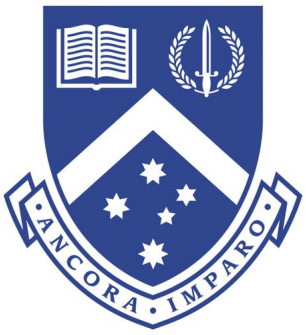


Post-Quantum Cryptographic Primitives

by

Wilson Abel Alberto Torres

A thesis submitted in total fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Information Technology
MONASH UNIVERSITY

September 2020

MONASH UNIVERSITY

Abstract

Faculty of Information Technology

Doctor of Philosophy

by Wilson Abel Alberto Torres

The necessity of cybersecurity is becoming apparent as we grow increasingly dependent on technology in our daily lives. cybersecurity allows us to protect our digital information that is stored and transferred via the Internet. Because of this growing dependence, this research project endeavours to design, construct and evaluate new post-quantum cryptographic primitives that not only secure such information against cyber attacks, but also protect it from the probable vulnerabilities that will emerge as a result of quantum computer technology.

To begin with, this thesis focuses on different variants of digital signatures such as ring, linkable-ring and threshold signatures schemes. These new constructions are then applied in a cryptocurrency protocol where further cryptographic techniques such as homomorphic commitments, zero knowledge proofs and range proofs are utilised to guarantee a high level of security. The results of the security evaluation of these constructions and protocols show that they are secure in terms of unforgeability, anonymity, linkability and non-slanderability.

A novel post-quantum Linkable Ring Signature scheme is first proposed, which enables the public to verify if two or more signatures were generated by the same signer, whilst still protecting the anonymity of this signer. The scheme achieves unconditional anonymity and security guarantees with the lattice-based hardness assumptions. The proposed scheme is generalised to be applied in a post-quantum cryptocurrency protocol based on the Ring Confidential transaction RingCT, which forms the foundation of the privacy-preserving protocol in any post-quantum secure cryptocurrency such as Hcash.

In further research, this thesis presents an innovated Lattice-based Ring Confidential Transactions supporting Multiple-Input and Multiple-Output wallets. This

construction is a fully functional underlying scheme for cryptocurrency applications. We extend the quantum resistant linkable ring signature scheme to support these multiple transactions. The security model provides stronger security for balance and anonymity properties, including the use of a lattice-based range proof.

In the end, another new post-quantum cryptographic mechanism is introduced, named the Lattice-based Linkable Ring Signature with Co-Signing, which offers a distributed authorisation feature to protect electronic wallets. It also covers a formal definition of a security model for this authorisation scheme, so accomplishes the security requirements to protect any privacy preserving applications like the blockchain cryptocurrency protocols (including the RingCT). To address key-generation security concerns, and to support compression of keys and signatures, this proposal incorporates a distributed key generation along with a solid public-key aggregation. Finally, we prove the security of this construction in the random oracle model and the standard lattice-based hardness assumption.

Declaration of Authorship

I, Wilson Abel Alberto Torres, declare that this thesis titled, "Post-Quantum Cryptographic Primitives" and the work presented in it are my own. I confirm that:

- This thesis comprises only my original work towards the Doctor of Philosophy award except where indicated;
- This thesis contains no material which has been accepted for the award of any other degree at any university or equivalent educational institution; and
- This thesis contains no material previously published or written by another person, except where due reference is made in the text of this thesis.

Print Name: **Wilson Abel Alberto Torres**

Date: **29 of September, 2020**

Publications During Enrolment

- Alberto Torres W.A., Steinfeld R., Sakzad A., Liu J.K., Kuchta V., Bhattacharjee, N., Au M.H., Cheng, J. (2018) *Post-Quantum One-Time Linkable Ring Signature and Application to Ring Confidential Transactions in Blockchain (Lattice RingCT v1.0).* In: Susilo W., Yang G. (eds) Information Security and Privacy. ACISP 2018. Lecture Notes in Computer Science, vol 10946. Springer, Cham.

