension required for $q = 2^{80}$.

An important advantage of using a formula to compute the required LWE dimension is due to the fact that the LWE estimator [APS15] does not run very fast. Therefore, when iterating over a large set of potential parameters, running the estimator over and over again takes too long whereas our simple formula provides accurate parameters *very quickly*.

Having fixed the module rank ℓ for LWE at Step 7, the remaining task is finding the *smallest* module rank n for SIS. We do this by iterating over increasing n values and checking if the SIS condition (i.e., the inequality (3.1)) holds. As the algorithm is designed to work for any d, including $d = 1$ (i.e., the case of plain LWE/SIS), n is incremented by 10 at each iteration. Otherwise, since n starts from 1, it would take too long until a suitable n is reached for small d values.

The SIS condition checks whether the SIS attack fails, i.e., the norm of the shortest vector computable by the attacker with a lattice reduction algorithm having root Hermite Factor δ is greater than β_{SIS}. To check this condition, one needs to compute β_{SIS} defined in Definition 3.1. That is, one should answer the question: How short should SIS solutions be to guarantee the security of the scheme? This can be determined from the underlying scheme's security proofs, where a statement similar to the following appears: "If M-$\text{SIS}_{n,m,q,\beta_{\text{SIS}}}$ is hard, then the proposed scheme is secure". The computation of this value is very much dependant on a particular construction, and hence Algorithm 3.1 requires a function $F_{\beta_{\text{SIS}}}$ that outputs β_{SIS} for a given $(n, \ell, d, \log q)$ tuple. For example, if we require a commitment scheme to be γ-binding, then we have $\beta_{\text{SIS}} = 2\gamma$.

For a given q (or $\log q$), there may be cases where there exists no suitable n because q may always fall smaller than β_{SIS}. Therefore, we increment n as long as $nd \leq 8192$, i.e., the overall SIS dimension is smaller than 8192. This is because most, if not all, of the practical lattice schemes have a SIS dimension less than or equal to 8192. On the other hand, once a suitable n value satisfying the SIS requirement is found at Step 12, we then start decreasing n as long as SIS condition is still satisfied. After that the smallest SIS and LWE module ranks are added to the list L with the corresponding $(d, \log q)$ pair. The LWE estimator should be run on the final chosen parameter sets to verify the accuracy of the linear extrapolation formula in Step 7. At the end, the list L of suitable parameter sets is returned. One can, for example, use this list to compute corresponding proof lengths to see what parameter sets give the optimal sizes. Note here that it is straightforward to compute the parameter m in the previous section given $(n, \ell, d, \log q)$ and the message dimension v. In particular, $m = n + \ell$ for HMC and $m = n + v + \ell$ for UMC.

We also note that simplicity of the algorithm is preferred over performance in its current form. For small optimisations, one may combine some of the 'if' and 'while' statements and change how much n increases at Step 20 depending on d. For example, if d is large, say $d \geq 256$, then n can be incremented one by one and the inner-most while loop can be removed in this case. Nevertheless, variants of Algorithm 3.1 are used for many constructions studied throughout the thesis, and the running time of the algorithm is already good enough for rapid selection of concrete parameter sets.

3.2.5 Ring signatures

The formal definitions of ring signatures were established in [BKM09], and we use variants of those. A ring signature consists of four algorithms (**RSetup**, **RKeygen**, **RSign**, **RVerify**) defined as follows.

- $pp \leftarrow \mathbf{RSetup}(1^\lambda)$: On input a security parameter λ, outputs the public parameters pp, which are available to everyone.
- $(pk, sk) \leftarrow \mathbf{RKeygen}(pp)$: Given pp, generates a public-secret key pair (pk, sk).
- $\sigma \leftarrow \mathbf{RSign}_{pp,sk}(M, \boldsymbol{L})$: On input a message M and a ring $\boldsymbol{L}$ of public keys and for a secret key sk generated by **RKeygen** and its corresponding public key $pk \in \boldsymbol{L}$, outputs a signature σ on M with respect to $\boldsymbol{L}$.
- $\{0,1\} \leftarrow \mathbf{RVerify}_{pp}(M, \boldsymbol{L}, \sigma)$: On input a purported signature σ, a message M and a ring $\boldsymbol{L}$, checks if σ is a valid signature on M with respect to $\boldsymbol{L}$. Outputs 1 when it is valid, and outputs 0 otherwise.

Definition 3.7 (Correctness)**.** *A ring signature scheme has* statistical correctness *if the following holds for any* $pp \leftarrow \mathbf{RSetup}(1^\lambda)$*, any* $(pk, sk) \leftarrow \mathbf{RKeygen}(pp)$*, any* $\boldsymbol{L}$ *with* $pk \in \boldsymbol{L}$*, and any* $M \in \{0,1\}^*$*,*

$$\Pr[\ \mathbf{RVerify}_{pp}(M, \boldsymbol{L}, \mathbf{RSign}_{pp,sk}(M, \boldsymbol{L})) = 1\ ] \geq 1 - \mathrm{negl}(\lambda).$$

Definition 3.8 (Anonymity)**.** *A ring signature scheme has* statistical anonymity *if the following holds for any PPT adversary* $\mathcal{A}$

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathbf{RSetup}(1^\lambda); (M, j_0, j_1, \boldsymbol{L}) \leftarrow \mathcal{A}^{\mathbf{RKeygen}(pp)} \\ b \leftarrow \{0,1\};\ \sigma \leftarrow \mathbf{RSign}_{pp,sk_{j_b}}(M, \boldsymbol{L});\ b' \leftarrow \mathcal{A}(\sigma) \end{array} : b' = b\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda),$$

where $(pk_{j_0}, sk_{j_0}), (pk_{j_1}, sk_{j_1}) \leftarrow \mathbf{RKeygen}(pp)$ *and* $pk_{j_0}, pk_{j_1} \in \boldsymbol{L}$*.*

Definition 3.9 (Unforgeability w.r.t. insider corruption)**.** *A ring signature scheme is* unforgeable with respect to insider corruption *if the following holds for all PPT adversary* $\mathcal{A}$

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathbf{RSetup}(1^\lambda); \\ (M, \boldsymbol{L}, \sigma) \leftarrow \mathcal{A}^{\mathrm{PKGen,Sign,Corrupt}}(pp) \end{array} : \mathbf{RVerify}(M, \boldsymbol{L}, \sigma) = 1\right] \leq \mathrm{negl}(\lambda),$$

where

- PKGen : *on the i-th query, runs* $(pk_i, sk_i) \leftarrow \mathbf{RKeygen}(pp)$ *and returns* pk_i*.*
- Sign$(i, M, \boldsymbol{L})$: *returns* $\sigma \leftarrow \mathbf{RSign}_{pp,sk_i}(M, \boldsymbol{L})$ *if* $(pk_i, sk_i) \leftarrow$ PKGen *and* $pk_i \in \boldsymbol{L}$*. Otherwise, returns* $\perp$*.*
- Corrupt(i) : *returns* sk_i *if* $(pk_i, sk_i) \leftarrow$ PKGen*. Otherwise, returns* $\perp$*.*
- *For* $\mathcal{A}$*'s output* $(M, \boldsymbol{L}, \sigma)$*,* Sign$(\cdot, M, \boldsymbol{L})$ *has never been queried, all public keys in* $\boldsymbol{L}$ *are generated by* PKGen *and no public key in* $\boldsymbol{L}$ *is corrupted.*

3.3 Mathematical Background

3.3.1 Representative matrices

For a vector $\boldsymbol{p} = (p_0, \ldots, p_{m-1})$ of polynomials in $R = \mathbb{Z}[X]/(X^d + 1)$ with $m \geq 1$, we denote the vector of all coefficients in $\boldsymbol{p}$ by $\mathrm{Coeff}(\boldsymbol{p}) \in \mathbb{Z}^{md}$. For any $f, g \in R$, there exists a matrix $\mathrm{Rot}(f)$, called the *Rot matrix* of f, such that $\mathrm{Rot}(f) \cdot \mathrm{Coeff}(g) = \mathrm{Coeff}(f \cdot g)$. This notion generalises to the case where $(\mathrm{Rot}(f) \otimes \boldsymbol{I}_m) \cdot \mathrm{Coeff}(\boldsymbol{p}) = \mathrm{Coeff}(f \cdot \boldsymbol{p})$ for $f \in R$ and $\boldsymbol{p} \in R^m$ for $m \geq 1$ where $\otimes$ denotes the Kronecker product.

3.3.2 Singular values

For a rank-n matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, there exists orthogonal matrices $\boldsymbol{U}, \boldsymbol{V}$ and a diagonal matrix $\boldsymbol{\Lambda}$ with the non-negative diagonal entries $\sigma_1 \geq \cdots \geq \sigma_n$ such that $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$. The values $\sigma_1(\boldsymbol{A})$ and $\sigma_n(\boldsymbol{A})$ are called the largest and the least singular values of $\boldsymbol{A}$, respectively.

Fact 3.10. *For square matrices $\boldsymbol{A}, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_s \in \mathbb{R}^{n\times n}$, $s \geq 1$, and $c \in \mathbb{R}$, the following holds*

- $\sigma_1(\boldsymbol{A}_1 \cdots \boldsymbol{A}_s) \leq \sigma_1(\boldsymbol{A}_1) \cdots \sigma_1(\boldsymbol{A}_s)$, *and* $\sigma_n(\boldsymbol{A}_1 \cdots \boldsymbol{A}_s) \geq \sigma_n(\boldsymbol{A}_1) \cdots \sigma_n(\boldsymbol{A}_s)$,
- $\sigma_1(c\boldsymbol{A}) = |c| \cdot \sigma_1(\boldsymbol{A})$ *and* $\sigma_n(c\boldsymbol{A}) = |c| \cdot \sigma_n(\boldsymbol{A})$,
- $\sigma_1(\boldsymbol{A} \otimes \boldsymbol{I}_m) = \sigma_1(\boldsymbol{A})$ *and* $\sigma_n(\boldsymbol{A} \otimes \boldsymbol{I}_m) = \sigma_n(\boldsymbol{A})$ *for any* $m \geq 1$ *where* $\otimes$ *denotes the Kronecker product,*
- $\boldsymbol{A}$ *and* $\boldsymbol{A}^\top$ *have the same singular values.*

3.3.3 Discrete Gaussian distribution and its properties

In this thesis, we always consider Gaussian distributions centred at zero, and thus restrict our definitions to that case. Let $\boldsymbol{S} \in \mathbb{R}^{m\times n}$ be a rank-n matrix. Define the *ellipsoid Gaussian function* on $\mathbb{R}^n$ centred at zero with parameter $\boldsymbol{S}$ (and covariance matrix $\boldsymbol{S}^\top\boldsymbol{S}$) as $\rho_{\boldsymbol{S}}(\boldsymbol{x}) = e^{-\pi \boldsymbol{x}^\top (\boldsymbol{S}^\top \boldsymbol{S})^{-1} \boldsymbol{x}}$ for all $\boldsymbol{x} \in \mathbb{R}^n$. The *ellipsoid discrete Gaussian distribution* over $\mathbb{Z}^n$ centred at zero with parameter $\boldsymbol{S}$ is then defined by the probability mass function $\mathcal{D}^n_{\boldsymbol{S}}(\boldsymbol{x}) = \rho_{\boldsymbol{S}}(\boldsymbol{x})/\rho_{\boldsymbol{S}}(\mathbb{Z}^n)$ where $\rho_{\boldsymbol{S}}(\mathbb{Z}^n) = \sum_{\boldsymbol{z}\in\mathbb{Z}^n} \rho_{\boldsymbol{S}}(\boldsymbol{z})$ is a normalisation factor. If the parameter $\boldsymbol{S} = s\boldsymbol{I}_n$ for $s \in \mathbb{R}^+$, then we obtain the *spherical* discrete Gaussian distribution, denoted by $\mathcal{D}^n_s$. We denote by D^n_σ the discrete normal distribution with standard deviation σ, defined as $\mathcal{D}^n_s$ with $s = \sigma\sqrt{2\pi}$.

Fact 3.11 (A result of [AGHS13, Fact 2])**.** *For an invertible $n\times n$ matrix $\boldsymbol{X}$, $\boldsymbol{X}\cdot\mathcal{D}^n_{\boldsymbol{S}} = \mathcal{D}^n_{\boldsymbol{S}\boldsymbol{X}^\top}$. That is, the distribution induced by sampling $\boldsymbol{v} \leftarrow \mathcal{D}^n_{\boldsymbol{S}}$ and outputting $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{v}$ is the same as $\mathcal{D}^n_{\boldsymbol{S}\boldsymbol{X}^\top}$.*

As defined in [MR07], for a lattice L and real $\epsilon > 0$, the *smoothing parameter*, $\eta_\epsilon(L)$, of L is the smallest s such that $\rho_{1/s}(L^* \setminus \{\boldsymbol{0}\}) \leq \epsilon$ where L^* is the "dual lattice". We skip the details, but for our purposes the following facts are enough.

Fact 3.12 ([MR07, Lemma 3.3])**.** $\eta_\epsilon(\mathbb{Z}^n) < 6$ *for* $\epsilon = 2^{-128}$ *and any* $1 \leq n \leq 2^{32}$.

Lemma 3.13 ([AGHS13, Lemma 3])**.** *Let $\sigma_1(\boldsymbol{S})$ and $\sigma_n(\boldsymbol{S})$ be the largest and the least singular values of a rank-n matrix $\boldsymbol{S}$, respectively. If $\sigma_n(\boldsymbol{S}) \geq \eta_\epsilon(\mathbb{Z}^n)$,*

$$\Pr_{\boldsymbol{v}\leftarrow\mathcal{D}^n_{\boldsymbol{S}}}\left[\, \|\boldsymbol{v}\| \geq \sigma_1(\boldsymbol{S})\sqrt{n} \,\right] \leq \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}.$$

In our protocols, we sometimes deal with sum of independent vectors from discrete normal distribution. To study the behaviour of such sums, we make use of the following lemma.

Lemma 3.14 (Special case of [MP13, Theorem 3.3])**.** *Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ be independent vectors with distribution D^d_σ for $d \geq 1$. If $\sigma \geq \eta(\mathbb{Z}^d)/\sqrt{\pi}$ for the smoothing parameter $\eta(\mathbb{Z}^d)$ of $\mathbb{Z}^d$, then the distribution of $\boldsymbol{z} := \boldsymbol{y}_1 + \cdots + \boldsymbol{y}_s$ is statistically close to $D^d_{\sigma\sqrt{s}}$.*

The standard deviations in our protocols are always much larger than 6, and thus discrete normal variables behave as its continuous counterpart when multiple samples are summed over. Finally, the lemma below summarises more concrete "tail-cut" bounds on discrete normal distribution.

Lemma 3.15 ([Lyu12, Lemma 4.4]). *The following holds for discrete normal distribution.*

1. *For any* $\alpha > 0$, $\Pr[|z| > \alpha \cdot \sigma \ : \ z \leftarrow D_\sigma] \leq 2 \cdot \exp\left(-\frac{\alpha^2}{2}\right)$,
2. *For any* $\alpha > 1$, $\Pr[\|\boldsymbol{z}\| > \alpha\sigma\sqrt{t} \ : \ \boldsymbol{z} \leftarrow D_\sigma^t] < \alpha^t e^{\frac{1-\alpha^2}{2}t}$.

In particular, we have

- $\Pr[|z| > 12\sigma \ : \ z \leftarrow D_\sigma] < 2^{-100}$,
- $\Pr[\|\boldsymbol{z}\| > 2\sigma\sqrt{t} \ : \ \boldsymbol{z} \leftarrow D_\sigma^t] < 2^{-100}$ *if* $t \geq 86$, *and*
- $\Pr[\|\boldsymbol{z}\| > 5\sigma\sqrt{t} \ : \ \boldsymbol{z} \leftarrow D_\sigma^t] < 2^{-100}$ *if* $t \geq 7$.

3.3.4 Rejection sampling

Rejection sampling is among the important tools used in many lattice-based protocols. Its use in lattice-based cryptography was introduced by Lyubashevsky [Lyu09, Lyu12]. There are two flavors of these rejection sampling techniques. The one in [Lyu09] makes the post-rejection distribution uniform over some set whereas that in [Lyu12] makes use of discrete Gaussian distribution. We summarise the results of the latter below and leave the discussion on the former when we make use of it in our protocols.

Algorithm 3.2 $\mathrm{Rej}(\boldsymbol{z}, \boldsymbol{c}, \phi, T)$

1: $\sigma = \phi T$
2: $\mu(\phi) = e^{12/\phi + 1/(2\phi^2)}$
3: $u \leftarrow [0, 1)$
4: **if** $u > \left(\frac{1}{\mu(\phi)}\right) \cdot \exp\left(\frac{-2\langle \boldsymbol{z}, \boldsymbol{c}\rangle + \|\boldsymbol{c}\|^2}{2\sigma^2}\right)$, **then return** 0 ▷ means "abort" in protocol
5: **else return** 1

Lemma 3.16 ([Lyu12]). *Let* h *be a probability distribution over* $V \subseteq \mathbb{Z}^s$ *(*$s \geq 1$*) where all the elements have norm less than* T. *Let* $\boldsymbol{c} \leftarrow h$ *and* $\phi > 0$, *and consider the algorithm* $\mathcal{F}$ *that samples* $\boldsymbol{y} \leftarrow D_\sigma^s$ *and outputs Rej(*$\boldsymbol{z}, \boldsymbol{c}, \phi, T$*) (Algorithm 3.2) for* $\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{c}$. *The probability that* $\mathcal{F}$ *outputs 1 is within* 2^{-100} *of* $1/\mu(\phi)$ *for* $\mu(\phi) = e^{12/\phi + 1/(2\phi^2)}$, *and conditioned on the output being 1, the statistical distance between distribution of* $\boldsymbol{z}$ *and* D_σ^s *is at most* 2^{-100}.

3.3.5 Some basics of Linear Algebra and Vandermonde matrices

We recall some basics about Vandermonde matrices and from Linear Algebra relevant to our discussions (see, e.g., [HJ12] for more details). We assume that the matrices are defined over a ring $\mathfrak{R}$. Let $\boldsymbol{A}$ be a $n \times n$ square matrix and $\det(\boldsymbol{A})$ denote its determinant. The adjugate $\mathrm{adj}(\boldsymbol{A})$ of $\boldsymbol{A}$, defined as the transpose of the cofactor matrix of $\boldsymbol{A}$, satisfies the following property

$$\mathrm{adj}(\boldsymbol{A}) \cdot \boldsymbol{A} = \boldsymbol{A} \cdot \mathrm{adj}(\boldsymbol{A}) = \det(\boldsymbol{A}) \cdot \boldsymbol{I}_n. \tag{3.2}$$

Therefore, if $\boldsymbol{A}$ is non-singular, $\mathrm{adj}(\boldsymbol{A}) = \det(\boldsymbol{A}) \cdot \boldsymbol{A}^{-1}$. A $(k+1)$-dimensional Vandermonde matrix $\boldsymbol{V}$ is defined as below for some $x_0, \ldots, x_k \in \mathfrak{R}$,

$$\boldsymbol{V} = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^k \\ 1 & x_1 & x_1^2 & \cdots & x_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^k \end{pmatrix},$$

and its determinant satisfies the following property

$$\det(\boldsymbol{V}) = \prod_{0 \le i < j \le k} (x_j - x_i). \tag{3.3}$$

Further, the following is an easy consequence of (3.3).

Fact 3.17. *The Vandermonde determinant* $\det(\boldsymbol{V})$ *has* $\binom{k+1}{2}$ *multiplicands of the form* $x_j - x_i$ *with* $j \neq i$.

We observe from [Tur66] that the Vandermonde matrix inverse $\boldsymbol{V}^{-1}$, when it exists, has the following structure

$$\begin{pmatrix} \frac{*}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{*}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{*}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \\ \frac{*}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{*}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{*}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{-1}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{(-1)^k}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \end{pmatrix}, \tag{3.4}$$

where $*$ denotes some element in the ring $\mathfrak{R}$, computed as a function of x_i's. It is clear from this structure that $\boldsymbol{V}^{-1}$ exists over $\mathfrak{R}$ if and only if the differences $x_i - x_j$ for $0 \le i < j \le k$ are invertible over $\mathfrak{R}$. The structure in (3.4) helps us to visualise the structure of $\operatorname{adj}(\boldsymbol{V})$ using the fact that $\operatorname{adj}(\boldsymbol{V}) = \det(\boldsymbol{V}) \cdot \boldsymbol{V}^{-1}$ if $\boldsymbol{V}$ is non-singular. In particular, we have the following fact.

Fact 3.18. *Let* $(\Gamma_0, \ldots, \Gamma_k)$ *be the last row of* $\operatorname{adj}(\boldsymbol{V})$. *Then,*

$$\Gamma_i = (-1)^{i+k} \prod_{0 \le l < j \le k \,\wedge\, j,l \neq i} (x_j - x_l),$$

and Γ_i *has* $\left[\binom{k+1}{2} - k\right] = \frac{k(k-1)}{2}$ *multiplicands for all* $0 \le i \le k$.

Fact 3.18 follows by observing that k multiplicands in $\det(\boldsymbol{V})$ are cancelled out by the corresponding denominator in $\boldsymbol{V}^{-1}$.

3.3.6 Technical lemmas

We first summarise some results regarding different norms and a product of polynomials in $\mathbb{Z}[X]/(X^d + 1)$.

Lemma 3.19. *For any* $f, g \in R = \mathbb{Z}[X]/(X^d + 1)$, *we have the following relations*

1. $\|f\| \le \sqrt{d} \cdot \|f\|_\infty$,
2. $\|f\| \le \|f\|_1 \le \sqrt{d}\,\|f\|$,
3. $\|f \cdot g\| \le \sqrt{d} \cdot \|f\| \cdot \|g\|$,
4. $\|f \cdot g\|_\infty \le \|f\| \cdot \|g\|$,
5. $\|f \cdot g\|_\infty \le \|f\|_1 \cdot \|g\|_\infty$,
6. $\|\prod_{i=1}^n f_i\|_\infty \le \left(\prod_{i=1}^{n-1} \|f_i\|_1\right) \cdot \|f_n\|_\infty$ *where* $f_i \in R$ *for all* $1 \le i \le n$.

Proof. The first 5 relations are standard and we only provide a proof for the last one. If $n = 2$, the result is clear by the forth relation. Assume that the result holds for all $s < n$, and we want to show that it holds for $n > 2$.

$$\left\| \prod_{i=1}^{n} f_i \right\|_\infty \le \|f_1\|_1 \cdot \left\| \prod_{i=2}^{n} f_i \right\|_\infty \le \left(\prod_{i=1}^{n-1} \|f_i\|_1 \right) \cdot \|f_n\|_\infty ,$$

where the first inequality holds due to the forth relation and the second one follows by the inductive assumption. □

The lemma below shows that any difference of two (distinct) monomials in $R = \mathbb{Z}[X]/(X^d + 1)$ is of a special form and also has a small norm. The lemma will be useful when we make use of "monomial challenges" [BCK+14] in our protocols.

Lemma 3.20 ([BCK+14, Lemma 3.1])**.** *For $0 \leq i, j \leq 2d - 1$, all the coefficients of $2(X^i - X^j)^{-1} \in \mathbb{Z}[X]/(X^d + 1)$ are in $\{-1, 0, 1\}$. This implies that $\left\|2(X^i - X^j)^{-1}\right\| \leq \sqrt{d}$.*

Chapter 4

Multi-Shot Algebraic Proofs and Applications

As discussed in the previous chapters, it is not an easy task to design *efficient* lattice-based ZKPs that can prove *complex non-linear* relations. It seems even harder to make such proofs *one-shot*, where one is restricted to working with exponentially large challenge sets with possibly much less control. Therefore, as an initial step, this chapter investigates the design of algebraic proofs that do not necessarily reach a negligible soundness error in a single protocol execution. This investigation forms an important stepping stone in understanding the precise challenges that arise when one is concerned with proving non-linear relations. More precisely, this chapter studies multi-shot algebraic ZKPs that can prove non-linear relations.[1]

We first start with introducing new technical tools for the design and analysis of many-special sound protocols in Section 4.1. In order to understand the security requirements of such protocols, it is important to study the norm of an extracted witness, which is done in Section 4.1.2 for the monomial challenges. Then, with these new tools available, we show how to construct binary and one-out-of-many proofs in Section 4.2. Having these ZKPs as building blocks, we introduce a ring signature scheme based on standard lattice assumptions in Section 4.3.

Throughout this chapter, the commitment scheme is always instantiated with HMC in HNF.

4.1 New Technical Tools for Lattice-Based Proofs

In this section, we present a collection of technical tools we use in our constructions in this chapter. These new tools may be of independent interest for future works on algebraic lattice-based zero-knowledge proofs and signatures.

4.1.1 Proving a value binary in R_q

We first show a lemma that, in particular, enables one to guarantee that $b \in R_q$ is a bit when the equation $b \cdot (1-b) = 0$ holds over R_q. Our lemma does not put any additional assumption on q but its size, which enables one to use fast computation algorithms such as the number-theoretic transform (NTT) with $q \equiv 1 \bmod 2d$. In particular, we do not need number theoretic conditions on q that makes NTT less efficient. For example, such a condition is imposed in [dPLNS17] to ensure the invertibility of small elements in R_q.

Lemma 4.1. *For $b \in R_q = \mathbb{Z}_q[X]/(X^d+1)$, if $b \cdot (\alpha - b) = 0$ over R_q for some positive integer α, and $\|b\| + \alpha < \sqrt{q}$, then $b \in \{0, \alpha\}$.*

[1] This chapter is mainly based on [ESS+19].

Proof. Since $\|b\| + \alpha < \sqrt{q}$, we have $\|b\| < \sqrt{q}$. Then, we get

$$\|b \cdot (\alpha - b)\|_\infty \leq \|b\| \cdot \|\alpha - b\| \leq \|b\| \cdot (\|b\| + \alpha) < \sqrt{q} \cdot \sqrt{q} = q.$$

Therefore, $b \cdot (\alpha - b) = 0$ holds over R. Since $X^d + 1$ is irreducible over $\mathbb{Q}$, we get $b \in \{0, \alpha\}$. □

We remark that above we essentially use the fact that R is an integral domain. This is in particular true for our choice of R since X^d+1 is irreducible over $\mathbb{Q}$, but the result also generalises to any other integral domain rings (for instance, $\mathbb{Z}_q[X]/(f(X))$ for other irreducible f's).

4.1.2 Bounding the extracted witness norm for monomial challenges

Consider a Σ-protocol where the prover's initial commitments are $A_0, A_1, \ldots, A_k$ ($k \geq 1$), and he responds with $(\boldsymbol{f}_x, \boldsymbol{r}_x)$ for a given challenge x by the verifier. Then, the verifier checks whether $A_0 + A_1 x + A_2 x^2 + \cdots + A_k x^k = \mathrm{Com}(\boldsymbol{f}_x; \boldsymbol{r}_x)$ holds where Com is a homomorphic commitment scheme. Now, suppose A_k is the commitment of prover's witness and that the extractor obtains $k+1$ accepting protocol transcripts for the same initial commitments, represented as follows.

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^k \\ 1 & x_1 & x_1^2 & \cdots & x_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^k \end{pmatrix} \cdot \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_k \end{pmatrix} = \begin{pmatrix} \mathrm{Com}(\boldsymbol{f}_{x_0}; \boldsymbol{r}_{x_0}) \\ \mathrm{Com}(\boldsymbol{f}_{x_1}; \boldsymbol{r}_{x_1}) \\ \vdots \\ \mathrm{Com}(\boldsymbol{f}_{x_k}; \boldsymbol{r}_{x_k}) \end{pmatrix}.$$

Here, the matrix on the very left is a Vandermonde matrix $\boldsymbol{V}$, and the extractor can recover a *possible* opening of A_k via multiplying both sides by $\boldsymbol{V}^{-1}$, if exists, due to the homomorphic properties of the commitment scheme. Recalling from Section 3.3.5, the inverse matrix $\boldsymbol{V}^{-1}$ has the following form:

$$\begin{pmatrix} \frac{*}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{*}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{*}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \\ \frac{*}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{*}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{*}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{-1}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{(-1)^k}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \end{pmatrix}, \quad (4.1)$$

where $*$ denotes some element in the domain. Our main protocol to be described has this structure and, therefore, the Vandermonde matrix inverse plays a crucial role in the witness extraction. In particular, if we denote the entries in the last row of $\boldsymbol{V}^{-1}$ by $\alpha_0, \ldots, \alpha_k$ (from left to right), we have

$$A_k = \sum_{j=0}^{k} \alpha_j \mathrm{Com}(\boldsymbol{f}_{x_j}; \boldsymbol{r}_{x_j}) = \mathrm{Com}\left(\sum_{j=0}^{k} \alpha_j \boldsymbol{f}_{x_j}; \sum_{j=0}^{k} \alpha_j \boldsymbol{r}_{x_j}\right) =: \mathrm{Com}(\boldsymbol{m}_{\mathrm{ext}}; \boldsymbol{r}_{\mathrm{ext}}). \quad (4.2)$$

These arguments tell us that we need to make sure $\boldsymbol{V}^{-1}$ exists in the first place, which follows from the invertibility of pairwise differences of challenges. What is more important in the case of lattice-based proofs is that α_j's (and, in general, the entries in $\boldsymbol{V}^{-1}$) must have small norm so that extracted witness (particularly, $(\boldsymbol{m}_{\mathrm{ext}}; \boldsymbol{r}_{\mathrm{ext}})$) is a *valid* opening (of A_k). To that end, we choose the set of monomials as the challenge space and, thus, can make use of Lemma 3.20 to bound the entries in $\boldsymbol{V}^{-1}$, which brings us to our first method below. In the rest, we focus on the last row of $\boldsymbol{V}^{-1}$,

which is enough for our purposes, but our results can be extended to the cases related to the other entries of $\boldsymbol{V}^{-1}$.

Method 1

Taking the first entry α_0 as an example, we have

$$2^k\alpha_0 = \frac{2^k}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} = \frac{2}{x_0-x_1}\cdot\frac{2}{x_0-x_2}\cdots\frac{2}{x_0-x_k}.$$

For monomial challenges, using Lemma 3.19 and Lemma 3.20, we get

$$\left\|2^k\alpha_0\right\| = \left\|\prod_{i=1}^{k}\frac{2}{x_0-x_i}\right\| \le \left(\sqrt{d}\right)^{k-1}\prod_{i=1}^{k}\left\|2(x_0-x_i)^{-1}\right\| \le \left(\sqrt{d}\right)^{k-1}\left(\sqrt{d}\right)^{k} = d^{k-0.5}.$$

Since all the entries in the last row have a similar form and the bound does not depend on the particular choice of monomials, the same bound holds for all entries in the last row of $\boldsymbol{V}^{-1}$. Note that $\boldsymbol{V}^{-1}$ exists over R_q for odd q (though may not have small entries) since 2 is invertible for such q. We summarise these results in the following lemma, whose proof follows from the above discussion.

Lemma 4.2. *For $k \in \mathbb{Z}^+$, let $x_i = X^{\omega_i} \in R = \mathbb{Z}[X]/(X^d+1)$ for $0 \le \omega_i \le 2d-1$ and $0 \le i \le k$. Define the Vandermonde matrix $\boldsymbol{V}$ of dimension $k+1$ where i-th row is the vector $(1, x_i, x_i^2, \ldots, x_i^k)$. Then, $\boldsymbol{V}$ is invertible over R_q for odd q, and for any entry α_j $(0 \le j \le k)$ in the last row of $\boldsymbol{V}^{-1}$, we have $\left\|2^k\alpha_j\right\| \le d^{k-0.5}$.*

Using Lemma 4.2, we can now summarise the main result of Method 1.

Lemma 4.3. *For the extracted opening $(\boldsymbol{m}_{\text{ext}}, \boldsymbol{r}_{\text{ext}})$ of A_k in (4.2), we have*

$$\left\|2^k\boldsymbol{r}_{\text{ext}}\right\| \le (k+1)\cdot d^k \cdot \max_{0\le j\le k}\left\|\boldsymbol{r}_{x_j}\right\| \quad \textit{and} \quad \left\|2^k\boldsymbol{m}_{\text{ext}}\right\| \le (k+1)\cdot d^k \cdot \max_{0\le j\le k}\left\|\boldsymbol{f}_{x_j}\right\|.$$

Proof.

$$\begin{aligned}\left\|2^k\boldsymbol{r}_{\text{ext}}\right\| &= \left\|\sum_{j=0}^{k}2^k\alpha_j\boldsymbol{r}_{x_j}\right\| \le (k+1)\cdot\max_{0\le j\le k}\left\|2^k\alpha_j\boldsymbol{r}_{x_j}\right\| \\ &\le (k+1)\sqrt{d}\max_{0\le j\le k}\left\|2^k\alpha_j\right\|\max_{0\le j\le k}\left\|\boldsymbol{r}_{x_j}\right\| \le (k+1)\cdot d^k\max_{0\le j\le k}\left\|\boldsymbol{r}_{x_j}\right\|. \qquad (4.3)\end{aligned}$$

A similar result follows analogously for $\boldsymbol{m}_{\text{ext}}$. □

This initial attempt succeeds, but the result may not be optimal. Thus, we deepen our analysis to get a tighter bound.

Method 2

We observe that all entries in $\boldsymbol{V}^{-1}$ are constructed by challenge values, which are public. Therefore, independent of a protocol run, anyone can take a set of challenges and compute, in particular, $\left\|2^k\alpha_j\right\|$ for any entry α_j in the last row of $\boldsymbol{V}^{-1}$. The important part here is that one can efficiently indeed iterate through all the possible challenge sets (to be used in witness extraction) *if* the challenge space size and k are not too large. This means anyone can compute a global bound $\mathbb{B}_{d,k}$ on $\left\|2^k\alpha_j\right\|$ for

any given k and d independent of the index j and the challenges used in the witness extraction.

Observing from (4.1), the total search space will be of size at most $(k+1)\cdot|\mathcal{C}|^{k+1}$ where $|\mathcal{C}| = 2d$ denotes the monomial challenge space size. However, note that, assuming w.l.o.g. $i > j$,

$$\left\|(X^i - X^j)^{-1}\right\| = \left\|X^{2d-i}(1-X^{j-i})^{-1}\right\| = \left\|(1-X^{j-i})^{-1}\right\| \tag{4.4}$$

since multiplication by a monomial in R simply performs a nega-cyclic rotation of the coefficients. Therefore, for any given k, it is enough to iterate through all subsets of $\{1,\ldots,2d-1\}$ of size k, and compute $\left\|\prod_{\omega\in U_k} 2(1-X^{\omega})^{-1}\right\|$ for such a given subset U_k. As a result, the search space size is reduced to $\binom{|\mathcal{C}|-1}{k}$. In our parameter setting for practical ring sizes of $N \leq 2^{20}$, we have $k \leq 3$. Therefore, for example, for $d = 64$ and $k = 3$, this requires only $\binom{127}{3} < 2^{18.4}$ iterations to be performed only ever once. Below is the result of Method 2, where the proof follows by replacing $\max_{0\leq j\leq k}\left\|2^k\alpha_j\right\|$ in (4.3) by $\mathbb{B}_{d,k}$.

Lemma 4.4. *For the extracted opening $(\boldsymbol{m}_{\text{ext}}, \boldsymbol{r}_{\text{ext}})$ of A_k in (4.2), and any given d and k, there exists a constant $\mathbb{B}_{d,k} \leq d^{k-0.5}$ and an algorithm to compute $\mathbb{B}_{d,k}$ with a running time at most $(k-1)\cdot\binom{2d-1}{k}$ polynomial multiplications in R_q and $\binom{2d-1}{k}$ Euclidean norm computations of degree d polynomials such that*

$$\left\|2^k\boldsymbol{r}_{\text{ext}}\right\| \leq (k+1)\cdot\sqrt{d}\cdot\mathbb{B}_{d,k}\cdot\max_{0\leq j\leq k}\left\|\boldsymbol{r}_{x_j}\right\|, \text{ and} \tag{4.5}$$

$$\left\|2^k\boldsymbol{m}_{\text{ext}}\right\| \leq (k+1)\cdot\sqrt{d}\cdot\mathbb{B}_{d,k}\cdot\max_{0\leq j\leq k}\left\|\boldsymbol{f}_{x_j}\right\|. \tag{4.6}$$

Method 3

The above two methods give us ways to bound the extracted witness length independent of a protocol run. The question one may also ask is "How much additional information can we use from a protocol run?"

Assume that the prover's response follows a discrete Gaussian distribution, i.e., $\boldsymbol{r}_x \leftarrow \mathcal{D}_s^{md}$ for some $s \in \mathbb{R}^+, m \in \mathbb{Z}^+$. Instead of bounding $\left\|2^k\alpha_j\right\|$, we bound $\left\|2^k\alpha_j\boldsymbol{r}_{x_j}\right\|$ for all j's. The product $2^k\alpha_j\boldsymbol{r}_{x_j}$ can be represented as $(\text{Rot}(2^k\alpha_j)\otimes\boldsymbol{I}_m)\cdot\text{Coeff}(\boldsymbol{r}_{x_j}) = \text{Coeff}(2^k\alpha_j\boldsymbol{r}_{x_j})$ where $\otimes$ denotes the Kronecker product. Let us denote $\boldsymbol{R}_j = \text{Rot}(2^k\alpha_j)\otimes\boldsymbol{I}_m$. Since $\text{Coeff}(\boldsymbol{r}_{x_j}) \leftarrow \mathcal{D}_s^{md}$, by Fact 3.11, we have $\boldsymbol{R}_j\cdot\text{Coeff}(\boldsymbol{r}_{x_j}) \in \mathcal{D}_{s\boldsymbol{R}_j^\top}^{md}$. Hence, by Lemma 3.13, with high probability, we get

$$\left\|\text{Coeff}(2^k\alpha_j\boldsymbol{r}_{x_j})\right\| = \left\|\boldsymbol{R}_j\cdot\text{Coeff}(\boldsymbol{r}_{x_j})\right\| \leq \sigma_1(s\boldsymbol{R}_j^\top)\sqrt{md} = \sigma_1(\boldsymbol{R}_j)s\sqrt{md}, \tag{4.7}$$

if $\sigma_n(s\boldsymbol{R}_j^\top) \geq \eta_\epsilon(\mathbb{Z}^{md})$, which can be easily satisfied as shown in the proof of Lemma 4.5 below. We can now summarise the main result of Method 3 as below.

Lemma 4.5. *Let $\boldsymbol{r}_{\text{ext}} = \sum_{j=0}^k \alpha_j\boldsymbol{r}_{x_j}$ be the randomness opening of A_k as in (4.2). Assume that $s \geq 6$, $d \in \{4, 8, \ldots, 512\}$ and $md \leq 2^{32}$. If $\boldsymbol{r}_{x_j} \leftarrow \mathcal{D}_s^{md}$ for all $0 \leq j \leq k$, then with probability at least $1 - \frac{1+\epsilon}{1-\epsilon}2^{-md}$ for $\epsilon = 2^{-128}$,*

$$\left\|2^k\boldsymbol{r}_{\text{ext}}\right\| \leq (k+1)\cdot\max_{0\leq j\leq k}\sigma_1(\boldsymbol{S}_j)\cdot s\sqrt{md}, \tag{4.8}$$

where $\boldsymbol{S}_j = \text{Rot}(2^k\alpha_j)$ for $j = 0,\ldots,k$.

Proof. By Fact 3.10, $\sigma_n(\boldsymbol{R}_j^\top) = \sigma_n(\boldsymbol{R}_j) = \sigma_n(\boldsymbol{S}_j \otimes \boldsymbol{I}_m) = \sigma_n(\boldsymbol{S}_j)$ for any $0 \leq j \leq k$. Again, by Fact 3.10, we have

$$\sigma_n(\boldsymbol{S}_j) = \sigma_n\left(\prod_{i=0, i\neq j}^{k} \mathrm{Rot}\left(\frac{2}{x_j - x_i}\right)\right) \geq \prod_{i=0, i\neq j}^{k} \sigma_n\left(\mathrm{Rot}\left(\frac{2}{x_j - x_i}\right)\right).$$

We have verified by computation that $\sigma_n\left(\mathrm{Rot}\left(\frac{2}{x_j - x_i}\right)\right) \geq 1$ for any pair of monomial challenges x_j, x_i and any $d \in \{4, 8, \ldots, 512\}$. As a result, $\sigma_n(\boldsymbol{R}_j^\top) \geq 1$ is always satisfied with the given assumptions. Thus, using Fact 3.10 and Fact 3.12, we have

$$\sigma_n(s\boldsymbol{R}_j^\top) \geq s \cdot \sigma_n(\boldsymbol{R}_j^\top) \geq 6 > \eta_\epsilon(\mathbb{Z}^{md}).$$

Since $\sigma_1(\boldsymbol{S}_j) = \sigma_1(\boldsymbol{R}_j)$ by Fact 3.10, the rest follows from Lemma 3.13 as sketched in the description of Method 3. □

Similar to the idea in Method 2, one can iterate through all $\boldsymbol{S}_j$'s and compute a global bound $\mathbb{S}_{d,k}$ on possible $\sigma_1(\boldsymbol{S}_j)$'s for a given d and k. When $\boldsymbol{r}_{x_j} \leftarrow \mathcal{D}_s^{md}$, we have $\|\boldsymbol{r}_{x_j}\| \leq s\sqrt{md}$ (up to a small constant factor) by Lemma 3.13. As a result, we may reduce the comparison of the three methods to the comparison of the values d^k (Method 1), $\mathbb{B}'_{d,k} = \sqrt{d} \cdot \mathbb{B}_{d,k}$ (Method 2) and $\mathbb{S}_{d,k}$ (Method 3).

However, there is an important detail in Method 3: it only works when the prover's response follows a discrete Gaussian distribution and the verifier cannot simply check if that is the case. To solve this problem, we introduce a new tool called, Pseudo Witness Extraction in Algorithm 4.1. If Algorithm 4.1 is used in protocol's verification with an input bound β, then $\|2^k \boldsymbol{r}_{\text{ext}}\| \leq (k+1)\beta$ must hold. Hence, when the prover's responses $\boldsymbol{r}_{x_j}$'s are from $\mathcal{D}_s^{md}$, setting $\beta = \mathbb{S}_{d,k} s\sqrt{md}$ ensures both that an honest prover's proof will be accepted and also that the extracted randomness will satisfy the norm-bound as in Lemma 4.5.

In Table 4.1, we provide a comparison between the three methods introduced. As can be seen from the table, as k increases, the advantage of Method 2 and Method 3 over Method 1 grows larger. There are also obvious patterns that can be observed from the table such as $\mathbb{S}_{d,k}/\mathbb{B}'_{d,k} \approx \sqrt{2}$ for any d and k. We leave the investigation of these behaviours as an open problem. For larger values of k, for which it is infeasible to search the whole space, one can use Lemma 3.19 to upper-bound $\mathbb{B}_{d,k}$ (as $\mathbb{B}_{d,k}$ is an upperbound on the norm of a product of polynomials) and Fact 3.10 to upper-bound $\mathbb{S}_{d,k}$ (as $\mathbb{S}_{d,k}$ is an upperbound on the singular value of a product of matrices). These still give better results over Method 1.

Algorithm 4.1 Pseudo-witness-extraction

Input: a vector $\boldsymbol{r}$; a challenge $x_0 \in \mathcal{C}$; an integer $k \geq 1$; a norm bound $\beta \in \mathbb{R}^+$

1: **for** each k-tuple $(x_1, \ldots, x_k) \in \mathcal{C}^k$ s.t. $x_0 \neq x_1 \neq \cdots \neq x_k$ **do**
2: $\quad \boldsymbol{r}_{\text{p-ext}} = \left[\prod_{j=1}^{k} 2(x_0 - x_j)^{-1}\right] \cdot \boldsymbol{r}$
3: $\quad$ **if** $\|\boldsymbol{r}_{\text{p-ext}}\| > \beta$, **then return** False
4: **end for**
5: **return** True

TABLE 4.1: Comparison of Method 1, Method 2 and Method 3. * indicates that only a subset of the whole search space has been iterated through.

	$k=2$			$k=3$			$k=4$		
d	$\log(d^k)$	$\log(\mathbb{B}'_{d,k})$	$\log(\mathbb{S}_{d,k})$	$\log(d^k)$	$\log(\mathbb{B}'_{d,k})$	$\log(\mathbb{S}_{d,k})$	$\log(d^k)$	$\log(\mathbb{B}'_{d,k})$	$\log(\mathbb{S}_{d,k})$
16	8	7.21	6.70	12	9.56	9.06	16	11.92	11.42
32	10	9.21	8.70	15	12.55	12.05	20	15.90	15.40
64	12	11.21	10.70	18	15.55	15.05	24	19.90	19.40
128	14	13.21	12.70	21	18.55	18.05*	28	23.90*	23.40*
256	16	15.21	14.70	24	21.55	21.05*	32	-	-

4.2 Multi-Shot Sigma Protocols from Lattices

Now that we have established various tools that can be used in the witness extraction of algebraic protocols, we apply these techniques in ZKPs for important relations. The starting point of the protocols to be described is the works by Groth and Kohlweiss [GK15] and Bootle et al. [BCC+15]. However, as mentioned in Section 2.1, these DL-based proofs do not easily extend to the lattice setting, and one needs special tools as those introduced in the previous section.

4.2.1 Σ-protocol for commitment to a sequence of bits

In this section, we describe a lattice-based Σ-protocol showing that a commitment B opens to sequences of binary values where the Hamming weight of each sequence is exactly one. Let $N = \beta^k > 1$ and $\boldsymbol{r}, \hat{\boldsymbol{r}} \in R_q^m$, and define the relations to be proved in Definition 4.6.

Definition 4.6. *For positive real numbers $\mathcal{T}$ and $\hat{\mathcal{T}}$, we define the following relations to be used in Protocol 4.1.*

$$\mathcal{R}_{\text{bin}}(\mathcal{T}) = \left\{ \begin{array}{c} ((ck, B), (b_{0,0}, \ldots, b_{k-1,\beta-1}, \boldsymbol{r})) \ : \ \|\boldsymbol{r}\| \leq \mathcal{T} \ \wedge \ (b_{j,i} \in \{0,1\} \ \forall j, i) \\ \wedge \ B = \text{Com}_{ck}\left(b_{0,0}, \ldots, b_{k-1,\beta-1}; \ \boldsymbol{r}\right) \ \wedge \ \left(\sum_{i=0}^{\beta-1} b_{j,i} = 1 \ \forall j\right) \end{array} \right\}.$$

$$\mathcal{R}'_{\text{bin}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{c} ((ck, B), (b_{0,0}, \ldots, b_{k-1,\beta-1}, \hat{\boldsymbol{r}})) \ : \ \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \ \wedge \ (b_{j,i} \in \{0,1\} \ \forall j, i) \\ \wedge \ 2B = \text{Com}_{ck}\left(2b_{0,0}, \ldots, 2b_{k-1,\beta-1}; \ \hat{\boldsymbol{r}}\right) \ \wedge \ \left(\sum_{i=0}^{\beta-1} b_{j,i} = 1 \ \forall j\right) \end{array} \right\}.$$

Remark 4.7. *The conditions on the norms of $\boldsymbol{r}$ and $\hat{\boldsymbol{r}}$ in the relations $\mathcal{R}_{\text{bin}}$ and $\mathcal{R}'_{\text{bin}}$ play a very crucial role, and is one of the main differences of a lattice-based zero-knowledge proof over its classical counterpart. Without that control, one cannot easily tie the security of the protocol to a hard lattice problem.*

In the protocol, we first prove that each value in the sequences is binary, and then that the sum of each sequence equals one. This guarantees that there is only a single 1 in each sequence. The idea behind proving a value binary works as follows. Let b be the value we want to prove binary. Given a challenge x, the value b is multiplied by x and the resulting value is masked by a as $f = x \cdot b + a$ in the protocol (Step 10 in Protocol 4.1). Now observe that $f \cdot (x - f) = b(1-b) \cdot x^2 + a(1-2b) \cdot x - a^2$ and proving that the coefficient of x^2 is zero implies that $b(1-b) = 0$. Then, using Lemma 4.1, for a sufficiently large q, this statement over R_q implies that b is binary.

Similar to [BCK+14], we make use of an auxiliary commitment scheme aCom (which is assumed to be hiding and binding) in order to be able to simulate aborts

$\mathcal{P}_{\text{bin}}((ck, B), (\{b_{j,i}\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}))$		$\mathcal{V}_{\text{bin}}(ck, B)$
$a_{0,1}, \ldots, a_{k-1,\beta-1} \leftarrow D^d_{\phi_1\sqrt{k}}$		
$\boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$		
$\boldsymbol{r}_a, \boldsymbol{r}_d \leftarrow D^{md}_{\phi_2\mathcal{B}\sqrt{2md}}$		
for $j = 0, \ldots, k-1$ **do**		
$a_{j,0} = -\sum_{i=1}^{\beta-1} a_{j,i}$		
$A = \text{Com}_{ck}(a_{0,0}, \ldots, a_{k-1,\beta-1}; \boldsymbol{r}_a)$		
$C = \text{Com}_{ck}\left(\{a_{j,i}(1-2b_{j,i})\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}_c\right)$		
$D = \text{Com}_{ck}(-a_{0,0}^2, \ldots, -a_{k-1,\beta-1}^2; \boldsymbol{r}_d)$		
$(c_a, d_a) = \text{aCom}(A, C, D)$		
	$\xrightarrow{c_a}$	
	$\xleftarrow{x := X^\omega}$	$\omega \leftarrow \{0, \ldots, 2d-1\}$
$f_{j,i} = x \cdot b_{j,i} + a_{j,i}\ \forall j, \forall i \neq 0$		
$\boldsymbol{f}_1 := (f_{0,1}, \ldots, f_{k-1,\beta-1})$		
$\boldsymbol{b}_1 := (b_{0,1}, \ldots, b_{k-1,\beta-1})$		
$\text{Rej}(\boldsymbol{f}_1, x\boldsymbol{b}_1, \phi_1, \sqrt{k})$		
$\boldsymbol{z}_b = x \cdot \boldsymbol{r} + \boldsymbol{r}_a$		
$\boldsymbol{z}_c = x \cdot \boldsymbol{r}_c + \boldsymbol{r}_d$		
$\text{Rej}((\boldsymbol{z}_b, \boldsymbol{z}_c), x(\boldsymbol{r}, \boldsymbol{r}_c), \phi_2, \mathcal{B}\sqrt{2md})$		
Return $\perp$ if aborted.	$\xrightarrow{\substack{f_{0,1}, \ldots, f_{k-1,\beta-1}, \\ d_a, A, C, D, \boldsymbol{z}_b, \boldsymbol{z}_c}}$	
		for $j = 0, \ldots, k-1$ **do**
		$f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$
		$(c_a, d_a) \stackrel{?}{=} \text{aCom}(A, C, D)$
		$\|f_{j,i}\| \stackrel{?}{\leq} 5\phi_1\sqrt{dk}\ \ \forall j, \forall i \neq 0$
		$\|f_{j,0}\| \stackrel{?}{\leq} 5\phi_1\sqrt{dk(\beta-1)}\ \ \forall j$
		$\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \stackrel{?}{\leq} 2\sqrt{2}\phi_2\mathcal{B}md$
		$\boldsymbol{f} := (f_{0,0}, \ldots, f_{k-1,\beta-1})$
		$\boldsymbol{g} := \{f_{j,i}(x - f_{j,i})\}_{j,i=0}^{k-1,\beta-1}$
		$xB + A \stackrel{?}{=} \text{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z}_b)$
		$xC + D \stackrel{?}{=} \text{Com}_{ck}(\boldsymbol{g}; \boldsymbol{z}_c)$

PROTOCOL 4.1: Lattice-based Σ-protocol for $\mathcal{R}_{\text{bin}}$ and $\mathcal{R}'_{\text{bin}}$.

in the proof of zero-knowledge property.[2] One can treat aCom as a random oracle. However, if aCom is computationally binding, then the soundness of the protocol holds under the respective assumption and similarly if it is computationally hiding [BCK+14]. The full protocol is described in Protocol 4.1, which will later be used in the one-out-of-many proof. The parameters ϕ_1, ϕ_2 control the acceptance rate of two-step rejection sampling and can be adjusted as desired.

Remark 4.8. *The way the rejection sampling is done in Protocol 4.1 allows us to sample $f_{j,i}$'s from a narrower distribution, and to make their norm smaller. This as a result weakens the condition on the size of q.*

We summarise the result of Protocol 4.1 below.

Theorem 4.9. *For $T = (2d+2)\left(5^4\phi_1^4 d^3 k^3 \beta(\beta-1) + 12\phi_2^2\mathcal{B}^2 m^2 d^2\right)^{1/2}$, assume that the commitment scheme is T-binding and also hiding (i.e., M-LWE$_{m-n,n,q,\mathcal{B}}$ is hard). Let $d \geq 7$, $md \geq 86$, and $q > (10\phi_1 d\sqrt{kd(\beta-1)}+2)^2$. Then, Protocol 4.1 is a 3-special sound Σ-protocol (as in Definition 3.6) for relations $\mathcal{R}_{\text{bin}}(\mathcal{B}\sqrt{md})$ and $\mathcal{R}'_{\text{bin}}(4\sqrt{2}\phi_2\mathcal{B}md^2)$ with soundness error $1/d$ and a completeness error $1 - \frac{1}{\mu(\phi_1)\mu(\phi_2)}$ for $\mu(\cdot)$ defined in Lemma 3.16.*

Proof. **Completeness:** By Lemma 3.16, prover responds with probability statistically close to $1/(\mu(\phi_1)\mu(\phi_2))$, and distributions of $f_{j,i}$'s ($i \neq 0$) are statistically close to $D^d_{\phi_1\sqrt{k}}$ and that of $\boldsymbol{z}_b, \boldsymbol{z}_c$ are statistically close to $D^{md}_{\phi_2\mathcal{B}\sqrt{2md}}$ since

$$\|(x\cdot b_{0,1},\ldots,x\cdot b_{k-1,\beta-1})\| \leq \sqrt{k}, \quad \text{and} \quad \|(x\cdot \boldsymbol{r}, x\cdot \boldsymbol{r}_c)\| \leq \mathcal{B}\sqrt{2md}.$$

Since the standard deviation of all sampled discrete normal coefficients are much larger than 6, the sum of discrete normal samples behave as in the continuous case by Fact 3.12 and Lemma 3.14. That is, the distribution of $\sum_{i=1}^{\beta-1} f_{j,i}$ is statistically close to $D^d_{\phi_1\sqrt{k(\beta-1)}}$. Therefore, if the prover does not abort, and since $d \geq 7$ and $md \geq 86$, by Lemma 3.15 except with probability at most 2^{-100}, we have,

$$\|f_{j,i}\| \leq 5\cdot\phi_1\sqrt{k}\cdot\sqrt{d} = 5\phi_1\sqrt{dk}, \qquad \forall j\in[0,k-1], \forall i\in[1,\beta-1],$$
$$\|f_{j,0}\| = \left\|x - \sum_{i=1}^{\beta-1} f_{j,i}\right\| \leq 5\cdot\phi_1\sqrt{k(\beta-1)}\cdot\sqrt{d} = 5\phi_1\sqrt{dk(\beta-1)}, \quad \forall j\in[0,k-1],$$

and $\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \leq 2\cdot\phi_2\mathcal{B}\sqrt{2md}\cdot\sqrt{md} = 2\phi_2\sqrt{2}\mathcal{B}md$, proving the bounds on the norms. The other verification steps follow via straightforward investigation.

SHVZK: Given a challenge x, the simulator outputs $(\text{aCom}(0), x, \perp)$ indicating an abort with probability $1 - 1/(\mu(\phi_1)\mu(\phi_2))$. Otherwise, it picks $C \leftarrow R_q^n$, $f_{j,i} \leftarrow D^d_{\phi_1\sqrt{k}}$ for all $0 \leq j \leq k-1$ and $1 \leq i \leq \beta-1$, and also $\boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow D^{md}_{\phi_2\mathcal{B}\sqrt{2md}}$. Then, it sets $f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$ for all $j = 0,\ldots,k-1$. Finally, it computes $A = \text{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z}_b) - xB$, $D = \text{Com}_{ck}(\{f_{j,i}(x-f_{j,i})\}_{j,i}; \boldsymbol{z}_c) - xC$ and $(c_a, d_a) = \text{aCom}(A, C, D)$ where $\boldsymbol{f} = (f_{0,0},\ldots,f_{k-1,\beta-1})$. It outputs the simulated transcript $(c_a, x, (d_a, \{f_{j,i}\}_{j=0,i=1}^{k-1,\beta-1}, A, C, D, \boldsymbol{z}_b, \boldsymbol{z}_c))$.

Note that the narrowest distribution where a randomness coefficient is sampled from is $\mathcal{U}(\{-\mathcal{B},\ldots,\mathcal{B}\})$ and M-LWE$_{m-n,n,q,\mathcal{B}}$ is assumed to be hard. Therefore, by Lemma 3.3, all of the commitments are computationally indistinguishable from uniformly random elements in R_q^n. Hence, if the protocol is not aborted, the real and

[2] In protocol's application to a ring signature (and for other applications in general), simulation of aborts is not needed as the protocol is made non-interactive.

simulated transcripts are indistinguishable by Lemma 3.16 and the hiding property of the commitment scheme. If an abort occurs, then the indistinguishability is satisfied due to hiding property of aCom and the fact that the probability of having an abort is the same for all x.