- Alberto Torres W., Kuchta V., Steinfeld R., Sakzad A., Liu J.K., Cheng J. (2019) *Lattice RingCT V2.0 with Multiple Input and Multiple Output Wallets.* In: Jang-Jaccard J., Guo F. (eds) Information Security and Privacy. ACISP 2019. Lecture Notes in Computer Science, vol 11547. Springer, Cham.

- Alberto Torres W.A., Steinfeld R., Sakzad A., Kuchta V. (2020) *Post-Quantum Linkable Ring Signature Enabling Distributed Authorised Ring Confidential Transactions in Blockchain.* https://eprint.iacr.org/2020/1121.

Acknowledgements

I would like to thank my supervisors, Dr Ron Steinfeld and Dr Amin Sakzad, for their technical support, guidance and teaching experiences, particularly for mathematical concepts and cryptography. Similarly, I would like to thank all the Hcash team (Dr Joseph Liu and Dr Veronika Kuchta) to allow me to be part of this research project.

I would like to express my gratitude to Monash University and the Australian Government Research Training Program (RTP) since I was benefited with a full scholarship to undertake this PhD study.

I owe my deepest gratitude to all my friends within the IT faculty and other faculties who were there to share multiple and memorable experiences during the last three and a half years.

In the end, none of this would have been possible without the unconditional love and support of my parents, siblings, and wife. To them and the rest of the family, many thanks for offering their support from a distance.

Contents

List of Figures

List of Tables

Abbreviations

PKI	Public Key Infrastructure
RS	Ring Signature
LRS	Linkable Ring Signature
RingCT	Ring Confidential Transaction
SIS	Short Integer Solution
BLISS	Bimodal Lattice Signature Scheme
L2RS	Lattice-based one-time Linkable Ring Signature
SISO	Single-Input to Single-Output
MIMO	Multiple-Input to Multiple-Output
LRCT	Lattice Ring Confidential transaction
L2RS-CS	Lattice-based Linkable Ring Signature with Co-Signinge
DKG	Distributed Key Generation
TS	Threshold Signature
TRS	Threshold Ring Signatures
IFP	Integer Factorization Problem
DLP	Discrete Logarithm Problem
PPT	Probabilistic Polynomial Time
ROM	Random Oracle Model
IW	Input Wallet
OW	Output Wallet
PoK	Proof of Knowledge
LHL	Leftover Hash Lemma

Chapter 1

Introduction

Cryptology, the science of secret writing, has existed for many years. Different organisations including governments, businesses and the military use cryptology to keep their communication secure against adversaries. Cryptology is divided in two areas: *cryptography* which is concerned with the design of techniques to encrypt information, and *cryptanalysis* which is related to the mechanisms used to break such cryptographic techniques. Public-key (or *asymmetric*) cryptography has become increasingly relevant to modern society, mainly in terms of securing digital communications like the Internet. The main uses of this cryptographic system are *Encryption/Decryption*, *Digital Signatures* and *Key Exchange*.

Digital Signature schemes, are analogous to real handwriting signatures, as they have the same common goal of certifying the content of documents. However, digital signatures are capable of providing stronger security guarantees in terms

of authenticity, unforgeability, and non-repudiation [KL14, HPSS08]. These advantages make digital signatures practical and widely accepted in today's digital society. Digital signatures can be used, for instance, to certify documents, to authenticate individuals and enterprises, or to be employed as a component of other cryptographic protocols. They can also provide the foundation for Public Key Infrastructure (PKI), where digital signatures are used in public-key certificates and are implemented under the standard "$X.509$" [MMB17]. Certificates based on this standard are developed in current network security applications including Internet Protocol Security (IPSec), Transport Layer Security (TLS), Secure Shell (SSH) and Secure/Multipurpose Internet Mail Extension (S/MIME) [MMB17]. In a higher abstraction, these tools are employed to protect Internet communication, Virtual Private Networks (VPN), intranets, software updates, the Internet of Things (IoT) and sensitive data in different industries like healthcare and finance.

Digital Signature schemes have also represented the starting point of the development of new cryptographic primitives. One of these primitives is the ring signature scheme that was initially formalized in [RST01], where the authors defined it as having members of a group who do not want to cooperate. This means that signers in this scheme will no longer have a manager who can eventually reveal their identity, and thus the anonymity is unconditionally preserved. This approach was considered a security improvement when compared with group signature schemes [CVH91], where a group manager was part of the construction. A property called *"Linkability"* was afterwards introduced to the ring signature scheme [LWW04b], which is known as a Linkable Ring Signature (LRS) scheme. This new feature enables the public verifier to check whether two or more signatures are produced by same signer whilst preserving his anonymity property. These types of signatures enable practical privacy preserving applications, including cryptocurrencies, that can anonymously prevent double spending.

Cryptocurrencies are applications that use virtual assets and cryptographic mechanisms to conduct e-commerce operations such as electronic payments or money transfers. Those payments can be carried out among accounts or wallets, independently of a central party [CELR18]. This leads to some advantages like lower

transaction fees, theft resistance and anonymous payments. The Ring Confidential Transaction (RingCT) [Noe15] is a cryptographic protocol that is employed by Monero, which is one of the most popular cryptocurrencies to date. The RingCT performs e-commerce operations in a decentralised network while maintaining *complete-anonymity* of the parties involved in the transactions [CELR18]. In the RingCT framework, *complete-anonymity* provides a remarkable advantage since other cryptocurrencies, such as Bitcoin, are only *pseudo-anonymous* [KKM14].

These schemes, nevertheless, are based on the integer factorization and discrete logarithm assumptions, represented by Rivest, Shamir and Adleman (RSA) [RSA78] and Elgamal [ElG84, Elg85] asymmetric public-key cryptosystems, respectively. Conventional cryptographic schemes such as RSA, DSA and ECC, whose security guarantees are based on mathematical assumptions like discrete logarithm, factoring large numbers and/or number theory, are believed to be vulnerable with the onset of powerful quantum computers [Sho99]. This perceived vulnerability has motivated researchers in the area of post-quantum cryptography to construct approaches against these computers. Among the alternatives, lattice-based cryptography has attracted the attention of this field due to its distinguishing features and new applications. Algorithms based on this new lattice-based primitive are efficient, simple and highly parallelizable and provide strong, provable security guarantees under the worst-case hardness assumptions [MR09, CCJ+16].

Therefore, this thesis aims to design and evaluate new, compact and efficient post-quantum cryptographic schemes that seek to overcome the presence of powerful quantum computers. This research also focuses on the construction of linkable and threshold ring signature schemes due to their capabilities in relevant applied research cryptography, such as cryptocurrencies.

1.1 Research Methodology

1.1.1 Research Questions

1. *Main question*: How to design and evaluate a post-quantum *Linkable Ring Signature* scheme using lattice-based mathematical assumptions (such as Ring-SIS or Module-SIS).

2. *Sub-questions*:

   - Are the security assurances (*unforgeability*, *anonymity*, *linkability* and *non-slanderability*) of this scheme guaranteed?
   - Does this scheme provide better performance results compared to other *Linkable Ring Signature* schemes?
   - Can this *Linkable Ring Signature* scheme be extended to be securely used in applications like cryptocurrencies?
   - Can we design and incorporate particular functionalities in the proposed *Linkable Ring Signature* scheme?

1.1.2 Research Objectives

The aims of this thesis are detailed as follows:

1. *Design* a post-quantum *Linkable Ring Signature* scheme that uses lattice-based cryptographic assumptions (Ring-SIS and Module-SIS) in order to provide better security and performance assurances.

2. *Analyse* the security of the proposed schemes (in terms of *unforgeability*, *anonymity*, *linkability* and *non-slanderability*) in order to guarantee reliability.

3. *Evaluate* the performance of the proposed cryptographic constructions and *optimise* them.