3-special soundness: Given 3 accepting transcripts, by the binding property of aCom, we have the tuples $(A, C, D, x, f_{0,1}, \ldots, f_{k-1,\beta-1}, \boldsymbol{z}_b, \boldsymbol{z}_c)$, $(A, C, D, x', f'_{0,1}, \ldots, f'_{k-1,\beta-1}, \boldsymbol{z}'_b, \boldsymbol{z}'_c)$, $(A, C, D, x'', f''_{0,1}, \ldots, f''_{k-1,\beta-1}, \boldsymbol{z}''_b, \boldsymbol{z}''_c)$. Let $\boldsymbol{f} = (f_{0,0}, \ldots, f_{k-1,\beta-1})$, $\boldsymbol{f}' = (f'_{0,0}, \ldots, f'_{k-1,\beta-1})$, $\boldsymbol{f}'' = (f''_{0,0}, \ldots, f''_{k-1,\beta-1})$ where $f_{j,0}, f'_{j,0}, f''_{j,0}$'s are computed as in the verification. Then, by Step 7 in the verification, we have $xB + A = \mathrm{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z}_b)$ and $x'B + A = \mathrm{Com}_{ck}(\boldsymbol{f}'; \boldsymbol{z}'_b)$. By subtracting the equations and multiplying both sides by $2(x - x')^{-1}$, we get

$$2B = \mathrm{Com}_{ck}\left(2(x-x')^{-1}(\boldsymbol{f} - \boldsymbol{f}'); 2(x-x')^{-1}(\boldsymbol{z}_b - \boldsymbol{z}'_b)\right) =: \mathrm{Com}_{ck}\left(\hat{\boldsymbol{b}}; \hat{\boldsymbol{r}}_b\right).$$

This gives us openings of $2B$ as $\hat{\boldsymbol{b}} = (\hat{b}_{0,0}, \ldots, \hat{b}_{k-1,\beta-1})$ and $\hat{\boldsymbol{r}}_b$. Note that

$$\begin{aligned}\|\hat{\boldsymbol{r}}_b\| &= \left\|2(x-x')^{-1}(\boldsymbol{z}_b - \boldsymbol{z}'_b)\right\| \leq \sqrt{d} \cdot \left\|2(x-x')^{-1}\right\| \cdot \left\|(\boldsymbol{z}_b - \boldsymbol{z}'_b)\right\| \\ &\leq d \cdot \left\|(\boldsymbol{z}_b - \boldsymbol{z}'_b)\right\| \leq d \cdot 2 \cdot 2\sqrt{2}\phi_2 \mathcal{B} m d = 4\sqrt{2}\phi_2 \mathcal{B} m d^2,\end{aligned}$$

which proves the required norm-bound on the extracted randomness for $\mathcal{R}'_{\mathrm{bin}}$.

We can also recover openings of $2A$ by computing $\hat{a}_{j,i} = 2f_{j,i} - x \cdot \hat{b}_{j,i}$ and $\hat{\boldsymbol{r}}_a = 2\boldsymbol{z}_b - x \cdot \hat{\boldsymbol{r}}_b$. Similarly, by Step 8 of the verification, we get openings $\hat{c}_{j,i}$ and $\hat{d}_{j,i}$ of $2C$ and $2D$, respectively, such that $2g_{j,i} = x\hat{c}_{j,i} + \hat{d}_{j,i}$ and $g_{j,i} = f_{j,i}(x - f_{j,i})$. From here, by multiplying the former by 2, we get

$$\begin{aligned}2 \cdot \left(x \cdot \hat{c}_{j,i} + \hat{d}_{j,i}\right) &= 2 \cdot 2g_{j,i} = 2 \cdot (2f_{j,i}(x - f_{j,i})) = 2f_{j,i}(2x - 2f_{j,i}) \\ &= x^2\left[\hat{b}_{j,i}(2 - \hat{b}_{j,i})\right] + x\left[2\hat{a}_{j,i}(1 - \hat{b}_{j,i})\right] - \hat{a}^2_{j,i},\end{aligned}$$

which implies

$$x^2\left[\hat{b}_{j,i}(2 - \hat{b}_{j,i})\right] + x\left[2\hat{a}_{j,i}(1 - \hat{b}_{j,i}) - 2\hat{c}_{j,i}\right] - \hat{a}^2_{j,i} - 2\hat{d}_{j,i} = 0. \tag{4.9}$$

By Lemma 4.11 (further below), norms of the openings of $2A, 2B, 2C, 2D$ are all smaller than T. By the T-binding property of the commitment scheme, PPT prover cannot know other openings of $2A, 2B, 2C$ or $2D$. Thus, (4.9) also holds for the other challenges x' and x'' with the same $\hat{a}_{j,i}, \hat{b}_{j,i}, \hat{c}_{j,i}, \hat{d}_{j,i}$'s. Then, we can write this system of equations as

$$\begin{pmatrix} 1 & x & x^2 \\ 1 & x' & x'^2 \\ 1 & x'' & x''^2 \end{pmatrix} \cdot \begin{pmatrix} -\hat{a}^2_{j,i} - 2\hat{d}_{j,i} \\ 2\hat{a}_{j,i}(1 - \hat{b}_{j,i}) - 2\hat{c}_{j,i} \\ \hat{b}_{j,i}(2 - \hat{b}_{j,i}) \end{pmatrix} = \boldsymbol{0} \qquad \text{over } R_q.$$

The left-most matrix is a Vandermonde matrix $\boldsymbol{V}$, which is invertible by Lemma 4.2. Therefore, we get $\hat{b}_{j,i}(2 - \hat{b}_{j,i}) = 0$ over R_q. Further, we have

$$\begin{aligned}\left\|\hat{b}_{j,i}\right\| &= \left\|2(x-x')^{-1}(f_{j,i} - f'_{j,i})\right\| \leq \sqrt{d} \cdot \left\|2(x-x')^{-1}\right\| \cdot \left\|f_{j,i} - f'_{j,i}\right\| \\ &\leq d \cdot \left\|f_{j,i} - f'_{j,i}\right\| \leq d \cdot 2 \cdot (5\phi_1\sqrt{dk(\beta-1)}) = 10\phi_1 d\sqrt{dk(\beta-1)}.\end{aligned}$$

Since $q > \left(10\phi_1 d\sqrt{dk(\beta-1)}+2\right)^2 \geq \left(\left\|\hat{b}_{j,i}\right\|+2\right)^2$, we have $\hat{b}_{j,i} = 2b_{j,i}$ for $b_{j,i} \in \{0,1\}$ by Lemma 4.1. Moreover, by construction, for all $j = 0,\ldots,k-1$,

$$2x = \sum_{i=0}^{\beta-1} 2f_{j,i} = x \cdot \sum_{i=0}^{\beta-1} 2b_{j,i} + \sum_{i=0}^{\beta-1} \hat{a}_{j,i} = 2x \cdot \sum_{i=0}^{\beta-1} b_{j,i} + \sum_{i=0}^{\beta-1} \hat{a}_{j,i}.$$

If this is true for 2 distinct challenges x and x', then $\sum_{i=0}^{\beta-1} b_{j,i} = 1$ for all $j = 0,\ldots,k-1$ as desired. Finally, since the protocol is 3-special sound and $|\mathcal{C}| = 2d$, the soundness error is $2/(2d) = 1/d$. □

As evident from the discussions on lattice-based commitment schemes in Section 3.2.2, it is important to keep track of the norm of an opening of a commitment. For Protocol 4.1, we do that in the following lemmas.

Lemma 4.10. *The vector $\boldsymbol{g}$ defined in the verification of Protocol 4.1 satisfy the following $\|\boldsymbol{g}\|^2 \leq 5^4\phi_1^4 d^3k^3\beta(\beta-1)$.*

Proof. Since x is a monomial, we simply upper-bound $\|x - f_{j,i}\|$ by $\|f_{j,i}\|$ below.

$$\begin{aligned}
\|\boldsymbol{g}\|^2 &= \sum_{j=0}^{k-1}\sum_{i=0}^{\beta-1} \|f_{j,i}(x-f_{j,i})\|^2 \leq \sum_{j=0}^{k-1}\sum_{i=0}^{\beta-1} d\,\|f_{j,i}\|^2\,\|x-f_{j,i}\|^2 \\
&= \sum_{j=0}^{k-1}\sum_{i=1}^{\beta-1} d\,\|f_{j,i}\|^2\,\|x-f_{j,i}\|^2 + \sum_{j=0}^{k-1} d\,\|f_{j,0}\|^2\,\|x-f_{j,0}\|^2 \\
&\leq \sum_{j=0}^{k-1}\sum_{i=1}^{\beta-1} d\left(5\phi_1\sqrt{dk}\right)^2\left(5\phi_1\sqrt{dk}\right)^2 + \sum_{j=0}^{k-1} d\left(5\phi_1\sqrt{dk(\beta-1)}\right)^2\left(5\phi_1\sqrt{dk(\beta-1)}\right)^2 \\
&\leq dk(\beta-1)\left(5\phi_1\sqrt{dk}\right)^4 + dk\left(5\phi_1\sqrt{dk(\beta-1)}\right)^4 \\
&= 5^4\phi_1^4 d^3k^3(\beta-1) + 5^4\phi_1^4 d^3k^3(\beta-1)^2 \\
&= 5^4\phi_1^4 d^3k^3\beta(\beta-1)
\end{aligned}$$

□

Lemma 4.11. *The opening $(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_d)$ of $2D$ in the special soundness proof of Protocol 4.1 satisfy the following*

$$\left\|(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_d)\right\| \leq (2d+2)\left(5^4\phi_1^4 d^3k^3\beta(\beta-1) + 12\phi_2^2\mathcal{B}^2m^2d^2\right)^{1/2}.$$

Furthermore, the same bound applies to the openings of $2A, 2B$ and $2C$.

Proof. For distinct challenges x and x', recall the opening $(\hat{\boldsymbol{b}}, \hat{\boldsymbol{r}}_b)$ of $2B$ in the special soundness proof of Protocol 4.1. We have

$$\hat{\boldsymbol{b}} = 2(x-x')^{-1}(\boldsymbol{f}-\boldsymbol{f}') \quad \text{and} \quad \hat{\boldsymbol{r}}_b = 2(x-x')^{-1}(\boldsymbol{z}_b - \boldsymbol{z}_b'). \tag{4.10}$$

Similarly, recalling the opening $(\hat{\boldsymbol{a}}, \hat{\boldsymbol{r}}_a)$ of $2A$, we have

$$\hat{\boldsymbol{a}} = 2\boldsymbol{f} - x\hat{\boldsymbol{b}} \quad \text{and} \quad \hat{\boldsymbol{r}}_a = 2\boldsymbol{z}_b - x\hat{\boldsymbol{r}}_b. \tag{4.11}$$

TABLE 4.2: A summary of identifiers for Chapter 4.

Notation	Explanation
$N = \beta^k$	the number of public commitments for one-out-of-many proof (or the ring size for the ring signature)
β	base for the representation of user indices
q	an odd modulus
d	ring dimension (i.e., $R_q = \mathbb{Z}_q[X]/(X^d + 1)$)
$k \cdot \beta$	the number of packed messages in a commitment
m	the dimension of randomness in a commitment (i.e., $\boldsymbol{r} \in R_q^m$)
$n \times (m + k\beta)$	public commitment key dimensions (i.e., $\boldsymbol{G} \in R_q^{n\times(m+k\beta)}$)
$n \times 1$	commitment dimensions
$\mathcal{B}$	maximum absolute coefficient of a uniformly chosen fresh randomness
r	number of protocol repetitions to achieve negligible soundness error
ℓ	prover's index with $0 \leq \ell \leq N - 1$
$\mathcal{C}$	challenge space with $\mathcal{C} = \{X^\omega \ : \ 0 \leq \omega \leq 2d - 1\}$
ϕ_1, ϕ_2	parameters controlling the acceptance rate of rejection sampling

Following the same procedure using the last verification step of Protocol 4.1, we can get the following openings $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}_c)$ and $(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_d)$ of $2C$ and $2D$, respectively,

$$(\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}_c) = (2(x - x')^{-1}(\boldsymbol{g} - \boldsymbol{g}'), 2(x - x')^{-1}(\boldsymbol{z}_c - \boldsymbol{z}'_c)), \tag{4.12}$$

$$(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_d) = (2\boldsymbol{g} - x\hat{\boldsymbol{c}}, 2\boldsymbol{z}_c - x\hat{\boldsymbol{r}}_c). \tag{4.13}$$

We bound the norm of $(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_d)$, which also involves bounding the norm of $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}_c)$. Without loss of generality, assume that $\|\boldsymbol{g}\| \geq \|\boldsymbol{g}'\|$ and $\|\boldsymbol{z}_c\| \geq \|\boldsymbol{z}'_c\|$. We use a stronger bound from Protocol 4.2 (to be described in Section 4.2.2) in order to bound $\|\boldsymbol{z}_c\|$ below. This way, we can make use of the same result in both of the protocols.

$$\begin{aligned}
\left\|(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_d)\right\| &= \|(2\boldsymbol{g} - x\hat{\boldsymbol{c}}, 2\boldsymbol{z}_c - x\hat{\boldsymbol{r}}_c)\| \leq \|(2\boldsymbol{g}, 2\boldsymbol{z}_c)\| + \|(x\hat{\boldsymbol{c}}, x\hat{\boldsymbol{r}}_c)\| \\
&\leq 2\,\|(\boldsymbol{g}, \boldsymbol{z}_c)\| + \|(\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}_c)\| \\
&= 2\,\|(\boldsymbol{g}, \boldsymbol{z}_c)\| + \left\|(2(x - x')^{-1}(\boldsymbol{g} - \boldsymbol{g}'), 2(x - x')^{-1}(\boldsymbol{z}_c - \boldsymbol{z}'_c)\right\| \\
&\leq 2\,\|(\boldsymbol{g}, \boldsymbol{z}_c)\| + \sqrt{d}\left\|2(x - x')^{-1}\right\| \left\|((\boldsymbol{g} - \boldsymbol{g}'), (\boldsymbol{z}_c - \boldsymbol{z}'_c))\right\| \\
&\leq 2\,\|(\boldsymbol{g}, \boldsymbol{z}_c)\| + 2d\,\|(\boldsymbol{g}, \boldsymbol{z}_c)\| \\
&\leq (2d + 2)\left(5^4\phi_1^4 d^3 k^3 \beta(\beta - 1) + \left(2\sqrt{3}\phi_2 \mathcal{B} m d\right)^2\right)^{1/2} \\
&= (2d + 2)\left(5^4\phi_1^4 d^3 k^3 \beta(\beta - 1) + 12\phi_2^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}.
\end{aligned} \tag{4.14}$$

The bounds on openings of $2A$ and $2B$ are clearly weaker as they only involve $f_{j,i}$'s whereas opening of $2D$ involves the products $f_{j,i}(x - f_{j,i})$'s as part of $\boldsymbol{g}$. □

4.2.2 One-out-of-many protocol

We are now ready to describe our main protocol. Let $\delta_{j,i}$ denote the Kronecker's delta such that $\delta_{j,i} = 1$ if $j = i$, and $\delta_{j,i} = 0$ otherwise. The prover's goal in the protocol is to show that he knows the randomness within a commitment to zero among a list of N commitments. The commitments other than the prover's need not be commitments to zero, i.e., there is no need to assume that they are well-formed. Similar to the previous

works [GK15, BCC+15], we assume that the number of commitments satisfy $N = \beta^k$, which can be realised by using the same commitment multiple times until such an N is reached. Let c_ℓ be the prover's commitment for $0 \le \ell \le N-1$, and $\boldsymbol{L} = \{c_0, \ldots, c_{N-1}\}$ be the list of all commitments. The main idea is to prove knowledge of the index ℓ such that $\sum_{i=0}^{N-1} \delta_{\ell,i} c_i$ is a commitment to zero. Note that $\delta_{\ell,i} = \prod_{j=0}^{k-1} \delta_{\ell_j,i_j}$ where $\ell = (\ell_0, \ldots, \ell_{k-1})$ and $i = (i_0, \ldots, i_{k-1})$ are representations in base β. The relations for the protocol are given in Definition 4.12 and a set of identifier are given in Table 4.2.

Definition 4.12. *For positive real numbers $\mathcal{T}$ and $\hat{\mathcal{T}}$, we define the following relations to be used in Protocol 4.2.*

$$\mathcal{R}_{1/\mathrm{N}}(\mathcal{T}) = \left\{ \begin{array}{c} ((ck, (c_0, \ldots, c_{N-1})), (\ell, \boldsymbol{r})) \ : \ (c_i \in R_q^n \ \ \forall i \in [0, N-1]) \wedge \\ \ell \in \{0, \ldots, N-1\} \ \wedge \ \|\boldsymbol{r}\| \le \mathcal{T} \ \wedge \ c_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{r}) \end{array} \right\}.$$

$$\mathcal{R}'_{1/\mathrm{N}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{c} ((ck, (c_0, \ldots, c_{N-1})), (\ell, \hat{\boldsymbol{r}})) \ : \ (c_i \in R_q^n \ \ \forall i \in [0, N-1]) \wedge \\ \ell \in \{0, \ldots, N-1\} \ \wedge \ \|\hat{\boldsymbol{r}}\| \le \hat{\mathcal{T}} \ \wedge \ 2^k c_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; \hat{\boldsymbol{r}}) \end{array} \right\}.$$

For each $0 \le j \le k-1$, the prover commits to a sequence $(\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,\beta-1})$ and proves that it is a binary sequence with Hamming weight one using Protocol 4.1. As given in Protocol 4.1, the prover responds with $f_{j,i} = x \cdot \delta_{\ell_j,i} + a_{j,i}$ upon receiving a challenge x. Now, let us concentrate on the product $\prod_{j=0}^{k-1} f_{j,i_j} =: p_i(x)$. Observe that for all $i \in \{0, \ldots, N-1\}$,

$$p_i(x) = \prod_{j=0}^{k-1} \left(x\delta_{\ell_j,i_j} + a_{j,i_j} \right) = \prod_{j=0}^{k-1} x\delta_{\ell_j,i_j} + \sum_{j=0}^{k-1} p_{i,j} x^j = \delta_{\ell,i} x^k + \sum_{j=0}^{k-1} p_{i,j} x^j, \qquad (4.15)$$

for some coefficients $p_{i,j}$'s depending on ℓ and $a_{j,i}$. This means that $p_{i,j}$'s can be computed by the prover before receiving a challenge. Now, since $\delta_{\ell,i} = 1$ if and only if $i = \ell$, the only p_i of degree k is p_ℓ. Then, the idea is to send some E_j's in the initial message, which will later be used by the verifier to cancel out the coefficients of low order terms $1, x, \ldots, x^{k-1}$, and the coefficient of x^k will be $\sum_{i=0}^{N-1} \delta_{\ell,i} c_i = c_\ell$, which corresponds to the prover's commitment. The full protocol is described in Protocol 4.2, and its results are summarised in the following theorem.

Theorem 4.13. *For $T = (2d+2)\left(5^4 \phi_1^4 d^3 k^3 \beta(\beta-1) + 12\phi_2^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}$, assume that the commitment scheme is T-binding and also hiding (i.e., M-LWE$_{m-n,n,q,\mathcal{B}}$ is hard). Let $d \ge 7$, $md \ge 86$, and $q > (10\phi_1 d\sqrt{dk(\beta-1)} + 2)^2$. Then, Protocol 4.2 is a $(k'+1)$-special sound Σ-protocol (as in Definition 3.6) for the relations $\mathcal{R}_{1/\mathrm{N}}(\mathcal{B}\sqrt{md})$ and $\mathcal{R}'_{1/\mathrm{N}}(2\sqrt{3}\phi_2 \mathcal{B} md \cdot (k+1) \cdot d^k)$ with a soundness error $\frac{k'}{2d}$ and a completeness error $1 - 1/(\mu(\phi_1)\mu(\phi_2))$ where $k' = \max\{2, k\}$ and $\mu(\cdot)$ is defined in Lemma 3.16.*

Proof. **Completeness:** Note that multiplication by x in R_q simply performs a nega-cyclic rotation of the coefficients of a polynomial and thus the distribution of $\sum_{j=0}^{k-1} x^j \boldsymbol{\rho}_j$ is statistically close to $D^{md}_{\phi_2 \mathcal{B}\sqrt{3md}}$ by Lemma 3.14. From here the bounds on the norms of each component follow similar to the completeness proof of Theorem 4.9.

All the remaining but the last verification steps also follow straightforwardly. To prove that the last verification step holds for honestly generated values, we have, for $c_\ell = \mathrm{Com}_{ck}(\boldsymbol{m}_\ell; \boldsymbol{r})$,

$$\sum_{i=0}^{N-1} \left(\prod_{j=0}^{k-1} f_{j,i_j} \right) c_i - \sum_{j=0}^{k-1} E_j x^j = \sum_{i=0}^{N-1} p_i(x) c_i - \sum_{j=0}^{k-1} \left(\sum_{i=0}^{N-1} p_{i,j} c_i + \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{\rho}_j) \right) x^j$$

$\mathcal{P}(ck, (c_0, \ldots, c_{N-1}), (\ell, \boldsymbol{r}))$ $\qquad$ $\mathcal{V}(ck, (c_0, \ldots, c_{N-1}))$

1: $\boldsymbol{r}_b \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$

2: $\boldsymbol{\delta} = (\delta_{\ell_0,0}, \ldots, \delta_{\ell_{k-1},\beta-1})$

3: $B = \mathrm{Com}_{ck}(\boldsymbol{\delta}; \boldsymbol{r}_b)$

4: $A, C, D, \boldsymbol{r}_c \leftarrow \mathcal{P}_{\mathrm{bin}}(ck, B, (\boldsymbol{\delta}, \boldsymbol{r}_b))[1-8]$

5: **for** $j = 0, \ldots, k-1$ **do**

6: $\quad \boldsymbol{\rho}_j \leftarrow D^{md}_{\phi_2 \mathcal{B}\sqrt{3md/k}}$

7: $\quad E_j = \sum_{i=0}^{N-1} p_{i,j} c_i + \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{\rho}_j)$

using $p_{i,j}$'s from (4.15)

8: $(c_a, d_a) = \mathrm{aCom}(A, B, C, D, \{E_j\})$

$\xrightarrow{\quad c_a \quad}$

$\xleftarrow{\quad x = X^\omega \quad}$ $\omega \leftarrow \{0, \ldots, 2d-1\}$

9: $\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow \mathcal{P}_{\mathrm{bin}}(x)[10-13]$

10: $\boldsymbol{z} = x^k \cdot \boldsymbol{r} - \sum_{j=0}^{k-1} x^j \cdot \boldsymbol{\rho}_j$

11: $\mathrm{Rej}((\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c), (x^k \boldsymbol{r}, x\boldsymbol{r}_b, x\boldsymbol{r}_c), \phi_2, \mathcal{B}\sqrt{3md})$

Return $\perp$ if aborted.

$\xrightarrow{\quad d_a, \boldsymbol{f}_1, B, \boldsymbol{z}, \{E_j\}_{j=0}^{k-1} \quad \boldsymbol{R} := (A, C, D, \boldsymbol{z}_b, \boldsymbol{z}_c) \quad}$

1: $\mathcal{V}_{\mathrm{bin}}(ck, B, x, \boldsymbol{f}_1, \boldsymbol{R})[1,2,6,7] \stackrel{?}{=} 1$

2: $(c_a, d_a) \stackrel{?}{=} \mathrm{aCom}(A, B, C, D, \{E_j\})$

3: $\|f_{j,i}\| \stackrel{?}{\leq} 5\phi_1\sqrt{dk} \quad \forall j, \forall i \neq 0$

4: $\|f_{j,0}\| \stackrel{?}{\leq} 5\phi_1\sqrt{dk(\beta-1)} \quad \forall j$

5: $\|\boldsymbol{z}\|, \|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \stackrel{?}{\leq} 2\sqrt{3}\phi_2 \mathcal{B} md$

6: $\sum_{i=0}^{N-1} \left(\prod_{j=0}^{k-1} f_{j,i_j} \right) c_i - \sum_{j=0}^{k-1} E_j x^j \stackrel{?}{=} \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{z})$

for $i = (i_0, \ldots, i_{k-1})$.

PROTOCOL 4.2: Lattice-based Σ-protocol for $\mathcal{R}_{1/\mathrm{N}}$ and $\mathcal{R}'_{1/\mathrm{N}}$.

$\mathcal{P}_{\mathrm{bin}}(ck, B, (\boldsymbol{\delta}, \boldsymbol{r}_b))[1-8]$ denotes running the same steps from 1 to 8 done by $\mathcal{P}_{\mathrm{bin}}$ in Protocol 4.1. Similar notation is used for $\mathcal{V}_{\mathrm{bin}}$. $\boldsymbol{r}_a$ and $\boldsymbol{r}_d$ in $\mathcal{P}_{\mathrm{bin}}(ck, B, (\boldsymbol{\delta}, \boldsymbol{r}_b))[1-8]$ are drawn from $D^{md}_{\phi_2 \mathcal{B}\sqrt{3md}}$ instead of $D^{md}_{\phi_2 \mathcal{B}\sqrt{2md}}$ as the rejection sampling is now done on a $(3md)$-dimensional vector.

$$= \sum_{i=0}^{N-1} p_i(x)c_i - \sum_{j=0}^{k-1}\sum_{i=0}^{N-1} p_{i,j}c_i x^j - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}\left(\mathbf{0};\, \boldsymbol{\rho}_j\right)$$

$$= \sum_{i=0}^{N-1} c_i \left(p_i(x) - \sum_{j=0}^{k-1} p_{i,j}x^j\right) - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}\left(\mathbf{0};\, \boldsymbol{\rho}_j\right)$$

$$= \sum_{i=0}^{N-1} c_i\delta_{\ell,i}x^k - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}\left(\mathbf{0};\, \boldsymbol{\rho}_j\right) = x^k \cdot c_\ell - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}\left(\mathbf{0};\, \boldsymbol{\rho}_j\right)$$

$$= \mathrm{Com}_{ck}\left(x^k\boldsymbol{m}_\ell;\, x^k\boldsymbol{r} - \sum_{j=0}^{k-1} x^j\boldsymbol{\rho}_j\right) = \mathrm{Com}_{ck}\left(x^k\boldsymbol{m}_\ell;\, \boldsymbol{z}\right) = \mathrm{Com}_{ck}\left(\mathbf{0};\, \boldsymbol{z}\right) \quad \text{if } \boldsymbol{m}_\ell = \mathbf{0}.$$

SHVZK: Given a challenge x, the simulator outputs $(\mathrm{aCom}(0), x, \perp)$ indicating an abort with probability $1 - \frac{1}{\mu(\phi_1)\mu(\phi_2)}$. Otherwise, it picks $B, C, E_1, \ldots, E_{k-1} \leftarrow R_q^n$ and $f_{j,i} \leftarrow D^d_{\phi_1\sqrt{k}}$ for all $0 \le j \le k-1$ and $1 \le i \le \beta - 1$, and also picks $\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow D^{md}_{\phi_2\mathcal{B}\sqrt{3md}}$. Then, it calculates $f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$ for all $0 \le j \le k-1$, and computes E_0 so as to ensure that the last verification equation is satisfied. Similarly, it computes A and D so that the corresponding verification equations are satisfied. Then, it calculates $(c_a, d_a) = \mathrm{aCom}(A, B, C, D, \{E_j\}_{j=0}^{k-1})$ and outputs the simulated transcript

$$(c_a, x, (d_a, \{f_{j,i}\}_{i\neq 0}, A, B, C, D, \{E_j\}_{j=0}^{k-1}, \boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c)).$$

Note that the narrowest distribution where a randomness coefficient is sampled from is $\mathcal{U}(\{-\mathcal{B}, \ldots, \mathcal{B}\})$ and M-LWE$_{m-n,n,q,\mathcal{B}}$ is assumed to be hard. Therefore, by Lemma 3.3, all of the commitments are computationally indistinguishable from uniformly random elements in R_q^n. Hence, if the protocol is not aborted, the real and simulated transcripts are indistinguishable by Lemma 3.16, the hiding property of the commitment scheme and the fact that A, D, E_0 are uniquely determined by the verification equations given all the other components in both the real proof and the simulation. If an abort occurs, then the indistinguishability is satisfied due to hiding property of aCom and the fact that the probability of having an abort is the same for all x.

$(k'+1)$-special soundness: Assume that $k > 1$. Given $(k+1)$ distinct challenges $x_0, \ldots, x_k$, by aCom's binding property, we have $(k+1)$ accepting responses with the same $(A, B, C, D, \{E_j\})$. Suppose that $((f_{j,i}^{(0)}, \boldsymbol{z}^{(0)}), \ldots, (f_{j,i}^{(k)}, \boldsymbol{z}^{(k)}))$ are produced. We first use 3-special soundness of Protocol 4.1 to extract openings $\hat{b}_{j,i}$ and $\hat{a}_{j,i}$ of $2B$ and $2A$, respectively. We can also obtain $b_{j,i}$ such that $\hat{b}_{j,i} = 2b_{j,i}$, and it is guaranteed that $b_{j,i} \in \{0, 1\}$ and $\sum_{i=0}^{\beta-1} b_{j,i} = 1$. From here, we can obtain the digits ℓ_j by choosing $\ell_j = i^*$ for which $b_{j,i^*} = 1$. Then, we construct the index ℓ as $\ell = \sum_{j=0}^{k-1} \beta^j \ell_j$.

Using $b_{j,i}$ and $\hat{a}_{j,i}$, we can compute $\hat{p}_i(x) = 2^k \prod_{j=0}^{k-1} f_{j,i_j} = \prod_{j=0}^{k-1} 2f_{j,i_j} = \prod_{j=0}^{k-1}(x \cdot 2b_{j,i_j} + \hat{a}_{j,i_j})$. Note that $\hat{p}_\ell(x)$ is the only such polynomial of degree k in x by the construction of ℓ. Thus, the last verification step, when both sides are multiplied by 2^k, can be rewritten as $\sum_{i=0}^{N-1} \hat{p}_i(x)c_i - \sum_{j=0}^{k-1} 2^k E_j x^j = \mathrm{Com}_{ck}\left(\mathbf{0};\, 2^k\boldsymbol{z}\right)$. Separating the term of degree k with respect to x, we get

$$x^k \cdot 2^k c_\ell + \sum_{j=0}^{k-1} \tilde{E}_j x^j = \mathrm{Com}_{ck}\left(\mathbf{0};\, 2^k\boldsymbol{z}\right), \tag{4.16}$$

where $\tilde{E}_j$'s are the coefficients of the monomials x^j of degree strictly less than k. Now, we know that (4.16) holds for distinct challenges $x_0, \ldots, x_k$, which can be represented

as a system of equations where $x_0, \ldots, x_k$ form a Vandermonde matrix $\boldsymbol{V}$ as in Section 4.1.2. From the discussion in Section 4.1.2, $\boldsymbol{V}$ is invertible and we can obtain a linear combination $\alpha_0, \ldots, \alpha_k$ of copies of (4.16) with respect to different challenges that produces the vector $(0, \ldots, 0, 1)$. This gives

$$2^k c_\ell = \sum_{e=0}^{k} \alpha_e \left(x_e^k \cdot 2^k c_\ell + \sum_{j=0}^{k-1} \tilde{E}_j x_e^j \right) = \mathrm{Com}_{ck} \left(\mathbf{0}; 2^k \sum_{e=0}^{k} \alpha_e \boldsymbol{z}^{(e)} \right). \tag{4.17}$$

An opening of $2^k c_\ell$ to the message $\mathbf{0}$ with randomness $\boldsymbol{r}_{\text{ext}} = 2^k \sum_{e=0}^{k} \alpha_e \boldsymbol{z}^{(e)}$ is obtained. The bound on the norm of $\boldsymbol{r}_{\text{ext}}$ for $\mathcal{R}'_{1/\mathrm{N}}$ follows easily by Lemma 4.3.

Finally, we assumed that $k > 1$. If $k = 1$, then we still need at least 3 challenges to be able to prove special soundness due to the 3-special soundness of Protocol 4.1. Thus, Protocol 4.2 is $(k'+1)$-special sound for $k' = \max\{2, k\}$, and since $|\mathcal{C}| = 2d$, the soundness error is $k'/2d$. □

It is easy to see from the definition of $\mathcal{R}'_{1/\mathrm{N}}$ that the norm of the extracted randomness, and thus the size of q, grows with $d^k = d^{\log_\beta N}$. If one is to rely on Ring-SIS and use a base $\beta = 2$, then this growth would be very rapid, yielding a very inefficient scheme. This justifies our choice of working with M-SIS problem and choosing large base values β as given in Section 4.3.4. As discussed in Section 4.1.2, the bound on $\|\boldsymbol{r}_{\text{ext}}\|$ can be tightened using Method 2 or Method 3.

4.3 Application to Ring Signature

Let $N = \beta^k$ for $2 \leq \beta \leq N$, and n, m be fixed positive integers. As a single run of Protocol 4.2 does not provide a small enough soundness error, suppose that r non-aborting executions of Protocol 4.2 gives negligible soundness error of $2^{-\lambda}$.

Recall that a single run of Protocol 4.2 produces an accepting transcript with probability $1/(\mu(\phi_1)\mu(\phi_2))$. Therefore, when it is repeated r times, the overall acceptance rate reduces to $1/(\mu(\phi_1)\mu(\phi_2))^r$, which is too small. Therefore, we introduce the tweaks below to Protocol 4.2 in order to get an overall completeness error of $1 - 1/(\mu(\phi_1)\mu(\phi_2))$ for the r-repeated protocol.

4.3.1 Tweaks for r-repeated protocol

First, we apply the rejection sampling to r-concatenated vectors at once. That is, it is applied on $(\boldsymbol{f}_1^1, \ldots, \boldsymbol{f}_1^r)$ and $(\boldsymbol{z}^1, \boldsymbol{z}_b^1, \boldsymbol{z}_c^1, \ldots, \boldsymbol{z}^r, \boldsymbol{z}_b^r, \boldsymbol{z}_c^r)$. Thus, we need to sample $f_{j,i} \leftarrow D^d_{12\sqrt{kr}}$ $(i \neq 0)$ and $\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow D^{md}_{12\mathcal{B}\sqrt{3mdr}}$, and hence require $q > (10\phi_1 d\sqrt{dkr(\beta-1)} + 2)^2$ as in Assumption 4.14 below. Furthermore, since the extracted randomness norm will be larger, the relation $\mathcal{R}'_{1/\mathrm{N}}$ becomes $\mathcal{R}'_{1/\mathrm{N}}(24\sqrt{3r}\mathcal{B}md \cdot (k+1) \cdot d^k)$ and the commitment scheme is required to be binding in a larger domain. Therefore, the commitment scheme is set to be T_1-binding for

$$T_1 = (2d+2)\left(5^4\phi_1^4 d^3 k^3 \beta(\beta-1)r^2 + 12\phi_2^2\mathcal{B}^2 m^2 d^2 r\right)^{1/2}.$$

Note that these tweaks do not affect the soundness error of individual protocol runs as the extraction still works with $k+1$ accepting transcripts. Only the extracted witness norm is increased since the bound on $\|\boldsymbol{z}\|$ changes from $24\sqrt{3}\mathcal{B}md$ to $24\sqrt{3r}\mathcal{B}md$ in Protocol 4.2.

4.3.2 Construction

We now describe our lattice-based ring signature, which similarly builds on the one-out-of-many proof as in [GK15, BCC+15]. We summarise the assumptions on the parameters in Assumption 4.14, and let $CMT = (A, B, C, D, \{E_j\}_{j=0}^{k-1})$ and $RSP = (\{f_{j,i}\}_{j=0,i=1}^{k-1,\beta-1}, \boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c)$ be the corresponding values from Protocol 4.2.

Assumption 4.14. *Assume* $d \geq 7$, $md \geq 86$ *and* $q > (10\phi_1 d\sqrt{dkr(\beta-1)} + 2)^2$.

- **RSetup**(1^λ) : Run $\boldsymbol{G} \leftarrow \mathsf{CKeygen}(1^\lambda)$ and pick a hash function $H : \{0,1\}^* \rightarrow \mathcal{C}^r$ for $\mathcal{C} = \{X^\omega \ : \ \omega \in [0, 2d-1]\}$. Return $ck = \boldsymbol{G}$ and H as $pp = (ck, H)$.
- **RKeygen**(pp) : Run $\boldsymbol{r} \leftarrow \mathfrak{U}_{\mathcal{B}}^m$, $c = \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{r})$ and return $(pk, sk) = (c, \boldsymbol{r})$.
- **RSign**$_{pp,sk}(M, \boldsymbol{L})$: Parse $\boldsymbol{L} = (c_0, \ldots, c_{N-1})$ with $c_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; sk)$ where $\ell \in \{0, \ldots, N-1\}$. Continue as follows.
  1. Generate $(CMT_1, \ldots, CMT_r)$ by running $\mathcal{P}(ck, (c_0, \ldots, c_{N-1}), (\ell, sk))[1-7]$ r-times in parallel with the described modifications.
  2. Compute $\boldsymbol{x} = (x_1, \ldots, x_r) = H(ck, M, \boldsymbol{L}, (CMT_1, \ldots, CMT_r))$.
  3. Compute RSP_i by running $\mathcal{P}(x_i)[9-11]$ with CMT_i for all $i \in \{1, \ldots, r\}$.
  4. If $RSP_i \neq \perp$ for all $i \in \{1, \ldots, r\}$, return $\sigma = (\{CMT_i\}_{i=1}^r, \boldsymbol{x}, \{RSP_i\}_{i=1}^r)$.
  5. Otherwise go to Step 1.
- **RVerify**$_{pp}(M, \boldsymbol{L}, \sigma)$: Parse $\sigma = (\{CMT_i\}_{i=1}^r, \boldsymbol{x}, \{RSP_i\}_{i=1}^r)$, $\boldsymbol{x} = (x_1, \ldots, x_r)$ and $\boldsymbol{L} = (c_0, \ldots, c_{N-1})$. Proceed as follows.
  1. If $\boldsymbol{x} \neq H(ck, M, \boldsymbol{L}, (CMT_1, \ldots, CMT_r))$, return 0.
  2. For each $i \in \{1, \ldots, r\}$:
     (a) Run Protocol 4.2's verification with CMT_i, x_i and RSP_i except Step 2.
     (b) If verification fails, return 0.
  3. Return 1.

We can remove A, D, E_0 from the signature as they are uniquely determined by the remaining components, and Step 1 in **RVerify** ensures the relevant protocol verification steps hold. This is a standard technique and we skip the details.

4.3.3 Security proofs

The correctness and anonymity properties of the ring signature follow from the completeness and zero-knowledge properties of Protocol 4.2, respectively. In particular, the expected number of iterations in **RSign** is $\mu(\phi_1)\mu(\phi_2)$, which is upper-bounded by 3 in the parameter setting. However, the unforgeability proof of the ring signature is not straightforward due to the small challenge space and soundness gap issues. We prove the following.

Theorem 4.15. *If Assumption 4.14 holds and HMC in HNF defined in Section 3.2.2 is* T'*-binding where* $T' = \max\{T_1, \sqrt{\left(24\sqrt{3r} \cdot m\mathcal{B}(k+1)d^{k+1}\right)^2 + 2^{2k}}\}$ *for* T_1 *described with the tweaks, then the ring signature scheme described above is unforgeable with respect to insider corruption in the random oracle model.*

Proof. We prove the unforgeability by showing if there exists a PPT forger with a polynomial running time and a non-negligible success probability, then one can break the binding property of the commitment scheme for message and randomness of maximum Euclidean norms 2^k and $24\sqrt{3r} \cdot m\mathcal{B}(k+1)d^{k+1}$, respectively. This implies that one can find a solution to Module-SIS$_{n,m+k\beta,q,\beta_{\mathrm{SIS}}}$ problem for $\beta_{\mathrm{SIS}} =$

$2\sqrt{\left(24\sqrt{3r}\cdot m\mathcal{B}(k+1)d^{k+1}\right)^2+2^{2k}}$ by Lemma 3.3. For simplicity, we stick to the notation k and write $(k+1)$-special soundness instead of defining $k'=\max\{2,k\}$.

Let $\mathcal{C}^r$ be the range of H (i.e., each output component of H is in $\mathcal{C}$), Ψ be the set of all random tapes that could be used by a PPT adversary $\mathcal{A}$, and Φ be the set of all random tapes defining the random oracle H. Let $\boldsymbol{x}_j=(\boldsymbol{x}_{j,1},\ldots,\boldsymbol{x}_{j,r})$ be the output of j-th random oracle query. We partition Φ into Φ_{j_-}, $\boldsymbol{x}_j$ and Φ_{j_+} so that Φ_{j_-},Φ_{j_+} represent the sets of random tapes defining the random oracle outputs up to j-th query (i.e., $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{j-1}$) and after j-th query (i.e., $\boldsymbol{x}_{j+1},\ldots,\boldsymbol{x}_Q$), respectively. Therefore, the tuple $(\phi_{j_-},\boldsymbol{x}_j,\phi_{j_+})$ defines all the random oracle outputs. Further, assume that $\mathcal{A}$ makes q_P,q_S,q_H queries to PKGen, Sign and the random oracle, respectively. Hence, $\mathcal{A}$ makes at most $Q=q_S+q_H$ random oracle queries in total. Suppose that $\mathcal{A}$ has running time $T_A=\text{poly}(\lambda)$ and a probability $\varepsilon=1/\text{poly}(\lambda)>4Q\eta$ of generating a successful forgery where $\eta=(k/|\mathcal{C}|)^r$.

We construct an adversary $\mathcal{D}$ against the binding property of the commitment scheme with a running time $T_B=\text{poly}(\lambda)$ and non-negligible success probability $\varepsilon_B=1/\text{poly}(\lambda)$. On input a commitment key ck, $\mathcal{D}$ works as follows.

1. Pick $t\leftarrow\{1,\ldots,q_P\}$.
2. Set $pk_t=\text{Com}_{ck}(\mathbf{1};\boldsymbol{r}_t)$ for some randomness $\boldsymbol{r}_t\in\{-\mathcal{B},\ldots,\mathcal{B}\}^{md}$ where $\mathbf{1}=(1,0,\ldots,0)\in\text{BIN}\subseteq\{0,1\}^*$ (observe that $\|\boldsymbol{r}_t\|\leq\mathcal{B}\sqrt{md}$).
3. Pick $j\leftarrow\{1,\ldots,Q\}$.
4. Pick $\psi\leftarrow\Psi$.
5. Pick $(\phi_{j_-},\boldsymbol{x}_j,\phi_{j_+})\leftarrow\Phi_{j_-}\times\mathcal{C}\times\Phi_{j_+}$
6. Run 0: run $\mathcal{A}(\psi,\phi_{j_-},\boldsymbol{x}_j,\phi_{j_+})$ with access to the oracles PKGen, Sign, Corrupt and the random oracle $H(\phi_{j_-},\boldsymbol{x}_j,\phi_{j_+})$ simulated as follows. Whenever $\mathcal{A}$ queries PKGen, $\mathcal{D}$ answers as in the real case except for t-th query where pk_t is returned. If $\mathcal{A}$ ever queries Corrupt(t), $\mathcal{D}$ aborts (abort Type I). If $\mathcal{A}$ queries Sign$(t,M,\boldsymbol{L})$, it picks a random challenge vector $\boldsymbol{x}$ and uses SHVZK simulator of Protocol 4.2 to simulate the proof $(\{CMT_i\}_{i=1}^r,\{RSP_i\}_{i=1}^r)$ (note that only the simulation of non-aborted protocols is used here). Then, the random oracle is programmed as $H(ck,\mathcal{M},\boldsymbol{L},\{CMT_i\}_{i=1}^r)=\boldsymbol{x}$, except if $(ck,\mathcal{M},\boldsymbol{L},\{CMT_i\}_{i=1}^r)$ has been queried before (abort Type II).
   (a) If $\mathcal{A}$ outputs a forgery σ^0 using j-th random oracle query output $\boldsymbol{x}_j^0$, fix ψ and ϕ_{j_-}.
   (b) Otherwise, abort.
7. Pick $\phi'_1,\ldots,\phi'_{\mathcal{N}}\leftarrow\Phi_{j_+}$
8. Run i (for $i\in\{1,\ldots,\mathcal{N}\}$ and $\mathcal{N}$ defined below in the analysis): run $\mathcal{A}(\psi,\phi_{j_-},\boldsymbol{x}_j^i,\phi'_i)$ with access to oracles PKGen, Sign, Corrupt and the random oracle $H(\phi_{j_-},\boldsymbol{x}_j^i,\phi'_i)$ where $\boldsymbol{x}_j^i$ is the response of the j-th random oracle query at iteration i.
   (a) $\mathcal{A}$ outputs a forgery σ^i. We say that Run i is j-successful if σ^i was forged with respect to $\boldsymbol{x}_j^i$.
9. If there exists $i^*\in[1,r]$ and $S^*\subseteq\{0,\ldots,\mathcal{N}\}$ with $|S^*|=k+1$ such that $\mathcal{G}^*:=\{x^u_{j,i^*}\ :\ u\in S^*\}$ contains $k+1$ distinct challenges and σ^u is j-successful for all $u\in S^*$, then run $(k+1)$-special soundness extractor $\mathcal{E}$ of Protocol 4.2 on input $\{\sigma^u\}_{u\in S^*}$ to extract an opening of $2^kpk_{t'}$ to $(\mathbf{0},\boldsymbol{s}_{t'})$ for some $1\leq t'\leq q_P$ where $\|\boldsymbol{s}_{t'}\|\leq 24\sqrt{3r}\cdot m\mathcal{B}\cdot(k+1)\cdot d^{k+1}$.
10. If $t=t'$, return $((2^k\cdot\mathbf{1},2^k\cdot\boldsymbol{r}_t),(\mathbf{0},\boldsymbol{s}_{t'}))$ as a binding collision pair for the commitment scheme. Note that multiplication of $(\mathbf{1},\boldsymbol{r}_t)$ by 2^k gives a valid opening of 2^kpk_t, because $d^{k+1}>2^k$ since $d\geq 7$.

11. Otherwise, abort.

Note that when $\mathcal{D}$ returns a binding collision, there cannot be Type I aborts as the forged signature must be for a ring comprised only of uncorrupted users.

Now, let us analyse this procedure in more details and denote $\varepsilon_{\text{LWE}} = O(2^{-\lambda})$ as the advantage of solving M-LWE problem. First, we observe that in each run of $\mathcal{A}$, the view of $\mathcal{A}$ is simulated by $\mathcal{D}$ with the same distribution as in the real attack except for:

- pk_t is a commitment to $\mathbf{1}$ in the simulation by $\mathcal{D}$ whereas it is a commitment to $\mathbf{0}$ in the real attack. By the hiding property of the commitment scheme, this reduces the success probability of $\mathcal{A}$ by at most ε_{LWE}.
- There is a statistical distance of at most $O(q_S \cdot 2^{-\lambda})$ between the distribution of signing oracle simulator and that of the real signing oracle.
- A Type II abort occurs during a signing oracle query with probability at most $Q \cdot 2^{-\lambda}$.

By the simulation statistical distance argument above, each run of $\mathcal{A}$ with pk_t and signing oracle simulated by $\mathcal{D}$ succeeds with probability $\tilde{\varepsilon} \geq \varepsilon - O(Q \cdot 2^{-\lambda})$. We say that $(\psi, \phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+}, j)$ is 'winning' if $\mathcal{A}(\psi, \phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+})$ outputs a valid forgery using $\boldsymbol{x}_j$ after Q random oracle queries. Note that there exists a $j^* \in \{1, \ldots, Q\}$ such that $\Pr[\ (\psi, \phi_{j^*_-}, \boldsymbol{x}_{j^*}, \phi_{j^*_+}, j^*)$ winning $\ ] \geq \tilde{\varepsilon}/Q$. By the Splitting Lemma (Lemma 7 of [PS00]), there exists a subset $S \subseteq \Psi \times \Phi_{j^*_-}$ such that

$$\Pr_{\psi \in \Psi, \phi_{j^*_-} \in \Phi_{j^*_-}}[(\psi, \phi_{j^*_-}) \in S] \geq \tilde{\varepsilon}/(2Q), \text{ and}$$

$$\varepsilon' := \Pr_{\boldsymbol{x}_{j^*} \in \mathcal{C}, \phi_{j^*_+} \in \Phi_{j^*_+}}[(\psi, \phi_{j^*_-}, \boldsymbol{x}_{j^*}, \phi_{j^*_+}, j^*) \text{ winning }] \geq \tilde{\varepsilon}/(2Q) \qquad \forall (\psi, \phi_{j^*_-}) \in S.$$

Now, for $(\psi, \phi_{j^*_-}) \in S$, $c \in \mathcal{C}$ and $1 \leq i \leq r$, define $p_i(c)$ as the probability with respect to $\boldsymbol{x}_{j^*} \in \mathcal{C}$ and $\phi_{j^*_+} \in \Phi_{j^*_+}$ that $(\psi, \phi_{j^*_-}, \boldsymbol{x}_{j^*}, \phi_{j^*_+}, j^*)$ is winning and $\boldsymbol{x}_{j^*} = (x_{j^*,1}, \ldots, x_{j^*,r})$ with $x_{j^*,i} = c$.

Claim 4.16. *If $\varepsilon' > (k/|\mathcal{C}|)^r$, then there exists an $i^* \in [1, r]$ and $\mathcal{G} \subseteq \mathcal{C}$ with $|\mathcal{G}| = k+1$ such that*

$$p_{i^*}(c) \geq \frac{\varepsilon' - (k/|\mathcal{C}|)^r}{(|\mathcal{C}| - k) \cdot r} =: p \qquad \forall c \in \mathcal{G}.$$

If the claim holds, then a sample of $\mathcal{N} := (k+1) \cdot p^{-1}$ independent and identically distributed winning tuples $(\psi, \phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+}, j)$ will yield a set $\left\{\boldsymbol{x}_j^1, \ldots, \boldsymbol{x}_j^{k+1}\right\}$ such that $\mathcal{G} = \{x_{j,i^*}^1, \ldots, x_{j,i^*}^{k+1}\}$ with a probability at least $1 - (k+1)e^{-(k+1)}$, which is greater than $7/10$ for $k \geq 1$ (this comes from the fact that the probability that $\mathcal{N}$ samples do not contain c for some $c \in \mathcal{G}$ is at most $(k+1) \cdot (1-p)^{\mathcal{N}}$). That is, after $\mathcal{N}/\varepsilon'$ rewindings, we obtain a set of $(k+1)$ distinct challenge values of Protocol 4.2 with respect to the same initial commitment with a high probability.

Now, $\mathcal{N} = \text{poly}(\lambda)$ if $k, |\mathcal{C}|, r = \text{poly}(\lambda)$ and $(\varepsilon' - (k/|\mathcal{C}|)^r)^{-1} \leq \text{poly}(\lambda)$. It is easy to see that the first requirement holds since $|\mathcal{C}| = 2d$, $r = \frac{\lambda}{\log(2d) - \log k}$ and $k \leq \log N$. For the second requirement, we have

$$\left(\varepsilon' - (k/|\mathcal{C}|)^r\right)^{-1} = (\varepsilon' - \eta)^{-1} \leq (\varepsilon' - \varepsilon'/2)^{-1} = 2/\varepsilon' \leq \text{poly}(\lambda),$$

where the first inequality holds since $\varepsilon' > 2\eta$. Now, by $(k+1)$-special soundness of Protocol 4.2, we can use the set $\mathcal{G}$ to extract an opening of $2^k pk_{t'}$ to $(\mathbf{0}, \boldsymbol{s}_{t'})$ for some $t' \in \{1, \ldots, q_P\}$. By the hiding property of the commitment scheme, $t' = t$ with

probability at least $\frac{1}{q_P} - \varepsilon_{\mathrm{LWE}}$. Also, $j = j^*$ with probability $\frac{1}{Q}$. Hence, $\mathcal{D}$ succeeds to output a binding collision pair with probability

$$\Pr[j = j^*] \cdot \Pr[(\psi, \phi_{j_-}) \in S] \cdot \Pr\left[\begin{array}{l}\mathcal{N} \text{ runs contain } k+1 \\ j\text{-successful distinct challenges}\end{array}\right] \cdot \Pr[t = t'] \\ \geq \frac{1}{Q} \cdot \frac{\tilde{\varepsilon}}{2Q} \cdot \frac{7}{10} \cdot \left(\frac{1}{q_P} - \varepsilon_{\mathrm{LWE}}\right) = \frac{1}{\mathrm{poly}(\lambda)}.$$

This leaves us with the proof of the claim, which is based on a pigeonhole argument. For each $i \in [1, r]$, let M_i with $|M_i| = k$ be the set of $c \in \mathcal{C}$ such that $p_i(c') \leq p_i(c)$ for all $c' \notin M_i$ and all $c \in M_i$. Further, let B be the set of $(\boldsymbol{x}_j, \phi_{j_+}) \in \mathcal{C}^r \times \Phi_{j_+}$ for $\boldsymbol{x}_j = (x_{j,1}, \ldots, x_{j,r})$ such that $x_{j,i} \in M_i$ for all $i \in [1, r]$. Since $|M_i| = k$,

$$\Pr[(\boldsymbol{x}_j, \phi_{j_+}) \in B] \leq \Pr[x_{j,i} \in M_i \quad \forall i \in [1, r]] \leq (k/|\mathcal{C}|)^r .$$

For each $(\boldsymbol{x}_j, \phi_{j_+}) \in S \setminus B$, there exists $i \in [1, r]$ and $c \in \mathcal{C} \setminus M_i$ such that $x_{j,i} = c$. This implies that

$$\sum_{i=1}^{r} \sum_{c \in \mathcal{C} \setminus M_i} p_i(c) \geq \Pr[(\boldsymbol{x}_j, \phi_{j_+}) \in S \setminus B] \geq \Pr[(\boldsymbol{x}_j, \phi_{j_+}) \in S] - \Pr[(\boldsymbol{x}_j, \phi_{j_+}) \in B] \\ \geq \varepsilon' - (k/|\mathcal{C}|)^r .$$

From here, we can deduce that there exists $i^* \in [1, r]$ and $c^* \in \mathcal{C} \setminus M_i$ such that $p_{i^*}(c^*) \geq \frac{\varepsilon' - (k/|\mathcal{C}|)^r}{(|\mathcal{C}| - k) \cdot r}$. Hence, for all $c \in \mathcal{G} := M_{i^*} \cup \{c^*\}$, $p_{i^*}(c) \geq \frac{\varepsilon' - (k/|\mathcal{C}|)^r}{(|\mathcal{C}| - k) \cdot r}$, proving the claim. □

4.3.4 Parameter setting

First of all, we set $\phi_1 = \phi_2 = 22$ to get an acceptance rate of more than 1/3 for the two-step rejection sampling. Such an acceptance rate is greater than or equal to the most commonly used ones such as those in [dPLNS17, Lyu12, BLO18, BDL+18] and the expected number of iterations in **RSign** is 3 in this case. Also, we need to ensure that the commitment scheme T'-binding as in Theorem 4.15. Thus, from the discussion in Section 3.2.2, to make M-SIS secure against known lattice attacks, we ensure the following holds

$$\min\left\{q,\ 2^{2\sqrt{n \cdot d \cdot \log q \log \delta}}\right\} > \max\left\{2T_1,\ 2 \cdot 24\sqrt{3r}\mathcal{B}md \cdot (k+1) \cdot \mathbb{B}_{d,k}\right\}. \tag{4.18}$$

That is, we use Method 2 to bound the extracted witness norm, which does not require the use of Algorithm 4.1 in the protocol's verification. For the set of (d, k) pairs used in Table 4.3, the exact value of $\mathbb{B}_{d,k}$ is computed by iterating through the whole search space.

We also set $\mathcal{B} = 1$ as in previous works [BDL+18, LN17, dPLS18], and make sure that M-LWE$_{m-n,n,q,1}$ is hard using Albrecht et al.'s estimator [APS15]. The root Hermite factor δ is at most 1.0045 for both M-SIS and M-LWE security estimations. Finally, Assumption 4.14 is ensured to hold.

Table 4.3 shows several instances with respect to different ring sizes where the soundness error of the underlying (r-repeated) protocol is $2^{-\lambda}$ and we restrict $\log q \leq 64$. The calculations are done as given in Table 4.4 further below. Note that since r is rounded up, the security parameter λ may be slightly larger than 128. Also, the results from Lemma 3.15 used to bound the Euclidean norm of a discrete normal vector can

TABLE 4.3: Parameters and sizes of our lattice-based ring signature for a root Hermite factor $\delta \leq 1.0045$. The total challenge space size is around 2^{λ}. The signature sizes are rounded to the nearest integer.

N	64	256	1024	4096	$\sim 2^{16}$	$\sim 2^{20}$	2^{30}
(n, m)	$(5, 13)$	$(5, 13)$	$(11, 25)$	$(21, 50)$	$(20, 51)$	$(40, 101)$	$(41, 106)$
$(d, \log q)$	$(256, 50)$	$(256, 53)$	$(128, 46)$	$(64, 47)$	$(64, 50)$	$(32, 49)$	$(32, 52)$
(k, β)	$(2, 8)$	$(2, 16)$	$(2, 32)$	$(2, 64)$	$(3, 41)$	$(3, 102)$	$(5, 64)$
r	16	16	19	22	24	29	35
λ	128.0	128.0	133.0	132.0	129.96	128.04	128.73
Signature Size (KB)	774	881	1021	1178	1487	1862	3006
User PK Size (KB)	7.81	8.28	7.91	7.71	7.81	7.66	8.33
User SK Size (KB)	0.81	0.81	0.78	0.78	0.80	0.79	0.83

be adjusted with respect to the vector dimension. In particular, the constant 5^4 in T_1 (and also T) can be reduced depending on the choice of d. We take this optimisation into consideration when setting the parameters.

Remark 4.17. *When considering the r-repeated protocol, the bound in (4.14) becomes $(2d+2)\left(5^4\phi_1^4 d^3 k^3 \beta(\beta-1)r^2 + 12\phi_2^2 \mathcal{B}^2 m^2 d^2 r\right)^{1/2}$, and this bound is used when setting the parameters for the ring signature.*

TABLE 4.4: Calculation of parameters and sizes for the ring signature.

	Notation/Formula	**Notes**
Security parameter	λ	
Dimension of randomness vector	m	Chosen based on LWE estimator of [APS15]
Soundness error	$\eta = \frac{\max\{2, \log_\beta N\}}{2d}$	Recall that $k = \log_\beta N$
Number of protocol repetitions	$r = \lceil -\frac{\lambda}{\log \eta} \rceil$	
Num. of commitments	$N_c = k + 1$	$B, C, E_1, \ldots, E_{k-1}$
Number of $f_{j,i}$ values	$N_f = k \cdot (\beta - 1)$	$f_{0,1}, \ldots, f_{k-1,\beta-1} \in D_{\phi_1\sqrt{kr}}$
Num. of randomness	$N_R = 3$	$\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c \in D^{md}_{\phi_2\mathcal{B}\sqrt{3mdr}}$
Ring Signature size	$r \cdot [N_c \cdot (nd \log q)$ $+N_f \cdot d \cdot \log(12\phi_1\sqrt{kr})$ $+N_R \cdot (md \log(12\phi_2\mathcal{B}\sqrt{3mdr}))]$	
User public key size	$nd \log q$	A commitment in $R_q^{n\times 1}$
User secret key size	$md \cdot \log(2\mathcal{B}+1)$	A randomness vector in $\{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$. For $\mathcal{B} = 1$, we take $\log(2\mathcal{B}+1) = 2$.

4.4 Discussion

The investigation of many-special sound protocols in this chapter helps us to understand the challenges one needs to address in constructing lattice-based ZKPs for non-linear polynomial relations. For example, one of the things that we need to be

careful about is that we do not always have a *field* structure and therefore non-zero elements (in particular, non-zero challenge differences) may not always be invertible. Even more importantly, the challenge differences do not necessarily have a *short* inverse, restricting us to the set of monomials as the challenge space. As a result, the challenge space size is not exponentially large.

Another important aspect of the chapter is that it provides us with new techniques in handling relaxations in the context of multi-shot proofs, in particular, for the relations of interest in this work. For example, the unforgeability proof of the ring signature (Theorem 4.15) introduces useful tools when the challenge space size is small and the underlying ZKP of the signature is not exact.

It is not too hard to see the limitation in this chapter: the protocols require repetitions for soundness amplification. That is, the new technical tools do not work with an exponentially large challenge space (i.e., a challenge space of size about 2^{λ}). This is precisely one of the questions addressed in the next chapter, where new tools for the design and analysis of *one-shot* algebraic proofs are introduced.

On the other hand, the advantage of the ZKPs introduced in this chapter is that they have a *fixed publicly known* relaxation factor. For example, the relaxation factor is always 2 for the binary proof. That is, for any commitment B with a valid binary proof, anyone can conclude that $2B$ is well-formed. However, as we will see in the next chapter, the relaxation factor may also be a part of the witness. That is, the verifier may only be ensured of the fact that there exists a relaxation factor y such that yB is well-formed, but the exact value of y is unknown to the entities apart from the prover. The advantage of having such a fixed publicly known relaxation factor may prove useful particularly when extending Bulletproofs-like [BBB+18] recursive ZKPs to the lattice setting. Otherwise, when the relaxation factors are different in each recursion, witness extraction may become more complicated.

Chapter 5

One-Shot Algebraic Proofs and Applications

Having seen the challenges in constructing ZKPs for non-linear relations in the previous chapter, we set a more practice-oriented goal in this chapter by studying *one-shot* algebraic proofs for non-linear relations. Therefore, we will work with challenge spaces of size at least 2^λ (or more precisely, $2^{2\lambda}$ for λ-bit post-quantum security). It is unknown at the moment whether such a large suitable challenge space exists with the property that any pairwise challenge difference has short inverse. As a result, instead of enforcing challenge difference to have short inverses, we will aim to avoid any challenge difference inverse term in an extracted witness.[1]

Another goal of the chapter is to introduce new tools to overcome the $\widetilde{O}(\lambda^2)$ growth of the proof length. To that end, we first start the chapter with a discussion about the asymptotic costs of the existing lattice-based ZKPs in Section 5.1. Our new techniques, as summarised in Section 5.2, are aimed at constructing shorter and faster proofs both asymptotically and in practice. We first go into the details of our new one-shot proof techniques for non-linear polynomial relations and new tools for compact proofs in Section 5.3. The new techniques for *faster* lattice-based proofs are discussed in Section 5.4, where the CRT-packing technique supporting interslot operations and "NTT-friendly" tools are introduced. The former technique is accompanied by an application to an efficient (relaxed) range proof with concrete parameter settings. After that, we turn our attention in Section 5.5 to the applications of the new techniques to other useful relations such as a binary proof, one-out-of-many proof and set membership proof. Later in Section 5.6, we consider higher level applications of the new ZKPs. In particular, we introduce an efficient ring signature and a privacy-preserving credentials scheme based on standard lattice assumptions.

5.1 Asymptotic Costs of Existing Lattice-Based ZKP Techniques

First, let us assume that one relies on computational hardness assumptions, particularly, Module-SIS (M-SIS) and Module-LWE (M-LWE) for the security of a commitment scheme and let $d_{\mathrm{SIS}}, d_{\mathrm{LWE}}$ be the dimension parameters required for M-SIS and M-LWE security, respectively. Based on the state of the art in lattice cryptanalysis, it is known that one needs $d_{\mathrm{SIS}} = O(\lambda \frac{\log^2 \beta_{\mathrm{SIS}}}{\log q})$ for λ-bit security based on M-SIS where β_{SIS} is the norm of a valid M-SIS solution (see Section 5.6.1 for more discussion). Letting $\beta_{\mathrm{SIS}} = q^\varepsilon$ for $0 < \varepsilon \leq 1$, we get $\log \beta_{\mathrm{SIS}} = \varepsilon \log q$ and, for a balanced security,

$$d_{\mathrm{LWE}} \approx d_{\mathrm{SIS}} = O(\lambda \varepsilon^2 \log q). \tag{5.1}$$

[1] This chapter is mainly based on [ESLL19].

In lattice-based cryptography, the most commonly used commitment schemes for algebraic proofs are Unbounded-Message Commitment (UMC) and Hashed-Message Commitment (HMC). These commitment schemes have different tradeoffs as discussed in Section 3.2.2. Let n, m, d, v be the module rank for M-SIS, the randomness vector dimension in a commitment, the polynomial ring dimension and the message vector dimension in a commitment, respectively. The commitment vector is of dimension $n + v$ for UMC and n for HMC, which means the space costs of a commitment are $(n + v)d \log q$ and $nd \log q$ for UMC and HMC, respectively. Letting κ be the number of protocol repetitions, we get the formulae for space costs in Table 5.1.

The commitment matrix dimensions are $(n+v) \times m$ for UMC and $n \times (m+v)$ for HMC, and both of the commitments are computed as a matrix-vector multiplication.[2] Therefore, we also get the formulae for the time costs as given in Table 5.1 assuming a degree-d polynomial multiplication can be performed in time $\widetilde{O}(d)$ (more precisely, $O(d \log d)$) using, e.g., FFT-like methods.

Further, we have $d_{\mathrm{LWE}} = (m - n - v)d$ and thus $md > d_{\mathrm{LWE}}$ for UMC, and $d_{\mathrm{SIS}} = nd$ for both HMC and UMC. As a result, using (5.1), we get

$$md = O(\lambda \varepsilon^2 \log q) \text{ for UMC, and } \quad nd = O(\lambda \varepsilon^2 \log q) \text{ for UMC/HMC.} \tag{5.2}$$

Now, suppose that we want to prove a relation that involves commitment to $k = O(\log q)$ messages (for example, to prove knowledge of $m_1, \ldots m_k$ such that $\sum_{i=1}^{k} \alpha_i m_i = 0$ for public values $\alpha_1, \ldots, \alpha_k$). Clearly, if we commit to these messages independently, then the overall cost of both time and space increase by a factor of k. Alternatively, we can pack multiple messages in a commitment by setting $v = k$ and hope that this gives a better performance. If an existing multi-shot technique such as Stern-based proofs, or those using binary or monomial challenges, is used, the number of protocol repetitions κ will be $\widetilde{O}(\lambda)$, and thus we get the asymptotic costs in the "multi-shot" column of Table 5.1 (using (5.2)). On the other hand, if one can make the proof one-shot, then we get the complexities in the "one-shot" column of Table 5.1, where there is a clear saving of $\widetilde{O}(\lambda)$.

5.2 Overview of New Techniques

Before going into low-level technical details, we also present an overview of the main techniques introduced in this chapter.

5.2.1 One-shot witness extraction for non-linear polynomial relations

The main challenge in designing *efficient* lattice-based ZKPs is that the extracted witness is required to be *short* as mandated by computational lattice problems (in particular, *Short* Integer Solution – SIS problem). Traditional witness extraction techniques involve the inverse of challenge differences as a multiplicative factor in extracted witnesses, and such an approach is problematic in lattice-based protocols as these inverse terms need not be short in general. This causes one either to resort to more inefficient techniques such as aforementioned multi-shot proofs or to introduce relaxations in the proofs. Our solution in this chapter falls into the latter.

The target problem reduces to the question of extracting useful information from a system of equations of the form $\boldsymbol{V} \cdot \boldsymbol{c} = \boldsymbol{b}$ where $\boldsymbol{V}$ is a matrix (a Vandermonde

[2] Here, we overlook the fact that some parts of the commitment matrix are zero or identity, but this does not change the asymptotic behaviour in Table 5.1.

matrix in our case) constructed by challenges, $\boldsymbol{c}$ is a vector of commitments with unknown openings and $\boldsymbol{b}$ is a vector of commitments with known openings. Our idea is to introduce the use of *adjugate* matrices instead of inverse matrices in the "complex" witness extraction of lattice-based ZKPs. This technique, in one hand, enables us to extract *useful* information about the openings of the commitments in $\boldsymbol{c}$ without the involvement of inverse terms, and on the other hand, is the main cause of relaxations. Here, it is crucial that the relaxed proof proves a *useful* relation, is *sound*, and also *efficient*. These piece together nicely when the use of adjugate matrices is accompanied by a good choice of challenge space, and we provide an analysis of our technique with a family of commonly used challenge spaces. It is worth emphasising that straightforward soundness proofs do not work, and one needs special tools such as those introduced in this chapter to overcome the complications. Our one-shot proof approach is detailed in Section 5.3.

5.2.2 CRT-packing supporting inter-slot operations

Let $R = \mathbb{Z}[X]/(X^d + 1)$ and $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ for a usual choice of power-of-two d. It is known that $X^d + 1$ factors linearly (and thus R_q fully splits) for certain choices of q (e.g., a prime $q \equiv 1 \bmod 2d$) and, in that case, one can use NTT for polynomial multiplication in R_q in time $O(d \log d)$. Assume that we choose such an "NTT-friendly" q. For $1 \leq s \leq d$ where s is a power of two, let $R_q^{(0)}, \ldots, R_q^{(s-1)}$ be the polynomial rings of dimension d/s such that $R_q \cong R_q^{(0)} \times \cdots \times R_q^{(s-1)}$ and $R_q^{(i)} = \mathbb{Z}_q[X]/(P^{(i)}(X))$ for some polynomial $P^{(i)}(X)$ of degree d/s for all $0 \leq i < s$ (which is obtained by the Chinese Remainder Theorem – CRT). We use these CRT "slots" to store s messages in a single ring element. Thus, if we have k messages in total, we can set the message vector dimension in a commitment as $v = k/s$ (instead of $v = k$ in previous approaches).

This initial part of the CRT-packing idea seems easy, and indeed a possible application of CRT in lattice-based ZKPs is mentioned in [LS18] to perform parallel proofs, where there is no interaction between the messages in different slots. We are, on the other hand, interested in applications such as range proofs requiring "inter-slot" operations between messages in separate CRT slots, and get a *complete* set of operations (see [GHS12] for a discussion in the context of FHE).

First thing to note about the CRT-packing technique is that even if the messages to be stored in CRT slots are short, the resulting element in R_q representing s messages need not be so. This makes the technique inapplicable to HMC, which require short message inputs (at least in the general case). More importantly, there are two crucial hurdles we need to overcome: 1) it is not clear how to enable inter-slot operations and make the ZKP work in this setting, and 2) we need to make the proof one-shot in order not to lose the factor λ gained.

Let us write $m = \langle m_0, \ldots, m_{s-1} \rangle$ where $m \in R_q$ and $m_i \in R_q^{(i)}$ for $0 \leq i < s$ if m maps to $(m_0, \ldots, m_{s-1})$ under the CRT-mapping. In general, to prove knowledge of a message b, the prover in the protocol needs to send some "encoding" of the message as $f = \mathrm{Enc}_x(b) := x \cdot b + \rho$ where x is a challenge and ρ is a random masking value. Clearly, we do not want to send k encodings in R_q as it does not result in any savings. Instead, our idea is to send k/s elements in R_q, each encoding s messages, *in a way* that enables the verifier to "extract" all k messages out of them. When the prover sends $f = x \cdot m + \rho$ (there may be multiple such f's), for each $0 \leq i < s$, the verifier can compute $f_i = f \bmod (q, P^{(i)}(X)) = x_i \cdot m_i + \rho_i$ as the extracted encodings where $x = \langle x_0, \ldots, x_{s-1} \rangle$ and $\rho = \langle \rho_0, \ldots, \rho_{s-1} \rangle$. The main problem now is

TABLE 5.1: The (minimal) asymptotic time and space complexities of lattice-based protocols involving commitment to $k = O(\log q)$ messages. β_{SIS}: M-SIS solution norm, q: modulus, κ: the number of protocol repetitions, n: module rank for M-SIS, v: message vector dimension in a commitment, d: polynomial ring dimension, m: randomness vector dimension in a commitment. Assume: $\log q < \log^2 \beta_{\mathrm{SIS}}/2$ and degree-d polynomial multiplication costs $\widetilde{O}(d)$. To optimise both costs, one would set $n = v$ in all cases.

	Formula	**Multi-shot** $\kappa = \widetilde{O}(\lambda), v = k$	**One-shot** $\kappa = 1, v = k$	**One-shot + CRT** $\kappa = 1, v = O(1)$
Space UMC	$\kappa(n+v)d\log q$	$\widetilde{O}(\lambda^2 \log^2 \beta_{\mathrm{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\mathrm{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\mathrm{SIS}})$
Time UMC	$\kappa(n+v)md$	$\widetilde{O}(\lambda^2 \log^2 \beta_{\mathrm{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\mathrm{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\mathrm{SIS}}/\log q)$
Space HMC	$\kappa nd\log q$	$\widetilde{O}(\lambda^2 \log^2 \beta_{\mathrm{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\mathrm{SIS}})$	N/A
Time HMC	$\kappa n(m+v)d$	$\widetilde{O}(\lambda^2 \log^2 \beta_{\mathrm{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\mathrm{SIS}})$	N/A

that f_i's are encodings of m_i's, but under possibly *different* x_i's, which circumvents interoperability of distinct f_i's. For example, the sum $f_i + f_j$ for $i \neq j$ does not result in an encoding of the sum of messages under a common challenge x if $x_i \neq x_j$.

To overcome this problem, our idea is to choose the challenge $x = \langle x, \ldots, x \rangle$ for $x \in \bigcap_{i=0}^{s-1} R_q^{(i)}$ such that all extracted encodings are under the same challenge x. This means x must be of degree smaller than d/s and thus the challenge space size is possibly greatly decreased.[3] To make the proof one-shot, we choose the challenges to be polynomials of degree at most $d/s - 1$ with coefficients in $\mathbb{Z}_p$ such that $p^{d/s} = 2^{2\lambda}$ (i.e., there are $2^{2\lambda}$ challenges in total).[4] Therefore, we need $d/s \cdot \log p = 2\lambda$, which is satisfied by choosing $d/s = \lambda\varepsilon^2$ and $\log p = 2/\varepsilon^2$. We should also ensure $\log q > \log p = 2/\varepsilon^2 = 2\log^2 q/\log^2 \beta_{\mathrm{SIS}}$. This holds assuming $\log q < \log^2 \beta_{\mathrm{SIS}}/2$, which is easily satisfied in most of the practical applications.

To have fast computation, we also set $d = d_{\mathrm{SIS}} = O(\lambda\varepsilon^2 \log q)$, and hence get $s = O(\log q)$. Recall that we have k messages in total and s slots in a single ring element. As a result, for $k = O(\log q)$, it is enough to have $v = k/s = O(1)$. Overall, we end up with the asymptotic costs in the last column of Table 5.1, where our technique has a factor $\log q$ saving in asymptotic computational time in comparison to previous approaches *without* any compromise in communication.

An attractive example in practice where one would need a commitment to $k = O(\log q)$ messages is a range proof on $[0, 2^k - 1]$. Let us take a range proof on $\ell \in [0, 2^{64} - 1]$ as a running example. In this case, our proof proceeds as follows. We allow R_q to split into at least 64 factors, and thus use a *single* R_q element to commit to all the bits of ℓ (so committing to all the bits of ℓ only cost a single commitment with message vector dimension $v = 1$). In its initial move, the prover sends some commitments and gets a challenge from the verifier. Then, the prover responds with a *single* encoding in R_q (or 64 small encodings that costs as much as a single element in R_q). From here, the verifier extracts the encodings of all the bits, reconstructs the masked integer value ℓ and checks whether it matches the input commitment to ℓ. In this setting, it is clear that we require operability between different slots, and thus we set the encodings of all the bits to be under the same challenge x. For a

[3] We remark that earlier works [SSTX09, BKLP15] also considered choosing a challenge of degree d/s for some $s > 1$ for the purpose of invertibility of challenges. However, our motivation here is to make sure that x has the same element in all CRT slots.

[4] In this chapter, we consider a challenge space size of $2^{2\lambda}$ for λ-bit post-quantum security.

ring dimension $d = 512$, the infinity norm of a challenge can be as large as 2^{31}, which seems quite large.

An alternative to this approach is to use "norm-optimal" challenges from [LS18] (named "optimal" in [LS18]) such that the infinity norm of a challenge is set to 1, and thus the overall Euclidean norm of a challenge is minimised. In this case, one needs to set the ring dimension $d \geq 256$ to get a challenge space size of at least 2^{256}. However, this results in significantly longer proofs as shown in Table 5.2. The reason behind this phenomenon is that one needs to encode 64 values and with the "norm-optimal" challenges the cost of these encodings and the commitments grow too much. The use of challenges with larger (even much larger) norm does not seem to cause significant increase in the proof length, which can be explained as follows. To do a range proof on 64-bit range, the modulus q must be at least 2^{64}. Using UMC, where the message part does not affect the hardness of finding binding collisions (in particular, M-SIS hardness), such a large q already makes M-SIS very hard and M-LWE very easy. Therefore, having a challenge with a large norm only brings the hardness level of M-SIS to that of M-LWE, and results in a very compact proof.

We also add for comparison a hypothetical idealised range proof scheme optimised for proof length in Table 5.2, where for this scheme we only check two conditions: 1) $q \geq N$ and 2) M-SIS and M-LWE root Hermite factors are less than or equal to 1.0045. More specifically, we go over all the values of the ring dimension $d \in \{8, 16, \ldots, 1024\}$, $\log q \in \{\log N, \ldots, 100\}$ and initial noise distribution $\mathcal{U}(\{-\mathcal{B}, \ldots, \mathcal{B}\})$ for $\mathcal{B} \in \{1, 2, 3\}$, and set the remaining parameters so that the above security condition (2) is satisfied. Therefore, for the "ideal w/o CRT" scheme we do not check whether the soundness proof of the protocol works with the parameters set. Even with this advantage given, we see from Table 5.2 that our range proof, as expected, has approximately the same proof length as "ideal w/o CRT", and also achieves a significant speed-up as the ring dimension as well as the number of FFT levels supported is higher. One can see from [LS18, Table 2] that going from 2 levels of FFT to 6 levels of FFT alone results in a speedup of a factor more than 3.

When we allow the ring R_q to split into more than 64 factors, then the 64 subrings in which the message bits are encoded will not be fields and the structure of R_q is lost in these subrings. We are currently unable to make the soundness proof of the binary ZKP go through in these subrings, whose structure is unclear. On the other hand, we can make the binary proof work both in R_q using our new result (Lemma 5.5) and in any field. Thus, we allow R_q to split into exactly $\log N$ *fields* for a range proof of width N, which also gives the invertibility of challenges and challenge differences at no cost.

The reason why the scheme with "norm-optimal" challenges cannot split into more than $2^2 = 4$ factors is because the invertibility of polynomials with coefficients as large as 2^{16} is required when one relies solely on the results of [LS18].

5.2.3 "NTT-friendly" tools for fully-splitting rings

[LS18] studies in detail how cyclotomic rings split and the required invertibility conditions for short ring elements. A main motivation in [LS18] for the invertibility of short elements can be sketched as follows. In the hope of proving knowledge of a secret s (which is usually a message-randomness pair (m, r)) that satisfies a certain relation $g(s) = t$ for public homomorphic function g and public t, one-shot proofs can only convince the verifier of knowledge of $\bar{s}$ such that $g(\bar{s}) = \bar{x}t$, where $\bar{x} = x - x'$ for some (distinct) challenges x, x'. If g is a commitment scheme and one later opens t to a valid s' such that $g(s') = t$, then one can show that $s' = \bar{s}/\bar{x}$ using the binding

property of the commitment scheme provided that $\bar{x}$ is invertible. In our protocols, however, the relaxed relation proves knowledge of a secret *message* m such that

$$g'(\bar{x}m) = \bar{x}t'$$

where g' and t' are the parts dependent on the message (see Definitions 5.6 and 5.14). When one gets two relaxed openings $(\bar{x}_0, m_0)$ and $(\bar{x}_1, m_1)$, we have

$$\begin{matrix} g'(\bar{x}_0 m_0) = \bar{x}_0 t' \\ g'(\bar{x}_1 m_1) = \bar{x}_1 t' \end{matrix} \implies \begin{matrix} g'(\bar{x}_1 \bar{x}_0 m_0) = \bar{x}_1 \bar{x}_0 t' \\ g'(\bar{x}_0 \bar{x}_1 m_1) = \bar{x}_0 \bar{x}_1 t' \end{matrix} \implies \bar{x}_1 \bar{x}_0 m_0 = \bar{x}_0 \bar{x}_1 m_1, \quad (5.3)$$

due to the binding property of the commitment scheme. On contrary to the invertibility requirement, if the norm of each term is small relative to q, which is often the case, we use our new result Lemma 5.5 to show that,

$$\bar{x}_0 \bar{x}_1 (m_0 - m_1) = 0 \text{ in } \mathbb{Z}_q[X]/(X^d + 1) \implies m_0 = m_1. \quad (5.4)$$

That is, we can conclude the equality of two message openings even for non-invertible challenge differences. The lemma only requires q to be sufficiently large without putting any condition on its "shape", and thus enables the use of an "NTT-friendly" modulus q. Next, we initiate our detailed discussion on the new techniques.

5.3 One-Shot Proofs for Non-Linear Polynomial Relations

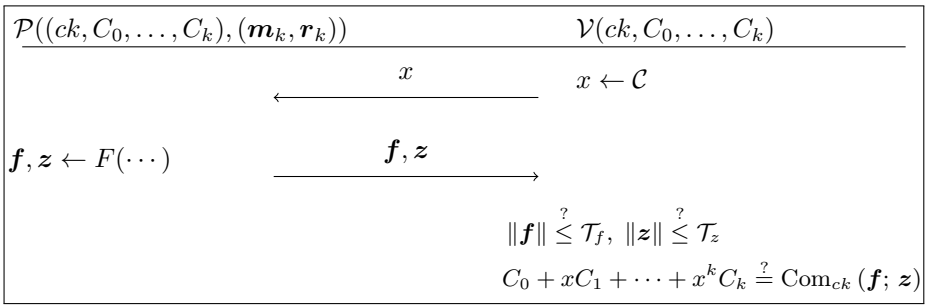
In this section, we focus on lattice-based zero-knowledge proofs in a general framework using homomorphic commitments, and introduce our techniques to get efficient proofs. Even though such a setting is also mostly shared with DL-based Σ-protocols using homomorphic commitments, the main challenges described here are not encountered in those cases. Since our main concern is about the soundness of the protocol, in this section, we omit the discussion about the zero-knowledge property, which is later obtained using a standard rejection sampling technique. We always consider homomorphic commitments when referring to "commitment" and assume that all the elements are in a ring $\mathfrak{R}$.

5.3.1 The case for linear relations (2-special soundness)

If we investigate the (underlying) one-shot Σ-protocols from [Lyu09, Lyu12, BKLP15, BDL+18], we see the following. The common input of the protocol is a commitment C_1 to the prover's witness and the prover sends an initial commitment C_0.[5] Then, the verifier sends a random challenge $x \leftarrow \mathcal{C}$, which is responded by the prover as $(\boldsymbol{f}, \boldsymbol{z})$, and $(\boldsymbol{f}, \boldsymbol{z})$ is used by the verifier as a message-randomness pair for a commitment computation.[6] More precisely, the verification checks if $C_0 + xC_1 = \mathrm{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z})$ holds and $\boldsymbol{f}, \boldsymbol{z}$ have small norm. This is equivalent to the structure represented in Protocol 5.1 for $k = 1$. From here, when the extractor gets two valid protocol transcripts $(C_0, x_0, \boldsymbol{f}_0, \boldsymbol{z}_0), (C_0, x_1, \boldsymbol{f}_1, \boldsymbol{z}_1)$ using the same initial message C_0, and different

[5] The reason behind indexing becomes clear in what follows.

[6] In certain proofs, the use of UMC allows the prover to respond only with the randomness part $\boldsymbol{z}$. In such a case, $\boldsymbol{f}$ need not be transmitted and can be assumed to be set appropriately by the verifier.



PROTOCOL 5.1: Structure of a $(k+1)$-special sound Σ-protocol. $\mathcal{T}_f, \mathcal{T}_z \in \mathbb{R}^+$ are some pre-determined values that vary among different proofs.

challenges x_0 and x_1, the extractor obtains

$$\left.\begin{aligned} C_0 + x_0 C_1 &= \mathrm{Com}_{ck}(\boldsymbol{f}_0; \boldsymbol{z}_0) \\ C_0 + x_1 C_1 &= \mathrm{Com}_{ck}(\boldsymbol{f}_1; \boldsymbol{z}_1) \end{aligned}\right\} \implies (x_1 - x_0) C_1 = \mathrm{Com}_{ck}(\boldsymbol{f}_1 - \boldsymbol{f}_0; \boldsymbol{z}_1 - \boldsymbol{z}_0)\,. \quad (5.5)$$

At this stage, it is not possible to obtain a *valid exact* opening of C_1 unless $(x_1 - x_0)^{-1}$ is guaranteed to be short due to the shortness requirements of valid openings for lattice-based commitment schemes.[7] Unless ensured by design, there is no particular reason why the inverse term $(x_1 - x_0)^{-1}$ would be short. In the current state of affairs, the largest set of challenges with short challenge difference inverses is monomial challenges [BCK+14] used with ring variants of lattice assumptions. Here, only $2(x_1 - x_0)^{-1}$ is guaranteed to be short and thus the extractor can only get the openings of $2C_1$. As discussed previously, for a ring dimension of d, the cardinality of the monomial challenge space is only $2d$, which is typically smaller than 2^{12} in practice. This small challenge space problem causes major efficiency drawbacks in terms of both computation and communication as the protocol is required to be repeated many times to get a negligible soundness error (that is, the same computation and communication steps are repeated multiple times, resulting in a multi-fold increase in both computation and communication). The situation is even worse in terms of the number of repetitions when binary challenges or Stern's framework [Ste96] is used where the protocol is required to be repeated at least λ times for λ-bit security.

The idea for a one-shot proof is to make use of (5.5) without any inverse computation by observing that $(\boldsymbol{f}_1 - \boldsymbol{f}_0, \boldsymbol{z}_1 - \boldsymbol{z}_0)$ is a valid opening of $(x_1 - x_0)C_1$ as long as $\boldsymbol{f}_1 - \boldsymbol{f}_0$ and $\boldsymbol{z}_1 - \boldsymbol{z}_0$ are short, which is ensured by norm checks on $\boldsymbol{f}, \boldsymbol{z}$ in each verification. If one can prove that having this *relaxed* case is sufficient and also violates the binding property of the commitment (i.e., that it allows one to solve a computationally hard problem), then the soundness of the protocol is achieved (with a relaxed relation $\mathcal{R}'$ as in Definition 3.6) with no challenge difference inverses involved. This eliminates the need for challenge differences to have short inverses and enables one to use exponentially large challenge spaces, resulting in *one-shot* proofs. The main technical difficulty here is handling the *soundness gap*, where the extractor only obtains an exact opening of $(x_1 - x_0)C_1$ (rather than C_1, which is the commitment to the prover's witness).

[7] Recall that UMC allows an unbounded *message* opening, but still the randomness is required to be short.

5.3.2 Generalisation to degree $k > 1$ ($(k+1)$-special soundness)

As can be seen from (5.5), the 2-special sound case is quite restrictive as it only allows witness extraction from linear (first degree) relations. On the other hand, the ability to work with non-linear relations is a must in recent efficient proofs [GK15, BCC+15, BCC+16, BBB+18], which renders the existing lattice-based one-shot techniques inapplicable. Therefore, we generalise our setting, and suppose that we have a degree-k polynomial relation ($(k+1)$-special sound Σ-protocol), $k \geq 1$, with the structure given in Protocol 5.1. Note that since the extractor only knows that verification steps hold, unaware of how any component is generated, other steps but those in the verification is not important. Therefore, we write all the C_i's as a common input whereas in the actual protocol a subset of them can be generated during a protocol run. The commitment to the prover's witness $(\boldsymbol{m}_k, \boldsymbol{r}_k)$ is C_k.

The witness extraction, in this case, works by the extractor obtaining $k+1$ accepting protocol transcripts for distinct challenges $x_0, \ldots, x_k$ with the same input $(C_0, \ldots, C_k)$, and responses $(\boldsymbol{f}_0, \boldsymbol{z}_0), \ldots, (\boldsymbol{f}_k, \boldsymbol{z}_k)$, represented as below.

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^k \\ 1 & x_1 & x_1^2 & \cdots & x_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^k \end{pmatrix} \cdot \begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ C_k \end{pmatrix} = \begin{pmatrix} \mathrm{Com}_{ck}\left(\boldsymbol{f}_0; \boldsymbol{z}_0\right) \\ \mathrm{Com}_{ck}\left(\boldsymbol{f}_1; \boldsymbol{z}_1\right) \\ \vdots \\ \mathrm{Com}_{ck}\left(\boldsymbol{f}_k; \boldsymbol{z}_k\right) \end{pmatrix}. \tag{5.6}$$

We have seen that using the aforementioned *relaxed* opening approach, one can extract a witness from a linear relation (5.5) in *one shot*. Now a natural generalisation is to ask "Can we extract a witness from a non-linear relation (5.6) as in Protocol 5.1 in *one shot*?"

Naive approach and previous *multi-shot* approach. Denoting (5.6) as $\boldsymbol{V} \cdot \boldsymbol{c} = \boldsymbol{b}$, the matrix $\boldsymbol{V}$ is a Vandermonde matrix. A straightforward idea to obtain the openings of C_i's is to multiply both sides of (5.6) by $\boldsymbol{V}^{-1}$, which gives $\boldsymbol{c} = \boldsymbol{V}^{-1} \cdot \boldsymbol{b}$. From here, using the homomorphic properties of the commitment scheme, we can get *potential* "openings" of C_i's. However, one needs to make sure that $\boldsymbol{V}^{-1}$ exists over $\mathfrak{R}$ and that it has *short* entries so that these "openings" are valid. This is exactly the problem addressed in Section 4.1.2.

The approach in Section 4.1.2 was by making use of monomial challenges from [BCK+14]. Using the structure of $\boldsymbol{V}^{-1}$ in (3.4), we argued that the entries in $2^k\boldsymbol{V}^{-1}$ are short by the fact that doubled inverse of challenge differences (i.e., $2(x_j - x_i)^{-1}$) are short *when* monomial challenges are used. Thus, this approach still maintains the drawback of requiring multiple protocol repetitions to achieve a negligible soundness error, and does not address our question in this chapter.

***One-shot* solution.** Now, let us see how we can develop a one-shot proof technique for non-linear relations. Using (3.2), we multiply both sides of (5.6) by $\mathrm{adj}(\boldsymbol{V})$, and obtain

$$\mathrm{adj}(\boldsymbol{V}) \cdot \boldsymbol{V} \cdot \boldsymbol{c} = \mathrm{adj}(\boldsymbol{V}) \cdot \boldsymbol{b} \quad \Longrightarrow \quad \det(\boldsymbol{V}) \cdot \boldsymbol{c} = \mathrm{adj}(\boldsymbol{V}) \cdot \boldsymbol{b}. \tag{5.7}$$

Note that $\det(\boldsymbol{V})$ is just some scalar in $\mathfrak{R}$, and we obtain *potential relaxed* "openings" of C_i's as a result of the multiplication $\mathrm{adj}(\boldsymbol{V}) \cdot \boldsymbol{b}$. In particular, for the commitment C_k of the *witness*, we have

$$\det(\boldsymbol{V}) \cdot C_k = \sum_{i=0}^{k} \Gamma_i \cdot \mathrm{Com}_{ck}\left(\boldsymbol{f}_i; \boldsymbol{z}_i\right) = \mathrm{Com}_{ck}\left(\sum_{i=0}^{k} \Gamma_i \cdot \boldsymbol{f}_i; \sum_{i=0}^{k} \Gamma_i \cdot \boldsymbol{z}_i\right), \tag{5.8}$$

where $\Gamma_i = (-1)^{i+k} \prod_{0 \leq l < j \leq k \wedge j, l \neq i} (x_j - x_l)$ by Fact 3.18. As a result, we get a *relaxed*

opening of C_k, or more precisely, an *exact* opening of $\det(\boldsymbol{V}) \cdot C_k$ as $(\hat{\boldsymbol{m}}_k, \hat{\boldsymbol{r}}_k) = \left(\sum_{i=0}^{k} \Gamma_i \boldsymbol{f}_i, \sum_{i=0}^{k} \Gamma_i \boldsymbol{z}_i\right)$. Provided that the norms of $\hat{\boldsymbol{m}}_k$ and $\hat{\boldsymbol{r}}_k$ are small, this gives a *valid* opening and thus can be related to a hard lattice problem (M-SIS, in particular). It is important to observe that $\hat{\boldsymbol{m}}_k$ and $\hat{\boldsymbol{r}}_k$ do not involve any inverse term and can be guaranteed to be short by ensuring that Γ_i's are short. The opening of other C_i's can also be recovered analogously, but the case for C_k is sufficient for our applications.

When $k = 1$, i.e., when the protocol is 2-special sound, $\det(\boldsymbol{V}) = (x_1 - x_0)$ and $(\Gamma_0, \Gamma_1) = (-1, 1)$. Therefore, we exactly obtain (5.5) as a special case of (5.8) with $k = 1$. That is, we get the results of the previous approaches from [Lyu09, Lyu12, BKLP15, BDL+18] as a special case of ours.

5.3.3 New tools for compact proofs

Let us analyse our generalised solution and introduce our new tools to get compact proofs. The results can be easily used in other protocols that use a challenge space of the form in (5.9) as they are independent of the low-level details of a protocol. Since the most commonly used challenge spaces (e.g., in [BDL+18, BLO18, dPLS18, LN17, LS18]) for one-shot proofs are special cases of (5.9), our results are widely applicable.

Let $\mathfrak{R} = R = \mathbb{Z}[X]/(X^d + 1)$ and $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ for $q \in \mathbb{Z}^+$. For $w \leq d$ and $p \leq q/2$, let $\mathcal{C}^d_{w,p}$ be the challenge space defined as

$$\mathcal{C}^d_{w,p} = \{\, x \in \mathbb{Z}[X] \,:\, \deg(x) < d \,\wedge\, \mathsf{HW}(x) = w \,\wedge\, \|x\|_\infty = p \,\}. \tag{5.9}$$

It is easy to observe that $\|x\|_1 \leq pw$ for any $x \in \mathcal{C}^d_{w,p}$ and $|\mathcal{C}^d_{w,p}| = \binom{d}{w} \cdot (2p)^w$, which is, for example, larger than 2^{256} for $(d, w, p) = (256, 60, 1)$. We define $\Delta\mathcal{C}^d_{w,p}$ to be the set of challenge differences excluding zero.

Bound on the product of challenge differences.

Lemma 5.1. *For any $y_1, \ldots, y_n \in \Delta\mathcal{C}^d_{w,p}$, the following holds*

$$\left\| \prod_{i=1}^{n} y_i \right\|_\infty \leq (2p)^n \cdot w^{n-1}, \quad \textit{and} \quad \left\| \prod_{i=1}^{n} y_i \right\| \leq \sqrt{d} \cdot (2p)^n \cdot w^{n-1}.$$

Proof. Since $\|x\|_\infty \leq p$ and $\|x\|_1 \leq pw$ for all $x \in \mathcal{C}^d_{w,p}$, we have $\|y\|_\infty \leq 2p$ and $\|y\|_1 \leq 2pw$ for all $y \in \Delta\mathcal{C}^d_{w,p}$. Therefore, using Lemma 3.19, we get

$$\left\| \prod_{i=1}^{n} y_i \right\|_\infty \leq \prod_{i=1}^{n-1} \|y_i\|_1 \cdot \|y_n\|_\infty \leq (2p)^n \cdot w^{n-1}.$$

Therefore, we also have

$$\left\| \prod_{i=1}^{n} y_i \right\| \leq \sqrt{d} \cdot \left\| \prod_{i=1}^{n} y_i \right\|_\infty \leq \sqrt{d} \cdot (2p)^n \cdot w^{n-1}.$$

□

Bound on the relaxation factor: $\det(\boldsymbol{V})$.

Lemma 5.2. *Let $\kappa = \binom{k+1}{2} = \frac{k(k+1)}{2}$. For the $(k+1)$-dimensional Vandermonde matrix $\boldsymbol{V}$ defined in (5.6) using the challenge space $\mathcal{C}^d_{w,p}$ in (5.9),*

$$\|\det(\boldsymbol{V})\|_\infty \leq (2p)^\kappa \cdot w^{\kappa-1}.$$

Proof. By Fact 3.17, $\det(\boldsymbol{V})$ has $\kappa = \binom{k+1}{2}$ multiplicands where each multiplicand is in $\Delta\mathcal{C}^d_{w,p}$. The result follows from Lemma 5.1. □

Bound on the extracted witness norm: $\mathrm{adj}(\boldsymbol{V}) \times$ **(openings of** $\boldsymbol{b}$**).**

Lemma 5.3. *For* $k \geq 1$ *and* $(\hat{\boldsymbol{m}}_k, \hat{\boldsymbol{r}}_k) = \left(\sum_{i=0}^{k} \Gamma_i \boldsymbol{f}_i, \sum_{i=0}^{k} \Gamma_i \boldsymbol{z}_i\right)$ *where* $\Gamma_i = \prod_{0 \leq l < j \leq k \wedge j, l \neq i}(x_j - x_l)$*, the following holds, for* $\kappa' = k(k-1)/2$*,*

- $\|\hat{\boldsymbol{m}}_k\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot \max_i \|\boldsymbol{f}_i\|$*, and*
- $\|\hat{\boldsymbol{r}}_k\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot \max_i \|\boldsymbol{z}_i\|$.

Proof. Let $\kappa' = \frac{k(k-1)}{2}$. We have

$$\|\hat{\boldsymbol{m}}_k\| = \left\|\sum_{i=0}^{k} \Gamma_i \boldsymbol{f}_i\right\| \leq (k+1) \cdot \max_i \|\Gamma_i \boldsymbol{f}_i\| \leq (k+1) \cdot \sqrt{d} \cdot \max_i \|\Gamma_i\| \cdot \max_i \|\boldsymbol{f}_i\|$$
$$\leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot \max_i \|\boldsymbol{f}_i\|\,, \quad \text{(by Fact 3.18 and Lemma 5.1).}$$

The bound for $\hat{\boldsymbol{r}}_k$ follows in a similar manner. □

Reducing extracted witness norm in proofs with non-linear relations. In some proofs with non-linear polynomial relations such as our one-out-of-many proof, the extractor obtains an opening with a relaxation factor y of some component that is witness of a sub-protocol. Since the invertibility of y is not ensured, when this opening is used in the non-linear polynomial relation, the relaxation factor also gets exponentiated by the degree $k > 1$. In the end, instead of getting $\det(\boldsymbol{V})$ as the overall relaxation factor, we end up with relaxation factor $y^k \cdot \det(\boldsymbol{V})$. We use the lemma below to show that even though we cannot completely eliminate the extra term y^k, we can eliminate its exponent k. This results in obtaining an extracted witness with a smaller norm, and in turn, helps in getting shorter proofs.

Lemma 5.4. *Let* $f, g \in R = \mathbb{Z}[X]/(X^d + 1)$*. If* $f \cdot g^k = 0$ *in* $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ *for some* $k \in \mathbb{Z}^+$*, then* $f \cdot g = 0$ *in* R_q*.*

Proof. If $k = 1$, then the result is clear. Assume that $k \geq 2$. Suppose that $X^d + 1$ factors into $n \leq d$ irreducible polynomials $\alpha_1, \ldots, \alpha_n$ modulo q. Let S be the set of indices i such that $g^k = 0 \bmod (q, \alpha_i)$ and $\varepsilon = |S|$. (Note that S may be an empty set and $\varepsilon = 0$).

(1) From the definition of S and the fact that $f \cdot g^k = 0$ over R_q, we have $f = 0 \bmod (q, \alpha_j)$ for all $j \notin S$.
(2) For any $i \in S$, $g^k = 0 \bmod (q, \alpha_i)$ by the definition of S. Since α_i is irreducible modulo q, it is impossible to have this property without having $g = 0 \bmod (q, \alpha_i)$.

Thus, for all $i \notin S$, $f \cdot g = 0 \bmod (q, \alpha_i)$ by (1). And, for all $i \in S$, $f \cdot g = 0 \bmod (q, \alpha_i)$ by (2). By the Chinese Remainder Theorem, $f \cdot g = 0$ over R_q. □

5.4 New Techniques for Faster Lattice-Based Proofs and Application to Range Proofs

In this section, we go into the details of our new techniques to get computation-efficient proofs. We first show a lemma that enables one to prove the following: if a product of polynomials is equal to zero in R_q and the norm of each factor is sufficiently small,

then there must be a factor which is exactly equal to zero. This result works for any sufficiently large q, enabling the use of a modulus suitable for fast computation such as an "NTT-friendly" modulus.

Lemma 5.5. *Let $f_1, \ldots, f_n \in R$ for some $n \geq 1$. If $\prod_{i=1}^{n} f_i = 0$ in R_q and $q/2 > \|f_1\|_\infty \cdot \prod_{i=2}^{n} \|f_i\|_1$, then there exists $1 \leq j \leq n$ such that $f_j = 0$.*

Proof. Using Lemma 3.19 and the assumption on q, we have

$$\left\| \prod_{i=1}^{s} f_i \right\|_\infty \leq \|f_1\|_\infty \cdot \prod_{i=2}^{n} \|f_i\|_1 < q/2.$$

Therefore, $\prod_{i=1}^{n} f_i = 0$ holds over R. Since $X^d + 1$ is irreducible over $\mathbb{Q}$, (at least) one of the multiplicand f_i's must be zero. □

Note that Lemma 5.5 requires all the multiplicands to have bounded norm whereas there is no such requirement in Lemma 5.4. Therefore, we are unable to use Lemma 5.5 for the purpose of the use of Lemma 5.4 described previously as there is no norm-bound on a multiplicand in the place Lemma 5.4 is used (see how these lemmas are used in the soundness proofs for more details). Lemma 5.5 is used in the binary proof to argue that $y_0 y_1 y_2 \hat{b}(y - \hat{b}) = 0$ in R_q for some (non-zero) challenge differences y, y_0, y_1, y_2 implies $\hat{b} = yb$ for a bit $b \in \{0, 1\}$ without requiring invertibility of any challenge difference (see Section 5.5.1).

5.4.1 Supporting inter-slot operations on CRT-packed messages

Now, we can go into the details of our CRT packing technique. Define $f = \mathrm{Enc}_x(m) = x \cdot m + \rho \in R_q$ as an encoding of a message m under a challenge x. This encoding is widely used in proofs of knowledge as a "masked" response to a challenge x. An important advantage of this encoding over a commitment is that the storage cost of an encoding is at most $d \log q$ whereas that of a commitment is $nd \log q$ for HMC and $(n+v)d \log q$ for UMC. Therefore, for a typical module rank of, say, 4, a commitment is $4\times$ more costly than an encoding.

There are known methods to choose a modulus q such that $X^d + 1$ splits into s factors, in which case, R_q splits into s fields and we get $R_q = R_q^{(0)} \times \cdots \times R_q^{(s-1)}$. In the case that X^d+1 splits into more than s factors, but we only want to use s slots, we still have $R_q = R_q^{(0)} \times \cdots \times R_q^{(s-1)}$ where $R_q^{(i)} = \mathbb{Z}_q[X]/(P^{(i)}(X))$ for some polynomial $P^{(i)}(X)$ of degree d/s. However, $R_q^{(i)}$'s are not a field in that case as $P^{(i)}(X)$'s are not irreducible over $\mathbb{Z}_q$.

As discussed previously, when we use these s slots to pack s messages in a single ring element, we have

$$f = \mathrm{Enc}_x(m) = x \cdot m + \rho = \langle x_0 m_0 + \rho_0, \ldots, x_{s-1} m_{s-1} + \rho_{s-1} \rangle, \tag{5.10}$$

where $x = \langle x_0, \ldots, x_{s-1} \rangle$, $m = \langle m_0, \ldots, m_{s-1} \rangle$ and $\rho = \langle \rho_0, \ldots, \rho_{s-1} \rangle$ in the CRT-packed representation. In this case, parallel additions are easy as

$$\mathrm{Enc}_x(\langle m_0, \ldots, m_{s-1} \rangle) + \mathrm{Enc}_x(\langle m'_0, \ldots, m'_{s-1} \rangle) = \mathrm{Enc}_x(\langle m_0 + m'_0, \ldots, m_{s-1} + m'_{s-1} \rangle).$$

Parallel multiplication is also possible as $\mathrm{Enc}_x(m) \cdot \mathrm{Enc}_x(m') = m \cdot m' \cdot x^2 + c_1 x + c_0$ for c_0, c_1 only dependent on m, m', ρ, ρ', all of which are known to the prover in advance of his first move. Therefore, the prover can prove that the coefficient of x^2

is the product of m and m', and thus proving the relation in parallel for all CRT slots.[8] Addition and multiplication alone, however, do not provide a complete set of operations (see [GHS12] for a discussion in the context of FHE). Given an encoding of m, our main requirement is to have the ability to extract all encodings in the CRT slots of m in a way that allows further operations among extracted encodings. That is, all extracted encodings must be under the same challenge x, which translates to requiring $x = \langle x, \ldots, x \rangle$ for $x \in \bigcap_{i=0}^{s-1} R_q^{(i)}$. As a result, when we use s slots, the degree of a challenge can be at most $d/s - 1$. With this, from an encoding $f = \mathrm{Enc}_x(\langle m_0, \ldots, m_{s-1} \rangle)$, anyone can extract encodings by computing

$$f_i := f \bmod (q, P^{(i)}(X)) = x \cdot m_i + \rho_i = \mathrm{Enc}_x(m_i)$$

for all $0 \leq i \leq s-1$. Conversely, given encoding $\mathrm{Enc}_x(m_i)$'s for all $0 \leq i \leq s-1$, anyone can compute an encoding $\mathrm{Enc}_x(\langle m_0, \ldots, m_{s-1} \rangle)$.

Even more, with this choice of the challenge $x = \langle x, \ldots, x \rangle$ for $x \in \bigcap_{i=0}^{s-1} R_q^{(i)}$, we get invariance of the challenge under *any* permutation σ on CRT slots. That is, for any permutation σ, we have $\sigma(\mathrm{Enc}_x(m)) = \mathrm{Enc}_x(\sigma(m))$. From here, one can perform any inter-slot operation, and may even not require packing/unpacking of the messages in some applications. In our application to the range proof, extraction of the slots is sufficient and we refer to [GHS12] for more on permutations. In our approach, an encoding and a commitment per message slot costs, respectively, at most $d \log q/s$ bits and $(n+v) \log q/s$ bits, which are much cheaper than a commitment to a single message.

5.4.2 Using CRT-packed inter-slot operations in relaxed range proof

In this section, we introduce the first application of our ideas to Σ-protocols where the proof is *relaxed* as described in Section 3.2.3. In all of our protocols in this chapter, the prover aborts if any rejection sampling step (Algorithm 3.2) returns 0, and our protocols in this chapter are honest-verifier zero-knowledge for *non-aborting* interactions. For most of the practical applications, the protocol is made non-interactive, and thus having only non-aborting protocols with the zero-knowledge property does not cause an issue. Nevertheless, the protocols can be easily adapted to be zero-knowledge for the aborting cases using a standard technique from [BCK+14] as done in Chapter 4.

Our first application is a range proof that allows an efficient aggregation in the sense that the prover can prove that a set of committed values packed in a *single* commitment falls within a set of certain ranges. Let $\psi \in \mathbb{Z}^+$, $\ell^{(i)} \in [0, N_i)$ be prover's values for $1 \leq i \leq \psi$ and $N_i = 2^{k_i}$ with $k = k_1 + \cdots + k_\psi$, and s be the smallest power of two such that $s \geq \max\{k_1, \ldots, k_\psi\}$. For simplicity, we use base $\beta = 2$, but the result can be generalized to other base values β. The binary case gives the most compact proofs in practice. Assume that $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ splits into exactly s fields such that $R_q = R_q^{(0)} \times \cdots \times R_q^{(s-1)}$ and $R_q^{(i)} = \mathbb{Z}_q[X]/(P^{(i)}(X))$ for some *irreducible* polynomial $P^{(i)}(X)$ of degree d/s for all $0 \leq i < s$. Write $\ell^{(i)} = (b_0^{(i)}, \ldots, b_{k_i-1}^{(i)})$ in the binary representation and define $\ell_{\mathrm{crti}}^{(i)} = \langle b_0^{(i)}, \ldots, b_{k_i-1}^{(i)} \rangle$. The exact relations proved by our "simultaneous" range proof is given in Definition 5.6. We show in Section 5.6.2 that the relaxed range proof is sufficient for an application in anonymous credentials. Such a "simultaneous" range proof is useful when showing a credential that a set of attributes such as age, expiry date, residential postcode etc. fall into some respective

[8] We believe this is the application of CRT mentioned in [LS18].

ranges, and this can be achieved with a single commitment and a single proof using our techniques.

Definition 5.6. *The following defines the relations for Protocol 5.2 for* $\mathcal{T}, \hat{\mathcal{T}} \in \mathbb{R}^+$.

$$\mathcal{R}_{\text{range}}(\mathcal{T}) = \left\{ \begin{array}{c} ((ck, V), (\ell^{(1)}, \ldots, \ell^{(\psi)}, \boldsymbol{r})) \ : \ \|\boldsymbol{r}\| \leq \mathcal{T} \wedge \\ V = \text{Com}_{ck}\left(\ell^{(1)}, \ldots, \ell^{(\psi)}; \boldsymbol{r}\right) \ \wedge \ \ell^{(i)} \in [0, N_i) \ \forall 1 \leq i \leq \psi \end{array} \right\},$$

$$\mathcal{R}'_{\text{range}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{c} ((ck, V), (\bar{x}, \ell^{(1)}, \ldots, \ell^{(\psi)}, \hat{\boldsymbol{r}})) \ : \ \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \ \wedge \ \bar{x} \in \Delta\mathcal{C}^{d/s}_{w,p} \ \wedge \\ \bar{x}V = \text{Com}_{ck}\left(\bar{x}\ell^{(1)}, \ldots, \bar{x}\ell^{(\psi)}; \hat{\boldsymbol{r}}\right) \wedge \ell^{(i)} \in [0, N_i) \ \forall 1 \leq i \leq \psi \end{array} \right\}.$$

The full description of the range proof is given in Protocol 5.2 where the commitment scheme is instantiated with UMC and ϕ_1, ϕ_2 are parameters determining the rejection sampling rate. The first part of the proof (Steps 4 and 5 in the verification, and its relevant components) uses the binary proof idea from [BCC+15] to show that $f_j^{(i)}$'s are encodings of bits, but the proof is done in parallel CRT slots. Observe in Protocol 5.2 that $f^{(i)} = x \cdot \langle b_0^{(i)}, \ldots, b_{k_i-1}^{(i)}, \mathbf{0}^{s-k_i} \rangle + \langle a_0^{(i)}, \ldots, a_{k_i-1}^{(i)}, \mathbf{0}^{s-k_i} \rangle = x \cdot \ell_{\text{crti}}^{(i)} + a_{\text{crti}}^{(i)}$ where $\mathbf{0}^{s-k_i}$ denotes a zero vector of dimension $s - k_i$. Therefore, we have, for each $1 \leq i \leq \psi$,

$$f^{(i)}(x - f^{(i)}) = x^2 \cdot \ell_{\text{crti}}^{(i)}(1 - \ell_{\text{crti}}^{(i)}) + x \cdot a_{\text{crti}}^{(i)}(1 - 2\ell_{\text{crti}}^{(i)}) - (a_{\text{crti}}^{(i)})^2.$$

Since there is no x^2 term (i.e., the coefficient of x^2 is zero) on the left hand side of Step 5 in the verification, we get $\ell_{\text{crti}}^{(i)}(1 - \ell_{\text{crti}}^{(i)}) = 0$ when Step 5 is satisfied for 3 distinct challenges x. This gives us

$$\langle b_0^{(i)}(1 - b_0^{(i)}), \ldots, b_{k_i-1}^{(i)}(1 - b_{k_i-1}^{(i)}), \mathbf{0}^{s-k_i} \rangle = 0 \implies b_j^{(i)}(1 - b_j^{(i)}) = 0 \text{ in } R_q^{(j)} \quad (5.11)$$

for each $0 \leq j < s - k_i$. This fact is then used to prove that $b_j^{(i)}$'s are binary. However, since the proof is relaxed, we need to deal with more complicated issues and give the full details in the proof of Theorem 5.8 below. The second part of the proof is a standard argument to show that the bits $b_0^{(i)}, \ldots, b_{k_i-1}^{(i)}$ construct a value $\ell^{(i)}$ for each $1 \leq i \leq \psi$.

Remark 5.7. *The first rejection sampling at Step 14 of Protocol 5.2 is not necessary as UMC allows unbounded-length messages. However, when rejection sampling is done, the bitsize of* $f_j^{(i)}$*'s are smaller than* $d \log q/s$*, which is the bitsize of a random element in* $R_q^{(j)}$*. Further, there is no mod* q *reduction in the prover's response, and also no mod* $P^{(j)}(X)$ *at Step 13 of Protocol 5.2 since* $b_j^{(i)}$*'s are binary.*

Theorem 5.8. *Let* $\gamma_{\text{range}} = 4\sqrt{3}\phi_2 pw\mathcal{B}md$*. Assume* $q > \max\{N_1, \ldots, N_\psi\}$*,* $d \geq 128$*,*[9] R_q *splits into exactly* s *fields and UMC is hiding and* γ_{range}*-binding. Then, Protocol 5.2 is a* 3*-special sound* Σ*-protocol (as in Definition 3.6) for the relations* $\mathcal{R}_{\text{range}}(\mathcal{B}\sqrt{md})$ *and* $\mathcal{R}'_{\text{range}}(\gamma_{\text{range}})$ *with a completeness error* $1 - 1/(\mu(\phi_1)\mu(\phi_2))$ *for* $\mu(\cdot)$ *defined in Lemma 3.16.*

Proof. Let $k = k_1 + \cdots + k_\psi$.

Completeness: The prover responds with probability $1/(\mu(\phi_1)\mu(\phi_2)) + \varepsilon$ for $|\varepsilon| \leq 2 \cdot 2^{-100}$ by Lemma 3.16. Since there is at most $k_1 + \cdots + k_s = k$-many 1's in $\boldsymbol{b}$ and the

[9] The assumption $d \geq 128$ is put merely to use a constant factor of 2 as in Lemma 3.15 when bounding the Euclidean norm of a vector following normal distribution.