4. *Construct* a cryptocurrency protocol based on the *Extended Linkable Ring Signature* scheme, and *analyse* its security properties (in terms of *balance*, *anonymity*, and *non-slanderability*) and performance.

5. *Design* new *authorisation* functionality that can be incorporated into both the previously devised *Linkable Ring Signature* and the *cryptocurrency* protocol.

1.1.3 Research Contributions

The main *contribution* of this thesis is the construction of practical post-quantum cryptographic primitives using lattice-based cryptography that combine desirable security, efficiency and functionality properties. The list of contributions of this thesis is summarised as follows:

1. Design and construction of the first post-quantum linkable ring signature, known as Lattice-based one-time Linkable Ring Signature (L2RS) along with the security evaluation (in terms of *unforgeability*, *anonymity*, *linkability* and *non-slanderability*) and performance analysis. This construction is a generalisation of the BLISS [DDLL13] scheme which is currently one of the practical and secure lattice-based digital signatures. The L2RS provides unconditional anonymity as well as security guarantees under the hardness of the Ring Short Integer Solution (Ring-SIS) standard lattice assumption. We also devised and constructed a new cryptocurrency privacy-preserving protocol, called Lattice Ring Confidential transaction (LRCT). This protocol employs the proposed post-quantum L2RS as a fundamental building block along with a homomorphic commitment primitive to form the foundation of any privacy-preserving protocols for blockchain cryptocurrencies, such as Hcash. The first version of these novel schemes supports transactions from Single-Input to Single-Output (SISO) wallets. These two contributions were published in [ATSS+18].

2. Construction of the *Multiple-Input Multiple-Output Lattice RingCT v2.0* (MIMO.LRCT) cryptocurrency protocol which was extended from the previous L2RS and SISO.LRCT schemes [ATSS+18]. This proposed construction supports multiple-input and multiple-output wallets in crytocurrency transactions. The MIMO.LRCT inherits the post-quantum security guarantees from SISO.LRCT, which are the hardness of lattice mathematical assumption (the Ring-SIS), and the unconditional anonymity. This newer version enhances the LRCT's security model, particularly the anonymity and balance properties. In the case of anonymity, it includes analyses of both user and amount privacy, contrary to another similar work [SALY17] which only considered user anonymity. The balance security property now includes the out-of-range attacks [BBB+18] and the security proofs which were overlooked by previous RingCT's proposal [ATSS+18, SALY17]. The security analysis illustrates how to incorporate a lattice-based range proof in the MIMO.LRCT protocol, which was a missing component in former proposals [ATSS+18, SALY17]. To begin with, this protocol deals with the difficulties stemming from the imperfection of lattice-based zero-knowledge proofs. To be more specific, the range proofs follow the approach based on 1-of-2 OR-proofs, but our analysis shows that directly applying lattice-based OR-proofs from [dPLNS17] does not provide soundness for the range proof. This argument leads us to carefully select the challenge space, as we describe in Chapter 5. Although these challenges are smaller -in norm- than the ones used in the OR-proofs, they are still larger than the challenges in [LLNW18]. In this framework, we achieve a lower soundness error than the previous lattice-based range proof as in [LLNW18]. Moreover, a thorough concrete performance analysis of the MIMO.LRCT protocol is provided by including this range proof analysis. Concrete bounds are applied to derive preliminary scheme parameters for regular RingCT transactions that support operations of 64-bit amounts along with fewer Multiple Input and Output wallets. The work of this contribution was published in [ATKS+19].

3. Construction of the first post-quantum *Multiple-Input Multiple-Output*

Lattice-based Linkable Ring Signature with *Co-Signing* (MIMO.L2RS-CS) scheme, which can be adapted to a *post-quantum* cryptocurrency protocol such as the LRCT [ATKS+19]. The L2RS-CS offers *complete-anonymity*, and can support MIMO feature for transactions between wallets. The L2RS-CS is built on top of the *post-quantum* LRS from [ATSS+18] and integrates a Distributed Key Generation (DKG) together with a solid public-key aggregation (in the *post-quantum* settings) which bring a high level of security and compression for the cosigners' keys. Additionally, we formalise another new security model, called Linkable Ring Signature with Co-Signing (LRS-CS), having a special combination of two Threshold Signature (TS) constructions, the (N_{CS}-out-of-N_{CS})-TS and (1-out-of-w)-LRS schemes (which are used in Monero [Alo18, GN18]). Although Threshold Ring Signatures (TRS) can be seen as a combination of TS and RS schemes, it is a *different* type of primitive to our proposed LRS-CS. Namely, in TRS any subset of t out of n signers can cooperate to generate a signature while hiding the signers' subset. In contrast, under our LRS-CS model, there are w groups of N_{CS} cosigners, so that all the N_{CS} signing keys within the signing group cooperate to produce the signature while hiding the signers among the w groups. Furthermore, in LRS-CS the N_{CS} cosigners interactively generate and share a *single* public-key, whereas in TRS each cosigner has an individual public-key generated with a non-interactive key-generation algorithm. Therefore, LRS-CS can be viewed as a more specialised primitive than the TRS; however, one that suffices for RingCT authorisation and can also be implemented with much shorter signatures than existing lattice-based TRS schemes, as we demonstrate in the performance evaluation of our L2RS-CS scheme. The security of the L2RS-CS scheme is proven in the classical random oracle model where the properties of *unforgeability*, *linkability* and *non-slanderability* are demonstrated to be computationally secure from the standard lattice-based Module-SIS hardness assumption. In terms of *anonymity*, we show that this construction is unconditionally secure under the Leftover Hash Lemma (LHL) [DDLL13].

Table 1.1 provides a general description of the contributions of this thesis along with its corresponding research objectives.

TABLE 1.1: Thesis Contributions

Contribution	Description	Objectives
1	Construction of the L2RS with security and performance evaluation	1, 2 and 3
2	Construction of the cryptocurrency SISO.LRCT as an application of the L2RS	4
3	Construction of the cryptocurrency MIMO.LRCT with security and performance analyses	2, 3 and 4
4	Construction of the MIMO.L2RR-CS with security and performance analyses	2, 3, and 5

1.2 Outline of the thesis

The organisation of the remaining parts of this thesis are summarised as follows:

- Chapter 2 presents the literature review associated with the thesis' objectives. It reviews concepts of public key cryptography, post-quantum cryptography and their applications to cryptocurrencies.
- Chapter 3 describes general definitions and notations that are employed throughout this thesis.
- Chapter 4 illustrates the first cryptographic construction, the Lattice-based Linkable Ring Signature (L2RS) with its corresponding security and performance analyses. It also extends this initial proposal to a cryptocurrency protocol that is called Lattice-based Ring Confidential Transaction (LRCT).
- Chapter 5 extends the initial L2RS and LRCT constructions. They now support transfers from multiple input wallets to multiple output wallets. It provides a comprehensive security analysis for a cryptocurrency protocol that supports operations of 64-bits amounts.

- Chapter 6 shows how we modified the L2RS to incorporate an authorisation functionality which uses threshold signatures. Under this functionality, several co-signers interact to generate a signature.

- Chapter 7 provides the conclusion of this thesis and possibilities of future research.

Chapter 2

Literature Review

Cryptography is the area of computer science which is responsible for protecting private information and communications. We can easily observe how it is used in many aspects of our daily lives. For instance, when we log onto our personal computers or email accounts, our passwords are protected by one-way hash functions. Moreover, multiple applications exploit the public-key cryptography to devise and implement several protocols, like digital certificates to protect our online financial transactions. Another example of applied cryptography includes blockchain and cryptocurrencies where the user's privacy is preserved during a transaction. On the other hand, classical computers were invented more than 70 years ago. Since then, many improvements have been proposed to speed up their hardware resources. Different technologies of advanced computation have been investigated to achieve such optimisations. Quantum computing will likely be one of those technologies that perform faster computations by far. Unfortunately, classical

cryptography would be vulnerable in the event of powerful quantum computers becoming practical because they would break the hardness of the mathematical assumptions which are used to secure many cryptographic applications. Therefore, this chapter will delve into the significant background of cryptography, certain applications to digital currency, and the threat of quantum attacks.