$\mathcal{P}_{\text{range}}((ck, V), (\ell^{(1)}, \ldots, \ell^{(\psi)}; \boldsymbol{r}))$ | $\mathcal{V}_{\text{range}}(ck, V)$

$\boldsymbol{r}_b, \boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$

$\boldsymbol{r}_a, \boldsymbol{r}_d, \boldsymbol{r}_e \leftarrow D^{md}_{\phi_2 T_2}$ for $T_2 = pw\mathcal{B}\sqrt{3md}$

for $i = 1, \ldots, \psi$ **do**

  $a^{(i)}_0, \ldots, a^{(i)}_{k_i - 1} \leftarrow D^{d/s}_{\phi_1 T_1}$ for $T_1 = p\sqrt{kw}$

  $a^{(i)}_{\text{crti}} = \texttt{CRT}^{-1}(a^{(i)}_0, \ldots, a^{(i)}_{k_i-1}, \boldsymbol{0}^{s-k_i})$

  $\ell^{(i)}_{\text{crti}} = \texttt{CRT}^{-1}(b^{(i)}_0, \ldots, b^{(i)}_{k_i-1}, \boldsymbol{0}^{s-k_i})$

$B = \text{Com}_{ck}\left(\ell^{(1)}_{\text{crti}}, \ldots, \ell^{(\psi)}_{\text{crti}}; \boldsymbol{r}_b\right)$

$A = \text{Com}_{ck}\left(a^{(1)}_{\text{crti}}, \ldots, a^{(\psi)}_{\text{crti}}; \boldsymbol{r}_a\right)$

$C = \text{Com}_{ck}\left(a^{(1)}_{\text{crti}}(1 - 2\ell^{(1)}_{\text{crti}}), \ldots, a^{(\psi)}_{\text{crti}}(1 - 2\ell^{(\psi)}_{\text{crti}}); \boldsymbol{r}_c\right)$

$D = \text{Com}_{ck}\left(-(a^{(1)}_{\text{crti}})^2, \ldots, -(a^{(\psi)}_{\text{crti}})^2; \boldsymbol{r}_d\right)$

$E = \text{Com}_{ck}(\boldsymbol{e}; \boldsymbol{r}_e)$ —— A, B, C, D, E ——→

←—— x —— $x \leftarrow \mathcal{C}^{d'}_{w,p}$ for $d' = d/s$

for $i \in [1, \psi], j \in [0, k_i)$ **do**

  $f^{(i)}_j = x \cdot b^{(i)}_j + a^{(i)}_j$

$\boldsymbol{f}_{\text{crt}} := (f^{(1)}_0, \ldots, f^{(\psi)}_{k_\psi - 1})$

$\boldsymbol{b} := (b^{(1)}_0, \ldots, b^{(\psi)}_{k_\psi - 1})$

$\text{Rej}(\boldsymbol{f}_{\text{crt}}, x\boldsymbol{b}, \phi_1, p\sqrt{kw})$

$\boldsymbol{z}_b = x \cdot \boldsymbol{r}_b + \boldsymbol{r}_a,\ \boldsymbol{z}_c = x \cdot \boldsymbol{r}_c + \boldsymbol{r}_d$

$\boldsymbol{z} = x \cdot \boldsymbol{r} + \boldsymbol{r}_e$

$\text{Rej}((\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}), x(\boldsymbol{r}_b, \boldsymbol{r}_c, \boldsymbol{r}), \phi_2, T_2)$

If aborted, return $\perp$. —— $\boldsymbol{f}_{\text{crt}}, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}$ ——→

for $i = 1, \ldots, \psi$ **do**

  $f^{(i)} = \texttt{CRT}^{-1}(f^{(i)}_0, \ldots, f^{(i)}_{k_i-1}, \boldsymbol{0}^{s-k_i})$

$\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\|, \|\boldsymbol{z}\| \stackrel{?}{\leq} 2\phi_2 T_2\sqrt{md}$

$xB + A \stackrel{?}{=} \text{Com}_{ck}\left(f^{(0)}, \ldots, f^{(\psi)}; \boldsymbol{z}_b\right)$

$\boldsymbol{g} := (f^{(0)}(x - f^{(0)}), \ldots, f^{(\psi)}(x - f^{(\psi)}))$

$xC + D \stackrel{?}{=} \text{Com}_{ck}(\boldsymbol{g}; \boldsymbol{z}_c)$

$xV + E \stackrel{?}{=} \text{Com}_{ck}(\boldsymbol{v}; \boldsymbol{z})$

PROTOCOL 5.2: Σ-protocol for $\mathcal{R}_{\text{range}}$ and $\mathcal{R}'_{\text{range}}$. The vectors $\boldsymbol{e}$ and $\boldsymbol{v}$ are defined below.

$$\boldsymbol{e} := \left(\sum_{j=0}^{k_1-1} 2^j a^{(1)}_j, \ldots, \sum_{j=0}^{k_\psi-1} 2^j a^{(\psi)}_j\right), \boldsymbol{v} := \left(\sum_{j=0}^{k_1-1} 2^j f^{(1)}_j, \ldots, \sum_{j=0}^{k_\psi-1} 2^j f^{(\psi)}_j\right) \quad \text{over } R_q.$$

rest is zero, there will be at most kw non-zero coefficients in $x\boldsymbol{b}$ where each coefficient is in $\{-p,\ldots,p\}$. Thus, we have

$$\|x\boldsymbol{b}\| = \left\|x(b_0^{(1)},\ldots,b_{k_\psi-1}^{(\psi)})\right\| \leq p\sqrt{kw} = T_1.$$

Also, we have, using Lemma 3.19,

$$\|x(\boldsymbol{r}_b,\boldsymbol{r}_c,\boldsymbol{r})\| \leq \|x(\boldsymbol{r}_b,\boldsymbol{r}_c,\boldsymbol{r})\|_\infty \sqrt{3md} \leq \|x\|_1 \|(\boldsymbol{r},\boldsymbol{r}_c)\|_\infty \sqrt{3md} \leq pw\mathcal{B}\sqrt{3md} = T_2.$$

Hence, by Lemma 3.16, the distributions of $f_j^{(i)}$'s are statistically close to $D_{\phi_1 T_1}^{d/s}$ and those of $\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z}$ are statistically close to $D_{\phi_2 T_2}^{md}$ (within statistical distance 2^{-100} in both cases). Therefore, since $d \geq 128$, by Lemma 3.15 except with probability at most 2^{-128}, we have

$$\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\|, \|\boldsymbol{z}\| \leq 2\cdot\phi_2 pw\mathcal{B}\sqrt{3md}\cdot\sqrt{md} = 2\sqrt{3}\phi_2 pw\mathcal{B}md.$$

Steps 4 and 5 of the verification follow by a straightforward investigation. For the last step of the verification, we have, for each $1 \leq i \leq \psi$,

$$\sum_{j=0}^{\psi} 2^j f_j^{(i)} = x\sum_{j=0}^{\psi} 2^j b_j^{(i)} + \sum_{j=0}^{\psi} 2^j a_j^{(i)} = x\cdot\ell^{(i)} + \sum_{j=0}^{\psi} 2^j a_j^{(i)}. \tag{5.12}$$

Therefore, we get

$$\mathrm{Com}_{ck}(\boldsymbol{v};\boldsymbol{z}) - E = \mathrm{Com}_{ck}(\boldsymbol{v}-\boldsymbol{e};\boldsymbol{z}-\boldsymbol{r}_e) = \mathrm{Com}_{ck}\left(x\ell^{(1)},\ldots,x\ell^{(\psi)};x\boldsymbol{r}\right) = xV.$$

SHVZK: Assume that the protocol is not aborted. The simulator sets $C = \mathrm{Com}_{ck}(\boldsymbol{0};\boldsymbol{r}_c)$ and $B = \mathrm{Com}_{ck}(\boldsymbol{0};\boldsymbol{r}_b)$ for $\boldsymbol{r}_c,\boldsymbol{r}_b \leftarrow \{-\mathcal{B},\ldots,\mathcal{B}\}^{md}$. Then, it picks $f_j^{(i)} \leftarrow D_{\phi_1 T_1}^{d/s}$ for all $1 \leq i \leq \psi$ and $0 \leq j \leq k_i - 1$, and also $\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z} \leftarrow D_{\phi_2 T_2}^{md}$. Then, it computes $\boldsymbol{f} = (f^{(1)},\ldots,f^{(\psi)})$ as in the verification of Protocol 5.2. Finally, it computes $A = \mathrm{Com}_{ck}(\boldsymbol{f};\boldsymbol{z}_b) - xB$, $D = \mathrm{Com}_{ck}(\boldsymbol{g};\boldsymbol{z}_c) - xC$ and $E = \mathrm{Com}_{ck}(\boldsymbol{v};\boldsymbol{z}) - xV$ where $\boldsymbol{g},\boldsymbol{v}$ are set as in Protocol 5.2. It outputs the simulated transcript $((A,B,C,D,E),x,(\boldsymbol{f}_{\mathrm{crt}},\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z}))$ where $\boldsymbol{f}_{\mathrm{crt}} := (f_0^{(1)},\ldots,f_{k_\psi-1}^{(\psi)})$.

The distribution of simulated $(\boldsymbol{f}_{\mathrm{crt}},\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z})$ is statistically close to the real distribution by Lemma 3.16 as argued in the completeness proof. Conditioned on $(\boldsymbol{f}_{\mathrm{crt}},\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z},x)$ and (B,C,V), simulated (A,D,E)'s distribution is exactly the same as in the real case. Finally, the distribution of simulated (B,C) is computationally indistinguishable from the real one by the hiding property of the commitment scheme (i.e., due to M-LWE).

3-special soundness: Given 3 accepting protocol transcripts, we have

$$\begin{aligned}&(A,B,C,D,E,x,\boldsymbol{f}_{\mathrm{crt}},\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z}),\\ &(A,B,C,D,E,x',\boldsymbol{f}'_{\mathrm{crt}},\boldsymbol{z}'_b,\boldsymbol{z}'_c,\boldsymbol{z}'),\\ &(A,B,C,D,E,x'',\boldsymbol{f}''_{\mathrm{crt}},\boldsymbol{z}''_b,\boldsymbol{z}''_c,\boldsymbol{z}''),\end{aligned}$$

with $\boldsymbol{f} = (f^{(1)},\ldots,f^{(\psi)})$, $\boldsymbol{f}' = (f'^{(1)},\ldots,f'^{(\psi)})$ and $\boldsymbol{f}'' = (f''^{(1)},\ldots,f''^{(\psi)})$ computed as in the verification. We split the proof into two parts: binary proof and range proof.

Binary proof. By Step 4 in the verification, we have

$$xB + A = \mathrm{Com}_{ck}(\boldsymbol{f};\boldsymbol{z}_b), \tag{5.13}$$

$$x'B + A = \mathrm{Com}_{ck}\left(\boldsymbol{f}'; \boldsymbol{z}'_b\right), \tag{5.14}$$
$$x''B + A = \mathrm{Com}_{ck}\left(\boldsymbol{f}''; \boldsymbol{z}''_b\right). \tag{5.15}$$

Subtracting (5.14) from (5.13), we get $(x - x') \cdot B = \mathrm{Com}_{ck}\left(\boldsymbol{f} - \boldsymbol{f}'; \boldsymbol{z}_b - \boldsymbol{z}'_b\right)$. Thus, for $y := x - x'$, we get exact valid openings of yB such that

$$yB = \mathrm{Com}_{ck}\left(\boldsymbol{f} - \boldsymbol{f}'; \boldsymbol{z}_b - \boldsymbol{z}'_b\right) =: \mathrm{Com}_{ck}\left(\hat{\boldsymbol{b}}; \hat{\boldsymbol{r}}_b\right). \tag{5.16}$$

Note that $\|\hat{\boldsymbol{r}}_b\| = \|\boldsymbol{z}_b - \boldsymbol{z}'_b\| \leq 4\sqrt{3}\phi_2 pw\mathcal{B}md = \gamma_{\text{range}}$, proving the claimed bound for $\mathcal{R}'_{\text{range}}$. Multiplying (5.13) by y and using (5.16) gives

$$\begin{aligned} yA &= \mathrm{Com}_{ck}\left(y\boldsymbol{f}; y\boldsymbol{z}_b\right) - xyB = \mathrm{Com}_{ck}\left(y\boldsymbol{f} - x\hat{\boldsymbol{b}}; y\boldsymbol{z}_b - x\hat{\boldsymbol{r}}_b\right) \\ &= \mathrm{Com}_{ck}\left(x\boldsymbol{f}' - x'\boldsymbol{f}; x\boldsymbol{z}'_b - x'\boldsymbol{z}_b\right) =: \mathrm{Com}_{ck}\left(\hat{\boldsymbol{a}}; \hat{\boldsymbol{r}}_a\right). \end{aligned} \tag{5.17}$$

Observe that $y\boldsymbol{f} = x\hat{\boldsymbol{b}} + \hat{\boldsymbol{a}}$ by the definition of $\hat{\boldsymbol{a}}$. By the Chinese Remainder Theorem, the equality holds in each CRT slot. Using Step 5 of the verification in a similar manner, we get exact message openings $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{d}}$ of yC and yD such that $y\boldsymbol{g} = x\hat{\boldsymbol{c}} + \hat{\boldsymbol{d}}$. Writing these equations coordinate-wise in each CRT slot, we have the following for all $1 \leq i \leq \psi$ and $0 \leq j \leq s - 1$

$$yf_j^{(i)} = x\hat{b}_j^{(i)} + \hat{a}_j^{(i)} \quad \text{in } R_q^{(j)}, \text{ and} \tag{5.18}$$
$$yg_j^{(i)} = yf_j^{(i)}(x - f_j^{(i)}) = x\hat{c}_j^{(i)} + \hat{d}_j^{(i)} \quad \text{in } R_q^{(j)}, \tag{5.19}$$

since all the challenges and their differences are the same in each CRT slot. Now, by the γ_{range}-binding property of UMC, except with negligible probability, the PPT prover cannot output a new valid exact opening of yA, yB, yC or yD in any of its rewinds. Thus, except with negligible probability, responses with respect to x' and x'' will have the same form. That is, the following holds

$$\begin{aligned} yf_j'^{(i)} &= x'\hat{b}_j^{(i)} + \hat{a}_j^{(i)}, & yf_j'^{(i)}(x' - f_j'^{(i)}) &= x'\hat{c}_j^{(i)} + \hat{d}_j^{(i)}, \\ yf_j''^{(i)} &= x''\hat{b}_j^{(i)} + \hat{a}_j^{(i)}, & yf_j''^{(i)}(x'' - f_j''^{(i)}) &= x''\hat{c}_j^{(i)} + \hat{d}_j^{(i)}, \end{aligned} \quad \text{in } R_q^{(j)}. \tag{5.20}$$

Now, multiplying (5.19) by y and using (5.18), we get

$$\begin{aligned} y\cdot\left(x \cdot \hat{c}_j^{(i)} + \hat{d}_j^{(i)}\right) &= y \cdot \left(yf_j^{(i)}(x - f_j^{(i)})\right) = yf_j^{(i)}(yx - yf_j^{(i)}) \\ &= (x\hat{b}_j^{(i)} + \hat{a}_j^{(i)})(yx - x\hat{b}_j^{(i)} - \hat{a}_j^{(i)}) = (x\hat{b}_j^{(i)} + \hat{a}_j^{(i)})(x(y - \hat{b}_j^{(i)}) - \hat{a}_j^{(i)}) \\ &= x^2\left[\hat{b}_j^{(i)}(y - \hat{b}_j^{(i)})\right] + x\left[\hat{a}_j^{(i)}(y - 2\hat{b}_j^{(i)})\right] - (\hat{a}_j^{(i)})^2, \end{aligned} \tag{5.21}$$

and thus

$$x^2\left[\hat{b}_j^{(i)}(y - \hat{b}_j^{(i)})\right] + x\left[\hat{a}_j^{(i)}(y - 2\hat{b}_j^{(i)}) - y\hat{c}_j^{(i)}\right] - (\hat{a}_j^{(i)})^2 - y\hat{d}_j^{(i)} = 0 \quad \text{in } R_q^{(j)}. \tag{5.22}$$

Repeating the same steps of (5.21) with the equations in (5.20), we get two copies of (5.22) where x is replaced with x' in one and with x'' in the other. That is, we have

the following system

$$\begin{pmatrix} 1 & x & x^2 \\ 1 & x' & x'^2 \\ 1 & x'' & x''^2 \end{pmatrix} \cdot \begin{pmatrix} -(\hat{a}_j^{(i)})^2 - y\hat{d}_j^{(i)} \\ \hat{a}_j^{(i)}(y - 2\hat{b}_j^{(i)}) - y\hat{c}_j^{(i)} \\ \hat{b}_j^{(i)}(y - \hat{b}_j^{(i)}) \end{pmatrix} = \mathbf{0} \qquad \text{in } R_q^{(j)}. \tag{5.23}$$

Since $R_q^{(j)}$ is a field, the Vandermonde matrix on the left is invertible for distinct challenges, and we get $\hat{b}_j^{(i)}(y - \hat{b}_j^{(i)}) = 0$, which implies $\hat{b}_j^{(i)} \in \{0, y\}$ in a field, i.e.,

$$\hat{b}_j^{(i)} = y b_j^{(i)} \quad \text{for } b_j^{(i)} \in \{0, 1\}. \tag{5.24}$$

Range proof. By Step 6 of the verification, we have $yV = \mathrm{Com}_{ck}(\boldsymbol{v} - \boldsymbol{v}'; \boldsymbol{z} - \boldsymbol{z}')$. Multiplying $\boldsymbol{v} - \boldsymbol{v}'$ by y, we also know that

$$\begin{aligned} y \cdot (\boldsymbol{v} - \boldsymbol{v}') &= \left(\sum_{j=0}^{k_1-1} 2^j (y f_j^{(1)} - y f_j'^{(1)}), \ldots, \sum_{j=0}^{k_\psi-1} 2^j (y f_j^{(\psi)} - y f_j'^{(\psi)}) \right) \\ &= \left(\sum_{j=0}^{k_1-1} 2^j (x - x') \hat{b}_j^{(i)}, \ldots, \sum_{j=0}^{k_\psi-1} 2^j (x - x') \hat{b}_j^{(\psi)} \right) \quad \text{(by (5.18) and (5.20))}, \\ &= \left(\sum_{j=0}^{k_1-1} 2^j y^2 b_j^{(i)}, \ldots, \sum_{j=0}^{k_\psi-1} 2^j y^2 b_j^{(\psi)} \right) \qquad \text{(by (5.24))}. \end{aligned}$$

Let us focus on a coordinate $y\hat{\ell}^{(i)}$ of $y(\boldsymbol{v} - \boldsymbol{v}')$ for any $1 \le i \le \psi$. Since $y = \langle y, \ldots, y \rangle$ and it is invertible in all $R_q^{(j)}$'s, it is invertible in R_q. Then, we have

$$y\hat{\ell}^{(i)} = y^2 \underbrace{\sum_{j=0}^{k_i-1} 2^j b_j^{(i)}}_{\in [0, N_i - 1]} \quad \Longrightarrow \quad \hat{\ell}^{(i)} = y\ell^{(i)} \text{ for some } \ell^{(i)} \in [0, N_i - 1]. \tag{5.25}$$

As a result, $yV = \mathrm{Com}_{ck}\left(y\ell^{(1)}, \ldots, y\ell^{(\psi)}; \boldsymbol{z} - \boldsymbol{z}'\right)$ where $\ell^{(1)}, \ldots \ell^{(\psi)}$ are in the ranges $[0, N_1), \ldots, [0, N_\psi)$, respectively. Note that since $q > \max\{N_1, \ldots, N_\psi\}$ there is no modular reduction performed when computing $\sum_{j=0}^{k_i-1} 2^j b_j^{(i)}$. □

Extension to arbitrary ranges.

We assumed that a range is of the form $[0, N)$ for $N = 2^k$. Our range proof can be extended to work for arbitrary ranges using standard techniques as follows. For simplicity, let us assume that $\psi = 1$, i.e., V is a commitment to a single value ℓ. Our discussion easily generalises to the case of committing to a set of values. Suppose that we want to prove $\ell \in [a, b)$ for $b > a+1$ and $a, b \in \mathbb{Z}$. First, using $V' = V - \mathrm{Com}_{ck}(a; 0)$ in the protocol proves that $\ell - a \in [0, N)$, i.e., $\ell \in [a, N + a)$ (this implies that we can "shift" the range in any suitable way). Now, if $b - a$ can be set so that $b - a = N = 2^k$, then we are done. Otherwise, we set $2^k = N > b - a$, and run another range proof for $V'' = \mathrm{Com}_{ck}(b; 0) - V$. This proves that $b - \ell \in [0, N)$, i.e., $\ell \in [b - N, b)$. As a result, ℓ must be in the intersection of $[a, N + a)$ and $[b - N, b)$, i.e., $\ell \in [a, b)$. Note that the proved relations are relaxed as in $\mathcal{R}'_{\text{range}}$, but they indeed work in this sense. Suppose that we have a range proof for $V_1 = V - \mathrm{Com}_{ck}(a; 0)$ for the range $[0, N_1)$ and another range proof for $V_2 = V - \mathrm{Com}_{ck}(b; 0)$ for the range $[0, N_2)$. Then, these

prove knowledge of $(\bar{x}_1, \ell_1, \hat{\boldsymbol{r}}_1)$ and $(\bar{x}_2, \ell_2, \hat{\boldsymbol{r}}_2)$ such that

$$\begin{aligned}\mathrm{Com}_{ck}\left(\bar{x}_1\ell_1;\, \hat{\boldsymbol{r}}_1\right) &= \bar{x}_1 V_1 = \bar{x}_1(V - \mathrm{Com}_{ck}\left(a;\, 0\right)) \;\wedge\; \ell_1 \in [0, N_1),\\ \mathrm{Com}_{ck}\left(\bar{x}_2\ell_2;\, \hat{\boldsymbol{r}}_2\right) &= \bar{x}_2 V_2 = \bar{x}_2(V - \mathrm{Com}_{ck}\left(b;\, 0\right)) \;\wedge\; \ell_2 \in [0, N_2).\end{aligned}$$

Therefore, we have

$$\begin{aligned}\bar{x}_1 V = \mathrm{Com}_{ck}\left(\bar{x}_1(\ell_1 + a);\, \hat{\boldsymbol{r}}_1\right) \quad &\Longrightarrow \quad \bar{x}_2\bar{x}_1 V = \mathrm{Com}_{ck}\left(\bar{x}_2\bar{x}_1(\ell_1 + a);\, \hat{\boldsymbol{r}}_1\right),\\ \bar{x}_2 V = \mathrm{Com}_{ck}\left(\bar{x}_2(\ell_2 + b);\, \hat{\boldsymbol{r}}_2\right) \quad &\Longrightarrow \quad \bar{x}_1\bar{x}_2 V = \mathrm{Com}_{ck}\left(\bar{x}_1\bar{x}_2(\ell_2 + b);\, \hat{\boldsymbol{r}}_2\right).\end{aligned}$$

By the binding property on $\bar{x}_1\bar{x}_2 V$, the committed messages $\bar{x}_2\bar{x}_1(\ell_1+a)$ and $\bar{x}_1\bar{x}_2(\ell_2+b)$ must be the same. That is, $\bar{x}_2\bar{x}_1(\ell_1 + a) = \bar{x}_1\bar{x}_2(\ell_2 + b)$, which implies that $\ell := \ell_1 + a = \ell_2 + b$ since $\bar{x}_1, \bar{x}_2 \in \Delta\mathcal{C}^{d/s}_{w,p}$ and thus are invertible when R_q splits into exactly s fields. Hence, $\ell \in [a, N_1 + a) \cap [b, N_2 + b)$, which concludes that the combination of the proofs behave in an expected manner.

Practical aspects of the range proof.

Let $N = \beta^k$, and assume we want to prove knowledge of an opening ℓ of V such that $V = \mathrm{Com}(\ell)$ and $\ell \in [0, N)$. The generic way for such a range proof works as follows. The prover publishes the commitments $\mathrm{Com}(\ell_j)$ to the digit ℓ_j's of ℓ, and proves that each digit is in $\{0, \ldots, \beta - 1\}$, namely a *set membership proof*. The last step is then to use the homomorphic properties of the commitment to check that these digits construct V, i.e., $\mathrm{Com}(\ell) = \sum_j \beta^j \mathrm{Com}(\ell_j)$. Such a proof involves sending at least k commitments and k masked randomnesses. The β value needs to be small, otherwise the set membership proof becomes cumbersome (especially in the case of lattice-based proofs), and in general $\beta = 2$ is set and thus $k = \log N$. Doing a range proof for ψ values, at best, would multiply the number of commitments by ψ. Therefore, the overall cost would be proportional to at least $\psi \log N nd \log q$ bits since a commitment is of size at least $nd \log q$ bits.

In our proof, on the other hand, the number of commitments and randomnesses communicated is constant. More precisely, always 2 commitments[10] and 3 randomnesses are sent, but their dimensions may vary. In total, the range proof length is $(2(n + v)d \log q + \psi \log N(d/s)B_f + 3mdB_z)$ bits where $B_f, B_z < \log q$ are the bit-lengths of $f^{(i)}_j$'s and $\boldsymbol{z}$'s, respectively. We have $v \approx \psi$ and $m \approx 2n + \psi$. Therefore, the overall proof length growth is slower in comparison to the generic approach. In Table 5.2, we provide a comparison of our CRT-packed range proof with an idealised scheme and one that uses the "norm-optimal" challenges with infinity norm 1 [LS18]. We can see easily that our range proof provides much better computational efficiency *without* any significant compromise in communication size. We also have the invertibility of the challenge differences in $\Delta\mathcal{C}^{d/s}_{w,p}$. In Tables 5.3, 5.4 and 5.5, we provide the full parameter setting of the compared range proof methods.

[10] Note that this happens in the non-interactive case where 5 commitments reduce to 2 commitments. It is standard to exclude from the proof output the commitments (A, D, E in our case) that are uniquely determined by the remaining components.

TABLE 5.2: Comparison of non-interactive range proof sizes (in KB). "Ideal w/o CRT" is a hypothetical scheme optimised for proof length. FFT denotes the maximum number of FFT levels supported. Our proof sizes can be slightly reduced at the cost of reducing the FFT levels.

range width (N)	$N = 2^{32}$				$N = 2^{64}$			
# of batched proofs (ψ)	1	5	10	(d, FFT)	1	5	10	(d, FFT)
with "norm-optimal" challenges from [LS18]	161	745	1484	(256, 1)	443	2131	4274	(256, 2)
Ideal w/o CRT	52	113	180	(32, 5)	86	201	302	(16, 4)
Our Work: CRT-packed	58	130	202	(512, 5)	93	216	319	(512, 6)

TABLE 5.3: The parameter setting of our range proof on $[0, 2^{\log N} - 1]$ with CRT-packing for 128-bit security. $\mathcal{B} = 1$ for all cases. The root Hermite factor for LWE varies in between 1.00399 and 1.00493, and for SIS is ≈ 1.0035. M-SIS and M-LWE dimension parameters are nd and $(m - n - v)d$, respectively, for Tables 5.3, 5.4 and 5.5. Here it is harder to balance the security as the dimension parameters increase as multiples of 512.

$\log N$		32	32	32	64	64	64
ψ		1	5	10	1	5	10
Range Proof Size (KB)		58	130	202	93	216	319
module rank for M-SIS	n	2	3	3	2	4	4
com. randomness vector dimension	m	7	12	17	9	14	19
poly. ring dimension	d	512	512	512	512	512	512
num. of slots in CRT	s	16	32	32	32	64	64
com. message vector dimension	v	2	5	10	2	5	10
modulus	q	$\approx 2^{43}$	$\approx 2^{46}$	$\approx 2^{47}$	$\approx 2^{67}$	$\approx 2^{66}$	$\approx 2^{67}$
Hamming weight of challenges	w	32	16	16	16	8	8
max. abs. coefficient of challenges	p	128	32768	32768	32768	2^{31}	2^{31}

TABLE 5.4: The parameter setting of "Ideal w/o CRT" range proof on $[0, 2^{\log N} - 1]$ for 128-bit security. $\mathcal{B} = 1$ and $v = \psi \log N$ for all cases. The root Hermite factor for SIS and LWE are ≈ 1.0045. There is no additional assumption on the parameters to make sure that the binary proof works. The only assumption is $q \geq 2^{\log N}$ and the parameters are set to make UMC hiding and γ_{range}-binding.

$\log N$	32	32	32	64	64	64
ψ	1	5	10	1	5	10
Range Proof Size (KB)	52	113	180	86	201	302
n	34	89	92	52	210	213
m	107	345	508	275	847	1170
d	32	16	16	16	8	8
q	$\approx 2^{32}$	$\approx 2^{38}$	$\approx 2^{38}$	$\approx 2^{64}$	$\approx 2^{64}$	$\approx 2^{64}$
w	32	16	16	16	8	8
p	128	32768	32768	32768	2^{31}	2^{31}

TABLE 5.5: The parameter setting of range proof on $[0, 2^{\log N} - 1]$ using "norm-optimal" challenges with infinity norm 1 for 128-bit security. $\mathcal{B} = 1$ and $v = \psi \log N$ for all cases. The root Hermite factor for LWE is ≈ 1.0045. For SIS, the root Hermite factor is in between 1.0030 and 1.0045. When the invertibility results of [LS18] are used, q may actually need to be larger to ensure that $\hat{b}(y - \hat{b}) = 0$ in R_q implies $\hat{b} \in \{0, y\}$. But, we ignore this in favor of the method.

$\log N$	32	32	32	64	64	64
ψ	1	5	10	1	5	10
Range Proof Size (KB)	161	745	1484	443	2131	4274
n	3	4	2	2	4	3
m	40	169	332	76	332	653
d	256	256	256	256	256	256
q	$\approx 2^{32}$	$\approx 2^{32}$	$\approx 2^{32}$	$\approx 2^{64}$	$\approx 2^{64}$	$\approx 2^{64}$
w	60	60	60	60	60	60
p	1	1	1	1	1	1

5.5 Efficient One-Shot Proofs for Other Useful Relations

In this section, we apply our one-shot proof techniques to construct efficient ZKPs for useful relations such as a binary proof, a one-out-of-many proof and a set membership proof. We instantiate the commitment scheme with HMC in HNF in this section.

5.5.1 Relaxed proof of commitment to sequences of bits

Using our new techniques, we extend the multi-shot proof of commitment to bits from Section 4.2.1 to a one-shot proof. The new protocol proves a weaker relation but, the relaxation is tailored in a way that the soundness proof of the higher level proofs (Protocol 5.4) still work. The protocol proves that a commitment B opens to sequences of binary values such that there is a single 1 in each sequence, i.e., Hamming weight of each sequence is exactly 1. The relations of our one-shot binary proof are defined in Definition 5.9 where $\boldsymbol{b} = (b_{0,0}, \ldots, b_{k-1,\beta-1})$ for $k \geq 1, \beta \geq 2$.

Definition 5.9. *The following defines the relations for Protocol 5.3 for $\mathcal{T}, \hat{\mathcal{T}} \in \mathbb{R}^+$.*

$$\mathcal{R}_{\text{bin}}(\mathcal{T}) = \left\{ \begin{array}{c} ((ck, B), (\boldsymbol{b}, \boldsymbol{r})) \ : \ \|\boldsymbol{r}\| \leq \mathcal{T} \ \wedge \ (b_{j,i} \in \{0,1\} \ \forall j, i) \\ \wedge \ B = \text{Com}_{ck}(\boldsymbol{b}; \boldsymbol{r}) \ \wedge \ (\sum_{i=0}^{\beta-1} b_{j,i} = 1 \ \forall j) \end{array} \right\}.$$

$$\mathcal{R}'_{\text{bin}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{c} ((ck, B), (y, \boldsymbol{b}, \hat{\boldsymbol{r}})) : \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \ \wedge \ (b_{j,i} \in \{0,1\} \ \forall j, i) \ \wedge \\ y \in \Delta\mathcal{C}^d_{w,p} \ \wedge \ yB = \text{Com}_{ck}(y\boldsymbol{b}; \hat{\boldsymbol{r}}) \ \wedge \ (\sum_{i=0}^{\beta-1} b_{j,i} = 1 \ \forall j) \end{array} \right\}.$$

The idea of the binary proof (combined with the CRT-packing technique) is already used in Protocol 5.2. The condition on the Hamming weight is the difference to Protocol 5.2 and is handled with a small modification. We describe the full protocol in Protocol 5.3 where the commitment scheme is instantiated with HMC in HNF and also summarise the results below. As before, the norms of some components in the protocol are bounded separately in Lemmas 5.11 and 5.12.

Theorem 5.10. *Let $\gamma_{\text{bin}} = 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^2 w^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}$. Assume that $d \geq 128$, $q/2 > 2^7 \phi_1^2 p^5 w^3 d^2 k\beta$ and HMC in HNF is hiding and γ_{bin}-binding. Then, Protocol 5.3 is a 3-special sound Σ-protocol (as in Definition 3.6) for the relations $\mathcal{R}_{\text{bin}}(\mathcal{B}\sqrt{md})$ and $\mathcal{R}'_{\text{bin}}(4\sqrt{2}\phi_2 pw\mathcal{B}md)$ with a completeness error $1 - \frac{1}{\mu(\phi_1)\mu(\phi_2)}$ for $\mu(\cdot)$ defined in Lemma 3.16.*

$\mathcal{P}_{\text{bin}}((ck, B), (\boldsymbol{b}; \boldsymbol{r}))$		$\mathcal{V}_{\text{bin}}(ck, B)$
$a_{0,1}, \ldots, a_{k-1,\beta-1} \leftarrow D^d_{\phi_1 T_1}$ for $T_1 = p\sqrt{kw}$		
$\boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$		
$\boldsymbol{r}_a, \boldsymbol{r}_d \leftarrow D^{md}_{\phi_2 T_2}$ for $T_2 = pw\mathcal{B}\sqrt{2md}$		
for $j = 0, \ldots, k-1$ **do**		
$a_{j,0} = -\sum_{i=1}^{\beta-1} a_{j,i}$		
$A = \text{Com}_{ck}\left(\{a_{j,i}\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}_a\right)$		
$C = \text{Com}_{ck}\left(\{a_{j,i}(1-2b_{j,i})\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}_c\right)$		
$D = \text{Com}_{ck}\left(\{-(a_{j,i})^2\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}_d\right)$		
	$\xrightarrow{A, C, D}$	
	$\xleftarrow{x}$	$x \leftarrow \mathcal{C}^d_{w,p}$
for $j \in [0, k), i \in [1, \beta)$ **do**		
$f_{j,i} = x \cdot b_{j,i} + a_{j,i}$		
$\boldsymbol{f}_1 := (f_{0,1}, \ldots, f_{k-1,\beta-1})$		
$\boldsymbol{b}_1 := (b_{0,1}, \ldots, b_{k-1,\beta-1})$		
$\text{Rej}(\boldsymbol{f}_1, x\boldsymbol{b}_1, \phi_1, T_1)$		
$\boldsymbol{z}_b = x \cdot \boldsymbol{r} + \boldsymbol{r}_a$		
$\boldsymbol{z}_c = x \cdot \boldsymbol{r}_c + \boldsymbol{r}_d$		
$\text{Rej}((\boldsymbol{z}_b, \boldsymbol{z}_c), x(\boldsymbol{r}, \boldsymbol{r}_c), \phi_2, T_2)$		
If aborted, return $\perp$.	$\xrightarrow{\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c}$	
		for $j = 0, \ldots, k-1$ **do**
		$f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$
		$\|f_{j,i}\| \overset{?}{\leq} 2\phi_1 T_1 \sqrt{d} \ \ \forall j, \forall i \neq 0$
		$\|f_{j,0}\| \overset{?}{\leq} 2\phi_1 T_1 \sqrt{\beta d} \ \ \forall j$
		$\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \overset{?}{\leq} 2\phi_2 T_2 \sqrt{md}$
		$\boldsymbol{f} := \{f_{j,i}\}_{j,i=0}^{k-1,\beta-1}$
		$\boldsymbol{g} := \{f_{j,i}(x - f_{j,i})\}_{j,i=0}^{k-1,\beta-1}$
		$xB + A \overset{?}{=} \text{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z}_b)$
		$xC + D \overset{?}{=} \text{Com}_{ck}(\boldsymbol{g}; \boldsymbol{z}_c)$

PROTOCOL 5.3: Σ-protocol for $\mathcal{R}_{\text{bin}}$ and $\mathcal{R}'_{\text{bin}}$.

Proof. **Completeness:** The main difference from Protocol 5.2 is that there is a sum of $f_{j,i}$'s, which follow a normal distribution. By the discussion in Section 3.3.3, the sum of discrete normal variables behaves as its continuous counterpart. That is, the distribution of $\sum_{i=1}^{\beta-1} f_{j,i}$ is statistically close to $D^d_{\phi_1 T_1 \sqrt{\beta-1}}$. Hence, we have

$$\begin{aligned}
&\|f_{j,i}\| \le 2 \cdot \phi_1 p\sqrt{kw} \cdot \sqrt{d} = 2\phi_1 p\sqrt{dkw}, \qquad \forall j \in [0,k),\ \forall i \in [0,\beta),\ \text{and} \\
&\|f_{j,0}\| = \left\| x - \sum_{i=1}^{\beta-1} f_{j,i} \right\| \le \|x\| + \left\| \sum_{i=1}^{\beta-1} f_{j,i} \right\| \\
&\qquad \le \sqrt{w} + 2 \cdot \phi_1 p\sqrt{kw(\beta-1)} \cdot \sqrt{d} \approx 2\phi_1 p\sqrt{\beta dkw}.
\end{aligned}$$

The rest is analogous to the completeness proof of Protocol 5.2.
SHVZK: Assume the protocol is not aborted. The simulator sets $C = \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{r}_c)$ for $\boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$, and picks $f_{j,i} \leftarrow D^d_{\phi_1 T_1}$ for all $0 \le j \le k-1$ and $1 \le i \le \beta - 1$, and also $\boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow D^{md}_{\phi_2 T_2}$. Then, given x, it sets $f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$ for all $j = 0, \ldots, k-1$. Finally, it computes $A = \mathrm{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z}_b) - xB$ and $D = \mathrm{Com}_{ck}(\boldsymbol{g}; \boldsymbol{z}_c) - xC$ where $\boldsymbol{f}, \boldsymbol{g}$ are set as in Protocol 5.3. It outputs the simulated transcript $((A, C, D), x, (\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c))$ where $\boldsymbol{f}_1$ is set as in Protocol 5.3. The indistinguishability argument is as in SHVZK of Theorem 5.8.
3-special soundness: The proof proceeds almost identical to the soundness proof of Theorem 5.8 up to Equation (5.23) except that when the commitment scheme is instantiated with HMC in HNF, we need to bound the norm of the message-randomness opening pair together to use the binding property argument. The exact opening of yD has the largest norm-bound of

$$\gamma_{\text{bin}} = 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^2 w^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}$$

as shown in Lemma 5.12. As in (5.18), (5.19), and (5.20), we have the following equations, but now holding in R_q

$$\begin{aligned}
y f_{j,i} &= x\hat{b}_{j,i} + \hat{a}_{j,i}, & y f_{j,i}(x - f_{j,i}) &= x\hat{c}_{j,i} + \hat{d}_{j,i}, \\
y f'_{j,i} &= x'\hat{b}_{j,i} + \hat{a}_{j,i}, & y f'_{j,i}(x' - f'_{j,i}) &= x'\hat{c}_{j,i} + \hat{d}_{j,i}, \quad \text{in } R_q. \\
y f''_{j,i} &= x''\hat{b}_{j,i} + \hat{a}_{j,i}, & y f''_{j,i}(x'' - f''_{j,i}) &= x''\hat{c}_{j,i} + \hat{d}_{j,i},
\end{aligned} \tag{5.26}$$

Then, by the γ_{bin}-binding property of HMC in HNF, we get below the same system of equations as in (5.23) of the soundness proof of Theorem 5.8

$$\begin{pmatrix} 1 & x & x^2 \\ 1 & x' & x'^2 \\ 1 & x'' & x''^2 \end{pmatrix} \cdot \begin{pmatrix} -(\hat{a}_{j,i})^2 - y\hat{d}_{j,i} \\ \hat{a}_{j,i}(y - 2\hat{b}_{j,i}) - y\hat{c}_{j,i} \\ \hat{b}_{j,i}(y - \hat{b}_{j,i}) \end{pmatrix} = \mathbf{0} \qquad \text{in } R_q.$$

Note that again all equations now hold in R_q, and there is no use of any invertibility argument. Now, multiplying both sides of the above system of equations by $\mathrm{adj}(\boldsymbol{V})$ where $\boldsymbol{V}$ is the Vandermonde matrix on the left, we get

$$\det(\boldsymbol{V})\hat{b}_{j,i}(y - \hat{b}_{j,i}) = (x'' - x')(x' - x)(x'' - x)\hat{b}_{j,i}(y - \hat{b}_{j,i}) = 0 \quad \text{in } R_q. \tag{5.27}$$

We also know that

$$\left\|\hat{b}_{j,i}\right\|_1 = \left\|f_{j,i} - f'_{j,i}\right\|_1 \le 2 \cdot \sqrt{d} \cdot 2\phi_1 p\sqrt{\beta dkw} = 4\phi_1 pd\sqrt{\beta kw}, \text{ and}$$

$$\left\|y-\hat{b}_{j,i}\right\|_1 \leq \|y\|_1 + \left\|\hat{b}_{j,i}\right\|_1 \leq pw + 4\phi_1 pd\sqrt{\beta kw} \approx 4\phi_1 pd\sqrt{\beta kw}.$$

From here, we can further get a bound as

$$\begin{aligned}\left\|(x''-x')(x'-x)(x''-x)\hat{b}_{j,i}(y-\hat{b}_{j,i})\right\|_\infty \leq \\ \left\|x''-x'\right\|_\infty \cdot \left\|x'-x\right\|_1 \cdot \left\|x''-x\right\|_1 \cdot \left\|\hat{b}_{j,i}\right\|_1 \cdot \left\|y-\hat{b}_{j,i}\right\|_1 \\ \leq 2p\cdot(2pw)^2\cdot(4\phi_1 pd\sqrt{\beta kw})^2 \\ = 2^7\phi_1^2 p^5 w^3 d^2 k\beta. \qquad (5.28)\end{aligned}$$

Since $q/2 > 2^7\phi_1^2 p^5 w^3 d^2 k\beta$, one of the factors in (5.27) must be zero by Lemma 5.5. As challenge differences are non-zero, this gives either $\hat{b}_{j,i}$ or $y-\hat{b}_{j,i}$ is zero. Thus, we get $\hat{b}_{j,i} \in \{0, y\}$. That is, $\hat{b}_{j,i} = yb_{j,i}$ for $b_{j,i} \in \{0,1\}$ as needed for $\mathcal{R}'_{\text{bin}}$.

Finally, for all $j = 0, \ldots, k-1$, by multiplying Step 2 of the verification by y, we have the following

$$yx = \sum_{i=0}^{\beta-1} yf_{j,i} = \sum_{i=0}^{\beta-1} x\hat{b}_{j,i} + \sum_{i=0}^{\beta-1}\hat{a}_{j,i} = yx\cdot\sum_{i=0}^{\beta-1} b_{j,i} + \sum_{i=0}^{\beta-1}\hat{a}_{j,i}.$$

This holds for 2 distinct challenges x and x', and therefore

$$\left(\sum_{i=0}^{\beta-1} b_{j,i} - 1\right) y(x-x') = \left(\sum_{i=0}^{\beta-1} b_{j,i} - 1\right) y^2 = 0 \quad \text{in } R_q.$$

Using Lemma 5.5 as above (the condition on the size of q here is much weaker), we get $\sum_{i=0}^{\beta-1} b_{j,i} = 1$ for all $0 \leq j \leq k-1$ as required. □

Lemma 5.11. *The vector $\boldsymbol{g}$ defined in Protocol 5.3 satisfies the following*

$$\|\boldsymbol{g}\|^2 \leq 16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1).$$

Proof. We use the bounds on the norm of $f_{j,i}$'s in the sequel (see Protocol 5.3 definition). For simplicity, we bound $\|x - f_{j,0}\|$ by the bound on $\|f_{j,0}\|$ as $\|x\|$ is much smaller in comparison.

$$\begin{aligned}\|\boldsymbol{g}\|^2 &= \sum_{j=0}^{k-1}\sum_{i=0}^{\beta-1}\|f_{j,i}(x-f_{j,i})\|^2 = \sum_{j=0}^{k-1}\sum_{i=1}^{\beta-1}\|f_{j,i}(x-f_{j,i})\|^2 + \sum_{j=0}^{k-1}\|f_{j,0}(x-f_{j,0})\|^2 \\ &\leq \sum_{j=0}^{k-1}\sum_{i=1}^{\beta-1} d\,\|f_{j,i}\|^2\,\|x-f_{j,i}\|^2 + \sum_{j=0}^{k-1} d\,\|f_{j,0}\|^2\,\|x-f_{j,0}\|^2 \\ &\leq dk(\beta-1)\left(2\phi_1 p\sqrt{dkw}\right)^4 + dk\left(2\phi_1 p\sqrt{\beta dkw}\right)^4 \\ &\leq dk\left(2\phi_1 p\sqrt{dkw}\right)^4\cdot\left[(\beta-1)+\beta^2\right] \leq 16\phi_1^4 p^4 d^3 k^3 w^2\beta(\beta+1).\end{aligned}$$

□

Lemma 5.12. *The exact opening $(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_{\boldsymbol{d}})$ of yD in the soundness proof of Theorem 5.10 satisfies the following*

$$\left\|(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_{\boldsymbol{d}})\right\| \leq 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^2 w^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}.$$

Proof. For $y = x - x'$, we have

$$yC = \mathrm{Com}_{ck}\left(\boldsymbol{g} - \boldsymbol{g}'; \boldsymbol{z}_c - \boldsymbol{z}'_c\right). \tag{5.29}$$

Then, the exact opening $(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_{\boldsymbol{d}})$ of yD is obtained as follows

$$\begin{aligned} yD &= \mathrm{Com}_{ck}\left(y\boldsymbol{g}; y\boldsymbol{z}_c\right) - xyC = \mathrm{Com}_{ck}\left(y\boldsymbol{g}; y\boldsymbol{z}_c\right) - x\mathrm{Com}_{ck}\left(\boldsymbol{g} - \boldsymbol{g}'; \boldsymbol{z}_c - \boldsymbol{z}'_c\right) \\ &= \mathrm{Com}_{ck}\left(x\boldsymbol{g} - x'\boldsymbol{g}; x\boldsymbol{z}_c - x'\boldsymbol{z}_c\right) - \mathrm{Com}_{ck}\left(x\boldsymbol{g} - x\boldsymbol{g}'; x\boldsymbol{z}_c - x\boldsymbol{z}'_c\right) \\ &= \mathrm{Com}_{ck}\left(x\boldsymbol{g}' - x'\boldsymbol{g}; x\boldsymbol{z}'_c - x'\boldsymbol{z}_c\right). \end{aligned} \tag{5.30}$$

Without loss of generality, assume that $\|(x\boldsymbol{g}', x\boldsymbol{z}'_c)\| \geq \|(x'\boldsymbol{g}, x'\boldsymbol{z}_c)\|$.

$$\begin{aligned} &\left\|(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_{\boldsymbol{d}})\right\| = \left\|(x\boldsymbol{g}' - x'\boldsymbol{g}, x\boldsymbol{z}'_c - x'\boldsymbol{z}_c)\right\| \leq 2\left\|(x\boldsymbol{g}', x\boldsymbol{z}'_c)\right\| \leq 2\sqrt{d}\,\|x\| \cdot \left\|(\boldsymbol{g}', \boldsymbol{z}'_c)\right\| \\ &\leq 2p\sqrt{dw} \cdot \left\|(\boldsymbol{g}', \boldsymbol{z}'_c)\right\| = 2p\sqrt{dw}\left(\left\|\boldsymbol{g}'\right\|^2 + \left\|\boldsymbol{z}'_c\right\|^2\right)^{1/2} \\ &\leq 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + \left(2\sqrt{3}\phi_2 pw\mathcal{B}md\right)^2\right)^{1/2} \quad \text{(by Lemma 5.11)} \\ &= 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^2 w^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}. \end{aligned} \tag{5.31}$$

□

Remark 5.13. *Note that, when bounding $\|\boldsymbol{z}_c\|$ in the proof of Lemma 5.12, we use the norm bound in Protocol 5.2's verification, which is a stronger case than that in Protocol 5.3. The norm bound of $\boldsymbol{z}_c$ in Protocol 5.4 to be described is the largest one, and (5.31) becomes $2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^{2k} \mathcal{B}^2 m^2 d^2 w^{2k}\right)^{1/2}$ using that bound. We consider the strongest bound when instantiating the parameters for the ring signature in this chapter, which builds on Protocol 5.4.*

5.5.2 Relaxed one-out-of-many proof

Our one-shot one-out-of-many proof has the same structure as the one-out-of-many proof in Section 4.2.2. The main differences of the one-shot proof are the use of an exponentially large challenge set, the relation the verifier is convinced of and some tweaks to the rejection sampling. The challenging part here is the soundness proof of the protocol. We use our new tools, namely Lemmas 5.1, 5.3 and 5.4, from Section 5.3 to make the soundness proof work.

Let $\boldsymbol{L} = \{P_0, \ldots, P_{N-1}\}$ be a set of public commitments for some $N \geq 1$. The prover's goal is to show that he knows an opening of one of these P_i's. As before, we assume that $N = \beta^k$, which can be easily satisfied by adding dummy values to $\boldsymbol{L}$ when needed. Suppose that the prover's commitment is P_ℓ for some $0 \leq \ell < N$. Observe that $\sum_{i=0}^{N-1} \delta_{\ell,i} P_i = P_\ell$. The idea for the proof is then to prove knowledge of the index ℓ with $\sum_{i=0}^{N-1} \delta_{\ell,i} P_i$ being a commitment to zero. Writing $\ell = (\ell_0, \ldots, \ell_{k-1})$ and $i = (i_0, \ldots, i_{k-1})$ as the representations in base β, we have $\delta_{\ell,i} = \prod_{j=0}^{k-1} \delta_{\ell_j, i_j}$. The prover first commits to the sequences $(\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,\beta-1})$ for all $0 \leq j \leq k-1$, and then

uses Protocol 5.3 to show that they are well-formed (i.e., they construct an index in the range $[0, N)$ as in the range proof). Let us define the proved relations next.

Definition 5.14. *The following defines the relations for Protocol 5.4 for* $\mathcal{T}, \hat{\mathcal{T}} \in \mathbb{R}^+$.

$$\mathcal{R}_{1/\mathrm{N}}(\mathcal{T}) = \left\{ \begin{array}{c} ((ck, (P_0, \ldots, P_{N-1})), (\ell, \boldsymbol{r})) \ : \\ \ell \in [0, N) \ \wedge \ \|\boldsymbol{r}\| \leq \mathcal{T} \ \wedge \ P_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{r}) \end{array} \right\},$$

$$\mathcal{R}'_{1/\mathrm{N}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{c} ((ck, (P_0, \ldots, P_{N-1})), (y, \ell, \hat{\boldsymbol{r}})) \ : \ \ell \in [0, N) \ \wedge \ \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \wedge \\ yP_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; \hat{\boldsymbol{r}}) \ \wedge \ y \text{ is a product of elements in } \Delta\mathcal{C}^d_{w,p} \end{array} \right\}.$$

From Protocol 5.3, the prover's response contains $f_{j,i} = x\delta_{\ell_j,i} + a_{j,i}$ for a challenge x. Considering the product $p_i(x) := \prod_{j=0}^{k-1} f_{j,i_j}$, we see that, for all $i \in [0, N-1]$,

$$p_i(x) = \prod_{j=0}^{k-1} \left(x\delta_{\ell_j,i_j} + a_{j,i_j}\right) = \prod_{j=0}^{k-1} x \cdot \delta_{\ell_j,i_j} + \sum_{j=0}^{k-1} p_{i,j}x^j = \delta_{\ell,i}x^k + \sum_{j=0}^{k-1} p_{i,j}x^j, \quad (5.32)$$

for some ring element $p_{i,j}$'s as a function of ℓ and $a_{j,i}$'s (independent of the challenge x). Since ℓ and $a_{j,i}$'s are known to the prover before receiving a challenge, he can compute $p_{i,j}$'s prior to sending the initial commitment. Since p_ℓ is the only such polynomial of degree k, in his first move, the prover sends some E_j's that are tailored to cancel out the coefficients of the terms $1, x, \ldots, x^{k-1}$, and the coefficient of x^k is set to the prover's commitment P_ℓ using $\sum_{i=0}^{N-1} \delta_{\ell,i}P_i$. The full description is given in Protocol 5.4.

Theorem 5.15. *Let* $\gamma_{1/\mathrm{N}} = (k+1)2^{\kappa'+2}\sqrt{3}\phi_2\mathcal{B}md^2w^\kappa p^{\kappa+1}$ *for* $\kappa' = k(k-1)/2$ *and* $\kappa = k(k+1)/2$. *Assume* $d \geq 128$, $q > 2^7\phi_1^2p^5w^3d^2k\beta$ *and HMC in HNF is hiding and* γ*-binding for* $\gamma = \max\{\gamma_{\mathrm{bin}}, \gamma_{1/\mathrm{N}}\}$. *For* $\mu(\cdot)$ *defined in Lemma 3.16, Protocol 5.4 is a* $(k'+1)$*-special sound* Σ*-protocol (as in Definition 3.6) for the relations* $\mathcal{R}_{1/\mathrm{N}}(\mathcal{B}\sqrt{md})$ *and* $\mathcal{R}'_{1/\mathrm{N}}(\gamma_{1/\mathrm{N}})$ *with a completeness error* $1 - 1/(\mu(\phi_1)\mu(\phi_2))$ *where* $k' = \max\{2, k\}$.

Proof. **Completeness:** Step 1 of verification follows from the completeness of Protocol 5.3. For bounding the maximum norm of masked randomnesses in Step 11 of prover's computation, we have

$$\begin{aligned} \left\| x^k\boldsymbol{r} - \sum_{j=1}^{k-1} x^j\boldsymbol{\rho}_j \right\| &\leq \left\|x^k\boldsymbol{r}\right\| + \sum_{j=1}^{k-1} \left\|x^j\boldsymbol{\rho}_j\right\| \leq \sqrt{md} \cdot \left(\left\|x^k\boldsymbol{r}\right\|_\infty + \sum_{j=1}^{k-1} \left\|x^j\boldsymbol{\rho}_j\right\|_\infty \right) \\ &\leq \sqrt{md} \cdot \left(\|x\|_1^k \|\boldsymbol{r}\|_\infty + \sum_{j=1}^{k-1} \|x\|_1^j \left\|\boldsymbol{\rho}_j\right\|_\infty \right) \\ &\leq \sqrt{md} \cdot \left((pw)^k\mathcal{B} + \sum_{j=1}^{k-1} (pw)^j\mathcal{B} \right) = \mathcal{B}\sqrt{md} \sum_{j=1}^{k} (pw)^j. \end{aligned}$$

Denote $\boldsymbol{r}' = x^k\boldsymbol{r} - \sum_{j=1}^{k-1} x^j\boldsymbol{\rho}_j$. Then, we have

$$\begin{aligned} \left\|(x\boldsymbol{r}_b, x\boldsymbol{r}_c, \boldsymbol{r}')\right\| &= \left(\|(x\boldsymbol{r}_b, x\boldsymbol{r}_c)\|^2 + \left\|\boldsymbol{r}'\right\|^2 \right)^{1/2} \\ &\leq \left(\left(pw\mathcal{B}\sqrt{2md}\right)^2 + \left(\mathcal{B}\sqrt{md} \sum_{j=1}^{k} (pw)^j \right)^2 \right)^{1/2} \end{aligned}$$

$\mathcal{P}_{1/\mathrm{N}}((ck,(P_0,\ldots,P_{N-1})),(\ell,\boldsymbol{r}))$ | $\mathcal{V}_{1/\mathrm{N}}(ck,(P_0,\ldots,P_{N-1}))$

$\boldsymbol{r}_b \leftarrow \{-\mathcal{B},\ldots,\mathcal{B}\}^{md}$

$\boldsymbol{\delta} = (\delta_{\ell_0,0},\ldots,\delta_{\ell_{k-1},\beta-1})$

$B = \mathrm{Com}_{ck}(\boldsymbol{\delta};\boldsymbol{r}_b)$

$A,C,D \leftarrow \mathcal{P}_{\mathrm{bin}}((ck,B),(\boldsymbol{\delta},\boldsymbol{r}_b))$

$\boldsymbol{\rho}_0 \leftarrow D^{md}_{\phi_2 T_2}$ for $T_2 = \mathcal{B}p^k w^k\sqrt{3md}$

for $j = 0,\ldots,k-1$ **do**

  $\boldsymbol{\rho}_j \leftarrow \{-\mathcal{B},\ldots,\mathcal{B}\}^{md}$ if $j \neq 0$

  $E_j = \sum_{i=0}^{N-1} p_{i,j}P_i + \mathrm{Com}_{ck}(\mathbf{0};\boldsymbol{\rho}_j)$

using $p_{i,j}$'s from (5.32)

$\xrightarrow{A,B,C,D,E_0,\ldots,E_{k-1}}$

$\xleftarrow{x}$ $x \leftarrow \mathcal{C}^d_{w,p}$

$\boldsymbol{f}_1,\boldsymbol{z}_b,\boldsymbol{z}_c \leftarrow \mathcal{P}_{\mathrm{bin}}(x)$

$\mathrm{Rej}(\boldsymbol{f}_1, x\boldsymbol{\delta}_1, \phi_1, p\sqrt{kw})$ for $\boldsymbol{\delta}_1 := (\delta_{\ell_0,1},\ldots,\delta_{\ell_{k-1},\beta-1})$

$\boldsymbol{z} = x^k\boldsymbol{r} - \sum_{j=0}^{k-1} x^j\boldsymbol{\rho}_j$

$\mathrm{Rej}((\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z}),(x\boldsymbol{r}_b, x\boldsymbol{r}_c, x^k\boldsymbol{r} - \sum_{j=1}^{k-1} x^j\boldsymbol{\rho}_j),\phi_2,T_2)$

If aborted, return $\perp$.