We start firstly by defining cryptographic digital signature schemes, in Section 2.1. In Section 2.2, several extensions based on digital signatures, that is, the group and ring signature schemes, are discussed. Then, linkable ring signatures are then introduced in Section 2.3. The discussion of the quantum attacks on classical cryptography is demonstrated in Section 2.4, where then lattice-based cryptography (Section 2.5) is presumed to be a promising solution to that threat. In Section 2.6, cryptocurrencies are studied to understand how linkable ring signatures are employed to provide security to such applications. Threshold-signature primitives are reviewed in Section 2.7, which are important in cryptocurrencies since they enable an extra layer security for the transaction of wallets. Finally, the chapter concludes with a summary in Section 2.8.

2.1 Digital Signature – Conventional Approaches

Public-key (or *asymmetric*) cryptography has become increasingly relevant to modern society, mainly in terms of securing digital communications like the Internet. The main uses of this cryptographic system are: *Encryption/Decryption*, *Digital Signatures* and *Key Exchange*. *Digital Signature* schemes, are analogous to real handwriting signatures, as they have the same common goal of certifying the content of documents. However, digital signatures are capable of providing stronger security guarantees in terms of authenticity, unforgeability, non-repudiation, public verifiability, transferability and, perhaps in some cases, anonymity [KL14, HPSS08]. These characteristics make digital signatures practical and widely accepted in today's digital society.

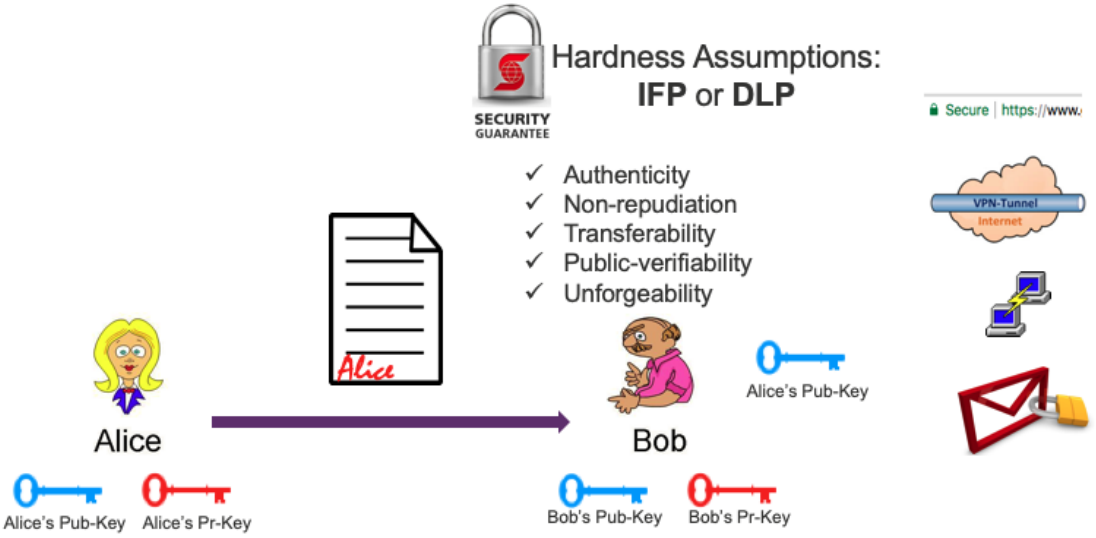
Digital signatures can be used to certify documents, to authenticate individuals and enterprises and as a component of other cryptographic protocols. They can also provide the foundation for public-key cryptography, where these digital signatures are used in public-key certificates and are widely accepted under several international standards, such as the *X.509* [MMB17]. Certificates based on this standard are implemented in current network security applications including Internet Protocol Security (IPSec), Transport Layer Security (TLS), Secure Shell (SSH) and Secure/Multipurpose Internet Mail Extension (S/MIME) [MMB17]. In a higher abstraction, these tools can be used to protect Internet communication, Virtual Private Networks (VPN) and intranets, software updates, the Internet of Things (IoT) and sensitive data in industries like healthcare and finance.

The concept of a *Digital Signature* was proposed (1976) in [DH76], where it was described that every *user* who has a couple of keys, *Private-Key* denoted as "Pr-Key" (a.k.a. secret or signing key) and *Public-Key* "Pub-Key" (a.k.a. verification key), will only publish their "Pub-Key". The digital signature σ is then generated based on the message μ and the *signer*'s "Pr-Key", and ultimately it will be verified by anyone who knows the *signer*'s "Pub-Key". A successful verification signature means that a third party approves the authenticity of the message μ. Figure 2.1 illustrates the concept of digital signatures where *Alice* plays the role of a signer and *Bob* plays the roles of the verifier. The security of such signatures is usually guaranteed on certain mathematical hardness assumptions.

This digital signature idea was later made practical by Rivest, Shamir and Adleman (RSA) in [RSA78], where the security of this RSA cryptographic scheme relied on the hardness of the Integer Factorization Problem (IFP). More precisely, the RSA public key is a product ($n = pq$) of two secret, large prime numbers p and q, and the security relies on finding the factors p and q of n. This RSA signature scheme has three Probabilistic Polynomial Time (PPT) algorithms:

1. Key generation "KeyGen": this algorithm creates the pair of keys, "Pub-Key": (n, e) and "Pr-Key": d. This algorithm depends on the security parameter λ (the number of bits in n), which is the input of this

FIGURE 2.1: Digital Signature Idea



algorithm.

2. Signing "Sign": in this algorithm, the sender will generate a signature σ based on a message μ, $0 \leq \mu \leq n$. It is computed as $\sigma = \mu^d \bmod n$, and (σ, μ) will be sent to the recipient.

3. Verification "Ver": the recipient verifies the message μ's authenticity by checking $\mu \stackrel{?}{=} \sigma^e \bmod n$. This is a deterministic algorithm as it only outputs *accept* or *reject*.

Another well-known digital signature scheme *Elgamal* was proposed in [ElG84, Elg85]. Its security is based on the hardness of the Discrete Logarithm Problem (DLP), which is the difficulty of finding an exponent x such that $y \equiv g^x \bmod p$ where y, q and p are given. Some variants of this scheme were later proposed, like the *Digital Signature Algorithm* (DSA) and the *Digital Signature Standard* (DSS) [KG13]. This scheme also has three main algorithms, which are described as follows:

1. KeyGen: given a security parameter λ, a pair of keys are generated by the sender, the "Pub-Key": (p, g, y) and the "Pr-Key": x, where p is a prime, g is a primitive root modulo p, $x \hookleftarrow U(Z_q*)$, where $U(Z_q*)$ is a uniform distribution over Z_q* and $y = g^x \bmod p$.

2. Sign: a message μ, where $0 \leq \mu \leq p$, is signed by the sender by choosing a random k, where $1 \leq k \leq p-2$ satisfying $\gcd(k, p-1) = 1$. Then, the sender computes $\sigma_1 = g^k \bmod p$ and $\sigma_2 = k^{-1}(\mu - x\sigma_1) \bmod p - 1$. The signature σ of the message μ is (σ_1, σ_2).

3. Ver: any potential verifier checks $0 < \sigma_1 < q$ and $0 < \sigma_2 < q$, then computes $w = {\sigma_2}^{-1} \mod p$, $u_1 = \mu \cdot w \mod p$, $u_2 = \sigma_1 \cdot w \mod p$, and $v = g^{u_1} \cdot (y^{u_1}) \mod p$. The signature σ is authentic iff $v = \sigma_1$.