$\xrightarrow{\boldsymbol{f}_1,\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z}}$

$\mathcal{V}_{\mathrm{bin}}(ck,B,x,A,C,D,\boldsymbol{f}_1,\boldsymbol{z}_b,\boldsymbol{z}_c) \stackrel{?}{=} 1$

$\|\boldsymbol{z}\|, \|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \stackrel{?}{\leq} 2\sqrt{3}\phi_2\mathcal{B}mdp^k w^k$

$\sum_{i=0}^{N-1}\left(\prod_{j=0}^{k-1} f_{j,i_j}\right)P_i - \sum_{j=0}^{k-1} E_j x^j \stackrel{?}{=} \mathrm{Com}_{ck}(\mathbf{0};\boldsymbol{z})$

PROTOCOL 5.4: Σ-protocol for $\mathcal{R}_{1/\mathrm{N}}$ and $\mathcal{R}'_{1/\mathrm{N}}$. Step 5 of the verification (norm checks on $\boldsymbol{z}_b,\boldsymbol{z}_c$) in Protocol 5.3 is skipped when $\mathcal{V}_{\mathrm{bin}}(ck,B,x,A,C,D,\boldsymbol{f}_1,\boldsymbol{z}_b,\boldsymbol{z}_c)$ is run.

$$\leq \left(2w^2p^2\mathcal{B}^2md + \mathcal{B}^2md\left(\sum_{j=1}^{k}(pw)^j\right)^2\right)^{1/2}$$

$$\leq \left(\mathcal{B}^2md\cdot\left[2w^2p^2 + \left(\sum_{j=1}^{k}(pw)^j\right)^2\right]\right)^{1/2} \leq \mathcal{B}w^kp^k\sqrt{3md}.$$

Therefore, for $T_2 = \mathcal{B}w^kp^k\sqrt{3md}$, the distribution of $\boldsymbol{z},\boldsymbol{z}_b,\boldsymbol{z}_c$ are statistically close to $D^{md}_{\phi_2 T_2}$ by Lemma 3.16. Hence, by Lemma 3.15, we have

$$\|\boldsymbol{z}\|, \|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \leq 2\cdot\phi_2\mathcal{B}w^kp^k\sqrt{3md}\cdot\sqrt{md} = 2\sqrt{3}\phi_2\mathcal{B}mdw^kp^k.$$

For the last verification, we have, for $P_\ell = \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{r})$,

$$\begin{aligned}
\sum_{i=0}^{N-1}\left(\prod_{j=0}^{k-1} f_{j,i_j}\right)P_i - \sum_{j=0}^{k-1} E_j x^j &= \sum_{i=0}^{N-1} p_i(x)P_i - \sum_{j=0}^{k-1}\left(\sum_{i=0}^{N-1} p_{i,j}P_i + \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{\rho}_j)\right)x^j \\
&= \sum_{i=0}^{N-1} p_i(x)P_i - \sum_{j=0}^{k-1}\sum_{i=0}^{N-1} p_{i,j}P_i x^j - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{\rho}_j) \\
&= \sum_{i=0}^{N-1} P_i\left(p_i(x) - \sum_{j=0}^{k-1} p_{i,j}x^j\right) - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{\rho}_j) \\
&= \sum_{i=0}^{N-1} P_i \delta_{\ell,i} x^k - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{\rho}_j) = x^k \cdot P_\ell - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{\rho}_j) \\
&= \mathrm{Com}_{ck}\left(\mathbf{0}; x^k \cdot \boldsymbol{r} - \sum_{j=0}^{k-1} x^j \cdot \boldsymbol{\rho}_j\right) = \mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{z}).
\end{aligned}$$

SHVZK: Assume that the protocol is not aborted. $A, B, C, D, \boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}$ are simulated as in the case of Protocol 5.2 where $\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}$ are sampled from $D^{md}_{\phi_2 T_2}$ for $T_2 = \mathcal{B}p^k w^k \sqrt{3md}$. The simulator also samples $E_1, \ldots, E_{k-1} \leftarrow \mathcal{U}(R_q^n)$ and computes E_0 in the way to ensure that the last verification step is satisfied. Then, the simulated transcript is output as below

$$((A, B, C, D, \{E_j\}_{j=0}^{k-1}), x, (\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z})).$$

The simulation of $E_1, \ldots, E_{k-1}$ is computationally indistinguishable from the real case by M-LWE assumption. The rest of the indistinguishability argument is the same as in SHVZK of Protocol 5.2.

$(k'+1)$-special soundness: Assume that $k > 1$. Given $(k+1)$ distinct challenges $x_0, \ldots, x_k$, we have $(k+1)$ accepting responses with same $(A, B, C, D, E_0, \ldots, E_{k-1})$. Suppose that $(\boldsymbol{f}_1^{(0)}, \boldsymbol{z}^{(0)}), \ldots, (\boldsymbol{f}_1^{(k)}, \boldsymbol{z}^{(k)})$ are part of the responses with respect to challenges $x_0, \ldots, x_k$, respectively. Setting $y = x_1 - x_0$, we first use 3-special soundness of Protocol 5.3 to extract exact valid message openings $\hat{b}_{j,i}$ and $\hat{a}_{j,i}$ of yB and yA, respectively. We know that $\hat{b}_{j,i} = y b_{j,i}$ for $b_{j,i} \in \{0, 1\}$ and only a single one of $\{b_{j,0}, \ldots, b_{j,\beta-1}\}$ is 1 for each $j \in \{0, \ldots, k-1\}$. Now, we construct the representation of ℓ in base β as follows. For each $0 \le j \le k-1$, the j-th digit ℓ_j is the integer c such that $b_{j,c} = 1$. It is easy to construct the index ℓ from here using its digit ℓ_j's.

Recalling equations in (5.26) from the soundness proof of Protocol 5.3 that use γ_{bin}-binding property of the commitment scheme, we have, for all $0 \le \eta \le k-1$,

$$y f_{j,i}^{(\eta)} = x_\eta \hat{b}_{j,i} + \hat{a}_{j,i} = x_\eta \cdot y b_{j,i} + \hat{a}_{j,i}.$$

Now compute $\hat{p}_i(x_\eta) = y^k \prod_{j=0}^{k-1} f_{j,i_j}^{(\eta)} = \prod_{j=0}^{k-1} y f_{j,i_j}^{(\eta)} = \prod_{j=0}^{k-1}\left(y x_\eta b_{j,i_j} + \hat{a}_{j,i_j}\right)$ for each $i = 0, \ldots, N-1$. By the construction of ℓ, $\hat{p}_\ell(x_\eta)$ is the only polynomial of degree k in x_η for all $0 \le \eta \le k-1$. Then, we can multiply the both sides of the last verification step by y^k and re-write it as below

$$\sum_{i=0}^{N-1} \hat{p}_i(x_\eta) P_i - \sum_{j=0}^{k-1} y^k E_j x_\eta^j = x_\eta^k \cdot y^k P_\ell + \sum_{j=0}^{k-1} \tilde{E}_j x_\eta^j = \mathrm{Com}_{ck}\left(\mathbf{0}; y^k \boldsymbol{z}^{(\eta)}\right), \qquad (5.33)$$

where $\tilde{E}_j$'s are the terms multiplied by the monomials x_η^j's of degree at most $k-1$ and are independent of x_η. Equation (5.33) is exactly the case described in (5.6) and the verification of Protocol 5.1 in Section 5.3 with $C_k = y^k P_\ell$. By the discussion in Section 5.3, we obtain exact openings of $\det(\boldsymbol{V})y^k P_\ell$ as $(\mathbf{0}, y^k\hat{\boldsymbol{r}})$ where $\hat{\boldsymbol{r}} = \sum_{i=0}^{k} \Gamma_i \boldsymbol{z}^{(i)}$ for $\Gamma_i = (-1)^{i+k}\prod_{0\le l<j\le k\wedge j,l\neq i}(x_j - x_l)$, i.e., we have

$$\begin{aligned}
\det(\boldsymbol{V})y^k P_\ell = \mathrm{Com}_{ck}\left(\mathbf{0}; y^k\hat{\boldsymbol{r}}\right) &\implies y^k\cdot(\det(\boldsymbol{V})P_\ell - \mathrm{Com}_{ck}(\mathbf{0};\hat{\boldsymbol{r}})) = 0\\
\text{(by Lemma 5.4)} &\implies y\cdot(\det(\boldsymbol{V})P_\ell - \mathrm{Com}_{ck}(\mathbf{0};\hat{\boldsymbol{r}})) = 0\\
&\implies \det(\boldsymbol{V})yP_\ell = \mathrm{Com}_{ck}(\mathbf{0}; y\hat{\boldsymbol{r}})\,. \qquad (5.34)
\end{aligned}$$

In the end, we have an exact opening of $\det(\boldsymbol{V})yP_\ell$ as $(\mathbf{0}, y\hat{\boldsymbol{r}})$. This randomness opening is a factor $y \in \Delta\mathcal{C}^d_{w,p}$ larger than what we have in Lemma 5.3. Thus, using Lemmas 5.1 and 5.3, we conclude, for $\kappa' = k(k-1)/2$ and $\kappa = k(k+1)/2$,

$$\begin{aligned}
\|y\hat{\boldsymbol{r}}\| &\le (k+1)d(2p)^{\kappa'+1}w^{\kappa'}\max_i\left\|\boldsymbol{z}^{(i)}\right\| \le (k+1)d(2p)^{\kappa'+1}w^{\kappa'}\cdot 2\sqrt{3}\phi_2\mathcal{B}mdw^k p^k\\
&\le (k+1)2^{\kappa'+2}\sqrt{3}\phi_2\mathcal{B}md^2w^\kappa p^{\kappa+1}.
\end{aligned}$$

Recall that we assumed $k > 1$. When $k = 1$, Protocol 5.3 still needs 3 challenges for its soundness property. As a result, Protocol 5.4 is at least 3-special sound. □

5.5.3 Relaxed set membership proof

Suppose the prover has a commitment C and wants to prove knowledge of a message opening $\boldsymbol{m}$ of C such that $\boldsymbol{m} \in \mathcal{S} = \{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{N-1}\}$. Such a *set membership proof* can be constructed from one-out-of-many proof as in [GK15]. The protocol works as follows. Both the prover and the verifier set $P_i = C - \mathrm{Com}_{ck}(\boldsymbol{v}_i; 0)$ in the one-out-of-many proof, and run that protocol. This proves knowledge of $(y, \ell, \hat{\boldsymbol{r}})$ such that $yP_\ell = \mathrm{Com}_{ck}(\mathbf{0}; \hat{\boldsymbol{r}})$, which gives

$$\begin{aligned}
& yP_\ell = y(C - \mathrm{Com}_{ck}(\boldsymbol{v}_\ell; 0)) = \mathrm{Com}_{ck}(\mathbf{0}; \hat{\boldsymbol{r}})\,,\\
\implies\quad & yC = \mathrm{Com}_{ck}(\mathbf{0}; \hat{\boldsymbol{r}}) + y\mathrm{Com}_{ck}(\boldsymbol{v}_\ell; 0) = \mathrm{Com}_{ck}(y\boldsymbol{v}_\ell; \hat{\boldsymbol{r}})\,. \qquad (5.35)
\end{aligned}$$

As a result, our set membership proof convinces the verifier of the following statement, for some $\hat{\mathcal{T}} \in \mathbb{R}^+$,

$$\mathcal{R}'_{\mathrm{mem}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{l} ((ck, (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{N-1}), C), (y, \ell, \hat{\boldsymbol{r}})) : \ell \in [0, N) \ \wedge\ \|\hat{\boldsymbol{r}}\| \le \hat{\mathcal{T}} \wedge \\ yC = \mathrm{Com}_{ck}(y\boldsymbol{v}_\ell; \hat{\boldsymbol{r}}) \ \wedge y \text{ is a product of elements in } \Delta\mathcal{C}^d_{w,p} \end{array} \right\}. \qquad (5.36)$$

5.6 Applications to Advanced Cryptographic Schemes

5.6.1 Ring signature

The construction of ring signature from one-out-of-many proof follows the same strategy as in [GK15, BCC+15] and Section 4.3. The users commit to their secret keys and these commitments represent the public keys. A set of public keys is then used as the set of public commitments in the one-out-of-many proof. The prover proves knowledge of an opening of one of the commitments (i.e., knowledge of a secret key corresponding to one of the public keys used to construct the signature). The main difference from [GK15, BCC+15] and Section 4.3 is that we show that our relaxed proof is still sufficient.

Construction

Let $N = \beta^k$ for $2 \leq \beta \leq N$, $k \geq 1$ and $q, \mathcal{B}$ with $\mathcal{B} < q$ be positive integers. Let $\mathsf{CMT} = (A, B, C, D, E_0, \ldots, E_{k-1})$ and $\mathsf{RSP} = \left(\{f_{j,i}\}_{j=0,i=1}^{k-1,\beta-1}, \boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c\right)$ be the initial commitment and prover's response from Protocol 5.4, respectively. Also, denote $\mathcal{C}_{w,p}^d$ as the challenge space defined in (5.9) and $\mathsf{CMT}^* = (B, C, E_1, \ldots, E_{k-1})$ as a subset of CMT. Our ring signature uses Fiat-Shamir heuristic [FS86] to make Protocol 5.4 non-interactive and is defined as follows.

- **RSetup**(1^λ): Run $\boldsymbol{G} \leftarrow \mathsf{CKeygen}(1^\lambda)$ of HMC in HNF and pick a hash function $H : \{0,1\}^* \rightarrow \mathcal{C}_{w,p}^d$. Return the commitment key $ck = \boldsymbol{G}$ and H as $pp = (ck, H)$.
- **RKeygen**(pp): Sample $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$ and compute $P = \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{r})$ for the all-zero vector $\boldsymbol{0}$. Return $(pk, sk) = (P, \boldsymbol{r})$.
- **RSign**$_{pp,sk}(M, \boldsymbol{L})$: Parse $\boldsymbol{L} = (P_0, \ldots, P_{N-1})$ with $P_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; sk)$ for $\ell \in [0, N)$. Proceed as follows.
  1. Generate CMT by running $\mathcal{P}_{1/\mathrm{N}}((ck, (P_0, \ldots, P_{N-1})), (\ell, sk))$.
  2. Compute $x = H(ck, M, \boldsymbol{L}, \mathsf{CMT})$.
  3. Compute RSP by running $\mathcal{P}_{1/\mathrm{N}}(x)$ with CMT.
  4. If $\mathsf{RSP} = \perp$, go to Step 1.
  5. Otherwise, return $\sigma = (\mathsf{CMT}^*, x, \mathsf{RSP})$.
- **RVerify**$_{pp}(M, \boldsymbol{L}, \sigma)$: Parse $\boldsymbol{L} = (P_0, \ldots, P_{N-1})$, $\sigma = (\mathsf{CMT}^*, x, \mathsf{RSP})$ and $\mathsf{CMT}^* = (B, C, E_1, \ldots, E_{k-1})$. Continue as follows.
  1. Compute A, D and E_0 so that Steps 6 and 7 in Protocol 5.3, and Step 3 in Protocol 5.4 are satisfied.
  2. Set $\mathsf{CMT} = (A, B, C, D, E_0, \ldots, E_{k-1})$.
  3. If $x \neq H(ck, M, \boldsymbol{L}, \mathsf{CMT})$, return 0.
  4. Return the output of $\mathcal{V}_{1/\mathrm{N}}(ck, (P_0, \ldots, P_{N-1}), (\mathsf{CMT}, x, \mathsf{RSP}))$.

Correctness and anonymity properties of the ring signature follows easily from the completeness and SHVZK of Protocol 5.4, respectively. In particular, for $\phi_1 = \phi_2 = 15$, we have $1/(\mu(\phi_1)\mu(\phi_2)) > 1/5$. Therefore, an accepting transcript is produced by Protocol 5.4 with probability at least $1/5$. Thus, the expected number of iterations in **RSign** is 5, which is $O(1)$, for $\phi_1 = \phi_2 = 15$. Unforgeability of the ring signature is stated in Theorem 5.16.

Theorem 5.16. *If HMC in HNF is hiding and γ-binding where $\gamma = \max\{\gamma_{\mathrm{bin}}, \gamma_{1/\mathrm{N}}\}$ for γ_{bin} and $\gamma_{1/\mathrm{N}}$ defined in Theorem 5.10 and Theorem 5.15, respectively, then the ring signature scheme described by* (**RSetup**, **RKeygen**, **RSign**, **RVerify**) *is unforgeable with respect to insider corruption in the random oracle model.*

Proof (Sketch). The idea for the proof is similar to that in Proof of Theorem 4.15, but the challenge space is exponentially large in this case and no parallel repetition of the underlying protocol is required in the ring signature. Assume that there exists a PPT adversary $\mathcal{F}$ that can forge a ring signature in polynomial time and non-negligible probability. This gives rise to an adversary $\mathcal{A}$ which can break the binding property of the commitment scheme, and thus solve M-SIS problem.

$\mathcal{A}$ creates an invalid public key pk_ℓ such that $pk_\ell = \mathrm{Com}_{ck}(1, 0, \ldots, 0; \boldsymbol{r})$ for $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$, which cannot be detected by $\mathcal{F}$ due to the hiding property of the commitment scheme. Then, it runs $\mathcal{F}$ until $k+1$ forgeries in total with distinct challenges are obtained where CMT^* part of the signature (and thus CMT) is the same for all forgeries and pk_ℓ is not corrupted. This can be done in polynomial time using

TABLE 5.6: The parameter setting of our ring signature with a root Hermite factor ≤ 1.0045 for both M-LWE and M-SIS. $\mathcal{B} = 1, \phi_1 = \phi_2 = 15$ for all cases.

N	2	8	64	2^{12}	2^{21}
$(d,\ w,\ p)$	$(256, 60, 1)$	$(256, 60, 1)$	$(128, 66, 2)$	$(128, 66, 2)$	$(128, 66, 2)$
$(n,\ m)$	$(4, 12)$	$(4, 13)$	$(10, 28)$	$(13, 32)$	$(22, 46)$
$(k,\ \beta)$	$(1, 2)$	$(1, 8)$	$(1, 64)$	$(2, 64)$	$(3, 128)$
q	$\approx 2^{53}$	$\approx 2^{58}$	$\approx 2^{59}$	$\approx 2^{60}$	$\approx 2^{77}$
Signature Length (KB)	36	41	58	103	256
Public Key Length (KB)	6.63	7.25	9.22	12.19	26.47
Secret Key Length (KB)	0.38	0.41	0.44	0.50	0.72

the Forking Lemma in [BPVY00]. Then, $\mathcal{A}$ runs the extractor of Protocol 5.4 to get an *exact* valid opening $(\mathbf{0}; \boldsymbol{s})$ of $y \cdot pk_i$ for some public key pk_i where y is the relaxation factor in Definition 5.14. With $1/\text{poly}(\lambda)$ probability, $i = \ell$ as $\mathcal{F}$ can only make polynomially many queries to PKGen oracle. As a result, $((y, 0, \ldots, 0; y\boldsymbol{r}), (\mathbf{0}; \boldsymbol{s}))$ is binding collision pair for the commitment scheme since $(y, 0, \ldots, 0) \neq \mathbf{0}$. □

Concrete parameters

As mentioned before, we set our parameters to make the easiest cases of M-SIS and M-LWE hard against known attacks in practice and aim for 128-bit "post-quantum" security $\lambda = 128$. We set (d, w, p) so that $|\mathcal{C}^d_{w,p}| > 2^{256}$. Similar to recent lattice-based proposals [BDL+18, LN17, dPLS18], we too consider an error distribution of $\mathcal{U}(\{-1, 0, 1\}^d)$, i.e., $\mathcal{B} = 1$ for M-LWE problem. We set $\phi_1 = \phi_2 = 15$ so that the acceptance rate of the rejection sampling is more than 1/5. Practical security estimations are done as summarised in Section 3.2.2 for a root Hermite factor of around 1.0045. We make sure that the commitment scheme is γ_{bin}-binding and $\gamma_{1/\text{N}}$-binding. The full parameter setting is given in Table 5.6 for various ring sizes.

Asymptotic signature length

We neglect any $\log\log$ (multiplicative/additive) terms (both in λ and N) throughout our analysis in this section and work with the challenge space $\mathcal{C}^d_{w,1}$. Security of β_{SIS}-M-SIS in dimension nd and modulus q with $\beta_{\text{SIS}} \approx q$ against BKZ-reduction attack with root Hermite factor δ [MR09] requires $nd \geq \log q/(4 \log \delta)$. Using the BKZ root Hermite factor from [ADPS16] with $\log \delta = \Omega(1/\lambda)$ for security parameter λ, we get $nd = \Omega(\lambda \log q)$. Balancing the same security level for "dual" attack on LWE [ADPS16], we get $m = O(n)$ and $md = \Omega(\lambda \log q)$ (recall that the LWE dimension parameter is proportional to $(m - n) \cdot d$). Take $k = O((\log N)/t)$ for a parameter $1 \leq t \leq \log N$ to be optimised. As a result, we have $\beta = N^{1/k} = 2^{\log N/k} = O(2^t)$. To be a one-shot proof, we require $2^w \binom{d}{w} = 2^{O(\lambda)}$. Then, choosing $d = O(\lambda)$ and $w = O(\lambda/\log \lambda)$ is sufficient. Finally, we need $q = O(w^{k^2} k m d^2)$ for M-SIS security[11]. Therefore, $\log q = O(k^2 \log w + \log(md)) = O(((\log N)/t)^2 \log(\lambda/\log \lambda) + \log(\lambda \log q)) = O((\log^2 N \log \lambda)/t^2)$. Using these, we can also find

$$\log(kw) = O(\log(\log N \lambda/(t \log \lambda))) = O(\log \lambda), \text{ and}$$
$$\log(wmd) = O(\log(\lambda^2/(\log \lambda \log q)) = O(\log \lambda).$$

[11] This is due to $\gamma_{1/\text{N}}$, which grows asymptotically faster than γ_{bin}.

Now, for the signature size $|\sigma|$, we have

$$\begin{aligned}|\sigma| &= O(knd\log q + k\beta d\log(kw) + md\log(wmd))\\ &= O(\lambda\log q(\log N\log q/t + \log\lambda) + \lambda\log\lambda\log N2^t/t)\\ &= O(\lambda\log N(\log^2 q/t + 2^t\log\lambda/t))\\ &= O(\lambda\log N(\log^4 N\log^2\lambda/t^5 + 2^t\log\lambda/t))\\ &= O(\lambda\log\lambda\log N(\log^4 N\log\lambda/t^5 + 2^t/t)).\end{aligned}$$

Taking $t = O(1)$, we can get $|\sigma|$ to be quasi-linear in λ and poly-logarithmic in N, i.e., $|\sigma| = O(\lambda\log^2\lambda\log^5 N)$. However, if we set $t = (\log N)^{2/3}$, then $\log^4 N\log\lambda/t^5 = \log\lambda(\log N)^{2/3}$ and this term is roughly of the same size as or larger than $2^t/t$ for all practical N values such as $N \leq 2^{30}$. Therefore, we can say that the signature length in practice is proportional to $\lambda\log^2\lambda\log^c N$ where $c \approx 1.67$ for $N \leq 2^{30}$. Hence, for a fixed security level, the signature length grows slightly faster than logarithmic in N, which also matches the signature length growth for the values provided in Table 5.6.

Computational efficiency

To estimate the computational efficiency of the ring signature, we look at that of the one-out-of-many proof in Section 5.5.2, and consider the efficiency in terms of degree-256 polynomial multiplications in R_q, denoted by `poly256_mult`. We assume that a standard PC has a CPU running at 3 GHz.

Let us take a medium-sized number of ring participants as $N = 2^{10}$. Our ring signature in this case can be instantiated with $(d, w, p) = (256, 60, 1)$, $n = 6$, $m = 15$, $q \approx 2^{56}$, $k = 2$, $\beta = 32$, $\mathcal{B} = 1$, $\phi_1 = \phi_2 = 15$ (signature length is 89 KB). Then, the commitment key dimensions are 6×79 where the first 6×6 part is the identity matrix. Therefore, each commitment computation requires at most $6\cdot 73 = 438$ `poly256_mult` (this can be further optimised when the committed message is zero or binary-valued).
Offline signing. To compute A, B, C, D, there will be $4\cdot 438 = 1752$ `poly256_mult` in total. To compute E_j's, we need to perform around $k\cdot n\cdot N + k\cdot n\cdot m = 2\cdot 6\cdot 2^{10} + 2\cdot 6\cdot 15 = 12468$ `poly256_mult` (the computation of $p_{i,j}$ takes at most 1 `poly256_mult` for $k = 2$). Setting $\phi_1 = \phi_2 = 15$, the expected number of iterations due to rejection sampling will be 5. Therefore, the initial step for the prover is dominated by $5\cdot(1752 + 12468) \approx 2^{16}$ `poly256_mult` on average. According to the NTT implementation of [Chu18], for polynomial degree 256 and 51-bit modulus, NTT transformation costs about 8000 cycles and pointwise multiplication costs about 1000 cycles. Note that the user public keys and the commitment matrix can be stored in the NTT domain, and thus the number of NTT transformations are much less than that of the pointwise multiplications in our scheme. In particular, the number of NTT transformations is in the order of $k\cdot(m+\beta)$ ($k\beta$ transformations for $a_{0,0},\ldots,a_{k-1,\beta-1} \in R_q$'s and $(k+4)m$ transformations for the randomnesses $\boldsymbol{r}_a, \boldsymbol{r}_b, \boldsymbol{r}_c, \boldsymbol{r}_d, \rho_0, \ldots, \rho_{k-1} \in R_q^m$) whereas that of pointwise multiplications is in the order of $kn(N+m)$ (due to Step 8 of the prover). Therefore, the cost of pointwise multiplication is the dominant part in the computational cost of signing, and it can be done in about $1000\cdot 2^{16} \approx 2^{26}$ cycles, i.e., in about 20 ms on a standard PC. This phase can be easily computed offline.

Note that we need to sample $3md$ coefficients from a wide discrete normal distribution to construct the vectors $\boldsymbol{\rho}_0, \boldsymbol{r}_a, \boldsymbol{r}_d \in R_q^m$, which is repeated 5 times on average due to rejection sampling. For the concrete parameters with $N = 2^{10}$, this means sampling less than 58000 coefficients in total with a standard deviation around $2^{22.5}$.

Figure 3 and Table 4 in [ZSS19] show that one can sample 1024 Gaussian coefficients in less than 2^{18} cycles *independent* of the standard deviation. Thus, Gaussian sampling with a cost of about $58 \cdot 2^{18} < 2^{24}$ cycles will not be a bottleneck for offline running time.

Online signing. The response phase of the prover requires about $(k+2)m$ polynomial multiplications, which is only $4 \cdot 15 = 60$ `poly256_mult`. When repeated 5 times on average, it would take only around 100 μs on a standard PC. This phase can be treated as the online signing phase and is very fast.

Verification. The verification time of the ring signature is dominated by the last verification step in Protocol 5.4, which takes around $(k-1+n) \cdot N + k \cdot n + n \cdot m = 7270$ `poly256_mult`. Note that there is no additional factor due to rejection sampling here. This would take about 2-3 ms with the same assumptions.

The reported signing/verification running times of [KKW18] with the same $N = 2^{10}$ is 2.8 seconds. Also, extrapolating the computational efficiency results of NTRU-based ring signature from [LAZ19] (without linkability), the running time for signing/verification would be around 700-800 ms (where the signature length is about 14 times larger than ours). Therefore, our ring signature scheme also outperforms [KKW18] and [LAZ19], which are the only two works providing concrete running times, by a large margin in terms of computational efficiency for medium-sized rings.

For smaller ring sizes, the scheme in [KKW18] does not seem to get noticeably faster. For example, for $N = 2^7$, the running times of signing and verification go down to 2 seconds, i.e., not even reduced by a factor of 2 over the case $N = 2^{10}$. In our case, the offline signing and verification times would be reduced by a factor of more than 8 as N is reduced by a factor 8 and k would be set to 1.

5.6.2 Privacy-preserving credentials

In an anonymous credential system, there are three entities: organisations, that are able to issue credentials, users, who can obtain and show credentials, and verifiers, who verify the user credentials. Our goal here is to enable an *efficient* way for users to get a credential containing a set of attributes and later use it to prove that some of the attributes satisfy certain properties without revealing the attribute itself. We provide privacy for credential attributes by revealing only that they satisfy a certain relation, but we do not provide unlinkability between multiple showings or issuing, which is left as an interesting open problem for future work. In fact, linkability is a desired property in some applications such as e-voting and e-cash systems. Also, user anonymity can be obtained to some degree by using pseudonyms. Let us a give a simple example scenario. The user, Alice, wants to apply for a job that only considers applicants of age between 18 and 33, and from a European country. Then, she obtains a credential from her state government with these (and possibly more) attributes. The goal in this scenario is for Alice to convince the employer that she is eligible for the application without revealing the full details as she may not end up getting the job. The privacy of the credential attributes is achieved by revealing only the fact that some attributes satisfy a certain relation. Let us first describe our security model and then see how we can tackle this problem using our tools.

Let $\mathsf{RPoK}(C)$ be a *relaxed* proof of knowledge where the verifier is convinced that the prover knows $(y, \hat{\boldsymbol{r}})$ such that $yC = \mathrm{Com}_{ck}(\mathbf{0}; \hat{\boldsymbol{r}})$ and y is invertible. Further, let $\Pi_{\mathbb{P}}(C)$ be a protocol that proves knowledge of $(y, \hat{\boldsymbol{m}}, \hat{\boldsymbol{r}})$ such that $yC = \mathrm{Com}_{ck}(y\hat{\boldsymbol{m}}; \hat{\boldsymbol{r}})$, y is invertible and $\hat{\boldsymbol{m}}$ satisfies some property $\mathbb{P}$, denoted by $\hat{\boldsymbol{m}} \in \mathbb{P}$ (recall that the relaxation factor in our range proof is invertible). Now, the game between a prover and a challenger works as follows.

1. The prover sends a message $\boldsymbol{m}$ and a commitment C along with $\mathsf{RPoK}(C')$ where $C' = C - \mathrm{Com}_{ck}(\boldsymbol{m}; \mathbf{0})$.
2. Then, the challenger chooses a property $\mathbb{P}$ that can be proven by some protocol $\Pi_{\mathbb{P}}$, and sends $\mathbb{P}$ to the prover.
3. Finally, the prover sends $\Pi_{\mathbb{P}}(C)$.

The prover wins if $\boldsymbol{m} \notin \mathbb{P}$.

Theorem 5.17. *If the commitment scheme* $\mathrm{Com}_{ck}(*; *)$ *is* γ*-binding (for appropriately set* γ*), and* $\mathsf{RPoK}(C')$ *and* $\Pi_{\mathbb{P}}(C)$ *are sound with an invertible relaxation factor, then no PPT adversary can win the above game with a non-negligible probability.*

Proof. Let $\mathcal{A}$ be a PPT adversary that plays the above game and sends $\boldsymbol{m}$ and C in the first move. Let $C' = C - \mathrm{Com}_{ck}(\boldsymbol{m}; \mathbf{0})$. By the soundness of $\mathsf{RPoK}(C')$, $\mathcal{A}$ must know $(y, \hat{\boldsymbol{r}})$ such that

$$\begin{aligned} yC' = \mathrm{Com}_{ck}(\mathbf{0}; \hat{\boldsymbol{r}}) &\implies y(C - \mathrm{Com}_{ck}(\boldsymbol{m}; \mathbf{0})) = \mathrm{Com}_{ck}(\mathbf{0}; \hat{\boldsymbol{r}}) \\ &\implies yC = \mathrm{Com}_{ck}(y\boldsymbol{m}; \hat{\boldsymbol{r}}). \end{aligned} \tag{5.37}$$

Now, by the soundness of $\Pi_{\mathbb{P}}(C)$, $\mathcal{A}$ must also know $(y', \hat{\boldsymbol{m}}', \hat{\boldsymbol{r}}')$ such that $\hat{\boldsymbol{m}}' \in \mathbb{P}$ and

$$y'C = \mathrm{Com}_{ck}(y'\hat{\boldsymbol{m}}'; \hat{\boldsymbol{r}}'). \tag{5.38}$$

Multiplying (5.37) by y' and (5.38) by y, we get

$$\begin{aligned} y' \cdot (yC) = \mathrm{Com}_{ck}(y'y\boldsymbol{m}; y'\hat{\boldsymbol{r}}) \quad \text{and} \quad y \cdot (y'C) = \mathrm{Com}_{ck}(yy'\hat{\boldsymbol{m}}'; y\hat{\boldsymbol{r}}') \\ \implies \quad \mathrm{Com}_{ck}(y'y\boldsymbol{m}; y'\hat{\boldsymbol{r}}) = \mathrm{Com}_{ck}(yy'\hat{\boldsymbol{m}}'; y\hat{\boldsymbol{r}}'). \end{aligned} \tag{5.39}$$

By the binding property of the commitment scheme, $y'y\boldsymbol{m} = yy'\hat{\boldsymbol{m}}'$, and thus $\boldsymbol{m} = \hat{\boldsymbol{m}}' \in \mathbb{P}$ since y, y' are invertible by assumption. □

Remark 5.18. *A similar argument as in Theorem 5.17 can be used to strengthen the proved relations in our relaxed protocols as follows. A relaxed proof for a property* $\mathbb{P}$ *combined with an* exact *proof of knowledge (i.e., proving knowledge of* $(\boldsymbol{m}, \boldsymbol{r})$ *such that* $C = \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r})$*) proves that the prover knows an* exact *valid opening of* C *and that this opening (without any relaxation) satisfies* $\mathbb{P}$*. However, such lattice-based exact proofs of knowledge are not currently very efficient. Furthermore, the invertibility assumption of the relaxation factor can be circumvented using Lemma 5.5 provided that the relaxation factor and the message* $\boldsymbol{m}$ *have bounded norm, i.e., we can infer* $\boldsymbol{m} = \boldsymbol{m}'$ *from* $yy'(\boldsymbol{m} - \boldsymbol{m}') = 0$ *in* R_q *using Lemma 5.5.*

Going back our scenario, $\boldsymbol{m}$ in the above game represents the set of attributes. Alice applies to get a credential with the set of attributes in $\boldsymbol{m}$, and obtains a signature on C after passing the relaxed proof of knowledge. In her application for the job, she first shows that C is signed by an authority and that her age attribute is the range $[18, 33)$, and the expiry date and country code attributes are in some valid ranges (using a single relaxed range proof). Here, the ranges for all the other attributes but these three are set so that they are trivially satisfied (for example, the range is $[0, 2^{32})$ if they are unsigned and represented by 32 bits). Seeing a signature on C and the range proof, the employer is convinced that Alice is eligible to apply.

All of our proofs except for one-out-of-many proof has the structure represented by $\Pi_{\mathbb{P}}$ where the property $\mathbb{P}$ changes for each proof. For example, a similar idea can be also used with the set membership proof. In that case, the public set S in the protocol needs to be set so that it covers all the possible options for the attributes that Alice does not want to reveal any information about.

5.7 Discussion

The clear advantage of the new techniques and tools introduced in this chapter is that they do not require protocol repetitions for soundness amplification. As a result, we are able to overcome the $\widetilde{O}(\lambda^2)$ barrier in both the asymptotic proof length growth and the computation time, and can construct very short and fast proofs from lattices. We believe that overcoming of this barrier is an important milestone in the construction of efficient advanced ZKPs from lattices.

Having efficient building blocks in turn paves the way for an efficient and scalable ring signature scheme based on standard lattice assumptions. One can see the effectiveness of the new techniques by looking at the dramatic improvements of the new ring signature scheme in comparison to the prior art (see Table 1.1).

A limitation, on the other hand, is that we can only get *relaxed* proofs. That is, the proved relations additionally involve a (possibly non-trivial) relaxation factor. However, we also show how to handle the relaxation in ZKPs' application to higher level constructions and show that they are still useful in practice despite their relaxed nature. Another example of this is given in the next chapter, where we show how to construct an efficient blockchain confidential transactions protocol based on similar relaxed proofs.

If we compare the proofs in this chapter and those in the previous chapter, we can see an important difference in relation to the relaxation factor. Firstly, as mentioned before, the relaxation factor in the previous chapter is a fixed publicly known value, whereas in this chapter, we only know that it is either a challenge difference or a product of challenge differences with its exact nature known only to the prover. Furthermore, the growth of the norm of the relaxation factor with the polynomial relation degree k in this chapter is greater than that in the previous chapter. This in turn creates an advantage for the methods in the previous chapter as the extracted witness length growth is slower in the polynomial relation degree k, but this advantage in favour of the prior methods from Chapter 4 is negated by the quadratic dependence of the proof length growth on the security parameter.

Throughout the chapter, our focus has mainly been on proving *relaxed* relations. As briefly touched upon in Remark 5.18, it is indeed possible to strengthen our relaxed proofs by combining them with an *exact* proof of knowledge. This method would enable one to construct, in particular, exact binary, range and set membership proofs (with larger proof sizes). Although current exact proof constructions from lattices do not seem to offer a satisfactory solution in terms of practical efficiency, there has been very recent exciting works [BLS19, YAZ+19] that aim to design more efficient exact proofs. It is therefore plausible to expect further developments in the construction of exact proofs, and hence strengthening our relaxed proofs would become cheaper as the cost of an exact proof of knowledge decreases.

Chapter 6

Blockchain Confidential Transactions from Lattices

As stated previously in the introduction, zero-knowledge proofs are a fundamental tool used in numerous privacy-preserving applications, and they have recently become a crucial part of privacy-aware blockchain-based applications such as private/anonymous cryptocurrencies, e.g., Monero and Zcash. As seen in the previous chapters, when coupled with commitments, ZKPs allow users to prove useful statements without leaking private information. For example, Monero uses the RingCT protocol [Noe15] to realise confidential transactions. However, the currently deployed solutions in these systems do not provide post-quantum security. Quoting again from Zcash's FAQ page [Tea19], the developers "plan to monitor developments in postquantum-secure components, and if/when they are mature and practical, update the Zcash protocol to use them." Therefore, there is an evident need to design quantum-secure alternatives of currently deployed privacy-preserving protocols, which brings us to the main goal of this chapter: introduce a *practical post-quantum* RingCT protocol based on computational lattice problems (in particular, M-SIS and M-LWE).[1]

As one may recall from Chapter 2, the main challenge in the construction of efficient lattice-based advanced protocols such as RingCT comes from the difficulty in realising suitable underlying ZKPs efficiently. Thankfully, the tools developed in the previous two chapters can be used as important building steps towards the construction of an efficient post-quantum RingCT protocol. Of course, straightforward combination of the already designed ZKPs does not result in a very efficient scheme and we still need to come up with new techniques specifically crafted for our target RingCT construction, which is exactly what is done in the upcoming sections.

We start the chapter by giving an overview of MatRiCT, our efficient, scalable and post-quantum RingCT protocol, in Section 6.1. Then, a high-level summary of the new techniques introduced in this chapter is given in Section 6.2. Our formal definitions for RingCT-like cryptocurrency protocols and comparative discussions between the new definitions and the prior ones are provided in Section 6.3. Later in Section 6.4, the full algorithmic details of MatRiCT are discussed, where we split MatRiCT into subprocedures, each of which is summarised as a pseudocode. We introduce an improved soundness proof for the underlying binary proof used in MatRiCT in Section 6.5. The security proofs of MatRiCT are given in Section 6.6, where we show that MatRiCT satisfies *correctness*, *anonymity* and *balance* properties defined in Section 6.3. We then discuss the parameter choices and implementation details in Section 6.7, where an evaluation of MatRiCT is also provided. Section 6.8 covers a discussion about a parameter choice regarding the serial numbers used in MatRiCT. The extension of MatRiCT to provide auditability is detailed in Section 6.9, where our novel extractable

[1] This chapter is mainly based on [EZS+19].

commitment scheme, the crucial tool enabling the auditability feature, is introduced. More discussion on the ring and group signatures, which are constructed as sub-blocks of MatRiCT, is provided in Section 6.10.

In this chapter, the commitment scheme is always instantiated with HMC (i.e., using a completely random commitment matrix $\boldsymbol{G}$).

6.1 Overview of MatRiCT

One of the most important features of MatRiCT is that no *wide* range proof is required. The general structure of the whole system is as follows. We work over two cyclotomic rings $R_q = \mathbb{Z}_q[X]/(X^d+1)$ and $R_{\hat{q}} = \mathbb{Z}_{\hat{q}}[X]/(X^d+1)$ where q is a small modulus (of about 31 bits) and $\hat{q}$ is large modulus (of about 53 bits). Note that both moduli are much smaller than 64 bits in bit-length even though we allow the transaction amount to be of 64 bits.

A user secret key sk is a random short vector over R_q and the user's public key is generated by using sk as the randomness of a commitment to zero. When minting a coin to represent an amount without revealing its value, we do not commit to the integer amount, but instead commit to the bits of the amount over R_q. Therefore, a standard argument stating that the sum of input coins equals the sum of output coins is not sufficient for the balance property as the addition is done over $\mathbb{Z}_q$. Instead, we introduce a novel balance proof as sketched in Section 6.2.

To spend some of her accounts, each of which is a pair of a public key and a coin, a user Alice proceeds as follows. She mints her coins and computes some "corrector" values to be used in the balance proof. These corrector values help Alice prove that the sum of input amounts equals the sum of output amounts, and they are binary when there is one input account and at most two output accounts. For simplicity, let us assume that is the case for now. To hide her identity, Alice gathers other accounts to be used in a ring signature. Suppose Alice wants to spend M of her accounts while hiding herself among N users, in which case she gathers $M \cdot N$ accounts (including those of her own), seen as an $M \times N$ matrix. Then, she chooses an index $\ell \in [0, N-1]$ and places her own accounts on the ℓ-th column. Below is an illustration of the matrix constructed by Alice.

$$\mathsf{A}_{\text{in}} = \begin{array}{|c|c|c|c|c|} \hline \mathsf{act}_{0,0} & \cdots & \mathsf{act}_{0,\ell} = (\mathsf{pk}_{0,\ell}, \mathsf{cn}_{0,\ell}) & \cdots & \mathsf{act}_{0,N-1} \\ \hline \vdots & \ddots & \vdots & \ddots & \vdots \\ \hline \mathsf{act}_{M-1,0} & \cdots & \mathsf{act}_{M-1,\ell} = (\mathsf{pk}_{M-1,\ell}, \mathsf{cn}_{M-1,\ell}) & \cdots & \mathsf{act}_{M-1,N-1} \\ \hline \end{array}$$

After this initial setting, Alice computes an *aggregated* binary proof over $R_{\hat{q}}$ where she proves that 1) all minted output coins are commitment to bits, 2) her index ℓ is properly encoded by some bits in some ring elements, and 3) the "corrector" values are properly encoded as bits in some ring elements. Here, our scheme crucially benefits from this efficient aggregation. Alice then provides M ring signatures for her accounts to be spent, and also proves that the "corrector" commitment is of special form such that the commitment does not contain any value (i.e., the committed message represents zero). Finally, she runs another ring signature on commitments $P_0, \ldots, P_{N-1}$ where

$$P_i = \sum(\text{output coins}) - \sum \begin{pmatrix} \text{input coins in} \\ i\text{-th column} \end{pmatrix} + (\text{“corrector” com.}).$$

Observe that Alice knows all the secret values that constructs P_ℓ, and the corrector commitment is constructed in a way that P_ℓ is a commitment to zero over R_q *when* the sum of input (integer) amounts equals the sum of output (integer) amounts. Recall that Alice also proves corrector commitment contains no value. Finally, she computes serial numbers as commitments to her secret keys used under a new commitment key and proves that this is indeed the case. This step is to prevent Alice from double-spending.

For auditability, we allow the auditor to extract Alice's index since she already proves that her index ℓ is properly encoded in some commitment. For this, we require an extra tool: an extractable commitment compatible with our construction (see Section 6.9.1).

An important feature of auditable MatRiCT is that users can choose a specific auditor from a set of possible auditors or can even choose to have no auditors, all within the same environment. Therefore, the user chooses an auditing option i, where $i = 0$ indicates no auditor (i.e., full anonymity) and $i > 0$ indicates auditing by the i-th authority (i.e., conditional anonymity).

6.2 Overview of New Techniques

6.2.1 Improved ring signature

The ring signatures from previous chapters consist mainly of two parts: 1) a binary proof on the signer's index, and 2) a one-out-of-many proof on the set of public keys. We show that the two verification equations in the binary proof can be batched together, which reduces the number of commitments and the randomnesses to be communicated by half and thus the binary proof's cost almost by half. Let us recall the general idea for the binary proof from previous chapters that build on the ideas from [GK15, BCC+15].

Suppose that we want to show b is a bit. Let $B = \mathrm{Com}(b; *)$, $A = \mathrm{Com}(a; *)$ and $f = x \cdot b + a$ for some masking value a and a challenge x where Com is homomorphic commitment.[2] Then, one verification equation shows that f is well-formed by checking $xB + A = \mathrm{Com}(f; *)$. The other equation proves that the coefficient of x^2 in the product $f \cdot (x - f) = x^2\,[b(1-b)] + x\,[a(1-2b)] - a^2$ is zero by checking $\mathrm{Com}(f \cdot (x - f); *) = xC + D$ where $C = \mathrm{Com}(a(1-2b); *)$ and $D = \mathrm{Com}(-a^2; *)$ are set by the prover. Thus, the latter verification equation proves that $b(1-b) = 0$, which implies (under certain conditions) that b is binary.

In the ring signature, we do not make use of the commitments A, B, C, D, other than for showing that f "encodes" a bit b. Therefore, the prover can set $B = \mathrm{Com}(b, a(1-2b); *)$ and $A = \mathrm{Com}(a, -a^2; *)$, and the verifier can equivalently check $xB + A = \mathrm{Com}(f, f(x-f); *)$. That is, since both verification equations are effectively linear in x, they can be batched together. This ensures again that $f = xb + a$ and that the coefficient of x^2 in the product $f \cdot (x - f)$ is zero. Now we do not need to have the commitments C, D at all, and also do not need to communicate a masked randomness for a second verification. The gain in the communication cost follows from here. This idea works both in the DL setting (and thus applies to all protocols using the proof systems from [GK15, BCC+15]), and in the lattice setting as we do not exploit any special property of the commitment scheme other than the standard binding property.

[2] In the real proof, multiple binary proofs are batched by committing to all the bits b_i's together as $B = \mathrm{Com}(b_0, b_1, \ldots; *)$, which we ignore here for simplicity.

Second, we show that by using two sets of *compatible* parameters for the two parts of the ring signature, one can significantly reduce the signature length. Here, it is important to choose the parameters carefully as the two parts are not completely independent. In our setting, the binary proof requires a much bigger modulus than the one-out-of-many proof. This is due to both the hardness of the underlying M-SIS problem and also to make the binary proof go through in a *ring* $R_{\hat{q}}$ with zero divisors where $b(1-b)=0$ may not imply $b \in \{0,1\}$ (unlike in the field $\mathbb{Z}_q$). Therefore, we use a large modulus $\hat{q}$ for the binary proof, and a small modulus q for the one-out-of-many proof (and also the other parts of the protocol). In addition to reducing the proof length, this also reduces the user public key size. Since public keys play a central role in the whole blockchain system, the overall advantage is two-fold. The new binary proof additionally has the advantage that the condition on the modulus to make the binary proof go through in the ring $R_{\hat{q}}$ is much weaker than the one in Chapter 5. Using the soundness proof from Chapter 5, one would need to set $\hat{q}$ to be of more than 70 bits whereas we use around a 53-bit modulus $\hat{q}$ in this chapter.

6.2.2 Efficient rejection sampling for binary secrets of fixed Hamming weight

Recall that the prover's binary secrets are encoded as $f = x \cdot b + a$ where b is a secret bit, $a \in R_{\hat{q}}$ is a masking element and $x \in R_{\hat{q}}$ is a challenge received (computed) after a is sampled. For our commitment scheme to be binding, f needs to be of small norm and thus we cannot choose a randomly from $R_{\hat{q}}$. In this case, a standard technique to make sure that f does not reveal information about the secret is using rejection sampling [Lyu09]. Suppose that we sample $a \leftarrow \{-\mathcal{B}_a, \ldots, \mathcal{B}_a\}^d$ and $\|x \cdot b\|_\infty \leq p$ for all possible x and b values where $\mathcal{B}_a \gg p \in \mathbb{Z}^+$. The idea for the rejection sampling in [Lyu09] is to make the distribution of f uniform in a box by aborting the interactive protocol (or starting over in the non-interactive case) if the maximum absolute coefficient of f is greater than $\mathcal{B}_a - p$.

Now, when $b = 0$, we know independent of the challenge x that f will be equal to a. Therefore, in this case, one may sample a directly from $\{-(\mathcal{B}_a - p), \ldots, \mathcal{B}_a - p\}^d$ in the first place to make sure that $f = x \cdot b + a$ is not rejected. Still, the distribution of f conditioned on passing the rejection sampling check is identical to the uniform distribution on $\{-(\mathcal{B}_a - p), \ldots, \mathcal{B}_a - p\}^d$, thus simulation-based security aspects remain untouched. However, the number of zero secrets affects the overall acceptance probability and thus such a rejection sampling leaks side-channel information. For example, proving knowledge of secret bits $1, 1, 1, 1, 1$ is likely to take longer than proving knowledge of secret bits $0, 0, 0, 0, 0$ as the latter is never rejected while the former is rejected with some non-negligible probability.

In our protocol, the user index is represented in unary, i.e., the bit sequence representing the user index has a *fixed* number of zeros and ones. Therefore, the above technique of sampling the masking value from the accepted distribution in advance does not leak additional information as the prover's goal is to prove that there are exactly k ones in the secret bit sequence for some publicly known $k \in \mathbb{Z}^+$. Hence, the technique is applicable and allows us to increase the acceptance rate significantly without needing to sample these components from a wider interval. To illustrate, when $N = 200$, the bit sequence representing the user index ℓ is the ℓ-th unit vector, i.e., has 199 zeros and a single one. Therefore, if the acceptance probability for a single secret bit is P, then the overall acceptance probability using our technique is still P instead of P^{200}, which would be the case using the previous standard technique.

We also note that the technique trivially extends to the case where the secret sequence has a fixed number of zeros and some other elements (which may not be binary) as we do not make use of the fact that nonzero secrets are equal to 1.

6.2.3 Novel balance proof

Suppose we want to prove that

$$\sum_{i=0}^{M-1} a_{\text{in},i} = \sum_{i=0}^{S-1} a_{\text{out},i} \tag{6.1}$$

for some input amounts $a_{\text{in},0}, \ldots, a_{\text{in},M-1}$ and output amounts $a_{\text{out},0}, \ldots, a_{\text{out},S-1}$ where $M, S \in \mathbb{Z}^+$. The general idea to prove (6.1) while hiding the amounts is to commit to each amount value using a homomorphic commitment scheme, and then show that 1) each committed value is in a valid positive range, and 2) the sum of output commitments minus the sum of the input commitments is a commitment to zero. For the lattice-based schemes, there do not exist a range proof that is significantly shorter than the generic approach: first, prove that some masked values encode bits, and then that these bits construct the committed integer. One important detail that especially has an effect for the lattice-based schemes is that (6.1) must hold over $\mathbb{Z}$, not just $\mathbb{Z}_q$.

Now, let us see how our balance proof works. Assume we want to work in base $\beta \geq 2$ and the amounts are represented by r digits. Then, we can write $a = \sum_{j=0}^{r-1} \beta^j a[j]$ for any amount a with the digits $a[j]$'s. Hence, we get

$$\begin{aligned}
\sum_{i=0}^{M-1} a_{\text{in},i} = \sum_{i=0}^{S-1} a_{\text{out},i} &\iff \sum_{i=0}^{M-1}\sum_{j=0}^{r-1} \beta^j a_{\text{in},i}[j] = \sum_{i=0}^{S-1}\sum_{j=0}^{r-1} \beta^j a_{\text{out},i}[j], \\
&\iff \sum_{j=0}^{r-1} \beta^j \sum_{i=0}^{M-1} a_{\text{in},i}[j] = \sum_{j=0}^{r-1} \beta^j \sum_{i=0}^{S-1} a_{\text{out},i}[j], \\
&\iff 0 = \sum_{j=0}^{r-1} \beta^j \left(\sum_{i=0}^{S-1} a_{\text{out},i}[j] - \sum_{i=0}^{M-1} a_{\text{in},i}[j] \right), \qquad (6.2) \\
&\iff 0 = \sum_{j=0}^{r-1} \beta^j \left(\sum_{i=0}^{S-1} a_{\text{out},i}[j] - \sum_{i=0}^{M-1} a_{\text{in},i}[j] + c_j - \beta c_{j+1} \right), \qquad (6.3)
\end{aligned}$$

for $c_0 = c_r = 0$ and any corrector values $c_1, \ldots, c_{r-1} \in \mathbb{Z}$. Therefore, instead of using the general idea that mandates a very large modulus, we can proceed as follows. Setting $\beta = 2$, for each amount, we commit to its bits as

$$\begin{aligned}
C_{\text{in},i} &= \text{Com}(a_{\text{in},i}[0], \ldots, a_{\text{in},i}[r-1]; *), \\
C_{\text{out},i} &= \text{Com}(a_{\text{out},i}[0], \ldots, a_{\text{out},i}[r-1]; *).
\end{aligned}$$

Then, we also create a "corrector" commitment $C = \text{Com}(c_0 - 2c_1, \ldots, c_{r-1} - 2c_r; *)$ with $c_0 = c_r = 0$. Finally, we prove that 1) $C_{\text{in},i}$'s and $C_{\text{out},i}$'s are commitments to bits, 2) $\sum_{i=0}^{S-1} C_{\text{out},i} - \sum_{i=0}^{M-1} C_{\text{in},i} + C$ is a commitment to zero, and 3) C is well-formed as above. These guarantee that 1) the opening message for any commitment

to an amount represents a unique value in a positive range $[0, 2^r - 1]$, 2)

$$0 = \sum_{i=0}^{S-1} a_{\text{out},i}[j] - \sum_{i=0}^{M-1} a_{\text{in},i}[j] + c_j - 2c_{j+1} \tag{6.4}$$

for any $j \in \{0, \ldots, r-1\}$, and 3) C does not add any value in this representation. Therefore, we prove (6.3) and equivalently (6.1). Since a range proof already decomposes a value to its bits, from a practical perspective, we replace the reconstruction part of the range proof by the proof that shows C is well-formed. Importantly, though, we do not need to use a very large modulus since the modulus just needs to be large enough to guarantee that (6.4) holds over $\mathbb{Z}$, which is a very weak condition in a practical RingCT system.

The reason why we cannot simply use (6.2) is that, when amounts are represented by commitments to their bits, the addition of the commitments adds the corresponding bits over $\mathbb{Z}_q$ where $q \gg 2$. Therefore, for $\mathsf{Bits}(a)$ denoting the bits of a positive integer a,

$$\text{Com}(\mathsf{Bits}(a_1); *) + \text{Com}(\mathsf{Bits}(a_2); *) \neq \text{Com}(\mathsf{Bits}(a_1 + a_2); *),$$

and hence the proof does not work without the corrector C.

6.2.4 New extractable commitment

Our extractable commitment can be seen as a bridge between an LWE-based encryption, and a SIS-based commitment scheme with a "full trapdoor", i.e., the commitment matrix $\boldsymbol{A}$ is constructed in a way that a trusted party knows a matrix $\boldsymbol{G}$ such that $\boldsymbol{G} \cdot \boldsymbol{A} = \boldsymbol{0} \bmod q$. The disadvantage of an encryption scheme is that it does not allow compression (since there is unique decryption). As a result, it is inefficient to encrypt long messages whereas we want to have a compact commitment to long message vectors, i.e., we require a compressing commitment. The most promising candidate for this task is HMC, which can be seen as a commitment to the hash of the message while still preserving the algebraic structure. Now, if one puts a full trapdoor to HMC, then the recovered information via annihilating the randomness part with the trapdoor would be only the hash of the message, not the message itself. Then, one still has to recover the original message from here. Additionally, putting a full trapdoor often requires more aggressive parameters than those sufficient for the system without the trapdoor.

These bring us to our idea of using a "mini trapdoor". Suppose that $C = \boldsymbol{A}\boldsymbol{r} + \boldsymbol{B}\boldsymbol{m}$ is a commitment to a message vector $\boldsymbol{m}$ with a randomness vector $\boldsymbol{r}$ and a uniformly random matrix $\boldsymbol{B}$. The idea now works as follows. We construct $\boldsymbol{A}$ an as LWE matrix such that $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A'} \\ \boldsymbol{t}^\top \end{bmatrix}$ where $\boldsymbol{t} = \boldsymbol{A'}^\top \boldsymbol{s} + \boldsymbol{e}$ for some secret $\boldsymbol{s}$ and error $\boldsymbol{e}$ known only to the message extractor and $\boldsymbol{A'}$ is uniformly random. To extract a message from C, the extractor computes $\langle (\boldsymbol{s}, -1), C \rangle$, which is equal to $-\langle \boldsymbol{e}, \boldsymbol{r} \rangle + \langle \boldsymbol{b}, \boldsymbol{m} \rangle$ for $\boldsymbol{b} = (\boldsymbol{s}, -1)^\top \boldsymbol{B}$. Then, the idea is to let the extractor iterate through all the possible messages $\boldsymbol{m'}$ and compute $\langle (\boldsymbol{s}, -1), C \rangle - \langle \boldsymbol{b}, \boldsymbol{m'} \rangle$. For the correct message, the result will be $e' = -\langle \boldsymbol{e}, \boldsymbol{r} \rangle$, and for an incorrect message, it will be a random element in the ring R_q since $\boldsymbol{B}$ is an independent uniformly random matrix and $\boldsymbol{m} - \boldsymbol{m'} \neq \boldsymbol{0}$. Therefore, we can set the parameters so that $\| \langle (\boldsymbol{s}, -1), C \rangle - \langle \boldsymbol{b}, \boldsymbol{m'} \rangle \|_\infty$ is small only for the correct message with an overwhelming probability, which allows the extractor to recover the message. Furthermore, from M-LWE problem, $\boldsymbol{A}$ is computationally indistinguishable from random, and thus hiding property of HMC can still be used.

TABLE 6.1: Notations for the RingCT formal model.

$\mathbb{S}$	the blockchain state
$\mathsf{act} = (\mathsf{pk}, \mathsf{cn})$	an account comprised of a public key and a coin
$M, S \geq 1$	the number of spender's input and output accounts, resp.
$N \geq 2$	the number of accounts to hide a single input account
R_{in}	the set of spender's real accounts
$\mathsf{K}_{\text{in}} = (\mathsf{SK}_{\text{in}}, \mathsf{CK}_{\text{in}}, \mathsf{Amt}_{\text{in}})$	the set of spender's account secret keys $\mathsf{ask} = (\mathsf{sk}, \mathsf{cnk}, \mathsf{amt})$ with a secret key, coin key & amount
A_{in}	the set of all input accounts arranged as a $M \times N$ matrix where the i-th row contains $\mathsf{R}_{\text{in}}[i]$
PK_{out}	the set of output public keys with $\lvert\mathsf{PK}_{\text{out}}\rvert = S$
CN_{out}	the set of output coins with $\lvert\mathsf{PK}_{\text{out}}\rvert = S$
$\mathsf{Amt}_{\text{out}}$	the set of output amounts with $\lvert\mathsf{Amt}_{\text{out}}\rvert = S$
CK_{out}	the set of output coin keys with $\lvert\mathsf{CK}_{\text{out}}\rvert = S$
A_{out}	the set of output accounts with $\lvert\mathsf{A}_{\text{out}}\rvert = S$
Π	the proof output
SN	the set of serial numbers
tx	a transaction $\mathsf{tx} = (\mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{CN}_{\text{out}}, \Pi, \mathsf{SN})$
V	the set of all valid amounts

6.3 Formal Definitions for RingCT-like Cryptocurrency Protocols

In this section, we describe our formal definitions for RingCT-like protocols. First, we introduce the notation used specifically for the security model in Table 6.1. The blockchain state $\mathbb{S}$ consists of two lists: 1) a list of registered accounts $\mathsf{act} = (\mathsf{pk}, \mathsf{cn})$, indicating a public key pk is paired with a coin cn, and 2) a list of all verified transactions. We assume that $\mathbb{S}$ is properly updated among all users at all times.[3] The following tuple of polynomial time algorithms define RingCT protocol.

- $pp \leftarrow \textbf{Setup}(1^\lambda)$: given the security parameter λ, output the system parameters pp, which is assumed to be an implicit input to all the remaining functions.
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow \textbf{KeyGen}()$: output a public-secret key pair $(\mathsf{pk}, \mathsf{sk})$.
- $\mathsf{s} \leftarrow \textbf{SerialGen}(\mathsf{sk})$: on input a secret key sk, output a serial number s associated to sk.
- $(\mathsf{cn}, \mathsf{cnk})/\ \perp \leftarrow \textbf{Mint}(\mathsf{amt})$: on input an amount amt, if $\mathsf{amt} \in \mathsf{V}$, output a coin cn and its coin key cnk. Otherwise, output $\perp$. If cnk is given as an input, then **Mint** computes a deterministic function such that $\mathsf{cn} = \textbf{Mint}(\mathsf{amt}, \mathsf{cnk})$.
- $(\mathsf{act}, \mathbb{S}) \leftarrow \textbf{AccountGen}(\mathsf{pk}, \mathsf{cn}, \mathbb{S})$: on input a public key pk and a coin cn, register an account $\mathsf{act} = (\mathsf{pk}, \mathsf{cn})$ to the blockchain state $\mathbb{S}$. Output act and updated state $\mathbb{S}$.
- $0/1 \leftarrow \textbf{CheckAct}(\mathsf{pk}, \mathsf{cn}, \mathbb{S})$: on input a public key pk, a coin cn and the blockchain state $\mathbb{S}$, output 1 if $(\mathsf{pk}, \mathsf{cn})$ is a registered account in $\mathbb{S}$. Otherwise, output 0. In the case that the input has a set of pairs of $(\mathsf{pk}, \mathsf{cn})$, then output 1 if all $(\mathsf{pk}, \mathsf{cn})$ pairs are registered accounts in $\mathbb{S}$. Otherwise, output 0.
- $(\mathsf{tx}, \mathsf{CK}_{\text{out}}) \leftarrow \textbf{Spend}(\mathsf{A}_{\text{in}}, \mathsf{R}_{\text{in}}, \mathsf{K}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{Amt}_{\text{out}})$: on input $\mathsf{A}_{\text{in}}, \mathsf{R}_{\text{in}}, \mathsf{K}_{\text{in}}, \mathsf{PK}_{\text{out}}$, and $\mathsf{Amt}_{\text{out}}$ as in Table 6.1, mint output coins by running $(\mathsf{CN}_{\text{out}}, \mathsf{CK}_{\text{out}}) \leftarrow$

[3] In practice, this is managed by a consensus algorithm, which is outside the scope of this work.

TABLE 6.2: Structure of the list $\mathcal{L}$ used in the RingCT security model.

$\mathcal{L}$:	pk	sk	s (serial #)	cn	cnk	amt	IsCrpt

Mint$(\mathsf{Amt}_{\text{out}})$. Generate the serial numbers by running $\mathsf{SN} \leftarrow$ **SerialGen**$(\mathsf{SK}_{\text{in}})$ and a proof Π. Output $(\mathsf{tx}, \mathsf{CK}_{\text{out}}) = ((\mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{CN}_{\text{out}}, \Pi, \mathsf{SN}), \mathsf{CK}_{\text{out}})$.[4]

- $0/1 \leftarrow$ **IsSpent**$(\mathsf{SN}, \mathbb{S})$: on input a set SN of serial numbers and the blockchain state $\mathbb{S}$, if there is a collision in SN or if a serial number appears both in SN and $\mathbb{S}$, output 1. Otherwise, output 0.
- $\emptyset/(\mathsf{A}_{\text{out}}, \mathbb{S}) \leftarrow$ **Verify**$(\mathsf{tx}, \mathbb{S})$: on input a transaction tx as in Table 6.1, if **IsSpent**$(\mathsf{SN}, \mathbb{S}) = 1$ or **CheckAct**$(\mathsf{A}_{\text{in}}, \mathbb{S}) = 0$, output $\emptyset$. Check the proof Π and output $\emptyset$ if not valid. Otherwise, run $(\mathsf{A}_{\text{out}}, \mathbb{S}) \leftarrow$ **AccountGen**$(\mathsf{PK}_{\text{out}}, \mathsf{CN}_{\text{out}}, \mathbb{S})$ and add tx to $\mathbb{S}$. Output $\mathsf{A}_{\text{out}} \neq \emptyset$ and updated $\mathbb{S}$.

One of the most important differences of our definitions to RingCT 2.0 [SALY17] and 3.0 [YSL+19] is that some of the functions take the blockchain state $\mathbb{S}$ as an input to capture the inherent stateful nature of a blockchain environment. However, this important piece is completely missing in RingCT 2.0 and 3.0, and sometimes used implicitly in the definitions without having it as an input. For the tuple of algorithms that define the protocol, we additionally have the functions **SerialGen**, **CheckAct** and **IsSpent**, which do not exist in RingCT 2.0 or 3.0. Therefore, in the correctness definitions of RingCT 2.0 and 3.0, there is no restriction on input accounts being unspent whereas there should be such a restriction (see our correctness definition further below).

We consider an account as a registered public key and coin pair on blockchain. Therefore, our **Spend** algorithm does not output *accounts* as the transaction would not have been validated at that point yet. Hence, **Verify** takes public key and coin pairs as inputs, and outputs the accounts if the input transaction is valid. On the other hand, **Spend** algorithms in RingCT 2.0 and 3.0 directly output *accounts*. Also, **Mint** algorithm in RingCT 2.0 and 3.0 take a public key as an input, but does not make use of it.

6.3.1 Security Definitions

Towards getting a "cleaner" model, we only use the single list $\mathcal{L}$ in Table 6.2 instead of five lists as in RingCT 2.0 and 3.0. The list $\mathcal{L}$ is seen as a database for which any of the following can be used as a unique identifier of a row: a public key, a secret key, a serial number, a coin or a coin key. Retrieving a row in $\mathcal{L}$ is denoted, for example, by $\mathcal{L}[\mathsf{pk}]$ for some public key pk. Then, $\mathcal{L}[\mathsf{pk}].\mathsf{cnk}$ denotes the coin key associated with the public key pk. IsCrpt denotes the "is corrupted" tag.

Oracles. The oracles accessed by an adversary $\mathcal{A}$ are defined below.

- PKGEN(i) : on the i-th query, run $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow$ **KeyGen**$()$, $\mathsf{s} \leftarrow$ **SerialGen**(sk_i) and output pk_i. Add $(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{s})$ to $\mathcal{L}$ where IsCrpt tag is set to zero and the remaining fields are left empty.
- MINT(amt) : run $(\mathsf{cn}, \mathsf{cnk}) \leftarrow$ **Mint**(amt), and output cn.
- ACTGEN$(\mathsf{pk}, \mathsf{amt}, \mathbb{S})$: run $(\mathsf{cn}, \mathsf{cnk}) \leftarrow$ **Mint**(amt) and $(\mathsf{act}, \mathbb{S}) \leftarrow$ **AccountGen**$(\mathsf{pk}, \mathsf{cn}, \mathbb{S})$. Insert $(\mathsf{cn}, \mathsf{cnk}, \mathsf{amt})$ to $\mathcal{L}[\mathsf{pk}]$ and output $(\mathsf{act}, \mathbb{S})$.
- CORRUPT(act) : For $\mathsf{act} = (\mathsf{pk}, \mathsf{cn})$, if $\mathcal{L}[\mathsf{pk}]$ cannot be found, return $\perp$, indicating failure. Otherwise, update $\mathcal{L}[\mathsf{pk}].\mathsf{IsCrpt}$ to 1, and output $\mathcal{L}[\mathsf{pk}].\mathsf{sk}$, $\mathcal{L}[\mathsf{pk}].\mathsf{cnk}$ and $\mathcal{L}[\mathsf{pk}].\mathsf{amt}$. Alternatively, the input may be either pk alone or cn alone. In the

[4] CK_{out} along with the output amounts are delivered to the recipient(s) privately.

former case, only $\mathcal{L}[\mathsf{pk}].\mathsf{sk}$ is returned, and in the latter, $\mathcal{L}[\mathsf{cn}].\mathsf{cnk}$ and $\mathcal{L}[\mathsf{cn}].\mathsf{amt}$ are returned.

- $\textsc{Spend}(\mathsf{A}_{\text{in}}, \mathsf{R}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{Amt}_{\text{out}})$: Retrieve from $\mathcal{L}$ all account secret keys K_{in} associated to R_{in}. Run $(\mathsf{tx}, \mathsf{CK}_{\text{out}}) \leftarrow \mathbf{Spend}(\mathsf{A}_{\text{in}}, \mathsf{R}_{\text{in}}, \mathsf{K}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{Amt}_{\text{out}})$ and $B \leftarrow \mathbf{Verify}(\mathsf{tx}, \mathbb{S})$. If $B = \emptyset$ (i.e., the verification fails), return $\perp$. Otherwise, return tx and, for each $1 \leq i \leq |\mathsf{PK}_{\text{out}}|$, update the coin, coin key and amount information in $\mathcal{L}[\mathsf{PK}_{\text{out}}[i]]$ with $\mathsf{CN}_{\text{out}}[i]$, $\mathsf{CK}_{\text{out}}[i]$ and $\mathsf{Amt}_{\text{out}}[i]$, respectively.

We let $\textsc{Orc}$ denote the set of all oracles defined above together with the random oracle. With respect to the positioning of the accounts in R_{in} inside A_{in}, we define two flavours of properties for RingCT: 1) *with shuffling* and 2) *without shuffling*. In the latter case, all the accounts in R_{in} are restricted to be in the same column, which provides a somewhat weaker level of anonymity as described in [YSL+19]. We give our definitions for the former case, and it is trivial to get the latter by imposing the aforementioned restriction on R_{in}.

Correctness

Informally, correctness requires that any user is able to spend any of her honestly generated *unspent* accounts, which has honestly generated keys and coins with a valid amount.

A RingCT protocol is said to be ϵ-*correct* if the following holds for any $pp \leftarrow \mathbf{Setup}(1^\lambda)$, any $M, N, S \in \mathbb{Z}^+$, $(\mathsf{pk}_0, \mathsf{sk}_0), \ldots, (\mathsf{pk}_{M-1}, \mathsf{sk}_{M-1}) \leftarrow \mathbf{KeyGen}(pp)$ such that $\mathbf{IsSpent}(\mathbf{SerialGen}(\mathsf{sk}_i)) = 0$ for all $i = 0, \ldots, M-1$, any $\mathsf{amt}_0, \ldots, \mathsf{amt}_{M-1}, \mathsf{amt}_{\text{out},0}, \ldots, \mathsf{amt}_{\text{out},S-1} \in \mathsf{V}$ such that $\sum_{i=0}^{M-1} \mathsf{amt}_i = \sum_{i=0}^{S-1} \mathsf{amt}_{\text{out},i}$, any set PK_{out} of arbitrarily generated output public keys and any set $\mathsf{A}_{\text{in}} \setminus \mathsf{R}_{\text{in}}$ of arbitrarily generated decoy accounts,

$$\Pr\left[\begin{array}{c}\mathbf{Verify}(\mathsf{tx}, \mathbb{S}) \neq \emptyset \ : \\ (\mathsf{tx}, \mathsf{CK}_{\text{out}}) \leftarrow \mathbf{Spend}(\mathsf{A}_{\text{in}}, \mathsf{R}_{\text{in}}, \mathsf{K}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{Amt}_{\text{out}})\end{array}\right] \geq 1 - \epsilon$$

where $\mathsf{cn}_i = \mathbf{Mint}(\mathsf{amt}_i, \mathsf{cnk}_i)$ for some cnk_i's in the domain of coin keys, $\mathsf{act}_i \leftarrow \mathbf{AccountGen}(\mathsf{pk}_i, \mathsf{cn}_i)$ for $i = 0, \ldots, M-1$, $\mathsf{R}_{\text{in}} = \{\mathsf{act}_0, \ldots, \mathsf{act}_{M-1}\}$, $\mathsf{Amt}_{\text{out}} = \{\mathsf{amt}_{\text{out},0}, \ldots, \mathsf{amt}_{\text{out},S-1}\}$, A_{in} and tx are as in Table 6.1, and $\mathsf{K}_{\text{in}} = \{(\mathsf{sk}_0, \mathsf{cnk}_0, \mathsf{amt}_0), \ldots, (\mathsf{sk}_{M-1}, \mathsf{cnk}_{M-1}, \mathsf{amt}_{M-1})\}$. If $\epsilon = 0$, then the protocol is said to be *perfectly correct*. If $\epsilon = \text{negl}(\lambda)$, then it is said to be *statistically correct*.

Observe from the above correctness definition that the spent input coins may not be generated honestly, but the input amounts and coin keys are in the correct domains. Indeed, the input coins spent by a user, say Alice, are the output coins of a previous transaction and thus are generated by another user, say Bob. Therefore, Alice cannot guarantee that Bob generated the coins honestly. However, Alice also receives the coin keys and amounts for these coins and can easily check if they are in the correct domains and whether the coins can be spent. Therefore, the correctness property alone does not guarantee that any received coins can be spent. This aspect can be captured in a *security* property (such as availability), which could require any coin output by a *verified* transaction to be "spendable". As in prior models RingCT 2.0 and 3.0, our model does not include such an availability property.

In the correctness definition of RingCT 3.0, the amounts are randomly sampled from $\mathbb{Z}_p$. However, in our case, the correctness requires any amount in the valid range V to be able to be spent.

Anonymity

Informally, anonymity requires that the real spender's accounts are hidden among the uncorrupted (i.e., never been queried to CORRUPT) accounts as long as there are at least two sets of uncorrupted input accounts that can be successfully spent.

A RingCT protocol is said to be *anonymous* if the following holds for all PPT adversaries $\mathcal{A}$ and $pp \leftarrow \mathbf{Setup}(1^\lambda)$

$$\Pr\left[\mathcal{A} \text{ wins the game } \texttt{Exp:Anonymity}\right] \leq 1/2 + \mathrm{negl}(\lambda),$$

where `Exp:Anonymity` is defined as follows.

1. $(\mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{Amt}_{\text{out}}, \mathsf{R}^0_{\text{in}}, \mathsf{R}^1_{\text{in}}, \mathsf{st}) \leftarrow \mathcal{A}^{\text{ORC}}(pp)$: $\mathcal{A}$ is given pp and access to all oracles, and then outputs two target sets of accounts to be spent as $(\mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{Amt}_{\text{out}}, \mathsf{R}^0_{\text{in}}, \mathsf{R}^1_{\text{in}}, \mathsf{st})$ where
   - st is some state information to be used by $\mathcal{A}$ in the next stage,
   - $\mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}$ and $\mathsf{Amt}_{\text{out}}$ are as in Table 6.1,
   - $\mathsf{R}^0_{\text{in}}, \mathsf{R}^1_{\text{in}} \subset \mathsf{A}_{\text{in}}$ such that both R^0_{in} and R^1_{in} contain exactly one account from each row of A_{in}.
2. $(\mathsf{tx}_i, \mathsf{CK}^i_{\text{out}}) \leftarrow \mathbf{Spend}(\mathsf{A}_{\text{in}}, \mathsf{R}^i_{\text{in}}, \mathsf{K}^i_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{Amt}_{\text{out}})$ for $i = 0, 1$: Both sets R^0_{in} and R^1_{in} of real input accounts are spent with the arguments specified by $\mathcal{A}$ where
   - K^i_{in} is the set of account secret keys of the accounts in R^i_{in} retrieved from $\mathcal{L}$ for $i = 0, 1$.

   If $\mathbf{Verify}(\mathsf{tx}_i, \mathbb{S}) = \emptyset$ for some $i \in \{0, 1\}$, then set $\mathsf{tx}_0 = \mathsf{tx}_1 = \perp$.
3. $b \leftarrow \{0, 1\}$
4. $b' \leftarrow \mathcal{A}^{\text{ORC}}(\mathsf{tx}_b, \mathsf{CK}^b_{\text{out}}, \mathsf{Amt}^0_{\text{in}}, \mathsf{Amt}^1_{\text{in}}, \mathsf{st})$: $\mathcal{A}$ is given access to all the oracles, the state st, one of the **Spend** outputs, and the input amounts in K^0_{in} and K^1_{in}. Then, $\mathcal{A}$ outputs a guess for the real input of the **Spend** output provided.

$\mathcal{A}$ wins the game `Exp:Anonymity` if the following holds

- all public keys and coins in R^0_{in} and R^1_{in} are generated by PKGEN and MINT, respectively, and all accounts in R^0_{in} and R^1_{in} are generated by ACTGEN,
- all public keys in PK_{out} are generated by PKGEN,
- $\mathsf{tx}_0 \neq \perp$ and $\mathsf{tx}_1 \neq \perp$,
- no account (including its public key and coin) in R^0_{in} or R^1_{in} has been corrupted (i.e., queried to CORRUPT),
- $(\cdot, \mathsf{R}^i_{\text{in}} \cdot, \cdot)$ has never been queried to SPEND for $i = 0, 1$,
- $b' = b$.

Note that the adversary is restricted to corrupting at most $N - 2$ accounts in any row of A_{in} by making sure that R^0_{in} and R^1_{in} have all uncorrupted accounts. Further, instead of having two sub-definitions as in RingCT 3.0, we define a single anonymity experiment that covers different attack scenarios. Our definition is based on an indistinguishability argument, which makes it easier to extend the anonymity proofs of the ring signature used as a building block. Moreover, in our anonymity definition, only the accounts in R_{in} are assumed to be honestly generated, not all those in A_{in}.

Balance

Informally, balance requires that no adversary can spend a set A of accounts under his control such that the sum of output amounts is more than the sum of the amounts in A.

A RingCT protocol is said to be *balanced* if the following holds for all PPT adversaries $\mathcal{A}$ and $pp \leftarrow \mathbf{Setup}(1^\lambda)$

$$\Pr[\mathcal{A} \text{ wins the game } \texttt{Exp:Balance}] \leq \mathrm{negl}(\lambda),$$

where `Exp:Balance` is defined as follows.

1. $(\mathsf{tx}_1, \mathsf{Amt}^1_{\text{out}}, \mathsf{CK}^1_{\text{out}}), \cdots, (\mathsf{tx}_t, \mathsf{Amt}^t_{\text{out}}, \mathsf{CK}^t_{\text{out}}) \leftarrow \mathcal{A}^{\textsc{Orc}}(pp)$: The adversary $\mathcal{A}$ is given access to all the oracles $\textsc{Orc}$ together with pp, and outputs a set of t transactions where
   - $\mathsf{tx}_i = (\mathsf{A}^i_{\text{in}}, \mathsf{PK}^i_{\text{out}}, \mathsf{CN}^i_{\text{out}}, \Pi_i, \mathsf{SN}_i)$ for $i = 1, \ldots, t$,
   - $\mathsf{Amt}^i_{\text{out}}$'s and $\mathsf{CK}^i_{\text{out}}$'s are sets of output amounts and coin keys, respectively, for *uncorrupted* output public keys with $|\mathsf{CK}^i_{\text{out}}| = |\mathsf{Amt}^i_{\text{out}}| \leq |\mathsf{PK}^i_{\text{out}}| = |\mathsf{CN}^i_{\text{out}}|$ for all $i \in \{1, \ldots, t\}$.
2. $B_i \leftarrow \mathbf{Verify}(\mathsf{tx}_i, \mathbb{S})$ for $i = 1, \ldots, t$.

$\mathcal{A}$ wins the game `Exp:Balance` if the following holds

- for all $i \in \{1, \ldots, t\}$, all public keys and coins in A^i_{in} are generated by $\textsc{PkGen}$ and $\textsc{Mint}$, respectively, and all accounts in A^i_{in} are generated by $\textsc{ActGen}$,
- $\bigcap_{i=1}^t \mathsf{SN}_i = \emptyset$,
- $B_i \neq \emptyset$ for all $i = 1, \ldots, t$,
- there exists a $j^* \in [1, t]$ such that $\sum_{i=0}^{S'-1} \mathsf{Amt}^{j^*}_{\text{out}}[i] > \sum_{i=0}^{M-1} \mathsf{amt}_{\text{in},i}$ where $S' = |\mathsf{Amt}^{j^*}_{\text{out}}|$, $M = |\mathsf{SN}_{j^*}|$, $\mathsf{amt}_{\text{in},i} = \mathcal{L}[\mathsf{s}_i].\mathsf{amt}$ for all $\mathsf{s}_i \in \mathsf{SN}_{j^*}$ if $\mathsf{s}_i \in \mathcal{L}$ and $\mathcal{L}[\mathsf{s}_i].\mathsf{IsCrpt} = 1$, and $\mathsf{amt}_{\text{in},i} = 0$ otherwise,
- for any $i \in [1, t]$ and $0 \leq j < |\mathsf{PK}^i_{\text{out}}|$, if $\mathcal{L}[\mathsf{pk}_{i,j}].\mathsf{IsCrpt} = 0$ for $\mathsf{pk}_{i,j} = \mathsf{PK}^i_{\text{out}}[j]$, then $\mathsf{CK}^i_{\text{out}}[j] = \mathcal{L}[\mathsf{pk}_{i,j}].\mathsf{cnk}$, $\mathsf{Amt}^i_{\text{out}}[j] = \mathcal{L}[\mathsf{pk}_{i,j}].\mathsf{amt}$ and $\mathsf{CN}^i_{\text{out}}[j] = \mathbf{Mint}(\mathsf{Amt}^i_{\text{out}}[j], \mathsf{CK}^i_{\text{out}}[j])$.[5] That is, for all *uncorrupted* output public keys, the corresponding output coin key, output amount and output coin provided by the adversary are correct.

In `Exp:Balance`, the output of the adversary does not include information about the output coin key or output amount for the *corrupted* output public keys. The reason that the adversary needs to output such information for *uncorrupted* output public keys is that the honest recipient checks whether the output coin key and output amount are in the correct domains and construct the coin. Clearly, the adversary may corrupt all the output public keys, in which case, he would not need to output CK_{out} or $\mathsf{Amt}_{\text{out}}$.

Attack scenarios of the balance model.

1. **Forgery:** The attacker tries to create a valid proof where (at least) one of the real spent accounts is not corrupted. This is captured by setting an input amount to zero if no corruption occurs with respect to a certain serial number.
2. **Unbalanced input and output amounts:** The attacker tries to create a transaction where the sum of input amounts being spent does not match the sum of output amounts. This is captured by letting the attacker corrupt all the input accounts.

[5] Without loss of generality, we assume that the indices for corrupted public keys are the last ones so that the indexing matches.

3. **Double spending:** The attacker tries to spend an account twice with distinct serial numbers. This is captured by setting an input amount to zero if the respective serial number is not in $\mathcal{L}$.

Our balance definition is presented as a single experiment rather than having sub-definitions. For example, RingCT 3.0 has three sub-cases of balance: unforgeability, equivalence and linkability. Further, the unforgeability definition in RingCT 3.0 requires *all* input accounts in A_{in} to be uncorrupted. However, in this case, a natural forgery attack where the attacker has control over a single input account (which may not even be the real spent one) is excluded from the model. In our balance definition, on the other hand, the adversary wins the game -among other cases- if there is only a single uncorrupted account for which a valid serial number is generated by the adversary. Further, our balance definition allows an adversary to output a set of transactions (where one transaction can possibly be an input to the other), while only the linkability definition in RingCT 3.0 allows just two transactions to be output.

In RingCT 2.0/3.0 formal models (and also in LRCT v2.0), there is an additional property, *non-slanderability*. It states "it is infeasible for any malicious user to produce a valid spending that shares at least one serial number with a *previously* generated *honest* spending" [SALY17]. Although non-slanderability could be a requirement in some applications of a linkable ring signature where the users are punished if a signature is detected to be generated twice using the same secret, we do not believe that is the case in the RingCT setting. The reason is that if someone generates a spending that has the same serial number with a *previously* generated one, then simply the second spending does not verify and thus ignored with no punishment in regards to the first spending. Hence, even if an attacker succeeds in winning the above non-slanderability game, there is no harm to honest users nor is there any gain for the attacker.

In our formal definitions, we aimed to explicitly state our assumptions so that the model can further be easily strengthened in future by removing some of them. For example, a potential extension is to remove the assumption in `Exp:Balance` that the input coins are generated honestly. This would require using the soundness of the preceding transaction proofs (in addition to the "current" one), complicating the balance analysis even further. Thus, in this work, such an assumption is included as in RingCT 2.0 and 3.0.

6.4 MatRiCT: Efficient, Scalable and Post-Quantum Confidential Transactions Protocol

Table 6.3: Identifiers for MatRiCT.

Notation	Explanation
R_q, $R_{\hat{q}}$	Cyclotomic rings of degree d : $R_q = \mathbb{Z}_q[X]/(X^d+1)$, $R_{\hat{q}} = \mathbb{Z}_{\hat{q}}[X]/(X^d+1)$
$n, \hat{n}$	height of commitment matrices in R_q and $R_{\hat{q}}$, resp
$m, \hat{m}$	randomness vector dimensions in R_q and $R_{\hat{q}}$, resp
$N = \beta^k$	ring size of ring signature (and anonymity set size)
ℓ	spender's column index with $0 \leq \ell < N$
r	bit-length of an amount, i.e., $\mathsf{amt} \in \mathsf{V} = [0, 2^r - 1]$
$\mathcal{B}$	max. absolute coefficient of initial randomness

6.4.1 Description of MatRiCT

We describe the full details of MatRiCT in this section. We specify the functions **Setup**, **KeyGen**, **SerialGen**, **Mint**, **Spend** and **Verify**. To simplify presentation, the part concerning the corrector values in **Spend** is shown for the case $(M, S) = (1, 2)$ in Algorithm 6.8 and we discuss in the text how the general case can be easily accomplished. Let us go over the description of each algorithm one-by-one and fix the notation in Table 6.3. We assume that r-bit precision is always sufficient for the amounts (even when they are summed)[6] and that valid amounts are used to call the functions. It is trivial to return an error when that is not the case.

In general, there are 3 commitment keys $\boldsymbol{G}, \hat{\boldsymbol{G}}$ and $\boldsymbol{H}$ used in the system where $\boldsymbol{H}$ (defined over R_q) is used only in the serial number generation, $\hat{\boldsymbol{G}}$ is used for the commitments over $R_{\hat{q}}$ in binary proof and $\boldsymbol{G}$ is used elsewhere for the commitments over R_q.

Algorithm 6.1 $\mathsf{SamMat}(\rho', v, q', n', m', \mathrm{str})$

INPUT: ρ' for some seed $\rho' \in \{0,1\}^{256}$; $v, q', n', m' \in \mathbb{Z}^+$; str is an optional auxiliary input string.

1: $\boldsymbol{G} \leftarrow \mathsf{Sam}(\rho', \mathrm{str})$ where $\boldsymbol{G} \in R_{q'}^{n' \times (m'+v)}$
2: **return** $\boldsymbol{G}$ ▷ $\boldsymbol{G}$ can be output in the NTT domain.

Algorithm 6.1 generates a random matrix from a small seed ρ using an extendable output function Sam (modelled as a random oracle in the security analysis). It also has an auxiliary input string str so that different matrices from the same seed can be generated. Calling the function with the same seed ρ and same string str results in the generation of the same entries. For example, running $\boldsymbol{A} \leftarrow \mathsf{SamMat}(\rho, v_1, q, n_1, m_1, \mathrm{str})$ and $\boldsymbol{B} \leftarrow \mathsf{SamMat}(\rho, v_2, q, n_2, m_2, \mathrm{str})$ with $n_1 \leq n_2$, $m_1 + v_1 \leq m_2 + v_2$ results in two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ where $\boldsymbol{A}$ is a sub-matrix of $\boldsymbol{B}$.

Algorithm 6.2 $\mathbf{Setup}(1^\lambda)$ λ is the security parameter

1: Choose integer parameters $k, \beta, r, n, m, \hat{n}, \hat{m}, d, q, \hat{q}$ such that $N = \beta^k$
2: Set w, p such that $|\mathcal{C}^d_{w,p}| > 2^{256}$
3: $\rho \leftarrow \{0,1\}^{256}$
4: Pick a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathcal{C}^d_{w,p}$
5: **return** $pp = (\rho, \mathcal{H}, k, \beta, r, n, m, \hat{n}, \hat{m}, n_s, d, w, p, q, \hat{q})$

Algorithm 6.2 sets the system parameters. Here, for ease of presentation, we assume the ring size to be fixed to some N. The range of the hash function is defined as the following challenge space

$$\mathcal{C}^d_{w,p} = \{x \in \mathbb{Z}[X] : \deg(x) = d-1 \wedge \mathsf{HW}(x) = w \wedge \|x\|_\infty = p\}.$$

This is the same set defined in (5.9) and $|\mathcal{C}^d_{w,p}| = \binom{d}{w}(2p)^w$. Thus, given d, one can easily set (w, p) such that $|\mathcal{C}^d_{w,p}| > 2^{256}$.

[6] We do not assume here that summing multiple amounts of, say, $2^{64}-1$ is impossible. To be more explicit, r can be set as the smallest integer such that $\mathsf{MAX}_{\mathsf{amt}} \cdot \mathsf{MAX}_{\mathsf{io}} \leq 2^r - 1$ where $\mathsf{MAX}_{\mathsf{amt}}$ is the maximum amount possible and $\mathsf{MAX}_{\mathsf{io}}$ is the maximum number of input/output accounts allowed. But, the amount is still represented in r bits. Recall that our protocol does not have the disadvantage of requiring a modulus greater than $2^r - 1$ and a few more bits of precision can be added at almost no cost.

Algorithm 6.3 KeyGen(pp)

1: $\boldsymbol{G} \leftarrow \mathsf{SamMat}(\rho, 0, q, n, m, \text{“G”})$
2: $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{d \cdot m}$
3: $\boldsymbol{c} = \boldsymbol{G} \cdot \boldsymbol{r}$ in R_q^n ▷ Commitment to zero under $ck = \boldsymbol{G}$
4: **return** $(\mathsf{pk}, \mathsf{sk}) = (\boldsymbol{c}, \boldsymbol{r})$ ▷ Can be output in the NTT domain

Algorithm 6.3 generates a public-secret key pair. The secret key is random vector over R_q with infinity norm $\mathcal{B}$, and the public key is a commitment to zero with the secret key used as the randomness.

Algorithm 6.4 generates a serial number for a given secret key. The serial number is a commitment to zero using the secret key as the randomness under the commitment key $\boldsymbol{H}$. Observe that the height of the commitment matrix here is set to n_s.

Algorithm 6.5 implements minting a coin, which is computed as a commitment to the bits of an input amount. The commitment key used here is the same as the one in **KeyGen**.

Algorithm 6.4 SerialGen(sk) for a secret key $\mathsf{sk} \in R_q^m$

1: $\boldsymbol{H} \leftarrow \mathsf{SamMat}(\rho, 0, q, n_s, m, \text{“H”})$
2: $\boldsymbol{c} = \boldsymbol{H} \cdot \boldsymbol{r}$ in $R_q^{n_s}$ where $\boldsymbol{r} = \mathsf{sk} \in R_q^m$ ▷ Com. to zero under $\boldsymbol{H}$
3: **return** $\mathsf{s} = \boldsymbol{c}$ ▷ s can be output in the NTT domain

Algorithm 6.5 Mint(amt) for $\mathsf{amt} \in [0, 2^r - 1]$

1: $\boldsymbol{G} \leftarrow \mathsf{SamMat}(\rho, r, q, n, m, \text{“G”})$
2: $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{d \cdot m}$, $(b_0, \ldots, b_{r-1}) \leftarrow \mathsf{Bits}(\mathsf{amt})$
3: $C = \mathrm{Com}_{ck}(b_0, \ldots, b_{r-1}; \boldsymbol{r})$ in R_q^n where $ck = \boldsymbol{G}$
4: **return** $(\mathsf{cn}, \mathsf{cnk}) = (C, \boldsymbol{r})$

Since **Spend** algorithm is very long, we split it into multiple parts, Algorithms 6.6, 6.7, 6.8 and 6.9. **Spend** starts by setting some parameters in Algorithm 6.8. These settings are done to accommodate different parameters while keeping the acceptance rate of the rejection sampling similar. In Step 7, we compute the corrector values c_i where the division by two is always exact by the following lemma.

Lemma 6.1. *Let A, B be two sets of non-negative integers and $a = (a[0], \ldots, a[r-1])$ be the representation of a non-negative integer a in base $\beta \geq 2$. If $\sum_{a \in A} a = \sum_{b \in B} b$, then for any $\beta \geq 2$, there exists $c_0, \ldots, c_r \in [-(|B|-1), |A|-1]$ with $c_0 = 0$ such that*

$$\sum_{a \in A} a[i] - \sum_{b \in B} b[i] = \beta c_{i+1} - c_i.$$

Further, $c_r = 0$ if the result of the sum is of at most r digits in base β.

Proof. Let $\beta \geq 2$. For any set A of non-negative integers and any $0 \leq i < r$ where r is the maximum number of digits needed to represent elements in A, we can write

$$\left(\sum_{a \in A} a\right)[i] - c_i + \beta c_{i+1} = \sum_{a \in A} a[i] \tag{6.5}$$

for the carries $c_1, \ldots, c_r$ and $c_0 = 0$ since there is no carry for the least significant bit. Now, fix A, B as two sets of non-negative integers where the sum over A equals the

sum over B. Clearly, the following holds

$$\left(\sum_{a\in A} a\right)[i] = \left(\sum_{b\in B} b\right)[i] \tag{6.6}$$

for any $0 \leq i < r$. Using (6.5) and (6.6), for any $0 \leq i < r$, we get

$$\begin{aligned}
&\sum_{a\in A} a[i] - \sum_{b\in B} b[i] \\
&\quad = \left(\sum_{a\in A} a\right)[i] - c''_i + \beta c''_{i+1} - \left(\sum_{b\in B} b\right)[i] + c'_i - \beta c'_{i+1} \\
&\quad = -(c''_i - c'_i) + \beta(c''_{i+1} - c'_{i+1}),
\end{aligned}$$

for some carries $c'_0, c''_0, \ldots, c'_r, c''_r$ where $c'_0 = c''_0 = 0$. Defining $c_i := c''_i - c'_i$ with $c_0 = 0$ concludes the first part. Note that due to c''_i, c'_i being carry values, $c''_i \in [0, |A| - 1]$ and $c'_i \in [0, |B| - 1]$, and thus $c_i \in [-(|B| - 1), |A| - 1]$.

When neither of the sums exceed r digits, $c'_r = c''_r = 0$, and thus $c_r = 0$. □

After computation of the corrector values, the spender mints the output coins, and runs an aggregated binary proof using Algorithm 6.6. The idea for the binary proof is the same as in Section 5.5.1, but we apply our efficient rejection sampling technique for binary secrets of fixed Hamming weight and the binary proof here proves a slightly different relation given in Lemma 6.3. In general, Algorithm 6.6 takes t sequences of bits where each sequence has s_j elements, and the masking values for each sequence is sampled from $\mathfrak{U}_{\mathcal{B}_j}$. Also, each sequence has a flag Bool_j to indicate whether the sequence has a fixed Hamming weight, in which case the masking values for zero bits are sampled directly from the accepted distribution of rejection sampling. Note that for the case of Hamming weight equal to 1, there is always exactly one element in $\{a_1^{(j)}, \ldots, a_{s_j}^{(j)}\}$ not sampled from the accepted distribution.

Defining $\delta_{i,j}$ as the Kronecker's delta, Step 14 of Algorithm 6.8 computes the unary representation of the spender's index ℓ in base β, which has a fixed Hamming weight. Therefore, the flag Bool_j is set to True for these sequences. Then, we also add the corrector values and the output amount bits to the array $\boldsymbol{b}$, which is then input to the binary proof. The most common cases for the number of input/output accounts are $(M, S) = (1, 2)$ and $(M, S) = (2, 2)$. For the former case, the corrector values are binary as they are simply the carries in the sum of two output amounts. Therefore, the steps given in Algorithm 6.8 are sufficient. In the latter case, we can prove that the corrector values are differences of some bits $c_{\text{in},i}$, $c_{\text{out},i}$, which are the carries from the sum of two inputs and the sum of two outputs, respectively.

In general, however, the corrector values can fall in a larger interval $[-(M-1), S-1]$, and in that case, one needs to prove that they are indeed in that interval. This can be done using a standard range proof, where the range width is only $M + S - 1$, which is expected to be very small. In fact, we do not need to prove that they fall exactly in $[-(M-1), S-1]$, but can alternatively prove that they are in a range of width 2^l for $l = \lceil \log(M + S - 1) \rceil$. There are standard methods to "shift" the range at no cost (see Section 5.4.2). As mentioned in Section 6.2, as long as (6.4) is ensured to hold over $\mathbb{Z}$, the corrector values can be set freely. The final part of Algorithm 6.6 is committing to all the values.

Steps 26 and 27 of Algorithm 6.8 are used to prove that the corrector commitment C is well-formed, i.e., does not contain any value with respect to the representation

Algorithm 6.6 BinaryCommit ▷ Commitment Step of Binary Proof

INPUT: $t \in \mathbb{Z}^+$; $\{(s_j, \text{Bool}_j, (b_0^{(j)}, \ldots, b_{s_j-1}^{(j)}), \mathcal{B}_j)\}_{j=0}^{t-1}$ where $s_j \in \mathbb{Z}^+, \text{Bool}_j \in \{\text{True}, \text{False}\}, b_i^{(j)} \in \{0,1\}$; $\mathcal{B}, \hat{\mathcal{B}}_{\text{big}} \in \mathbb{Z}^+$.

OUTPUT: $(\boldsymbol{r}_a, \boldsymbol{r}_b), (A, B), \{(a_0^{(j)}, \ldots, a_{s_j-1}^{(j)})\}_{j=0}^{t-1}$ where $\boldsymbol{r}_a, \boldsymbol{r}_b \in R_{\hat{q}}^{\hat{m}}$, $A, B \in R_{\hat{q}}^{\hat{n}}$ and $a_i^{(j)} \in R_{\hat{q}}$.

1: $ck = \hat{\boldsymbol{G}} \leftarrow \mathsf{SamMat}(\rho, v, \hat{q}, \hat{n}, \hat{m}, \text{“Gbig”})$ for $v = 2 \cdot \left(\sum_{j=0}^{t-1} s_j\right)$
2: $\boldsymbol{r}_b \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{d \cdot \hat{m}}$
3: $\boldsymbol{r}_a \leftarrow \{-\hat{\mathcal{B}}_{\text{big}}, \ldots, \hat{\mathcal{B}}_{\text{big}}\}^{d \cdot \hat{m}}$
4: **for** $j = 0, \ldots, t-1$ **do** ▷ Iterate over each bit sequence
5: **if** $\text{Bool}_j = \text{True}$ **then** ▷ The case of HW being 1.
6: **if** $b_0^{(j)} = 0$, **then** $i^* = -1$ ▷ Out of $[1, s_j - 1]$ by default
7: **else** $i^* \leftarrow \{1, \ldots, s_j - 1\}$ and $a_{i^*}^{(j)} \leftarrow \{-\mathcal{B}_j, \ldots, \mathcal{B}_j\}^d$
8: **for** $i = 1, \ldots, s_j - 1$ and $i \neq i^*$ **do** ▷ i starts from 1.
9: **if** $b_i^{(j)} = 0$, **then** $a_i^{(j)} \leftarrow \{-(\mathcal{B}_j - p), \ldots, \mathcal{B}_j - p\}^d$
10: **else** $a_i^{(j)} \leftarrow \{-\mathcal{B}_j, \ldots, \mathcal{B}_j\}^d$
11: **end for**
12: $a_0^{(j)} = -\sum_{i=1}^{s_j-1} a_i^{(j)}$
13: **else**
14: **for** $i = 0, \ldots, s_j - 1$ **do** ▷ i starts from 0.
15: $a_i^{(j)} \leftarrow \{-\mathcal{B}_j, \ldots, \mathcal{B}_j\}^d$
16: **end for**
17: **end if**
18: **end for**
19: $\boldsymbol{b} = \left(b_0^{(0)}, \ldots, b_{s_{t-1}-1}^{(t-1)}\right)$, $\boldsymbol{a} = \left(a_0^{(0)}, \ldots, a_{s_{t-1}-1}^{(t-1)}\right)$
20: $\boldsymbol{c} = \left(a_0^{(0)}(1 - 2b_0^{(0)}), \ldots, a_{s_{t-1}-1}^{(t-1)}(1 - 2b_{s_{t-1}-1}^{(t-1)})\right)$
21: $\boldsymbol{d} = \left(-(a_0^{(0)})^2, \ldots, -(a_{s_{t-1}-1}^{(t-1)})^2\right)$
22: $B = \text{Com}_{ck}(\boldsymbol{b}, \boldsymbol{c}; \boldsymbol{r}_b)$, $A = \text{Com}_{ck}(\boldsymbol{a}, \boldsymbol{d}; \boldsymbol{r}_a)$ in $R_{\hat{q}}^{\hat{n}}$ with $ck = \hat{\boldsymbol{G}}$
23: **return** $(\boldsymbol{r}_a, \boldsymbol{r}_b), (A, B), \{(a_0^{(j)}, \ldots, a_{s_j-1}^{(j)})\}_{j=0}^{t-1}$

of the amounts. After that, the spender runs M ring signatures to prove ownership of an account from each row of A_{in}. Here, she also computes a serial number for each account spent. Finally, another ring signature is run to prove that the balance is preserved by showing $\sum_{i=0}^{S-1} \mathsf{cn}_{\text{out},i} - \sum_{i=0}^{M-1} \mathsf{cn}_{i,\ell} + C$ is a commitment to zero for the same index $\ell \in [0, N-1]$. Note that all ring signatures are run using the same vector $\boldsymbol{p}$, and thus the indices of the spender's accounts are the same in all rows (notice also that **Verify**, Algorithm 6.10, uses the same $f_{j,i}$'s in the verification of the ring signatures at Steps 22 and 28).

The main part of the ring signature is summarised in Algorithm 6.7, which follows the same blueprint as the one-out-of-many proof in Section 5.5.2, but again proves a slightly different relation given in Lemma 6.8. Additionally, when the ring signature is used to prove knowledge of a user secret key, Algorithm 6.7 also outputs elements $F_0, \ldots, F_{k-1}$ to be used in verification of the serial number. $p_{i,j} \in \boldsymbol{p}$ input to Algorithm 6.7 are defined as in (5.32). The computation of $p_{i,j}$'s is summarised in Algorithm 6.11 for $k \leq 2$. The final step of Algorithm 6.8 is hashing all the information up to

Algorithm 6.7 RingCommit ▷ Commitment Step of Ring Sign.

INPUT: GenSerial $\in \{\text{True}, \text{False}\}$; $(P_0, \ldots, P_{N-1})$ where $P_i \in R_q^n$; $(p_{0,0}, \ldots, p_{N-1,k-1})$ where $p_{i,j} \in R_q$; $\mathcal{B}, \mathcal{B}_{\text{big,k}} \in \mathbb{Z}^+$.
OUTPUT: $(\boldsymbol{\rho}_0, \ldots, \boldsymbol{\rho}_{k-1}), (E_0, F_0, \ldots, E_{k-1}, F_{k-1})$ where $\boldsymbol{\rho}_j \in R_q^m$, $E_j \in R_q^n$ and $F_j \in R_q^{n_s}$. F_j's are omitted when GenSerial = False.

1: $ck = \boldsymbol{G} \leftarrow \mathsf{SamMat}(\rho, 0, q, n, m, \text{“G”})$
2: **if** GenSerial = True, **then** $\boldsymbol{H} \leftarrow \mathsf{SamMat}(\rho, 0, q, n_s, m, \text{“H”})$
3: $\boldsymbol{\rho}_0 \leftarrow \{-\mathcal{B}_{\text{big,k}}, \ldots, \mathcal{B}_{\text{big,k}}\}^{d \cdot m}$
4: **for** $j = 0, \ldots, k-1$ **do**
5: $\quad \boldsymbol{\rho}_j \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{d \cdot m}$ **if** $j \neq 0$
6: $\quad R_j = \text{Com}_{ck}\left(\mathbf{0}; \boldsymbol{\rho}_j\right)$ in R_q^n
7: $\quad E_j = \sum_{i=0}^{N-1} p_{i,j} P_i + R_j$ in R_q^n
8: $\quad$ **if** GenSerial = True, **then** $F_j = \boldsymbol{H} \cdot \boldsymbol{\rho}_j$ in $R_q^{n_s}$
9: **end for**
10: **if** GenSerial = True, **then return** $(\boldsymbol{\rho}_0, \ldots, \boldsymbol{\rho}_{k-1}), (E_0, F_0, \ldots, E_{k-1}, F_{k-1})$
11: **return** $(\boldsymbol{\rho}_0, \ldots, \boldsymbol{\rho}_{k-1}), (E_0, \ldots, E_{k-1})$

that step, where the hash function is modelled as a random oracle.

The second part of **Spend** (Algorithm 6.9) is comprised of the spender's masked responses of the underlying ZKP. Each bit input to the binary proof is masked by the corresponding $a \in R_{\hat{q}}$, and the rejection sampling technique from [Lyu09] is applied. The general idea works as follows. If we have a vector $\boldsymbol{y} = \boldsymbol{s} + \boldsymbol{v}$ where $\boldsymbol{s}$ is the secret-dependent part and $\boldsymbol{v} \leftarrow \{-B, \ldots, B\}^t$ for some $B, t \in \mathbb{Z}^+$, then rejection happens when $\|\boldsymbol{y}\|_\infty > B - \|\boldsymbol{s}\|_\infty$. To have a small rejection probability, we set $B = c \cdot \|\boldsymbol{s}\|_\infty \cdot t$ for some constant c.

Additionally, **Spend** also restarts if the norm of some $f_{j,0}$ or $\boldsymbol{g}$ is "unexpectedly large". This is done in order to use tighter bounds when computing M-SIS hardness. The bound on $f_{j,0}$ comes from the fact that $a_{j,0}$ is the sum of uniformly sampled elements and thus its distribution converges to a Gaussian distribution. It is hard to formally bound the probability of having a rejection due to Step 20, and thus the bound T_g on $\|\boldsymbol{g}\|$ is computed experimentally so that the chance of restarting due to Step 20 is less than 1%.[7] However, this does not raise a security concern as the same bound is also checked by the verifier and thus ensured to hold for any accepting transcript. The masked randomnesses are similarly computed as in the one-out-of-many proof in Section 5.5.2 and the same rejection sampling idea as above is used. The output part CK_{out} of Algorithm 6.9 is transmitted to the recipient(s) privately along with corresponding output amounts and is not revealed publicly. This can be trivially accomplished by encrypting this information with the recipient's public key.

The verification (Algorithm 6.10) of a proof performs the same norm checks as in Algorithm 6.9, computes the "missing" components not output by **Spend** and then checks whether the hash output matches. The missing components are those that are uniquely determined by the rest and thus need not be transferred.

Remark 6.2. *For our concrete parameters, $\boldsymbol{f}_1$, $f_{j,0}$'s and $\boldsymbol{f}_r$ remain the same when seen as elements in either $R_{\hat{q}}$ or R_q since their infinity norm is smaller than $q/2 < \hat{q}/2$.*

[7] Observe here that T_g is a factor $4d$ smaller than the theoretical bound in Lemma 6.5.

Algorithm 6.8 Spend-I

INPUT: $M, S \in \mathbb{Z}^+$; $\mathsf{A}_{\text{in}} = (\mathsf{act}_{0,0}, \ldots, \mathsf{act}_{M-1,N-1})$ where $\mathsf{act}_{i,j} = (\mathsf{pk}_{i,j}, \mathsf{cn}_{i,j})$ is an account; $\ell \in [0, N-1]$; $(\mathsf{ask}_{0,\ell}, \ldots, \mathsf{ask}_{M-1,\ell})$ where $\mathsf{ask}_{i,\ell} = (\boldsymbol{r}_{i,\ell}, \mathsf{cnk}_{i,\ell}, \mathsf{amt}_{\text{in},i}) \in R_q^m \times R_q^m \times \mathbb{Z}^+$; $\mathsf{PK}_{\text{out}} = (\mathsf{pk}_{\text{out},0}, \ldots, \mathsf{pk}_{\text{out},S-1})$ where $\mathsf{pk}_{\text{out},i} \in R_q^n$; $(\mathsf{amt}_{\text{out},0}, \ldots, \mathsf{amt}_{\text{out},S-1})$ where $\mathsf{amt}_{\text{out},i} \in [0, 2^r - 1]$.

$\mathcal{B}_a = \lceil 20 \cdot pkd \rceil, \quad \mathcal{B}_r = \lceil p(S+1)rd \rceil$
$T_g = d^3 \left(\mathcal{B}_a^4 k\beta(\beta+1) + \mathcal{B}_r^4 r(S+1) \right) / (4d)$
$\mathcal{B}_{\text{big}} = \lceil 1.2 \cdot (M+S+1)\mathcal{B}pwmd \rceil$, $\hat{\mathcal{B}}_{\text{big}} = \lceil 8 \cdot (M+S+1)\mathcal{B}pw\hat{m}d \rceil$
$\mathcal{B}_{\text{big,k}} = \lceil 1.2 \cdot (M+S+1)\mathcal{B}(pw)^k md \rceil$
$\mathcal{B}'_{\text{big,k}} = \lceil 2.4 \cdot (M+S+1)\mathcal{B}(pw)^k md \rceil$
for $i = 0, \ldots, r-2$ **do** ▷ $c_0 = c_r = 0$
    $c_{i+1} = \left(c_i + \sum_{j=0}^{S-1} \mathsf{amt}_{\text{out},j}[i] - \sum_{j=0}^{M-1} \mathsf{amt}_{\text{in},j}[i] \right) / 2$
end for
for $i = 0, \ldots, S-1$ **do**
    $(\mathsf{cn}_{\text{out},i}, \mathsf{cnk}_{\text{out},i}) \leftarrow \textbf{Mint}(\mathsf{amt}_{\text{out},i})$
end for
$\mathsf{CN}_{\text{out}} = (\mathsf{cn}_{\text{out},0}, \ldots, \mathsf{cn}_{\text{out},S-1})$
$\mathsf{CK}_{\text{out}} = (\mathsf{cnk}_{\text{out},0}, \ldots, \mathsf{cnk}_{\text{out},S-1})$
$\boldsymbol{b} = \{(\beta, \text{True}, (\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,\beta-1}), \mathcal{B}_a)\}_{j=0}^{k-1}$
$\boldsymbol{b} = \boldsymbol{b} \cup (r-1, \text{False}, (c_1, \ldots, c_{r-1}), \mathcal{B}_r)$
for $j = 0, \ldots, S-1$ **do**
    $\boldsymbol{b} = \boldsymbol{b} \cup (r, \text{False}, \mathsf{Bits}(\mathsf{amt}_{\text{out},j}), \mathcal{B}_r)$
end for
$ck = \boldsymbol{G} \leftarrow \mathsf{SamMat}(\rho, r, q, n, m, \text{“G”})$
$\boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{d \cdot m}, \quad \boldsymbol{r}_d \leftarrow \{-\mathcal{B}_{\text{big}}, \ldots, \mathcal{B}_{\text{big}}\}^{d \cdot m}$
$(\boldsymbol{r}_a, \boldsymbol{r}_b), (A, B), (a_{0,0}, \ldots, a_{k-1,\beta-1}, a_{c,1}, \ldots, a_{c,r-1}, a_{\text{out},0}^{(0)}, \ldots, a_{\text{out},r-1}^{(S-1)}) \leftarrow$
$\mathsf{BinaryCommit}(k+1+S, \boldsymbol{b}, \mathcal{B}, \hat{\mathcal{B}}_{\text{big}})$ ▷ $a_{c,0} = a_{c,r} = 0$
for $i = 0, \ldots, S-1$ **do**
    $\boldsymbol{r}_g^{(i)} \leftarrow \{-\mathcal{B}_{\text{big}}, \ldots, \mathcal{B}_{\text{big}}\}^{d \cdot m}$
    $G_i = \text{Com}_{ck}\left(a_{\text{out},0}^{(i)}, \ldots, a_{\text{out},r-1}^{(i)}; \boldsymbol{r}_g^{(i)} \right)$ in R_q^n
end for
$C = \text{Com}_{ck}\left(c_0 - 2c_1, \ldots, c_{r-1} - 2c_r; \boldsymbol{r}_c \right)$ in R_q^n
$D = \text{Com}_{ck}\left(a_{c,0} - 2a_{c,1}, \ldots, a_{c,r-1} - 2a_{c,r}; \boldsymbol{r}_d \right)$ in R_q^n
Compute $\boldsymbol{p} = (p_{0,0}, \ldots, p_{N-1,k-1})$ using Alg. 6.11 with $(\ell, a_{0,0}, \ldots, a_{k-1,\beta-1})$
for $i = 0, \ldots, M-1$ **do**
    $\mathsf{s}_i = \textbf{SerialGen}(\boldsymbol{r}_{i,\ell})$ ▷ Not recomputed if restarted
    $(\boldsymbol{\rho}_0^{(i)}, \ldots, \boldsymbol{\rho}_{k-1}^{(i)}), (E_0^{(i)}, F_0^{(i)}, \ldots, E_{k-1}^{(i)}, F_{k-1}^{(i)}) \leftarrow$
    $\mathsf{RingCommit}(\text{True}, (\mathsf{pk}_{i,0}, \ldots, \mathsf{pk}_{i,N-1}), \boldsymbol{p}, \mathcal{B}, \mathcal{B}_{\text{big,k}})$
end for
for $j = 0, \ldots, N-1$ **do**
    $P_j = \sum_{i=0}^{S-1} \mathsf{cn}_{\text{out},i} - \sum_{i=0}^{M-1} \mathsf{cn}_{i,j} + C$ in R_q^n
end for
$(\boldsymbol{\rho}_0^{(M)}, \ldots, \boldsymbol{\rho}_{k-1}^{(M)}), (E_0^{(M)}, \ldots, E_{k-1}^{(M)}) \leftarrow$
$\mathsf{RingCommit}(\text{False}, (P_0, \ldots, P_{N-1}), \boldsymbol{p}, \mathcal{B}, \mathcal{B}'_{\text{big,k}})$
$x = \mathcal{H}(A, B, C, D, E_0^{(0)}, \ldots, E_{k-1}^{(M)}, F_0^{(0)}, \ldots, F_{k-1}^{(M)}, G_0, \ldots, G_{S-1}, \mathsf{s}_0, \ldots, \mathsf{s}_{M-1}, \mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{CN}_{\text{out}})$

Algorithm 6.9 Spend-II ▷ No mod q or $\hat{q}$ in this function!

OUTPUT: $\mathsf{CK}_{\text{out}}, \mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{CN}_{\text{out}}, \mathsf{SN}$ and Π as below where $F_1^{(0)}, \ldots, F_{k-1}^{(M)} \in R_q^{n_s}$; $B \in R_{\hat{q}}^{\hat{n}}$; $C, E_1^{(0)}, \ldots, E_{k-1}^{(M)} \in R_q^n$; $x \in \mathcal{C}_{w,p}^d$; $\boldsymbol{f}_1 \in R_q^{k(\beta-1)}$; $\boldsymbol{f}_r \in R_q^{r-1+Sr}$; $\boldsymbol{z}_b \in R_{\hat{q}}^{\hat{m}}$; $\boldsymbol{z}_c, \boldsymbol{z}_{\text{out},0}, \ldots, \boldsymbol{z}_{\text{out},S-1}, \boldsymbol{z}^{(0)}, \ldots, \boldsymbol{z}^{(M)} \in R_q^m$

for $j = 0, \ldots, k-1$ and $i = 0, \ldots, \beta - 1$ **do**
  $f_{j,i} = x\delta_{\ell_j,i} + a_{j,i}$
end for
for $i = 1, \ldots, r-1$ **do**
  $f_{c,i} = xc_i + a_{c,i}$
end for
for $j = 0, \ldots, S-1$ and $i = 0, \ldots, r-1$ **do**
  $f_{\text{out},i}^{(j)} = x \cdot \mathsf{amt}_{\text{out},j}[i] + a_{\text{out},i}^{(j)}$
end for
$\boldsymbol{f}_1 = (f_{0,1}, \ldots, f_{k-1,\beta-1})$ ▷ $f_{j,0}$'s are excluded
if $\|\boldsymbol{f}_1\|_\infty > \mathcal{B}_a - p$, **then** Go to Step 20 of Alg. 6.8
$\boldsymbol{f}_r = \left(f_{c,1}, \ldots, f_{c,r-1}, f_{\text{out},0}^{(0)}, \ldots, f_{\text{out},r-1}^{(S-1)}\right)$
if $\|\boldsymbol{f}_r\|_\infty > \mathcal{B}_r - p$, **then** Go to Step 20 of Alg. 6.8
for $j = 0, \ldots, k-1$ **do**
  if $\|f_{j,0}\| > \mathcal{B}_a\sqrt{d(\beta-1)}$, **then** Go to Step 20 of Alg. 6.8
end for
$\boldsymbol{g} = (f_{0,0}(x - f_{0,0}), \ldots, f_{k-1,\beta-1}(x - f_{k-1,\beta-1}))$
$\boldsymbol{g} = \boldsymbol{g} \cup (f_{c,1}(x - f_{c,1}), \ldots, f_{c,r-1}(x - f_{c,r-1}))$
$\boldsymbol{g} = \boldsymbol{g} \cup \left(f_{\text{out},0}^{(0)}(x - f_{\text{out},0}^{(0)}), \ldots, f_{\text{out},r-1}^{(S-1)}(x - f_{\text{out},r-1}^{(S-1)})\right)$
if $\|\boldsymbol{g}\| > \sqrt{T_g}$, **then** Go to Step 20 of Alg. 6.8
$\boldsymbol{z}_b = x\boldsymbol{r}_b + \boldsymbol{r}_a$ ▷ $\hat{m}$-dimensional
if $\|\boldsymbol{z}_b\|_\infty > \hat{\mathcal{B}}_{\text{big}} - \mathcal{B}pw$, **then** Go to Step 20 of Alg. 6.8
$\boldsymbol{z}_c = x\boldsymbol{r}_c + \boldsymbol{r}_d$
for $i = 0, \ldots, S-1$ **do**
  $\boldsymbol{z}_{\text{out},i} = x\boldsymbol{r}_{\text{out},i} + \boldsymbol{r}_g^{(i)}$ where $\boldsymbol{r}_{\text{out},i} = \mathsf{cnk}_{\text{out},i}$
end for
if $\|(\boldsymbol{z}_c, \boldsymbol{z}_{\text{out},0}, \ldots, \boldsymbol{z}_{\text{out},S-1})\|_\infty > \mathcal{B}_{\text{big}} - \mathcal{B}pw$, **then** Go to Step 20 of Alg. 6.8
for $i = 0, \ldots, M-1$ **do**
  $\boldsymbol{z}^{(i)} = x^k\boldsymbol{r}_{i,\ell} - \sum_{j=0}^{k-1} x^j\boldsymbol{\rho}_j^{(i)}$
  if $\left\|\boldsymbol{z}^{(i)}\right\|_\infty > \mathcal{B}_{\text{big,k}} - \mathcal{B}(pw)^k$, **then** Go to Step 20 of Alg. 6.8
end for
$\boldsymbol{z}^{(M)} = x^k\boldsymbol{r}_{M,\ell} - \sum_{j=0}^{k-1} x^j\boldsymbol{\rho}_j^{(M)}$ where $\boldsymbol{r}_{M,\ell} = \sum_{i=0}^{S-1}\boldsymbol{r}_{\text{out},i} - \sum_{i=0}^{M-1}\mathsf{cnk}_{i,\ell} + \boldsymbol{r}_c$
if $\left\|\boldsymbol{z}^{(M)}\right\|_\infty > \mathcal{B}'_{\text{big,k}} - (M+S+1)\mathcal{B}(pw)^k$, **then** Go to Step 20 of Alg. 6.8
return $\mathsf{CK}_{\text{out}}, \mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{CN}_{\text{out}}, \mathsf{SN} = (\mathsf{s}_0, \ldots, \mathsf{s}_{M-1})$ and
$\Pi = (B, C, E_1^{(0)}, \ldots, E_{k-1}^{(M)}, F_1^{(0)}, \ldots, F_{k-1}^{(M)}, x, \boldsymbol{f}_1, \boldsymbol{f}_r, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}^{(0)}, \ldots, \boldsymbol{z}^{(M)}, \boldsymbol{z}_{\text{out},0}, \ldots, \boldsymbol{z}_{\text{out},S-1})$

6.5 Improved Special Soundness Proof for the Binary Proof

Before going into the technical details of the soundness proofs, we remark the following. Even though we prove separate statements regarding the extracted openings of different components of the full protocol, the openings are indeed related. First,

Algorithm 6.10 Verify

INPUT: $M, S \in \mathbb{Z}^+$; $\mathsf{A}_{\text{in}} = (\mathsf{act}_{0,0}, \ldots, \mathsf{act}_{M-1,N-1})$ where $\mathsf{act}_{i,j} = (\mathsf{pk}_{i,j}, \mathsf{cn}_{i,j})$ is an account; $\mathsf{PK}_{\text{out}} = (\mathsf{pk}_{\text{out},0}, \ldots, \mathsf{pk}_{\text{out},S-1})$; $\mathsf{CN}_{\text{out}} = (\mathsf{cn}_{\text{out},0}, \ldots, \mathsf{cn}_{\text{out},S-1})$; $\Pi = (B, C, E_1^{(0)}, \ldots, E_{k-1}^{(M)}, F_1^{(0)}, \ldots, F_{k-1}^{(M)}, x, \boldsymbol{f}_1, \boldsymbol{f}_r, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}^{(0)}, \ldots, \boldsymbol{z}^{(M)}, \boldsymbol{z}_{\text{out},0}, \ldots, \boldsymbol{z}_{\text{out},S-1})$; $\mathsf{SN} = (\mathsf{s}_0, \ldots, \mathsf{s}_{M-1})$.
OUTPUT: True/False

---

if $\|\boldsymbol{f}_1\|_\infty > \mathcal{B}_a - p$, **then return** False
if $\|\boldsymbol{f}_r\|_\infty > \mathcal{B}_r - p$, **then return** False
Parse $\boldsymbol{f}_1 = (f_{0,1}, \ldots, f_{k-1,\beta-1})$ as in Alg. 6.9
Parse $\boldsymbol{f}_r = (f_{c,1}, \ldots, f_{c,r-1}, f_{\text{out},0}^{(0)}, \ldots, f_{\text{out},r-1}^{(S-1)})$ as in Alg. 6.9
for $j = 0, \ldots, k-1$ **do**
    $f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$
    if $\|f_{j,0}\| > \mathcal{B}_a\sqrt{d(\beta-1)}$, **then return** False
end for
Compute $\boldsymbol{g}$ as in Alg. 6.9
if $\|\boldsymbol{g}\| > \sqrt{T_g}$, **then return** False
if $\|\boldsymbol{z}_b\|_\infty > \hat{\mathcal{B}}_{\text{big}} - \mathcal{B}pw$, **then return** False
if $\|(\boldsymbol{z}_c, \boldsymbol{z}_{\text{out},0}, \ldots, \boldsymbol{z}_{\text{out},S-1})\|_\infty > \mathcal{B}_{\text{big}} - \mathcal{B}pw$, **then return** False
if $\left\|(\boldsymbol{z}^{(0)}, \ldots, \boldsymbol{z}^{(M-1)})\right\|_\infty > \mathcal{B}_{\text{big,k}} - \mathcal{B}(pw)^k$, **then return** False
if $\left\|\boldsymbol{z}^{(M)}\right\|_\infty > \mathcal{B}'_{\text{big,k}} - (M+S+1)\mathcal{B}(pw)^k$, **then return** False
$\boldsymbol{f} = (f_{0,0}, \ldots, f_{k-1,\beta-1}) \cup \boldsymbol{f}_r$ ▷ $f_{j,0}$'s are included.
$A = \text{Com}_{ck}(\boldsymbol{f}, \boldsymbol{g}; \boldsymbol{z}_b) - xB$ in $R_{\hat{q}}^{\hat{n}}$
$D = \text{Com}_{ck}(f_{c,0} - 2f_{c,1}, \ldots, f_{c,r-1} - 2f_{c,r}; \boldsymbol{z}_c) - xC$ in R_q^n where $f_{c,0} = f_{c,r} = 0$
for $i = 0, \ldots, S-1$ **do**
    $G_i = \text{Com}_{ck}\left(f_{\text{out},0}^{(i)}, \ldots, f_{\text{out},r-1}^{(i)}; \boldsymbol{z}_{\text{out},i}\right) - x\mathsf{cn}_{\text{out},i}$ in R_q^n
end for
for $l = 0, \ldots, M-1$ **do**
    $E_0^{(l)} = \left[\sum_{i=0}^{N-1}\left(\prod_{j=0}^{k-1} f_{j,i_j}\right)\mathsf{pk}_{l,i}\right] - \text{Com}_{ck}(\boldsymbol{0}; \boldsymbol{z}^{(l)}) - \sum_{j=1}^{k-1} E_j^{(l)} x^j$ in R_q^n
    where $i = (i_0, \ldots, i_{k-1})$ in base β
    $F_0^{(l)} = x^k \mathsf{s}_l - \boldsymbol{H} \cdot \boldsymbol{z}^{(l)} - \sum_{j=1}^{k-1} F_j^{(l)} x^j$ in $R_q^{n_s}$
end for
for $j = 0, \ldots, N-1$ **do**
    $P_j = \sum_{i=0}^{S-1} \mathsf{cn}_{\text{out},i} - \sum_{i=0}^{M-1} \mathsf{cn}_{i,j} + C$ in R_q^n
end for
$E_0^{(M)} = \left[\sum_{i=0}^{N-1}\left(\prod_{j=0}^{k-1} f_{j,i_j}\right)P_i\right] - \text{Com}_{ck}(\boldsymbol{0}; \boldsymbol{z}^{(M)}) - \sum_{j=1}^{k-1} E_j^{(M)} x^j$ in R_q^n
if $x \neq \mathcal{H}(A, B, C, D, E_0^{(0)}, \ldots, E_{k-1}^{(M)}, F_0^{(0)}, \ldots, F_{k-1}^{(M)}, G_0, \ldots, G_{S-1}, \mathsf{s}_0, \ldots, \mathsf{s}_{M-1}, \mathsf{A}_{\text{in}}, \mathsf{PK}_{\text{out}}, \mathsf{CN}_{\text{out}})$, **then return** False
return True

---

there is a single extraction procedure that is used to extract the openings of all components. Therefore, all the relaxation factors are determined by the same challenges. In particular, the relaxation factor is exactly the same element y when it is simply a challenge difference in $\Delta\mathcal{C}_{w,p}^d$. That is, for example, y in Lemmas 6.3, 6.6, and 6.7 are exactly the same. Moreover, when the relaxation factor is a product of elements in $\Delta\mathcal{C}_{w,p}^d$, then one of the multiplicands in the product is equal to the y in Lemmas 6.3, 6.6, and 6.7.

Also, as in the ring signature or one-out-of-many proof in Chapter 5, different parts

Algorithm 6.11 Compute $p_{i,j}$ for $k \in \{1, 2\}$

INPUT: $\ell, a_{0,0}, \ldots, a_{k-1,\beta-1}$
OUTPUT: $p_{0,0}, \ldots, p_{N-1,k-1} \in R_q$

1: **if** $k = 1$ **then**
2: **for** $i = 0, \ldots, N-1$ **do**
3: $p_{i,0} = a_{0,i_0}$ ▷ $i = (i_0, \ldots, i_{k-1})$ in base β
4: **end for**
5: **return** $(p_{0,0}, \ldots, p_{N-1,0})$
6: **else if** $k = 2$ **then**
7: **for** $i = 0, \ldots, N-1$ **do**
8: $p_{i,0} = a_{0,i_0} \cdot a_{1,i_1}$
9: $p_{i,1} = \delta_{\ell_0,i_0} \cdot a_{1,i_1} + \delta_{\ell_1,i_1} \cdot a_{0,i_0}$
10: **end for**
11: **return** $(p_{0,0}, p_{0,1}, \ldots, p_{N-1,0}, p_{N-1,1})$
12: **end if**

of the protocol uses the same components. For example, the binary proof proves that all f's in $(\boldsymbol{f}_1, \boldsymbol{f}_r)$ encode some bits, and they are later used both in the ring signature and also to prove that the corrector commitment C is well-formed. Since the full special soundness proofs for similar underlying protocols used here have already been shown in Chapter 5, our goal in the proofs here is to point to the main technical differences and to show how the extracted opening norms are bounded, which is important for choosing parameter.

Lemma 6.3. *Assume that the following holds*

- $\hat{q}/2 > \max\left\{2pwd\mathcal{B}_f(p + \mathcal{B}_f), 2pw\mathcal{B}_a^2 d\beta\right\}$ *for* $\mathcal{B}_f = \max\{\mathcal{B}_a, \mathcal{B}_r\}$,
- *HMC is* γ_{bin}*-binding for* $\gamma_{\text{bin}} = 2p\sqrt{dw}\left(T_g + \hat{\mathcal{B}}_{big}^2 \hat{n} d\right)^{1/2}$.

For an input commitment $B \in R_{\hat{q}}^{\hat{n}}$*, a commitment key ck and proof output* $(A, x, \boldsymbol{f}_1, \boldsymbol{f}_r, \boldsymbol{z}_b)$*, our binary proof in this chapter proves knowledge of* $(y, \boldsymbol{b}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{r}})$ *such that*

- $y \in \Delta\mathcal{C}_{w,p}^d$,
- $yB = \text{Com}_{ck}(y\boldsymbol{b}, \hat{\boldsymbol{c}}; \hat{\boldsymbol{r}})$,
- *All coordinates* b_i *of* $\boldsymbol{b}$ *is in* $\{0, 1\}$,
- $\hat{\boldsymbol{c}}$ *is uniquely determined by* $y, \boldsymbol{f}_1, \boldsymbol{f}_r, x$ *and* $\boldsymbol{b}$,
- *For the first* $k\beta$ *coordinates* $b_{0,0}, \ldots, b_{k-1,\beta-1}$ *of* $\boldsymbol{b}$,

$$\sum_{i=0}^{\beta-1} b_{j,i} = 0,$$

  i.e., there is only a single 1 in $\{b_{i,0}, \ldots, b_{i,\beta-1}\}$ *for all* $0 \leq i \leq k-1$,
- $\|(y\boldsymbol{b}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{r}})\| \leq \gamma_{\text{bin}}$.

Proof. The proof uses the standard norm relations in $R = \mathbb{Z}[X]/(X^d + 1)$ as given in Lemma 3.19. As in the special soundness proofs of Theorems 5.8 and 5.10, for 3 distinct challenges x, x', x'', the extractor of the *interactive* binary proof is given three accepting protocol transcripts as

$$(A, B, x, \boldsymbol{f}_1, \boldsymbol{f}_r, \boldsymbol{z}_b), \tag{6.7}$$

$$(A, B, x', \boldsymbol{f}'_1, \boldsymbol{f}'_r, \boldsymbol{z}'_b), \tag{6.8}$$
$$(A, B, x'', \boldsymbol{f}''_1, \boldsymbol{f}''_r, \boldsymbol{z}''_b). \tag{6.9}$$

Since they are accepting transcripts, we have from Step 16 of Algorithm 6.10

$$xB + A = \mathrm{Com}_{ck}\left(\boldsymbol{f}, \boldsymbol{g}; \boldsymbol{z}_b\right), \tag{6.10}$$
$$x'B + A = \mathrm{Com}_{ck}\left(\boldsymbol{f}', \boldsymbol{g}'; \boldsymbol{z}'_b\right), \tag{6.11}$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ are as defined in Algorithm 6.10. Let $y = x - x'$ and take the difference of the above two equations. We get

$$yB = \mathrm{Com}_{ck}\left(\boldsymbol{f} - \boldsymbol{f}', \boldsymbol{g} - \boldsymbol{g}'; \boldsymbol{z}_b - \boldsymbol{z}'_b\right), \tag{6.12}$$
$$yA = \mathrm{Com}_{ck}\left(x\boldsymbol{f}' - x'\boldsymbol{f}, x\boldsymbol{g}' - x'\boldsymbol{g}; x\boldsymbol{z}'_b - x'\boldsymbol{z}_b\right). \tag{6.13}$$

Define $\hat{\boldsymbol{b}} = \boldsymbol{f} - \boldsymbol{f}'$, $\hat{\boldsymbol{a}} = x\boldsymbol{f}' - x'\boldsymbol{f}$, $\hat{\boldsymbol{c}} = \boldsymbol{g} - \boldsymbol{g}'$ and $\hat{\boldsymbol{d}} = x\boldsymbol{g}' - x'\boldsymbol{g}$. From here, we have

$$yf_i = x\hat{b}_i + \hat{a}_i \in R_{\hat{q}}, \tag{6.14}$$
$$yg_i = yf_i(x - f_i) = x\hat{c}_i + \hat{d}_i \in R_{\hat{q}}, \tag{6.15}$$

for any coordinate f_i of $\boldsymbol{f}$ and any coordinate g_i of $\boldsymbol{g}$. Further, for any coordinate $f_i \neq f_{j,0}$ of $\boldsymbol{f}$, we have

$$\left\|yf_i(x - f_i)\right\|_\infty \leq \|y\|_1 \left\|f_i\right\|_1 \left\|x - f_i\right\|_\infty \leq 2pw \cdot d\mathcal{B}_f \cdot (p + \mathcal{B}_f),$$

where $\mathcal{B}_f = \max\{\mathcal{B}_a, \mathcal{B}_r\}$. Now, for the coordinates $f_{j,0}$ of $\boldsymbol{f}$, we have

$$\begin{aligned}\left\|yf_{j,0}(x - f_{j,0})\right\|_\infty &\leq \|y\|_1 \left\|f_{j,0}\right\| \left\|x - f_{j,0}\right\| \\ &\leq 2pw \cdot \mathcal{B}_a\sqrt{d(\beta - 1)} \cdot (p\sqrt{w} + \mathcal{B}_a\sqrt{d(\beta - 1)}) \\ &\approx 2pw\mathcal{B}_a^2 d\beta.\end{aligned}$$

Therefore, since the following holds

$$\hat{q}/2 > \max\left\{2pwd\mathcal{B}_f(p + \mathcal{B}_f), 2pw\mathcal{B}_a^2 d\beta\right\}, \tag{6.16}$$

the equations (6.14) and (6.15) hold over R for any coordinate of $\boldsymbol{f}$ and $\boldsymbol{g}$.

By Lemma 6.6 further below, we know that the Euclidean norm of $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{d}}, x\boldsymbol{z}_b - x'\boldsymbol{z}'_b)$ (i.e., the opening of yA) is bounded from above by $\gamma_{\mathrm{bin}} = 2p\sqrt{dw}\left(T_g + \hat{\mathcal{B}}_{\mathrm{big}}^2 \hat{m}d\right)^{1/2}$. The same bound clearly holds for the opening of yB by ignoring the factor x. Then, using γ_{bin}-binding of HMC, the steps continue exactly as in the soundness proofs of Theorems 5.8 and 5.10 with a crucial difference: the equations now hold in R, not $R_{\hat{q}}$. As a result, for any coordinate $\hat{a}$ of $\hat{\boldsymbol{a}}$, any coordinate $\hat{b}$ of $\hat{\boldsymbol{b}}$ and any coordinate $\hat{c}$ of $\hat{\boldsymbol{c}}$, we get

$$\begin{pmatrix} 1 & x & x^2 \\ 1 & x' & x'^2 \\ 1 & x'' & x''^2 \end{pmatrix} \cdot \begin{pmatrix} * \\ \hat{a}\left(y - 2\hat{b}\right) - y\hat{c} \\ \hat{b}(y - \hat{b}) \end{pmatrix} = \boldsymbol{0} \quad \text{in } R, \tag{6.17}$$

where the entries marked with $*$ are not relevant for our analysis. Denoting the left-most matrix by $\boldsymbol{V}$ and multiplying both sides by the adjugate matrix $\boldsymbol{V}$ as in Theorems 5.8 and 5.10, we get

$$\det(\boldsymbol{V})\hat{b}(y - \hat{b}) = (x'' - x')(x' - x)(x'' - x)\hat{b}(y - \hat{b}) = 0 \quad \text{in } R. \tag{6.18}$$

Again note the difference from the proofs of Theorems 5.8 and 5.10 that (6.18) hold in R, not $R_{\hat{q}}$. By Lemma 5.5, one of the factors in (6.18) needs to be zero. Since the challenge differences cannot be zero, we get either $\hat{b} = 0$ or $y - \hat{b} = 0$. That is, $\hat{b} \in \{0, y\}$. This proves that all coordinates of $\hat{\boldsymbol{b}}$ are in $\{0, y\}$ as required for the binary proof. In other words,

$$\hat{\boldsymbol{b}} = y\boldsymbol{b} \text{ for } \boldsymbol{b} \in \text{BIN} \subset \{0,1\}^*, \tag{6.19}$$

$$yf_i = yxb_i + \hat{a}_i \in R, \tag{6.20}$$

for any coordinate f_i of $\boldsymbol{f}$ where $b_i \in \{0, 1\}$. Further, by Step 6 of Algorithm 6.10, we have

$$\left.\begin{aligned} f_{j,0} &= x - \sum_{i=1}^{\beta-1} f_{j,i}, \\ f'_{j,0} &= x' - \sum_{i=1}^{\beta-1} f'_{j,i}, \end{aligned}\right\} \implies f_{j,0} - f'_{j,0} = x - x' - \sum_{i=1}^{\beta-1} \left(f_{j,i} - f'_{j,i}\right).$$

Since $\hat{\boldsymbol{b}} = \boldsymbol{f} - \boldsymbol{f}' = y\boldsymbol{b}$ with all the coordinates of $\boldsymbol{b}$ in $\{0, 1\}$, we get

$$yb_{j,0} = y - \sum_{i=1}^{\beta-1} yb_{j,i} \implies y\sum_{i=0}^{\beta-1} b_{j,i} = y \implies \sum_{i=0}^{\beta-1} b_{j,i} = 1, \tag{6.21}$$

since $y \neq 0$.

Now, using the fact that (6.17) holds over R in a similar fashion, we also get for any coordinate $\hat{a}$ of $\hat{\boldsymbol{a}}$, any coordinate $\hat{b}$ of $\hat{\boldsymbol{b}}$ and any coordinate $\hat{c}$ of $\hat{\boldsymbol{c}}$

$$\hat{a}\left(y - 2\hat{b}\right) - y\hat{c} = 0 \quad \implies \quad y\hat{c} = \hat{a}\left(y - 2\hat{b}\right) = \hat{a}\left(y - 2yb\right). \tag{6.22}$$

From here, by Lemma 5.5 and definition of $\hat{\boldsymbol{a}}$, we have

$$\begin{aligned} \hat{c} &= \hat{a}(1 - 2b) = \left(xf' - x'f\right)(1 - 2b) \\ &= \left(x\left(f - yb\right) - x'f\right)(1 - 2b) \quad \text{(by the definition of } \hat{\boldsymbol{b}} = y\boldsymbol{b}) \\ &= \left(xf - yxb - x'f\right)(1 - 2b) = \left(yf - yxb\right)(1 - 2b) \\ &= y(f - xb)(1 - 2b). \end{aligned}$$

The above proves that $\hat{\boldsymbol{c}}$ is determined by $y, \boldsymbol{f}, x$ and $\boldsymbol{b}$ where $\boldsymbol{f} = (\boldsymbol{f}_1, \boldsymbol{f}_r)$. □

6.6 Security Proofs for MatRiCT

We first prove/recall some auxiliary lemmas that will be used to prove the balance and anonymity properties of MatRiCT.

6.6.1 Auxiliary lemmas

The following lemma shows that any element in $\Delta\mathcal{C}^d_{w,p}$ is invertible in R_q when q is chosen in a certain way. The result is a direct corollary of [LS18, Corollary 1.2].

Lemma 6.4. *If $q > (2p\sqrt{K})^K$ and $q \equiv 2K + 1 \bmod 4K$ for some $1 < K \leq d$ where K is a power of 2, then any $y \in \Delta\mathcal{C}^d_{w,p}$ is invertible in R_q.*

Lemma 6.5. *The vector $\boldsymbol{g}$ defined in Algorithms 6.9 and 6.10 satisfies the following*

$$\|\boldsymbol{g}\|^2 \leq d^3 \left(\mathcal{B}_a^4 k\beta(\beta+1) + \mathcal{B}_r^4 r(S+1)\right).$$

Proof. We use the bounds on the norm of f_*'s in the sequel (see Algorithm 6.10). For simplicity, we bound $\|x - f_*\|$ by the bound on $\|f_*\|$ as $\|x\|$ is much smaller in comparison.

$$\begin{aligned}
\|\boldsymbol{g}\|^2 &= \sum_{j=0}^{k-1}\sum_{i=0}^{\beta-1} \|f_{j,i}(x - f_{j,i})\|^2 + \sum_{i=1}^{r-1} \|f_{c,i}(x - f_{c,i})\|^2 \\
&\quad + \sum_{j=0}^{S-1}\sum_{i=0}^{r-1} \left\|f_{\text{out},i}^{(j)}(x - f_{\text{out},i}^{(j)})\right\|^2 \\
&= \sum_{j=0}^{k-1}\sum_{i=1}^{\beta-1} \|f_{j,i}(x - f_{j,i})\|^2 + \sum_{j=0}^{k-1} \|f_{j,0}(x - f_{j,0})\|^2 \\
&\quad + \sum_{i=1}^{r-1} \|f_{c,i}(x - f_{c,i})\|^2 + \sum_{j=0}^{S-1}\sum_{i=0}^{r-1} \left\|f_{\text{out},i}^{(j)}(x - f_{\text{out},i}^{(j)})\right\|^2 \\
&\leq \sum_{j=0}^{k-1}\sum_{i=1}^{\beta-1} d\,\|f_{j,i}\|^2\,\|x - f_{j,i}\|^2 + \sum_{j=0}^{k-1} d\,\|f_{j,0}\|^2\,\|x - f_{j,0}\|^2 \\
&\quad + \sum_{i=1}^{r-1} d\,\|f_{c,i}\|^2\,\|x - f_{c,i}\|^2 + \sum_{j=0}^{S-1}\sum_{i=0}^{r-1} d\left\|f_{\text{out},i}^{(j)}\right\|\left\|x - f_{\text{out},i}^{(j)}\right\|^2 \\
&\leq dk(\beta-1)\left(\mathcal{B}_a\sqrt{d}\right)^4 + dk\left(\mathcal{B}_a\sqrt{d\beta}\right)^4 \\
&\quad + d(r-1)\left(\mathcal{B}_r\sqrt{d}\right)^4 + dSr\left(\mathcal{B}_r\sqrt{d}\right)^4 \\
&\leq dk\left(\mathcal{B}_a\sqrt{d}\right)^4\left[(\beta-1) + \beta^2\right] + d\left(\mathcal{B}_r\sqrt{d}\right)^4\left[r - 1 + Sr\right] \\
&\leq d^3\left(\mathcal{B}_a^4 k\beta(\beta+1) + \mathcal{B}_r^4 r(S+1)\right).
\end{aligned}$$

□

Lemma 6.6. *The extracted opening $(\hat{\mathbf{a}}, \hat{\boldsymbol{r}}_a)$ of A for A defined in Algorithms 6.8 and 6.10 satisfies the following*

$$\|(\hat{\mathbf{a}}, \hat{\boldsymbol{r}}_a)\| \leq 2p\sqrt{dw}\left(T_g + \hat{\mathcal{B}}_{big}^2 \hat{m} d\right)^{1/2}.$$

Proof. For two accepting transcripts with respect to different challenges $x, x' \in \mathcal{C}_{w,p}^d$, we have

$$xB + A = \text{Com}_{ck}\left(\boldsymbol{f}, \boldsymbol{g};\, \boldsymbol{z}_b\right), \tag{6.23}$$
$$x'B + A = \text{Com}_{ck}\left(\boldsymbol{f}', \boldsymbol{g}';\, \boldsymbol{z}_b'\right), \tag{6.24}$$

where $(\boldsymbol{f}, \boldsymbol{z}_b)$ and $(\boldsymbol{f}', \boldsymbol{z}_b')$ are responses with respect to challenges x and x', respectively, and $\boldsymbol{g}$ and $\boldsymbol{g}$ are constructed from $\boldsymbol{f}$ and $\boldsymbol{f}'$, respectively, as in Algorithm 6.10. From here, as in Lemma 5.12, the extracted opening $(\hat{\mathbf{a}}, \hat{\boldsymbol{r}}_a)$ of yA for $y = x - x'$ is as follows

$$yA = \text{Com}_{ck}\left(x\boldsymbol{f}' - x'\boldsymbol{f}, x\boldsymbol{g}' - x'\boldsymbol{g};\, x\boldsymbol{z}_b' - x'\boldsymbol{z}_b\right). \tag{6.25}$$

Assume $\|(x\boldsymbol{g}', x\boldsymbol{z}'_b)\| \geq \|(x'\boldsymbol{g}, x'\boldsymbol{z}_b)\|$ without loss of generality. Also, note that the norms of $\boldsymbol{f}$ and $\boldsymbol{f}'$ are much smaller than that of $\boldsymbol{g}$ and $\boldsymbol{g}'$, respectively. This is due to the fact that a coordinate of $\boldsymbol{g}$ is about the square of a coordinate of $\boldsymbol{f}$. Therefore, for simplicity, we neglect $x\boldsymbol{f}' - x'\boldsymbol{f}$ in the rest. We have

$$\begin{aligned}
&\|(\hat{\boldsymbol{a}}, \hat{\boldsymbol{r}}_a)\| \approx \left\|(x\boldsymbol{g}' - x'\boldsymbol{g}, x\boldsymbol{z}'_b - x'\boldsymbol{z}_b)\right\| \leq 2\left\|(x\boldsymbol{g}', x\boldsymbol{z}'_b)\right\| \\
&\leq 2\sqrt{d}\,\|x\| \cdot \left\|(\boldsymbol{g}', \boldsymbol{z}'_b)\right\| \leq 2p\sqrt{dw} \cdot \left\|(\boldsymbol{g}', \boldsymbol{z}'_b)\right\| \\
&= 2p\sqrt{dw}\left(\left\|\boldsymbol{g}'\right\|^2 + \left\|\boldsymbol{z}'_b\right\|^2\right)^{1/2} \\
&\leq 2p\sqrt{dw}\left(T_g + \left(\hat{\mathcal{B}}_{\text{big}}\sqrt{\hat{m}d}\right)^2\right)^{1/2} \\
&= 2p\sqrt{dw}\left(T_g + \hat{\mathcal{B}}^2_{\text{big}}\hat{m}d\right)^{1/2}. \qquad (6.26)
\end{aligned}$$

□

Lemma 6.7. *The extracted opening $(\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}_c)$ of C for C defined in Algorithms 6.8 and 6.10 satisfies the following*

$$\|(\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}_c)\| \leq 2\left(9r\mathcal{B}_r^2 d + \mathcal{B}_{big}^2 md\right)^{1/2}.$$

Further, the same bound as above also holds for the Euclidean norm of an extracted opening of $\mathsf{cn}_{\text{out},i}$ for any $0 \leq i \leq S-1$.

Proof. Analogues to Lemma 6.6, the extracted opening of C satisfies

$$yC = \text{Com}_{ck}\left(\boldsymbol{f}_c - \boldsymbol{f}'_c;\ \boldsymbol{z}_c - \boldsymbol{z}'_c\right), \qquad (6.27)$$

where $f_{c,0} = f_{c,r} = f'_{c,0} = f'_{c,r} = 0$, $\boldsymbol{f}_c = (f_{c,0} - 2f_{c,1}, \ldots, f_{c,r-1} - 2f_{c,r})$, and $\boldsymbol{f}'_c = (f'_{c,0} - 2f'_{c,1}, \ldots, f'_{c,r-1} - 2f'_{c,r})$.

Note the fact that all of polynomials $f_{c,0}, f'_{c,0}, \ldots, f_{c,r-1}, f'_{c,r-1}$ are upper-bounded by the same real value, which is used in the sequel. Assume without loss of generality that $\|(\boldsymbol{f}_c, \boldsymbol{z}_c)\| \geq \left\|(\boldsymbol{f}'_c, \boldsymbol{z}'_c)\right\|$.

$$\begin{aligned}
&\|(\hat{\boldsymbol{c}}, \hat{\boldsymbol{r}}_{\boldsymbol{c}})\| = \left\|(\boldsymbol{f}_c - \boldsymbol{f}'_c, \boldsymbol{z}_c - \boldsymbol{z}'_c)\right\| \leq 2\,\|(\boldsymbol{f}_c, \boldsymbol{z}_c)\| \\
&= 2\left(\|f_{c,0} - 2f_{c,1}, \ldots, f_{c,r-1} - 2f_{c,r}\|^2 + \|\boldsymbol{z}_c\|^2\right)^{1/2} \\
&\leq 2\left(3^2\,\|f_{c,1}, \ldots, f_{c,r-1}\|^2 + \|\boldsymbol{z}_c\|^2\right)^{1/2} \\
&\leq 2\left(9(r-1)(\mathcal{B}_r\sqrt{d})^2 + (\mathcal{B}_{\text{big}}\sqrt{md})^2\right)^{1/2} \\
&\leq 2\left(9r\mathcal{B}_r^2 d + \mathcal{B}_{\text{big}}^2 md\right)^{1/2}. \qquad (6.28)
\end{aligned}$$

For any $0 \leq i \leq S-1$, an extracted opening of $\mathsf{cn}_{\text{out},i}$ is

$$y \cdot \mathsf{cn}_{\text{out},i} = \text{Com}_{ck}\left(\boldsymbol{f}_{\text{out},i} - \boldsymbol{f}'_{\text{out},i};\ \boldsymbol{z}_{\text{out},i} - \boldsymbol{z}'_{\text{out},i}\right). \qquad (6.29)$$

We have $\|\boldsymbol{z}_{\text{out},i}\|_\infty \leq \mathcal{B}_{\text{big}}$ and $\left\|\boldsymbol{f}_{\text{out},i}\right\|_\infty \leq \mathcal{B}_r$. Therefore, $(\boldsymbol{f}_{\text{out},i} - \boldsymbol{f}'_{\text{out},i}, \boldsymbol{z}_{\text{out},i} - \boldsymbol{z}'_{\text{out},i})$ can be upper-bounded as above easily. □

Lemma 6.8. *Assume that $q > (2p\sqrt{K})^K$ and $q \equiv 2K+1 \bmod 4K$ for some $1 < K \leq d$ where K is a power of 2. On input a commitment key ck and a set of commitments*

$(P_0, \ldots, P_{N-1})$, the underlying one-out-of-many proof of our ring signature proves knowledge of $(y, \ell, \hat{\boldsymbol{r}})$ such that

- $\ell \in \{0, \ldots, N-1\}$,
- $yP_\ell = \mathrm{Com}_{ck}(\mathbf{0}; \hat{\boldsymbol{r}})$,
- *y is a product of κ elements in $\Delta\mathcal{C}^d_{w,p}$ for $\kappa = k(k+1)/2$, and $\|y\| \leq \sqrt{d} \cdot (2p)^\kappa w^{\kappa-1}$,*
- $\|\hat{\boldsymbol{r}}\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} w^{\kappa'-1} \sqrt{md} \cdot \max\{\mathcal{B}_{big,k}, \mathcal{B}'_{big,k}\}$.

Further, the proof is k'-special sound where $k' = \max\{k+1, 3\}$.

Proof. The first three properties and the fact that one-out-of-many proof is k'-special sound directly follow from Theorem 5.15, and the remaining property is shown in Lemma 6.9 below. □

Lemma 6.9. *Let $\kappa' = k(k-1)/2$. For any $0 \leq i \leq M-1$, the extracted opening $(\mathbf{0}, \hat{\boldsymbol{r}}_i)$ of $\mathsf{pk}_{i,\ell}$ for the real spender's public key $\mathsf{pk}_{i,\ell}$ defined in Algorithms 6.8 and 6.10, and the extracted opening $(\mathbf{0}, \hat{\boldsymbol{r}}_M)$ of P_ℓ for P_ℓ defined in Algorithms 6.8 and 6.10 satisfies the following*

$$\|\hat{\boldsymbol{r}}_i\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} w^{\kappa'-1} \mathcal{B}_{big,k} \sqrt{md},$$
$$\|\hat{\boldsymbol{r}}_M\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} w^{\kappa'-1} \mathcal{B}'_{big,k} \sqrt{md},$$

provided that $q > (2p\sqrt{K})^K$ and $q \equiv 2K+1 \bmod 4K$ for some $1 < K \leq d$ where K is a power of 2. Further, if $k = 1$ and the same assumption on q holds, we have

$$\|\hat{\boldsymbol{r}}_i\| \leq 2\mathcal{B}_{big,k} \sqrt{md},$$
$$\|\hat{\boldsymbol{r}}_M\| \leq 2\mathcal{B}'_{big,k} \sqrt{md}.$$

Proof. By the assumption on q and Lemma 6.4, any $y \in \Delta\mathcal{C}^d_{w,p}$ is invertible. The extraction of our one-out-of-many proofs (the underlying ZKP of the ring signature) will not have an additional y factor that appears in the special soundness proof of Theorem 5.15. Therefore, we can directly use the results in Lemma 5.3, which gives for $\kappa' = k(k-1)/2$ and any $0 \leq i \leq M-1$

$$\|\hat{\boldsymbol{r}}_i\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot B_z,$$

where B_z is an upper-bound on the Euclidean norm of any $\boldsymbol{z}^{(i)}$. Hence, using the bound from Algorithm 6.10, we have

$$\|\hat{\boldsymbol{r}}_i\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot \mathcal{B}_{\mathrm{big,k}} \sqrt{md}.$$

Similarly, the following is obtained by replacing $\mathcal{B}_{\mathrm{big,k}}$ with $\mathcal{B}'_{\mathrm{big,k}}$ in the norm $\boldsymbol{z}^{(M)}$

$$\|\hat{\boldsymbol{r}}_M\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot \mathcal{B}'_{\mathrm{big,k}} \sqrt{md}.$$

When $k = 1$, then the verification equations for the ring signatures are just linear equations. Therefore, the extracted openings are simply obtained by looking at the difference of two verification equations with respect to different responses as in the proof of Lemma 6.6. That is, $\hat{\boldsymbol{r}}_i = \boldsymbol{z}_0^{(i)} - \boldsymbol{z}_1^{(i)}$ where $\boldsymbol{z}_0^{(i)}, \boldsymbol{z}_1^{(i)}$ are two responses with respect to different challenges in the protocol's witness extraction. Hence, when $k = 1$, we have

$$\|\hat{\boldsymbol{r}}_i\| \leq 2\mathcal{B}_{\mathrm{big,k}} \sqrt{md},$$

$$\|\hat{\boldsymbol{r}}_M\| \leq 2\mathcal{B}'_{\text{big,k}}\sqrt{md}.$$

Observe that when $k = 1$, $\kappa' = 0$. Therefore, the general bound gives $\|\hat{\boldsymbol{r}}_i\| \leq 2(d/w)\mathcal{B}_{\text{big,k}}\sqrt{md}$, which is slightly looser then the above bound (note that we always have $1 \leq w \leq d$ since w is the Hamming weight of degree $d-1$ polynomials). □

6.6.2 Correctness

The correctness of MatRiCT follows from the completeness of the underlying ZKP, and MatRiCT is perfectly correct. The settings of $A, D, E_0^{(l)}, F_0^{(l)}, E_0^{(M)}$ for all $0 \leq l \leq M-1$ are all done analogous to Protocol 5.4. All the norm checks will be successful as they are all also done in **Spend** algorithm. Also, the underlying one-out-of-many proof allows decoy public commitments not to be well-formed as in Protocol 5.4, and therefore the given correctness requirements are satisfied.

6.6.3 Anonymity

Lemma 6.10. *(Anonymity) Let $\mathcal{A}$ be a PPT adversary, $\text{Adv}_{\mathcal{A}}^{LWE}$ be the advantage of $\mathcal{A}$ over solving M-LWE$_{m-n-n_s,m,q,\mathcal{B}}$ and $\text{Adv}_{\mathcal{A}}^{LWE_2}$ be the advantage of $\mathcal{A}$ over solving M-LWE$_{\hat{m}-\hat{n},\hat{m},\hat{q},\mathcal{B}}$. The advantage of $\mathcal{A}$ against* `Exp:Anonymity` *without shuffling is at most*

$$\text{Adv}_{\mathcal{A}}^{Ano} \leq \text{Adv}_{\mathcal{A}}^{LWE_2} + k(M+1)\cdot \text{Adv}_{\mathcal{A}}^{LWE}.$$

Proof. The proof uses the simulation of the underlying ZKP of our construction where the indistinguishability is either due to an M-LWE assumption or rejection sampling. We use the following succession of games.

Game$_0$: This is identical to `Exp:Anonymity` without shuffling.

Game$_1$: First, the challenger simulates the response where the rejection sampling is applied. In Algorithm 6.9, it replaces all the coordinates of $\boldsymbol{f}_1$ by uniformly random elements in $\mathfrak{U}_{\mathcal{B}_a-p}$, all the coordinates of $\boldsymbol{z}_b$ by uniformly random elements in $\mathfrak{U}_{\hat{\mathcal{B}}_{\text{big}}-\mathcal{B}pw}$, all the coordinates of $\boldsymbol{z}_c$ by uniformly random elements in $\mathfrak{U}_{\mathcal{B}_{\text{big}}-\mathcal{B}pw}$, all the coordinates of $\boldsymbol{z}^{(i)}$ by uniformly random elements in $\mathfrak{U}_{\mathcal{B}_{\text{big,k}}-\mathcal{B}(pw)^k}$ for all $0 \leq i \leq M-1$, and all the coordinates of $\boldsymbol{z}^{(M)}$ by uniformly random elements in $\mathfrak{U}_{\mathcal{B}'_{\text{big,k}}-(M+S+1)\mathcal{B}(pw)^k}$. This game is perfectly indistinguishable from the previous game due to rejection sampling.

$$\text{Adv}_{\mathcal{A}}^{\textbf{Game}_0} - \text{Adv}_{\mathcal{A}}^{\textbf{Game}_1} = 0.$$

Game$_2$: In Algorithm 6.6, the challenger replaces B by a uniformly random element in $R_{\hat{q}}^{\hat{n}}$. This game is computationally indistinguishable from the previous game by M-LWE$_{\hat{m}-\hat{n},\hat{m},\hat{q},\mathcal{B}}$ hardness as in the hiding property of the commitment scheme.

$$\left|\text{Adv}_{\mathcal{A}}^{\textbf{Game}_1} - \text{Adv}_{\mathcal{A}}^{\textbf{Game}_2}\right| \leq \text{Adv}_{\mathcal{A}}^{\text{LWE}_2}.$$

Game$_3$: In Algorithm 6.8, the challenger replaces C by a uniformly random element in R_q^n. This game is computationally indistinguishable from the previous game by M-LWE$_{m-n,m,q,\mathcal{B}}$ hardness.

$$\left|\text{Adv}_{\mathcal{A}}^{\textbf{Game}_2} - \text{Adv}_{\mathcal{A}}^{\textbf{Game}_3}\right| \leq \text{Adv}_{\mathcal{A}}^{\text{LWE}}.$$

Game$_4$: In Algorithm 6.4, the challenger replaces serial number s_i's by a uniformly random element in $R_q^{n_s}$ and the public keys $\mathsf{pk}_{i,\ell}$'s in R_{in} (i.e., the ℓ-th column of A_{in}) by a uniformly random element in R_q^n for $i = 0, \ldots, M-1$. This game is computationally

indistinguishable from the previous game by M-LWE$_{m-n-n_s,m,q,\mathcal{B}}$ hardness due to the following observation.

Let $\boldsymbol{G}' := \begin{pmatrix} \boldsymbol{G} \\ \boldsymbol{H} \end{pmatrix}$. We have $\begin{pmatrix} \mathsf{pk}_{i,\ell} \\ \mathsf{s}_i \end{pmatrix} = \boldsymbol{G}' \cdot \boldsymbol{r}_{i,\ell}$ where $\boldsymbol{r}_{i,\ell}$ is the secret key corresponding to the public key $\mathsf{pk}_{i,\ell}$. Since $\boldsymbol{G}'$ has the same distribution as a commitment key ck output by $\mathsf{CKeygen}$, the hiding property argument for the commitment also holds with respect to the combined matrix $\boldsymbol{G}' \in R_q^{(n+n_s)\times m}$. Also, note that no CORRUPT or SPEND is allowed to be queried for these public keys, and the distribution of the secret keys $\boldsymbol{r}_{i,\ell}$ is identical to that in M-LWE definition since the public keys in R_{in} are assumed to be generated honestly by querying PKGEN.

$$\left|\text{Adv}_{\mathcal{A}}^{\textbf{Game}_3} - \text{Adv}_{\mathcal{A}}^{\textbf{Game}_4}\right| \leq M \cdot \text{Adv}_{\mathcal{A}}^{\text{LWE}}.$$

Game$_5$: In Algorithm 6.7, the challenger replaces R_j by a uniformly random element in R_q^n, and F_j by a uniformly random element in $R_q^{n_s}$ for all $1 \leq j \leq k-1$ (if GenSerial = False, then only R_j is replaced and the argument still works). This game is computationally indistinguishable from the previous game by M-LWE$_{m-n-n_s,m,q,\mathcal{B}}$ hardness due to a similar discussion as above.

$$\left|\text{Adv}_{\mathcal{A}}^{\textbf{Game}_4} - \text{Adv}_{\mathcal{A}}^{\textbf{Game}_5}\right| \leq (M+1)(k-1) \cdot \text{Adv}_{\mathcal{A}}^{\text{LWE}}.$$

Game$_6$: In Algorithm 6.7, the challenger replaces E_j by a uniformly random element in R_q^n for all $1 \leq j \leq k-1$. This game is perfectly indistinguishable from the previous game as R_j is uniformly random in R_q^n and independent of the summation in Step 7 of Algorithm 6.7.

$$\left|\text{Adv}_{\mathcal{A}}^{\textbf{Game}_5} - \text{Adv}_{\mathcal{A}}^{\textbf{Game}_6}\right| = 0.$$

Note that **Mint** is completely independent of all the inputs to **Spend** except for output amounts, which is already known to $\mathcal{A}$. Also, output coin keys CK_{out} are always generated independently and uniformly at random. Therefore, in **Game**$_6$, the output of **Spend** is independent of R_{in}, K_{in} and Amt_{in}, and thus also independent of b. Hence, $\mathcal{A}$ has probability $1/2$ of outputting $b' = b$ in **Game**$_6$. □

6.6.4 Balance

Lemma 6.11. *(Balance) Assume that* $q > \left(2p\sqrt{K}\right)^K$ *and* $q \equiv 2K+1 \bmod 4K$ *for some* $1 < K \leq d$ *where* K *is a power of 2. Let* $\kappa = k(k+1)/2$ *and* θ *be a positive real number such that the Euclidean norm of any product of* $\kappa - 1$ *elements in* $\Delta\mathcal{C}_{w,p}^d$ *is at most* θ*. If M-LWE*$_{m-n-n_s,m,q,\mathcal{B}}$*, M-SIS*$_{n,m+r,q,2\gamma}$ *and M-SIS*$_{\hat{n},\hat{m}+v,\hat{q},2\gamma_{\text{bin}}}$ *are hard where* $v = 2\left(k(\beta-1)+r-1+Sr\right)$*,*
$\gamma_{\text{bin}} = 2p\sqrt{dw}\left(T_g + \hat{\mathcal{B}}_{big}^2 \hat{m} d\right)^{1/2}$ *and*

$$\gamma = \max\left\{\begin{array}{c}(k+1)\cdot d\cdot(2p)^{\kappa'} w^{\kappa'-1}\sqrt{md}\cdot\max\{\mathcal{B}_{big,k}, \mathcal{B}'_{big,k}\}, \\ \theta\sqrt{d}\cdot(S+1)\cdot 2\left(9r\mathcal{B}_r^2 d + \mathcal{B}_{big}^2 md\right)^{1/2}\end{array}\right\},$$

then no PPT adversary can win `Exp:Balance` *without shuffling with non-negligible probability.*

Proof. First, due to the M-SIS assumptions, HMC is γ-binding when instantiated with parameters $n, m, q, \mathcal{B}$ and γ_{bin}-binding when instantiated with parameters $\hat{n}, \hat{m}, \hat{q}, \mathcal{B}$. We separate the proof into three cases.

Case 1 (forgery): Let $\mathcal{E}_{\text{forge}}$ be the event that $\mathcal{A}$ wins the game in a way that there exists $\mathsf{s}_{i^*} \in \mathsf{SN}_{j^*}$ with $0 \leq i^* \leq M-1$ and $1 \leq j^* \leq t$ such that $\mathsf{s}_{i^*} \in \mathcal{L}$ and $\mathcal{L}[\mathsf{s}_{i^*}].\mathsf{IsCrpt} = 0$. In this case, the proof follows as in the unforgeability proof of Theorem 5.16, which is sketched below.

$\mathcal{D}$ creates an invalid public key pk_ℓ for $\textsc{PkGen}(\ell)$ query such that $\mathsf{pk}_\ell = \text{Com}_{ck}(1, 0, \ldots, 0; \boldsymbol{r})$ for $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{d \cdot m}$. pk_ℓ is computationally indistinguishable from a valid public key by M-LWE$_{m-n,m,q,\mathcal{B}}$ hardness assumption. $\mathcal{D}$ runs $\mathcal{A}$ until $\mathcal{E}_{\text{forge}}$ occurs $k' = \max\{k+1, 3\}$ times in total with respect to distinct $\mathcal{H}$ outputs and the same $\mathcal{H}$ inputs where

- the indices j^* and i^* are the same for all k' $\mathcal{E}_{\text{forge}}$ events,
- $\mathsf{pk}_\ell \in \mathsf{A}_{\text{in}}^{j^*}$ and it is not corrupted.

k'-special soundness of the underlying one-out-of-many proof holds when HMC is γ- and γ_{bin}-binding, which is satisfied if M-SIS$_{n,m+r,q,2\gamma}$ and M-SIS$_{\hat{n},\hat{m}+v,\hat{q},2\gamma_{\text{bin}}}$ are hard. Therefore, there exists a PPT extractor that recovers an opening $(\mathbf{0}, \boldsymbol{s})$ of a public key pk_ψ in the i^* row of $\mathsf{A}_{\text{in}}^{j^*}$ such that $y \cdot \mathsf{pk}_\psi = \text{Com}_{ck}(\mathbf{0}; \boldsymbol{s})$ where $y \in \Delta\mathcal{C}_{w,p}^d$ is some relaxation factor with $\|y\| = \sqrt{d}(2p)^\kappa w^{\kappa-1} \ll \gamma$ and $\|\boldsymbol{s}\| \leq \gamma$ by Lemma 6.8. With probability $1/(M \cdot N)$, $\mathsf{pk}_\ell = \mathsf{pk}_\psi$. Hence, $y \cdot \mathsf{pk}_\ell = \text{Com}_{ck}(y, 0, \ldots, 0; y\boldsymbol{r}) = \text{Com}_{ck}(\mathbf{0}; \boldsymbol{s}) = y \cdot \mathsf{pk}_\psi$. Since $(y, 0, \ldots, 0, y\boldsymbol{r}) \neq (\mathbf{0}, \boldsymbol{s})$, this violates the γ-binding property of the commitment scheme and also gives a solution to M-SIS$_{n,m,q,2\gamma}$, which gives a contradiction.

Case 2 (double-spend): Let $\mathcal{E}_{\text{2xspend}}$ be the event that $\mathcal{A}$ wins the game in a way that there exists $\mathsf{s}_{i^*} \in \mathsf{SN}_{j^*}$ with $0 \leq i^* \leq M-1$ and $1 \leq j^* \leq t$ such that $\mathsf{s}_{i^*} \notin \mathcal{L}$. Assume that the assumptions in the statement of the lemma hold and $\mathcal{E}_{\text{2xspend}}$ happens. We show that this gives a contradiction. Since the transactions output by $\mathcal{A}$ are valid, we have from Algorithm 6.10

$$\boldsymbol{G} \cdot \boldsymbol{z}^{(i^*)} = \sum_{i=0}^{N} \left(\prod_{j=0}^{k-1} f_{j,i_j} \right) \mathsf{pk}_{i^*,i} - \sum_{j=0}^{k-1} E_j^{(i^*)} x^j, \tag{6.30}$$

$$\boldsymbol{H} \cdot \boldsymbol{z}^{(i^*)} = x^k \mathsf{s}_{i^*} - \sum_{j=0}^{k-1} F_j^{(i^*)} x^j, \tag{6.31}$$

where $\mathsf{pk}_{i^*,i}$ is an honestly generated public key for all $i \in [0, N-1]$. Again using the extractor of the underlying ZKP as in Case 1, $\mathcal{D}$, who runs $\mathcal{A}$ multiple times, obtains a witness $\boldsymbol{s}$ such that

$$y \cdot \mathsf{pk}_{i^*,\ell} = \boldsymbol{G} \cdot \boldsymbol{s}, \tag{6.32}$$

$$y \cdot \mathsf{s}_{i^*} = \boldsymbol{H} \cdot \boldsymbol{s}, \tag{6.33}$$

for some $0 \leq \ell \leq N-1$ where y is a product of κ elements in $\Delta\mathcal{C}_{w,p}^d$ with $\|y\| \ll \gamma$ and $\|\boldsymbol{s}\| \leq \gamma$ by Lemma 6.8. Since $\mathsf{pk}_{i^*,\ell}$ is an honestly generated public key, we also have

$$\begin{aligned} \mathsf{pk}_{i^*,\ell} &= \boldsymbol{G} \cdot \boldsymbol{r}_\ell \\ \mathsf{s}_\ell &= \boldsymbol{H} \cdot \boldsymbol{r}_\ell \end{aligned} \quad \Longrightarrow \quad y \cdot \mathsf{pk}_{i^*,\ell} = \boldsymbol{G} \cdot y\boldsymbol{r}_\ell \tag{6.34}$$

for some $\boldsymbol{r}_\ell \in R_q^m$ with $\|\boldsymbol{r}_\ell\|_\infty = \mathcal{B}$ where $\mathsf{s}_\ell = \mathcal{L}[\mathsf{pk}_{i^*,\ell}].\mathsf{s}$. Using (6.32), right side of (6.34) and γ-binding of HMC with respect to $\boldsymbol{G}$, we get $\boldsymbol{s} = y\boldsymbol{r}_\ell$. Then, from (6.33), we get

$$y \cdot \mathsf{s}_{i^*} = \boldsymbol{H} \cdot y\boldsymbol{r}_\ell \quad \Longrightarrow \quad \mathsf{s}_{i^*} = \boldsymbol{H} \cdot \boldsymbol{r}_\ell \tag{6.35}$$

since y is invertible because all its factors are invertible by Lemma 6.4. From right side of (6.35) and left side of (6.34), we conclude that $\mathsf{s}_{i^*} = \mathsf{s}_\ell \in \mathcal{L}$, which gives a contradiction.

Case 3 (unbalanced amounts): Let $\mathcal{E}_{\text{unbalanced}}$ be the event that $\mathcal{A}$ wins the game in a way that for all $\mathsf{s}_i \in \mathsf{SN}_{j^*}$ where $0 \leq i \leq M-1$, $\mathsf{s}_i \in \mathcal{L}$ and $\mathcal{L}[\mathsf{s}_i].\mathsf{IsCrpt} = 1$. Assume that the assumptions in the statement of the lemma hold and there exists a PPT $\mathcal{D}$ who runs $\mathcal{A}$.

As in Case 1, $\mathcal{D}$ runs $\mathcal{A}$ until $\mathcal{E}_{\text{unbalanced}}$ occurs $k' = \max\{k+1, 3\}$ times with respect to distinct $\mathcal{H}$ outputs and the same $\mathcal{H}$ inputs where the index j^* is the same for all k' events. Then, it uses the extractor of the underlying ZKP of the j^*-th transaction to obtain the following, for all $i \in [0, S-1]$,

$$\bar{x} \cdot C = \mathrm{Com}_{ck}\left(\bar{x}c_0 - \bar{x}2c_1, \ldots, \bar{x}c_{r-1} - \bar{x}2c_r;\, \boldsymbol{r}_c\right), \tag{6.36}$$

$$\bar{x} \cdot \mathsf{cn}_{\text{out},i} = \mathrm{Com}_{ck}\left(\bar{x}b^{(i)}_{\text{out},0}, \ldots, \bar{x}b^{(i)}_{\text{out},r-1};\, \boldsymbol{r}_{\text{out},i}\right), \tag{6.37}$$

$$y \cdot P_\ell = \mathrm{Com}_{ck}\left(0;\, \boldsymbol{r}\right), \text{ with } P_\ell = \sum_{j=0}^{S-1} \mathsf{cn}_{\text{out},j} - \sum_{j=0}^{M-1} \mathsf{cn}_{j,\ell} + C \tag{6.38}$$

where

- $c_0 = c_r = 0$ and $c_1, \ldots, c_{r-1} \in [-(M-1), (S-1)]$,[8]
- $\bar{x} \in \Delta\mathcal{C}^d_{w,p}$,
- y is a product of κ elements in $\Delta\mathcal{C}^d_{w,p}$ where one of its factors is $\bar{x}$ by Lemma 6.8,
- $\|\boldsymbol{r}\| \leq \gamma$ by Lemma 6.9,
- $\|(\bar{x}c_0 - \bar{x}2c_1, \ldots, \bar{x}c_{r-1} - \bar{x}2c_r, \boldsymbol{r}_c)\| \leq \gamma/((S+1)\theta\sqrt{d})$,
  $\left\|\bar{x}b^{(i)}_{\text{out},0}, \ldots, \bar{x}b^{(i)}_{\text{out},r-1}, \boldsymbol{r}_{\text{out},i}\right\| \leq \gamma/((S+1)\theta\sqrt{d})$ by Lemma 6.7,
- $b^{(i)}_{\text{out},j} \in \{0,1\}$ for all $i \in \{0, \ldots, S-1\}$ and all $j \in \{0, \ldots, r-1\}$.

Multiplying (6.36) and (6.37) by $y' = y/\bar{x}$, we get

$$y \cdot C = \mathrm{Com}_{ck}\left(yc_0 - y2c_1, \ldots, yc_{r-1} - y2c_r;\, y'\boldsymbol{r}_c\right), \tag{6.39}$$

$$y \cdot \mathsf{cn}_{\text{out},i} = \mathrm{Com}_{ck}\left(yb^{(i)}_{\text{out},0}, \ldots, yb^{(i)}_{\text{out},r-1};\, y'\boldsymbol{r}_{\text{out},i}\right), \tag{6.40}$$

where

$$\left\|(yc_0 - y2c_1, \ldots, yc_{r-1} - y2c_r, y'\boldsymbol{r}_c)\right\| \leq \gamma/(S+1), \text{ and}$$
$$\left\|yb^{(i)}_{\text{out},0}, \ldots, yb^{(i)}_{\text{out},r-1}, y'\boldsymbol{r}_{\text{out},i}\right\| \leq \gamma/(S+1).$$

Since the input coins are generated honestly, we also have

$$\mathsf{cn}_{i,\ell} = \mathrm{Com}_{ck}\left(b_{i,0}, \ldots, b_{i,r-1};\, \boldsymbol{r}_i\right) \tag{6.41}$$

where $\|\boldsymbol{r}_i\|_\infty = \mathcal{B}$ and $b_{i,j} \in \{0,1\}$ for all $i \in \{0, \ldots, M-1\}$ and all $j \in \{0, \ldots, r-1\}$. Substituting (6.39), (6.40) and (6.41) into (6.38), we get

[8] Here, we assume the general case where the spender proves that c_i's are in $[-(M-1), (S-1)]$, and need not be necessarily binary.

$$\begin{aligned}\mathrm{Com}_{ck}(0;\boldsymbol{r}) = &\sum_{i=0}^{S-1}\left(\mathrm{Com}_{ck}\left(yb^{(i)}_{\mathrm{out},0},\ldots,yb^{(i)}_{\mathrm{out},r-1};\, y'\boldsymbol{r}_{\mathrm{out},i}\right)\right)\\ &-\sum_{i=0}^{M-1}\left(\mathrm{Com}_{ck}\left(yb_{i,0},\ldots,yb_{i,r-1};\, y\boldsymbol{r}_i\right)\right)\\ &+\mathrm{Com}_{ck}\left(yc_0-y2c_1,\ldots,yc_{r-1}-y2c_r;\, y'\boldsymbol{r}_c\right).\end{aligned}$$

Observe that the input of the commitment on the left hand side has Euclidean norm at most γ. Similarly, after using the homomorphic properties of the commitment scheme, the input of the commitment on the right hand side has norm at most γ (here we neglect the norm of $(yb_{i,0},\ldots,yb_{i,r-1},y\boldsymbol{r}_i)$ as that is much smaller in comparison). Then, using γ-binding property of HMC, we get

$$0 = y\sum_{i=0}^{S-1} b^{(i)}_{\mathrm{out},j} - y\sum_{i=0}^{M-1} b_{i,j} + yc_j - y2c_{j+1} \tag{6.42}$$

for all $j \in \{0,\ldots,r-1\}$ with $c_0 = c_r = 0$. By the assumption on q and Lemma 6.4, y is invertible in R_q, and thus we have[9]

$$0 = \sum_{i=0}^{S-1} b^{(i)}_{\mathrm{out},j} - \sum_{i=0}^{M-1} b_{i,j} + c_j - 2c_{j+1}, \tag{6.43}$$

where with $c_0 = c_r = 0$. Since HMC is γ-binding, we have $q > \gamma \gg \max\{4M, 4S\}$. Hence, (6.43) holds over R. Since all the values are just integers, (6.43) in fact holds over $\mathbb{Z}$.

By the definition of `Exp:Balance`, the sum of the amounts in $\mathsf{Amt}^{j^*}_{\mathrm{out}}$ (i.e., the amounts corresponding to *uncorrupted* output public keys) can be at most the sum of the amounts in all output coins. Using this fact, we look at the following sum where $\mathsf{amt}_{\mathrm{in},i} = \mathcal{L}[\mathsf{s}_i].\mathsf{amt}$ for $\mathsf{s}_i = \mathsf{SN}_{j^*}[i]$

$$\begin{aligned}\sum_{i=0}^{S'-1}\mathsf{Amt}^{j^*}_{\mathrm{out}}[i] - \sum_{i=0}^{M-1}\mathsf{amt}_{\mathrm{in},i} &\le \sum_{i=0}^{S-1}\sum_{j=0}^{r-1} 2^j b^{(i)}_{\mathrm{out},j} - \sum_{i=0}^{M-1}\sum_{j=0}^{r-1} 2^j b_{i,j}\\ &= \sum_{j=0}^{r-1} 2^j\left(\sum_{i=0}^{S-1} b^{(i)}_{\mathrm{out},j} - \sum_{i=0}^{M-1} b_{i,j}\right) = \sum_{j=0}^{r-1} 2^j\left(2c_{j+1}-c_j\right)\\ &= \sum_{j=0}^{r-1} 2^{j+1}c_{j+1} - \sum_{j=0}^{r-1} 2^j c_j = 2^r c_r - c_0\\ &= 0, \quad \text{since } c_0 = c_r = 0.\end{aligned}$$

The above implies that

$$\sum_{i=0}^{S'-1}\mathsf{Amt}^{j^*}_{\mathrm{out}}[i] \le \sum_{i=0}^{M-1}\mathsf{amt}_{\mathrm{in},i},$$

[9] We note here that (6.43) can also be obtained without using invertibility of y. In that case, one can argue that the infinity norm of the right-hand side of (6.42) is smaller than $q/2$, which would be easily satisfied. That implies that (6.42) holds over R. Then, by Lemma 5.5, either $y = 0$ or (6.43) holds. Since $y \neq 0$, (6.43) must hold.

which gives a contradiction with the winning assumption of $\mathcal{A}$ in `Exp:Balance`. □

Remark 6.12. *Note that in Lemma 6.11, the factor $\theta\sqrt{d}$ can be taken to be 1, when $k = 1$. This is due to fact that in this case, $\kappa = 1$, and thus $y = \bar{x}$. Hence, there is no need to do cross multiplication to have (6.36), (6.37) and (6.38) multiplied by the same relaxation factor y.*

6.7 Implementation and Parameters

In our implementation, we target any anonymity level $1/N$ for $N \leq 1000$, 64-bit precision for amounts (i.e., $r = 64$) and the most common transaction settings where there are at most two input/output accounts (i.e., $M, S \leq 2$). For all these settings, the following parameters are sufficient: $\mathcal{B} = 1$, $(d, w, p) = (64, 56, 8)$, $q = 2^{31} - 2^{18} + 2^3 + 1$, $\hat{q} = (2^{27} - 2^{11} + 1) \cdot (2^{26} - 2^{12} + 1)$, $k = 1$, $n_s = 1$, $(n, m) = (18, 38)$ and $(\hat{n}, \hat{m}) = (32, 65)$. With these parameters, a single public costs 4.36 KB and a single serial number costs 248 bytes. The rationale behind the parameter setting is as follows.

First, our experimental analysis shows that $d = 64$ is the best choice to optimise the proof length. Having set $d = 64$, we get $(w, p) = (56, 8)$ to have about 256-bit $\mathcal{H}$ output. Again we follow the same methodology from Section 3.2.2 to measure the practical security of our scheme, aim for a root Hermite factor of $\delta \approx 1.0045$ for both M-LWE and M-SIS, and choose $\mathcal{B} = 1$.

From our security assumption M-LWE$_{m-n-n_s,m,q,\mathcal{B}}$, we can see that the efficiency of our scheme degrades with increasing n_s as M-LWE gets easier. Indeed, having a small n_s does not affect the anonymity or balance properties. Therefore, we can simply set $n_s = 1$. We discuss the implications of this choice in more detail in Section 6.8. Then, we set the remaining parameters to make sure that M-LWE and M-SIS assumptions hold with $\delta \approx 1.0045$.

q is chosen to allow R_q to split into 4 factors while having $y = x - x'$ (challenge differences) invertible in R_q for any $x, x' \in \mathcal{C}^d_{w,p}$. This follows from the results of [LS18] as given in Lemma 6.4. The other modulus $\hat{q}$ is chosen to have two "NTT-friendly" prime factors p_1 and p_2 so that both R_{p_1} and R_{p_2} fully splits, allowing efficient polynomial multiplication using NTT. All these primes p_1, p_2 and q are chosen to have a form similar to $2^{k_1} - 2^{k_2} + 1$, which enables the fast modulo reduction technique in [Sei18] for the input smaller than $2^{k_1 - k_2}$ times the modulus. By using this technique, we only apply one modulo reduction at the end when computing $\sum x_i$, $x_i \in R_q$ or $x_i \in R_{\hat{q}}$, such as in the commitments.

To reduce the number of NTT transformations, SamMat samples uniformly at random directly from the NTT domain[10]. In addition, all commitment outputs (including pk and cn) are in the NTT domain (i.e., without any inverse NTT during commitment computations). However, since the secrets (notably a, $\boldsymbol{r}$, and $\boldsymbol{\rho}$) are involved in the norm checks of **Spend** in Algorithm 6.9, and norm checks are also required for the output $\boldsymbol{f}$ and $\boldsymbol{z}$ of **Spend** during **Verify**, we keep these elements in their normal domain and perform NTT when computing the commitments. Therefore, only forward NTT is needed and we eliminate all the inverse NTT from the implementation.

To accelerate the norm checks and avoid unnecessary overhead during the rejection of **Spend** in Algorithm 6.9, we adapt the early-evaluation rejection technique in [RGCB19]. In particular, we check the infinity or Euclidean norm and restart immediately during each ring element computation of $\boldsymbol{f}_r$, $\boldsymbol{g}$, and all $\boldsymbol{z}$'s. However, for $\boldsymbol{f}_1$, we

[10] What we mean by NTT domain for R_q is the four factors it splits into.

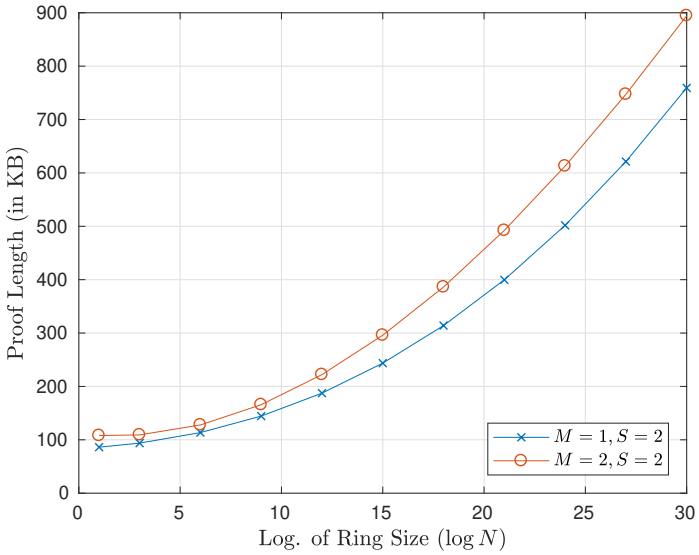


FIGURE 6.1: Proof length growth (including the cost of serial numbers) with anonymity set (ring) size.

need to hide what index may give a rejection due to the application of our rejection sampling technique for fixed Hamming weight binary secrets. Therefore, a restart happens only after $\boldsymbol{f}_1$ is completely iterated over. In addition, since T_g is larger than 64 bits, we use the GMP library [GT15] to compute $\|\boldsymbol{g}\|$ and make the comparison.

To implement the NTT efficiently, we adapt the techniques discussed in [Sco17] for both factors R_{p_1} and R_{p_2} of $R_{\hat{q}}$ during the NTT butterfly computations, notably the lazy Montgomery reduction. However, for multiplication in R_q, since the input would be reduced to $[0, 2q-1]$ in the lazy reduction, the intermediate value during multiplication reduction may exceed 64 bits for the input less than $4q^2$. Thus, we use the full Montgomery reduction for R_q instead. In addition, we also adapt the constant-time comparison techniques similar to [Sco17] in our NTT implementation and uniform samplers (e.g., $\{-\mathcal{B}, \ldots, \mathcal{B}\}$ or $\{-\mathcal{B}_{\text{big}}, \ldots, \mathcal{B}_{\text{big}}\}$) for the secrets.

In our implementation, we use the AES-NI hardware instructions on Intel CPUs [Gue09] to implement the pseudorandom generator and use the SHAKE-256 [NIS] to implement the hash function $\mathcal{H}$. We compile our implementation by using GCC 8.3.0 with the compiler switch `-O3 -march=native` during the benchmarks.

The concrete proof lengths of MatRiCT are compared with the prior art in Table 6.4, and the computational evaluation of MatRiCT is given in Table 6.5, where the running times are the average number of cycles in 1000 runs divided by $3 \cdot 10^6$. Asymptotically, the proof generation and verification times are $O(M \cdot N)$ as M ring

TABLE 6.4: Proof length comparison (in KB) of "post-quantum" RingCT proposals, supporting multiple inputs/outputs.

Anonymity level	1/10		1/100	
#inputs → #outputs	1 → 2	2 → 2	1 → 2	2 → 2
LRCT v2.0 [TKS+19]	>8000	>10000	>50000	>70000
MatRiCT: Chapter 6	**93**	**110**	**103**	**120**
LRCT v2.0 [TKS+19]	PK Size: 100 KB		Modulus: $\approx 2^{196}$	
MatRiCT: Chapter 6	PK Size: 4 KB		Modulus: $< 2^{53}$	

TABLE 6.5: Running times (in ms) of MatRiCT at 3 GHz.

Anonymity level	1/10		1/100		1/1000	
#inputs → #outputs	1 → 2	2 → 2	1 → 2	2 → 2	1 → 2	2 → 2
Key Gen.	2	2	2	2	2	2
Transaction Gen.	242	375	360	620	1858	3514
Verification	20	23	31	40	146	223

signatures are run, each with $O(N)$ computation. Further, we show in Figure 6.1 that MatRiCT proof length scales very slowly with anonymity set size. The proof length scales linearly with the number of input accounts as shown in Figure 6.2. Asymptotically, the proof length grows poly-logarithmically in N (due to the use of an improved variant of the ring signature from Chapter 5) and linear in M, i.e., $|\Pi| = O(M \cdot \log^c N)$ for a small constant c.

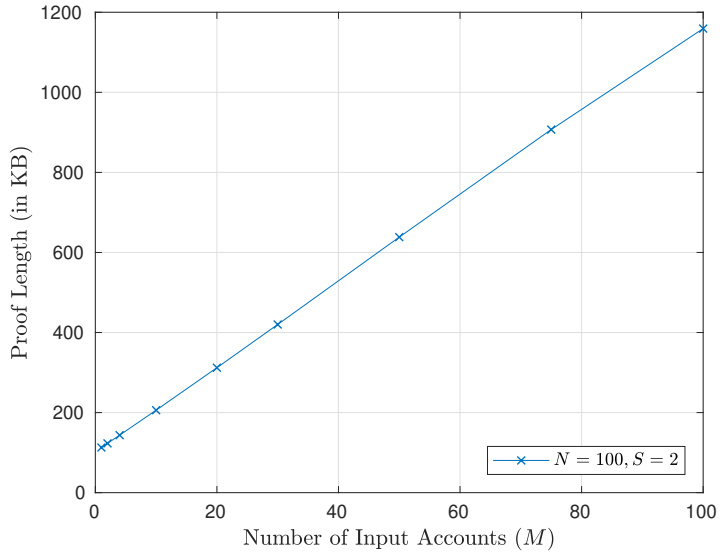


FIGURE 6.2: Proof length growth with the number of input accounts. Proof length also includes the cost of serial numbers.

6.8 Implications of Small Dimensional Serial Number

Recall that for the commitment matrix $\boldsymbol{H} \in R_q^{n_s \times m}$, a serial number is computed as $\mathsf{s} = \mathrm{Com}_{\boldsymbol{H}}(0; \mathsf{sk})$ where sk is a secret key. Choosing $n_s = 1$ here with $d = 64$ actually makes M-SIS easy to solve with respect to $\boldsymbol{H}$. However, one can see from the balance proof (proof of Lemma 6.11) that the adversary needs to find a witness either for $(\mathsf{pk}, \mathsf{s})$ *together* or for pk in order to break the balance property. Since we make sure that the public key commitments are binding, there is no issue in the balance proof. Similarly, from the anonymity perspective, the serial number alone will clearly hide the secret key as we show in the anonymity proof (proof of Lemma 6.10) that $(\mathsf{pk}, \mathsf{s})$ together still hides the secret key. Moreover, even with $n_s = 1$, the output space of 1-dimensional commitments is sufficiently large (in particular, $q^d \approx 2^{31 \cdot 64}$ for our parameters) that there is a negligible chance for two random secret keys to result in the same serial number.

What may indeed happen is that, for a given s, an adversary can find some *short* vector $\boldsymbol{r}'$ such that $\boldsymbol{r}' \neq \mathsf{sk}$ and $\mathsf{s} = \mathrm{Com}_{\boldsymbol{H}}(0; \boldsymbol{r}')$. In this case, it must hold that $\mathsf{pk} = \mathrm{Com}_{\boldsymbol{G}}(0; \mathsf{sk}) \neq \mathrm{Com}_{\boldsymbol{G}}(0; \boldsymbol{r}')$ as otherwise we would get a solution to M-SIS$_{n,m,q,2\gamma}$ with respect to $\boldsymbol{G}$ where $\gamma = \max\{\|\boldsymbol{r}'\|, \|\mathsf{sk}\|\}$. Hence, the adversary still cannot create a valid transaction without having $\mathsf{pk}' = \mathrm{Com}_{\boldsymbol{G}}(0; \boldsymbol{r}')$ as one of the public keys of the real spent accounts in R_{in}. Therefore, he first requires an account to be created with pk' and then he can spend that account with the serial number s. If he can succeed in this *before* the honest user whose secret key commits to the same serial number s, then this would prevent the honest user from being able to spend her account. Note that the attack works only when the attacker guesses/knows the serial number of an *unspent* account, and only results in a violation of availability (as the honest user can no longer spend her account). The chance of a correct guess is negligibly small as the output space is too large, which leaves knowledge of an account serial number before it is ever spent as the only viable option for the attacker. This whole attack scenario in general does not seem very likely to happen. Nevertheless, stronger security against such an attack can be easily accomplished by increasing n_s so that M-SIS is hard with respect to $\boldsymbol{H}$, which requires increasing m, and possibly n and/or q.

6.9 Extension to Auditable RingCT

In this section, we introduce our novel extractable commitment scheme from lattices and explain how to use it to add auditability to MatRiCT.

6.9.1 Extractable commitment scheme

We extend HMC to allow message extraction. All the previous algorithms, $\mathsf{CKeygen}$, Commit and COpen, that define the commitment remain the same, and we introduce how to put a trapdoor to a commitment key.

- $\mathsf{CAddTrapdoor}(ck)$: Let $ck = [\,\boldsymbol{A} \,\|\, \boldsymbol{B}\,] \in R_q^{n\times(m+v)}$ where $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}' \\ \boldsymbol{a}^\top \end{bmatrix}$ for $\boldsymbol{A}' \in R_q^{(n-1)\times m}$ and $\boldsymbol{a} \in R_q^m$. Sample $\boldsymbol{s}' \leftarrow R_q^{n-1}$, $\boldsymbol{e} \leftarrow \mathfrak{U}^m_{\mathcal{B}_\mathrm{e}}$, and set $\boldsymbol{A}^{\mathsf{td}} = \begin{bmatrix} \boldsymbol{A}' \\ \boldsymbol{t}^\top \end{bmatrix}$ where $\boldsymbol{t} = \boldsymbol{A}'^\top \boldsymbol{s}' + \boldsymbol{e}$. Output $(ck^{\mathsf{td}}, \mathsf{td}) = (\boldsymbol{A}^{\mathsf{td}}, (\boldsymbol{s}, \boldsymbol{e}))$ where $\boldsymbol{s} = (\boldsymbol{s}', -1)$.

We next introduce how the extraction works when the committed message comes from a relatively small set. Let $\Delta\mathcal{C}^d_{w,p}$ be the set of differences of all challenges in $\mathcal{C}^d_{w,p}$ except for the zero element. When a commitment key with a trapdoor is used to generate a proof, the ZKPs we use prove knowledge of an opening $(y, \boldsymbol{m}, \boldsymbol{r})$ of a commitment C such that

$$yC = \mathrm{Com}_{ck}\,(y\boldsymbol{m}; \boldsymbol{r}) = \boldsymbol{A}^{\mathsf{td}}\boldsymbol{r} + \boldsymbol{B}y\boldsymbol{m}. \tag{6.44}$$

From here, we can try to eliminate the randomness by multiplying both sides by the secret key $\boldsymbol{s}$. However, the message extractor does not know what y is. For an honest user, we simply have $y = 1$ and we restrict our discussion here to this case. However, we note that, for a similar Fiat-Shamir protocol, it has been shown in [LN17] that a valid approach in general is actually trying random $y \in \Delta\mathcal{C}^d_{w,p}$, and the expected number of iterations until an acceptable y is reached is the same as the number of random oracle queries made to generate the proof. We believe that the same technique

(which is also used in [dPLS18]) and [LN17, Lemma 3.2] can be applied in our case and we leave its detailed investigation to future work.

Now, suppose that $y = 1$. We can rewrite (6.44) as $C = \boldsymbol{A}^{\mathsf{td}}\boldsymbol{r} + \boldsymbol{B}\boldsymbol{m}$. From here, the extraction proceeds as given in Algorithm 6.12.

Algorithm 6.12 $\mathsf{CExtractSM}(C, \mathsf{td})$

INPUT: $C \in R_q^n$ a commitment; $\mathsf{td} = (\boldsymbol{s}, \boldsymbol{e})$ trapdoor

1: **for** $\boldsymbol{m}' \in \mathcal{M}$ **do** ▷ where $|\mathcal{M}| = \mathrm{poly}(\lambda)$
2: $\quad e' = \langle \boldsymbol{s}, C \rangle - \langle \boldsymbol{b}, \boldsymbol{m}' \rangle$ where $\boldsymbol{b} = \boldsymbol{s}^\top \boldsymbol{B}$
3: $\quad$ **if** $\|e'\|_\infty < q/8$, **then return** $\boldsymbol{m}'$
4: **end for**

We prove in Lemma 6.13 that, for a commitment C with a valid zero-knowledge proof of opening, the message output by Algorithm 6.12 is the same as the one used to create the commitment C for sufficiently large q.

Lemma 6.13. *Let* $ck = \boldsymbol{G} = [\,\boldsymbol{A}^{\mathsf{td}} \,\|\, \boldsymbol{B}\,] \in R_q^{n \times (m+v)}$ *be a commitment key with a trapdoor* $\mathsf{td} = (\boldsymbol{s}, \boldsymbol{e})$ *as in* $\mathsf{CAddTrapdoor}$. *Assume that* $(C, (\boldsymbol{m}, \boldsymbol{r}))$ *satisfy* $C = \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r})$, $\|\boldsymbol{r}\|_\infty \leq \mathcal{B}_r$ *and* $\boldsymbol{m} \in \mathcal{M}$ *with* $|\mathcal{M}| = s$, *and* $\boldsymbol{m}' = \mathsf{CExtractSM}(C, \mathsf{td})$. *If* $q > 8\mathcal{B}_e\mathcal{B}_r md$, *then* $\boldsymbol{m} = \boldsymbol{m}'$ *except for a probability at most* $s \cdot 2^{-d}$.

Proof. It is easy to observe that $\boldsymbol{s}^\top \cdot \boldsymbol{A}^{\mathsf{td}} = \boldsymbol{s}'^\top \boldsymbol{A}' - \boldsymbol{t}^\top = -\boldsymbol{e}^\top$. Let $\boldsymbol{b} = \boldsymbol{s}^\top \cdot \boldsymbol{B}$. Since $C = \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r})$ for $ck = [\,\boldsymbol{A}^{\mathsf{td}} \,\|\, \boldsymbol{B}\,]$, we have

$$\begin{aligned} & \langle \boldsymbol{s}, C \rangle = \langle -\boldsymbol{e}, \boldsymbol{r} \rangle + \boldsymbol{s}^\top \cdot \boldsymbol{B} \cdot \boldsymbol{m} = \langle -\boldsymbol{e}, \boldsymbol{r} \rangle + \langle \boldsymbol{b}, \boldsymbol{m} \rangle, \\ \iff \quad & \langle \boldsymbol{b}, \boldsymbol{m} \rangle = \langle \boldsymbol{s}, C \rangle + \langle \boldsymbol{e}, \boldsymbol{r} \rangle. \end{aligned} \tag{6.45}$$

Since $\boldsymbol{m}'$ is the output of $\mathsf{CExtractSM}(C, \mathsf{td})$, we further have $e' = \langle \boldsymbol{s}, C \rangle - \langle \boldsymbol{b}, \boldsymbol{m}' \rangle$ and $\|e'\|_\infty < q/8$. Now, consider the following

$$\begin{aligned} \langle \boldsymbol{b}, \boldsymbol{m} - \boldsymbol{m}' \rangle &= \langle \boldsymbol{b}, \boldsymbol{m} \rangle - \langle \boldsymbol{b}, \boldsymbol{m}' \rangle \\ &= (\langle \boldsymbol{s}, C \rangle + \langle \boldsymbol{e}, \boldsymbol{r} \rangle) - \big(\langle \boldsymbol{s}, C \rangle - e'\big) \\ &= \langle \boldsymbol{e}, \boldsymbol{r} \rangle + e'. \end{aligned}$$

Therefore, we have

$$\begin{aligned} & \big\|\langle \boldsymbol{b}, \boldsymbol{m} - \boldsymbol{m}' \rangle\big\|_\infty = \big\|\langle \boldsymbol{e}, \boldsymbol{r} \rangle + e'\big\|_\infty \\ & \leq \|\langle \boldsymbol{e}, \boldsymbol{r} \rangle\|_\infty + \big\|e'\big\|_\infty \\ & \leq \mathcal{B}_e\mathcal{B}_r md + q/8 < q/8 + q/8 < q/4. \end{aligned}$$

Since $\boldsymbol{B}$ is chosen independently and uniformly at random, $\boldsymbol{b} = \boldsymbol{s}^\top \cdot \boldsymbol{B}$ is uniformly random, and thus when $\boldsymbol{m} \neq \boldsymbol{m}'$, $\langle \boldsymbol{b}, \boldsymbol{m} - \boldsymbol{m}' \rangle$ is also uniformly random R_q. So, the above holds with probability about 2^{-d}. Thus, using a union bound on all $\boldsymbol{m} \in \mathcal{M}$, $\boldsymbol{m} = \boldsymbol{m}'$ except for a probability at most $s \cdot 2^{-d}$. □

6.9.2 Adding auditability

From the tools developed so far, it is now easy to add auditability to our RingCT construction. As part of **Spend**, the spender proves knowledge of an index $\ell \in [0, N-1]$ and the secret keys of the accounts in the ℓ-th column of A_{in}. Further, as

detailed in Section 6.5, the binary proof part proves knowledge of $(y, \boldsymbol{b}, \hat{\boldsymbol{c}}, \hat{\boldsymbol{r}})$ such that $y \in \Delta\mathcal{C}^d_{w,p}$, $yB = \mathrm{Com}_{ck}(y\boldsymbol{b}, \hat{\boldsymbol{c}}; \hat{\boldsymbol{r}}) \in R^{\hat{n}}_{\hat{q}}$ and the first $k\beta$ elements of $\boldsymbol{b}$ represents an index $\ell \in [0, N-1]$. Hence, we know that

$$yB = \boldsymbol{A}\hat{\boldsymbol{r}} + \boldsymbol{B}\begin{pmatrix} y\boldsymbol{b} \\ \boldsymbol{c} \end{pmatrix} = \boldsymbol{A}\hat{\boldsymbol{r}} + \boldsymbol{B}_0 y\boldsymbol{m} + \boldsymbol{B}_1\hat{\boldsymbol{m}} = [\,\boldsymbol{A} \,\|\, \boldsymbol{B}_1\,]\begin{pmatrix} \hat{\boldsymbol{r}} \\ \hat{\boldsymbol{m}} \end{pmatrix} + \boldsymbol{B}_0 y\boldsymbol{m},$$

where $\boldsymbol{m}$ is the part (the first $k\beta$ elements of $\boldsymbol{b}$) we want to recover and $\hat{\boldsymbol{m}}$ is the remaining part of the message opening. Therefore, restricting to the case $y = 1$, we can put a trapdoor for the concatenated matrix $[\,\boldsymbol{A} \,\|\, \boldsymbol{B}_1\,]$ and use Algorithm 6.12 to extract $\boldsymbol{m}$, which reveals the real spender's identity. The message space size here is equal to the anonymity set size N. Therefore, the extraction time (as in **Spend**) is linear in N. We know by Lemma 6.13 that for an appropriately chosen $\hat{q}$, the extracted index will be the same as the one used in the proof. A formal definition of auditability, which can be established similar to traceability in group signatures [BMW03], is left as a future work.

Note that multiple trapdoors can be put for the same matrix. If no auditing is desired, the last row of the commitment matrix remains as a uniformly random vector. If a user selects auditing option $i > 0$, then a vector released by the i-th authority is used in the last row of the commitment matrix.

6.10 More on Ring and Group Signature

We describe the full ring/group signature signing procedure in Algorithm 6.13. The verification works similar to Algorithm 6.10. In particular, the verifier checks the same norm bounds as in Algorithm 6.13, computes A and E_0, and finally checks whether $x = \mathcal{H}(A, B, E_0, \ldots, E_{k-1})$ for x provided as part of the signature. The opening algorithm for revealing user's identity in the group signature is realised using Algorithm 6.12 in Section 6.9.1. Since the signer proves knowledge of an index $\ell \in [0, N-1]$, which is encoded using $f_{j,i}$'s, and the secret key of the ℓ-th public key, the opener of a group signature just needs to extract the index from the commitment B.[11] As there are only N possibilities[12], the running time of the opening algorithm of the group signature is in the same order as the signing (i.e., $O(N)$). Since signature generation is likely to occur much more frequently than opening a group signature, this is completely acceptable in our setting.

In Tables 6.7 and 6.8, example sets of concrete parameters for our ring and group signature, respectively, are provided, where we optimise the proof length for a fixed ring dimension d. In the case of a group signature, we also need to make sure that M-LWE$_{\hat{n}-1,\hat{m},\hat{q},\mathcal{B}_e}$ is hard in order to argue that a commitment key with a trapdoor is indistinguishable from random. Therefore, the choice of parameters for the group signature is more restrictive. In Table 6.6, a comparison among existing post-quantum scalable ring/group signatures is provided.

The two limitations of our group signature are 1) the running time of the opening algorithm as well as the signing and verification is linear in the group size (the opening algorithm has linear time also in [KKW18]) and 2) the group public key length grows linearly in the group size. These are as expected since the group signature builds on

[11] Note that the binary proof in this case includes only the proof for the index ℓ and no additional bits.

[12] Note here that even though the commitment B includes another vector $\boldsymbol{c}$ as a committed message, by Lemma 6.3, this second part is uniquely determined by the information provided in the proof and recovered in Algorithm 6.12.

Algorithm 6.13 Signing of Ring/Group Signature

INPUT: $\mu, (\mathsf{pk}_0, \ldots, \mathsf{pk}_{N-1}), \ell, \mathsf{sk}$ where μ is a message, $\mathsf{sk} = \boldsymbol{r} \in R_q^m$, $\ell \in [0, N-1]$ and $\mathsf{pk}_i \in R_q^n$ for $i = 0, \ldots, N-1$.

OUTPUT: $\Pi = (B, E_1, \ldots, E_{k-1}, x, \boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z})$ where $B \in R_{\hat{q}}^{\hat{n}}$; $E_1, \ldots, E_{k-1} \in R_q^n$; $x \in \mathcal{C}_{w,p}^d$; $\boldsymbol{f}_1 \in R_{\hat{q}}^{k(\beta-1)}$; $\boldsymbol{z}_b \in R_{\hat{q}}^{\hat{m}}$; $\boldsymbol{z} \in R_q^m$.

$\mathcal{B}_a = \lceil 2 \cdot pkd \rceil$
$T_g = d^3 \mathcal{B}_a^4 k\beta(\beta+1)/(2d)$
$\hat{\mathcal{B}}_{\text{big}} = \lceil 1.5 \mathcal{B} pw \hat{m} d \rceil$
$\mathcal{B}_{\text{big,k}} = \lceil 1.5 \mathcal{B} (pw)^k m d \rceil$
$\boldsymbol{b} = \{(\beta, \text{True}, (\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,\beta-1}), \mathcal{B}_a)\}_{j=0}^{k-1}$
$ck = \boldsymbol{G} \leftarrow \mathsf{SamMat}(\rho, r, q, n, m, \text{“G”})$
$(\boldsymbol{r}_a, \boldsymbol{r}_b), (A, B), (a_{0,0}, \ldots, a_{k-1,\beta-1}) \leftarrow \mathsf{BinaryCommit}(k, \boldsymbol{b}, \mathcal{B}, \hat{\mathcal{B}}_{\text{big}})$
Compute $\boldsymbol{p} = (p_{0,0}, \ldots, p_{N-1,k-1})$ using Algorithm 6.11 with the input $(\ell, a_{0,0}, \ldots, a_{k-1,\beta-1})$ when $k \in \{1, 2\}$
$\boldsymbol{\rho}_0 \leftarrow \{-\mathcal{B}_{\text{big,k}}, \ldots, \mathcal{B}_{\text{big,k}}\}^{d \cdot m}$
for $j = 0, \ldots, k-1$ **do**
    $\boldsymbol{\rho}_j \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{d \cdot m}$ **if** $j \neq 0$
    $E_j = \sum_{i=0}^{N-1} p_{i,j} P_i + \text{Com}_{ck}\left(\boldsymbol{0}; \boldsymbol{\rho}_j\right)$ in R_q^n
end for

$x = \mathcal{H}(\mu, A, B, E_0, \ldots, E_{k-1})$

for $j = 0, \ldots, k-1$ and $i = 0, \ldots, \beta-1$ **do**
    $f_{j,i} = x\delta_{\ell_j,i} + a_{j,i}$
end for
$\boldsymbol{f}_1 = (f_{0,1}, \ldots, f_{k-1,\beta-1})$ ▷ $f_{j,0}$'s are excluded
if $\|\boldsymbol{f}_1\|_\infty > \mathcal{B}_a - p$, **then** Restart
for $j = 0, \ldots, k-1$ **do**
    if $\|f_{j,0}\| > \mathcal{B}_a \sqrt{d(\beta-1)}$, **then** Restart
end for
$\boldsymbol{g} = (f_{0,0}(x - f_{0,0}), \ldots, f_{k-1,\beta-1}(x - f_{k-1,\beta-1}))$
if $\|\boldsymbol{g}\| > \sqrt{T_g}$, **then** Restart
$\boldsymbol{z}_b = x\boldsymbol{r}_b + \boldsymbol{r}_a$ ▷ $\hat{m}$-dimensional
if $\|\boldsymbol{z}_b\|_\infty > \hat{\mathcal{B}}_{\text{big}} - \mathcal{B}pw$, **then** Restart
$\boldsymbol{z} = x^k \boldsymbol{r} - \sum_{j=0}^{k-1} x^j \boldsymbol{\rho}_j$
if $\|\boldsymbol{z}\|_\infty > \mathcal{B}_{\text{big,k}} - \mathcal{B}(pw)^k$, **then** Restart
return $\Pi = (B, E_1, \ldots, E_{k-1}, x, \boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z})$

a ring signature. However, the cost of the public key storage can be amortised over many signatures as one would expect many signatures to be generated by the same group. Consider a group of 1000 members generating 1000 signatures in total, the total storage cost of all public keys and all signatures would be about 65 MB using our group signature. In the case of the two state-of-the-art post-quantum group signatures [KKW18] and [dPLS18], this cost would exceed 400 MB. Therefore, in the settings where the number of group members is not significantly greater than the number of signatures, the overall cost is lower using our group signature.

Moreover, our group signature can be easily made *dynamic* in that the group manager can add or remove members by appending their individual public key to the

group public key gpk or deleting it from gpk. In this case, it is required that group signature is verified using the group public key at the time of signature generation.

We also note that our constructions in this chapter do not require any (discrete) Gaussian sampling. Replacing some of the uniform samplings with a discrete Gaussian in our schemes helps further reduce the signature length, but our goal in this chapter is to introduce an easy-to-implement scheme.

TABLE 6.6: Comparison of signature lengths (in KB) of "post-quantum" ring/group signatures. The last column represents whether ring or group signature is supported. "?" indicates that the signature length cannot be approximated using the results of the respective reference.

Ring/Group Size N :	2	8	64	4096	2^{21}	Ring/Group
[dPLS18]	581	581	581	581	581	Group
[KKW18]	?	?	250	456	?	Ring&Group
Chapter 5	36	41	58	103	256	Ring
This Chapter	**18**	**19**	**31**	**59**	**148**	Ring
This Chapter	**28**	**29**	**34**	**60**	**148**	Group

TABLE 6.7: Concrete parameters of our ring signature, optimised for proof length. Signature and key sizes are in KB. The hash output size (or challenge space size for the underlying protocol) is at least 2^{256}. The root Hermite factor of both M-SIS and M-LWE are about 1.0045. $\mathcal{B} = 1$ and $(d, w, p) = (128, 66, 2)$ always. The signature length can sometimes be slightly further reduced by choosing varying d values. For example, the same parameters of the group signature for $N = 2^{21}$ in Table 6.8 also works for the ring signature.

N	2	8	64	1024	4096	2^{21}
$(n, \hat{n})$	$(8, 9)$	$(8, 9)$	$(8, 11)$	$(13, 10)$	$(13, 12)$	$(21, 13)$
$(m, \hat{m})$	$(17, 22)$	$(17, 23)$	$(17, 25)$	$(27, 26)$	$(27, 26)$	$(42, 28)$
$(\log q, \log \hat{q})$	$(27, 42)$	$(27, 46)$	$(27, 44)$	$(45, 52)$	$(45, 46)$	$(69, 47)$
(k, β)	$(1, 2)$	$(1, 8)$	$(1, 64)$	$(2, 32)$	$(2, 64)$	$(3, 128)$
Sign. Size	18	20	31	48	59	156
PK Size	3.38	3.38	3.38	9.14	9.14	22.64
SK Size	0.27	0.27	0.27	0.42	0.42	0.66

TABLE 6.8: Concrete parameters of our group signature, optimised for proof length. Signature and key sizes are in KB. The hash output size (or challenge space size for the underlying protocol) is at least 2^{256}. The root Hermite factor of both M-SIS and M-LWE are about 1.0045. $\mathcal{B} = 1$, $\mathcal{B}_{\mathrm{e}} = 4$ and $(d, w, p) = (64, 56, 8)$ always. PK Size here refers to the per-user public key size. The group public key size would be $N\times$(PK Size).

N	2	8	64	1024	4096	2^{21}
$(n, \hat{n})$	$(17, 30)$	$(17, 30)$	$(17, 30)$	$(29, 30)$	$(29, 30)$	$(47, 30)$
$(m, \hat{m})$	$(36, 61)$	$(36, 61)$	$(36, 61)$	$(60, 61)$	$(60, 61)$	$(96, 61)$
$(\log q, \log \hat{q})$	$(29, 49)$	$(29, 49)$	$(29, 49)$	$(49, 49)$	$(49, 49)$	$(79, 49)$
(k, β)	$(1, 2)$	$(1, 8)$	$(1, 64)$	$(2, 32)$	$(2, 64)$	$(3, 128)$
Sign. Size	28	29	34	54	60	148
PK Size	3.85	3.85	3.85	11.10	11.10	29.01
SK Size	0.28	0.28	0.28	0.47	0.47	0.75

6.11 Discussion

As can be seen from Figure 6.2, a limitation of MatRiCT is that the proof length is linear in the number of input accounts (i.e., the parameter M), which means that the proof would be costly when there are many, say 1000, input accounts (though such a large M is quite rare in practice). However, this limitation seems hard to overcome because one needs to construct an *efficient* M-out-of-N proof with proof length growth sublinear in M and N in order to circumvent this limitation. Currently, such proofs seem difficult to construct from lattices.

Another effect of increasing number of input accounts is that it makes the parameter setting more difficult. More precisely, due to the use of rejection sampling, each ring signature run as part of the transaction generation reduces the overall acceptance probability by a certain factor. Therefore, this acceptance probability gets exponentially smaller for increasing M unless ring signature components are sampled from wider distributions. As a result, if one wants to allow large number of inputs, then the parameter setting needs to be done accordingly, which results in a level of overkill for small M values. There seems to be no way around this when multiple ring signatures with rejection samplings are used.

As mentioned in Section 6.10, our group signature also has certain limitations, which are indeed inherited from the ring signature. These limitations may be restrictive for the scheme's use in certain scenarios, for example, when there are large number of group members. In such a case, the group public key may grow very large. It would be very interesting to see if these limitations can be overcome, e.g., by compressing the group public key somehow.

On the positive side, we can see MatRiCT as an important step forward in the deployment of post-quantum cryptographic algorithms. Its communication and computation costs are within practical limits, and it has relatively small public key and modulus sizes. It allows other nice features such as auditability. In fact, the auditability feature is quite unique in that the users themselves can decide what level of anonymity to choose. In the scenarios where auditability is a must, then the authority (or authorities) can enforce users to opt in for auditing by having transaction validation done via commitment keys with trapdoors. That is, if a user does not use a commitment key with a trapdoor, then her transaction would fail in the verification.

An important feature of MatRiCT is that the modulus size is, in essence, determined by the security requirements, not the balance proof. As mentioned before, the prior balance proof approach requires the modulus to be larger than the amounts. Removing this requirement gives MatRiCT much more flexibility in parameter choices, and allows faster computation and shorter proof length.

A more low-level technical side of MatRiCT is that it only makes use of *relaxed* proofs, which are often more efficient to construct in lattice-based cryptography. Of course, the use of relaxed proofs is not so straightforward as one needs to handle the effect of the relaxations. Most notably, we see its impact on the balance proof. We believe that our techniques to handle relaxations may also prove useful for future works that rely on relaxed proofs.

Chapter 7

Conclusion

In this thesis, we studied the problems related to the design of lattice-based zero-knowledge proofs and their application to higher level protocols. We first approached the problem from a foundational perspective and studied it in a less restrictive setting of multi-shot proofs. This approach was an important starting point as there was no set of prior technical tools that could *efficiently* solve the problems of interest in the thesis. In particular, a set of new technical tools for the design of *multi-shot* proofs for non-linear relations was introduced, followed by their application in the construction of useful protocols in practice.

Having seen the main challenges in proving non-linear relations in the lattice setting, we later focused on the more practical setting of *one-shot* proofs. Again, the prior techniques were very limited in that they could only allow one to prove linear relations, which are not sufficient to construct efficient advanced cryptographic schemes. Therefore, our next objective was to introduce new techniques for the design of *one-shot* proofs for *non-linear* relations. With this objective completed, we sought for practical applications of the new techniques, and were able to construct substantially more efficient advanced tools thanks to our new one-shot proof techniques.

As the final objective of the thesis, we focused on a very practical problem of important real-life impact. That is, we targeted designing an *efficient*, *scalable* and *post-quantum* RingCT protocol. This problem is likely to be of significant interest for future privacy-oriented cryptocurrencies as the threat of quantum computers against the currently deployed cryptographic algorithms grows bigger. Towards addressing the final objective, further improvements over the developed techniques were introduced. In particular, we designed efficient ring signature, group signature and extractable commitment scheme, which may be of independent interest for other privacy-preserving protocols such as secure e-voting. The full practical embodiment of our novel techniques is MatRiCT, the first practical RingCT protocol from standard lattice assumptions.

7.1 Future Research Directions

7.1.1 More theoretical directions

An interesting open question for lattice-based ZKPs in general is finding a *large* set of challenges with desired properties. If we recall from Chapter 4, some desired properties are as follows: 1) challenges should have small norm, 2) challenge differences should be invertible, 3) challenges difference inverses should have small norm. The set of monomial challenges is such a set with these features, but the total set size is very small. Finding a larger set with similar properties would be a very interesting result.

Another open question is related to our CRT-packing technique from Chapter 5. Our technique allows to get faster proof, but does not help with the proof length. It

would be significantly helpful in reducing the proof length if one can come up with a similar CRT-packing technique without requiring the challenges to be of small degree. Another related question is to do with making the CRT-packing technique work in fully splitting rings when proving a binary relation.

If we recall from Section 6.9.1, the message extraction algorithm for our extractable commitment scheme works for relatively small message sets (in particular, sets that can be exhaustively iterated over in practice). A very interesting study would be to investigate whether message extraction could be made possible in other settings where the set of possible messages is not very small, but possibly has some special properties.

A somewhat orthogonal direction of research is to see whether our proofs can be strengthened at a low cost. That is, for our purposes in the thesis, proofs of *relaxed* relations were sufficient, but for some scenarios such as verifiable encryption, this may not be the case. Even the recent exact proofs, e.g., [BLS19, YAZ+19], which indeed make use of the new techniques introduced in the thesis, do not yet seem to offer a satisfactory solution in practice. But we believe that future research on this topic will offer more practically relevant solutions.

Although named "more theoretical directions", the answers to these questions would surely have implications for various applications as well. Therefore, they should not be seen as questions of pure theoretical interest.

7.1.2 More application-oriented directions

A clear future research direction is the investigation of the new techniques' application to other zero-knowledge proofs of practical interest. For example, the techniques may be helpful in extending advanced DL-based ZKPs such as Bulletproofs [BBB+18] to the lattice setting. More generally, a ZKP of arithmetic circuit satisfiability is an important tool for various applications, and for such proofs, there does not seem to be a satisfactory lattice-based solution. Therefore, making use of the new techniques in this setting seems to be an interesting research direction.

An important question to study is the extension of MatRiCT. First, MatRiCT in its current form does not support recipient anonymity. Incorporation of this feature into both the formal foundations and MatRiCT is one of our future works. Another property to be formally studied is the auditability feature. We also studied auditability in a simpler setting where the relaxation factor of the underlying proof was assumed to be 1. As mentioned in Chapter 6, in order to allow auditability from other relaxation factors in general, we believe that a potential path to pursue is the extension of the results of [LN17].

Another applied research direction is to build on our privacy-preserving credentials proposal from Section 5.6.2. Our proposal can be seen as an initial step forward in constructing *efficient post-quantum* anonymous credentials. But there are certainly interesting questions to be answered such as providing unlinkability between credential showings (i.e., two showings of the same credential cannot be linked to each other).

There are further various applications where our techniques and results may prove useful. Some examples are group signatures, secure e-voting systems, other blockchain protocols (than RingCT) and remote attestation protocols (such as Direct Anonymous Attestation, namely DAA).

Bibliography

[ABB+19] Erdem Alkim, Paulo S. L. M. Barreto, Nina Bindel, Patrick Longa, and Jefferson E. Ricardini. The lattice-based digital signature scheme qTESLA. Cryptology ePrint Archive, Report 2019/085, 2019. https://eprint.iacr.org/2019/085.

[ACJT00] Giuseppe Ateniese, Jan Camenisch, Marc Joye, and Gene Tsudik. A practical and provably secure coalition-resistant group signature scheme. In *CRYPTO*, pages 255–270. Springer, 2000.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange-a new hope. In *USENIX Security Symposium*, 2016.

[AG11] Sanjeev Arora and Rong Ge. New algorithms for learning in presence of errors. In *ICALP*, pages 403–415. Springer, 2011.

[AGHS13] Shweta Agrawal, Craig Gentry, Shai Halevi, and Amit Sahai. Discrete Gaussian leftover hash lemma over infinite domains. In *ASIACRYPT*, pages 97–116. Springer, 2013.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108. ACM, 1996.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[ARS+15] Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In *EUROCRYPT*, volume 9056 of *LNCS*, pages 430–454. Springer, 2015.

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *IEEE Symposium on Security and Privacy*, pages 315–334. IEEE, 2018.

[BBC+18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In *CRYPTO*, volume 10992 of *LNCS*, pages 669–699. Springer, 2018.

[BCC+15] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In *ESORICS*, pages 243–265. Springer, 2015.

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic

circuits in the discrete log setting. In *EUROCRYPT*, pages 327–357. Springer, 2016.

[BCK+14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *ASIACRYPT*, pages 551–572. Springer, 2014.

[BDL+18] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In *SCN*, pages 368–385. Springer, 2018.

[BKLP15] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS*, pages 305–325. Springer, 2015.

[BKM09] Adam Bender, Jonathan Katz, and Ruggero Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *Journal of Cryptology*, 22(1):114–138, 2009.

[BLO18] Carsten Baum, Huang Lin, and Sabine Oechsner. Towards practical lattice-based one-time linkable ring signatures. In *ICICS*, volume 11149 of *LNCS*, pages 303–322. Springer, 2018.

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 176–202. Springer, 2019.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In *EUROCRYPT*, volume 2656, pages 614–629. Springer, 2003.

[BPVY00] Ernest Brickell, David Pointcheval, Serge Vaudenay, and Moti Yung. Design validations for discrete logarithm based signature schemes. In *PKC*, pages 276–292. Springer, 2000.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM conference on Computer and communications security*, pages 62–73. ACM, 1993.

[CDG+19] Melissa Chase, David Derler, Steven Goldfeder, Jonathan Katz, Vladimir Kolesnikov, Claudio Orlandi, Sebastian Ramacher, Christian Rechberger, Daniel Slamanig, Xiao Wang, and Greg Zaverucha. The picnic signature scheme, 2019. Submission to NIST Post-Quantum Cryptography project.

[Cha85] David Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[Chu18] Chitchanok Chuengsatiansup. Private communication, 2018.

[CvH91] David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265. Springer, 1991.

[DFMS19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. Security of the fiat-shamir transformation in the quantum random-oracle model. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pages 356–383. Springer, 2019.

[DLL+18] Léo Ducas, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals–Dilithium: Digital signatures from module lattices. In *CHES*, volume 2018-1, 2018.

[dPLNS17] Rafaël del Pino, Vadim Lyubashevsky, Gregory Neven, and Gregor Seiler. Practical quantum-safe voting from lattices. In *CCS*, pages 1565–1581. ACM, 2017.

[dPLS18] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *ACM CCS*, pages 574–591. ACM, 2018.

[DRS18] David Derler, Sebastian Ramacher, and Daniel Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In *PQCrypto*, pages 419–440. Springer, 2018. (Extended version at https://eprint.iacr.org/2017/1154).

[dSGDOS19] Cyprien Delpech de Saint Guilhem, Lauren De Meyer, Emmanuela Orsini, and Nigel P. Smart. BBQ: using AES in picnic signatures. In *SAC*, volume 11959 of *Lecture Notes in Computer Science*, pages 669–692. Springer, 2019.

[ESLL19] Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 115–146. Springer, 2019. (Full version at https://eprint.iacr.org/2019/445).

[ESS+19] Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *ACNS*, volume 11464 of *LNCS*, pages 67–88. Springer, 2019. (Full version at https://eprint.iacr.org/2018/773).

[EZS+19] Muhammed F. Esgin, Raymond K. Zhao, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. MatRiCT: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, pages 567–584. ACM, 2019. (Full version at https://eprint.iacr.org/2019/1287).

[FG15] Jason Fulman and Larry Goldstein. Stein's method and the rank distribution of random matrices over finite fields. *The Annals of Probability*, 43(3):1274–1314, 2015.

[FHK+18] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor

Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over NTRU, 2018. https://falcon-sign.info/falcon.pdf.

[Fou18] Abelian Foundation. Abelian Coin (ABE) – a quantum-resistant cryptocurrency balancing privacy and accountability, 2018. https://www.abelianfoundation.org/wp-content/uploads/2018/08/Abelian-Whitepaper-CB20180615.pdf (June 15, 2018 version).

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194. Springer, 1986.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, volume 7237 of *LNCS*, pages 465–482. Springer, 2012.

[GK15] Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT*, volume 9057, pages 253–280. Springer, 2015.

[GKV10] S. Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. A group signature scheme from lattice assumptions. In *ASIACRYPT*, volume 6477, pages 395–412. Springer, 2010.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18(1):186–208, 1989. Preliminary version in STOC 1985.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. ACM, 2008.

[GT15] Torbjrn Granlund and Gmp Development Team. *GNU MP 6.0 Multiple Precision Arithmetic Library*. Samurai Media Limited, United Kingdom, 2015.

[Gue09] Shay Gueron. Intel's new AES instructions for enhanced performance and security. In *FSE*, LNCS, pages 51–66. Springer, 2009.

[HJ12] Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *ACM CCS*, pages 525–537. ACM, 2018.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 372–389. Springer, 2008.

[LAZ19] Xingye Lu, Man Ho Au, and Zhenfei Zhang. Raptor: A practical lattice-based (linkable) ring signature. In *ACNS*, volume 11464 of *Lecture Notes in Computer Science*, pages 110–130. Springer, 2019.

[LLLS13] Fabien Laguillaumie, Adeline Langlois, Benoît Libert, and Damien Stehlé. Lattice-based group signatures with logarithmic signature size. In *ASIACRYPT*, pages 41–61. Springer, 2013.

[LLM+16] Benoît Libert, San Ling, Fabrice Mouhartem, Khoa Nguyen, and Huaxiong Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT*, pages 373–403. Springer, 2016.

[LLNW14] Adeline Langlois, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-based group signature scheme with verifier-local revocation. In *Public Key Cryptography (PKC)*, volume 8383, pages 345–361, 2014.

[LLNW16] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT*, pages 1–31. Springer, 2016.

[LN17] Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT*, pages 293–323. Springer, 2017.

[LNW15] San Ling, Khoa Nguyen, and Huaxiong Wang. Group signatures from lattices: simpler, tighter, shorter, ring-based. In *Public Key Cryptography (PKC)*, pages 427–449. Springer, 2015.

[LNWX17] San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. Lattice-based group signatures: Achieving full dynamicity with ease. In *ACNS*, pages 293–312. Springer, 2017.

[LRR+19] Russell W. F. Lai, Viktoria Ronge, Tim Ruffing, Dominique Schröder, Sri Aravinda Krishnan Thyagarajan, and Jiafan Wang. Omniring: Scaling private payments without trusted setup. In *ACM CCS*, pages 31–48. ACM, 2019.

[LS15] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.

[LS18] Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *EUROCRYPT*, pages 204–224. Springer, 2018.

[LWW04] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In *ACISP*, volume 3108 of *LNCS*, pages 325–335. Springer, 2004.

[Lyu09] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616. Springer, 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755. Springer, 2012. (Full version).

[LZ19] Qipeng Liu and Mark Zhandry. Revisiting post-quantum fiat-shamir. In *CRYPTO (2)*, volume 11693 of *Lecture Notes in Computer Science*, pages 326–355. Springer, 2019.

[MBB+13] Carlos Aguilar Melchor, Slim Bettaieb, Xavier Boyen, Laurent Fousse, and Philippe Gaborit. Adapting Lyubashevsky's signature schemes to the ring signature setting. In *AFRICACRYPT*, pages 1–25, 2013.

[MM11] Daniele Micciancio and Petros Mol. Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In *CRYPTO*, LNCS, pages 465–484. Springer, 2011. (Full version).

[MP12] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.

[MP13] Daniele Micciancio and Chris Peikert. Hardness of SIS and LWE with small parameters. In *CRYPTO*, pages 21–39. Springer, 2013.

[MR07] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007. Preliminary version in FOCS 2004.

[MR09] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[NIS] NIST. SHA-3 standard: Permutation-based hash and extendable-output functions. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf. Accessed: 2019-05-15.

[NIS17] NIST. Post-quantum cryptography – call for proposals, 2017. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization/Call-for-Proposals.

[Noe15] Shen Noether. Ring signature confidential transactions for monero. Cryptology ePrint Archive, Report 2015/1098, 2015. https://eprint.iacr.org/2015/1098.

[NZZ15] Phong Q. Nguyen, Jiang Zhang, and Zhenfeng Zhang. Simpler efficient group signatures from lattices. In *Public Key Cryptography (PKC)*, pages 401–426. Springer, 2015.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009. Preliminary version in STOC 2005.

[RGCB19] Prasanna Ravi, Sourav Sen Gupta, Anupam Chattopadhyay, and Shivam Bhasin. Improving speed of Dilithium's signing procedure. Cryptology ePrint Archive, Report 2019/420, 2019. https://eprint.iacr.org/2019/420.

[RST01] Ronald Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. *ASIACRYPT*, pages 552–565, 2001.

[SALY17] Shifeng Sun, Man Ho Au, Joseph K. Liu, and Tsz Hon Yuen. RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In *ESORICS*, volume 10493 of *LNCS*, pages 456–474. Springer, 2017.

[Sco17] Michael Scott. A note on the implementation of the number theoretic transform. In *IMACC*, LNCS, pages 247–258. Springer, 2017.

[Sei18] Gregor Seiler. Faster AVX2 optimized NTT multiplication for Ring-LWE lattice cryptography. Cryptology ePrint Archive, Report 2018/039, 2018. https://eprint.iacr.org/2018/039.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT*, pages 617–635. Springer, 2009.

[Ste96] Jacques Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

[SV14] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic SIMD operations. *Des. Codes Cryptography*, 71(1):57–81, 2014.

[Tea19] Zcash Team. Frequently asked questions, 2019. https://z.cash/support/faq/#quantum-computers, accessed on April 23, 2019.

[TKS+19] Wilson Abel Alberto Torres, Veronika Kuchta, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Jacob Cheng. Lattice RingCT v2.0 with multiple input and multiple output wallets. In *ACISP*, volume 11547 of *LNCS*, pages 156–175. Springer, 2019. Full version at https://eprint.iacr.org/2019/569.

[TSS+18] Wilson Abel Alberto Torres, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, Veronika Kuchta, Nandita Bhattacharjee, Man Ho Au, and Jacob Cheng. Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice ringct v1. 0). In *ACISP*, pages 558–576. Springer, 2018.

[Tur66] L. Richard Turner. Inverse of the vandermonde matrix with applications. Technical Report NASA-TN-D-3547, Lewis Research Center, NASA, 1966.

[YAL+17] Rupeng Yang, Man Ho Au, Junzuo Lai, Qiuliang Xu, and Zuoxia Yu. Lattice-based techniques for accountable anonymity: Composition of abstract stern's protocols and weak prf with efficient protocols from lwr. Cryptology ePrint Archive, Report 2017/781, 2017. https://eprint.iacr.org/2017/781.

[YAZ+19] Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO (1)*, volume 11692 of *LNCS*, pages 147–175. Springer, 2019.

[YSL+19] Tsz Hon Yuen, Shi-feng Sun, Joseph K. Liu, Man Ho Au, Muhammed F. Esgin, Qingzhao Zhang, and Dawu Gu. RingCT 3.0 for blockchain confidential transaction: Shorter size and stronger security. Cryptology ePrint Archive, Report 2019, 2019. (To appear at Financial Cryptography and Data Security 2020).

[ZSS19] Raymond K. Zhao, Ron Steinfeld, and Amin Sakzad. FACCT: Fast, compact, and constant-time discrete Gaussian sampler over integers. *IEEE Transactions on Computers*, 2019.

[ZZTA18] Huang Zhang, Fangguo Zhang, Haibo Tian, and Man Ho Au. Anonymous post-quantum cryptocash. In *Financial Cryptography*, volume 10957 of *LNCS*, pages 461–479. Springer, 2018. https://eprint.iacr.org/2017/716.