The security of Digital Signatures is said to be computationally secure, meaning that these schemes cannot be secure against an adversary who has both unlimited time and unlimited computational power [Kat10]. In this situation, the attacker, who has the victim's "Pub-Key", will try all possible values of the signature σ by using the verification algorithm. The signature σ will ultimately be forged when the adversary has found one σ' that satisfies the verification test of the message μ. This level of security also implies that a digital signature scheme cannot even be perfectly secure against a weak adversary: an attacker, even without knowing a victim's "Pub-Key", can randomly forge a signature σ with negligible probability. Nevertheless, having a computationally secure notion will ensure that it is impossible to forge a signature scheme except with a small or negligible probability for any efficient adversary or with a PPT algorithm [Kat10], where the running time is measured based on the security parameter λ. The larger the λ, the more secure the signature scheme. Hence, in digital signatures schemes, the security relies on hardness assumptions, such as the IFP with RSA and DLP with Elgamal. The hardness assumptions of the current and practical digital signatures, which are based on classical cryptography, are briefly described in Table 2.1 below.

The security analysis and requirements of digital signature schemes were formally proposed in [GMR88], providing a definition of attack models (as Table 2.2) as well as how an adversary might be successful in breaking digital signature schemes (as in Table 2.3). In this explanation, the legitimate signer is denoted as $\mathcal{U}$ and the adversary as $\mathcal{A}$. We can say that this analysis provides an upper–layer security

TABLE 2.1: Summary of Digital Signature Schemes

Digital Signature Schemes	**Hardness Assumption**
Plain RSA – (1978) [RSA78]; RSA-FDH – (1993) [BR93, BR96]	IFP
Elgamal – (1984) [ElG84, Elg85]	DLP
DSA – (1991) [KG13]	DLP
Schnorr – (1989) [Sch89, Sch91]	DLP
DSS/ECDSA (Elliptive Curve DS Algorithm) – (2013) [KG13]	DLP

of digital signature schemes where the interaction between $\mathcal{U}$ and $\mathcal{A}$ is evaluated, guaranteeing that signatures are not forged.

TABLE 2.2: Digital Signature Attacks Models

Digital Signature Attacks	**Description**
Key-Only (KO)	$\mathcal{A}$ only knows $\mathcal{U}$'s Pub-Key.
Known Message (KM)	Besides knowing $\mathcal{U}$'s Pub-Key, $\mathcal{A}$ has a collection of $\mathcal{U}$'s message-signatures pairs: (μ_1, σ_1), ..., (μ_t, σ_t).
Generic Chosen Message (GCM)	$\mathcal{A}$ chooses a list of messages $(\mu_1, \ldots \mu_t)$ that is independent of $\mathcal{U}$'s Pub-Key prior to attempt in breaking $\mathcal{U}$'s DS. Under these conditions, $\mathcal{A}$ creates valid $\mathcal{U}$'s DS (μ_1, σ_1), ..., (μ_t, σ_t).
Directed Chosen Message (DCM)	This differs from a GCM attack in that the list of messages is chosen after knowing $\mathcal{U}$'s Pub-Key but before any signatures are seen.
Adaptive Chosen Message (ACM)	$\mathcal{U}$ is also used as an "*Oracle*", where $\mathcal{A}$ uses to sign messages that depend on previously obtained message–signature pairs.

TABLE 2.3: Success at Breaking a Digital Signature Scheme

Digital Signature Forgeries	**Description**
Total break	$\mathcal{A}$ is able to recover $\mathcal{U}$'s Pri-Key.
Universal forgery	$\mathcal{A}$ is able to forge the σ of any μ.
Selective forgery	$\mathcal{A}$ can forge a σ for a particular message chosen by $\mathcal{A}$
Existential forgery	$\mathcal{A}$ can forge a σ of at least one message whose signature was not already seen.

After analyzing the security of digital signatures, it is emphasized that their foundational schemes are computationally secure. In the case of the RSA and Elgamal

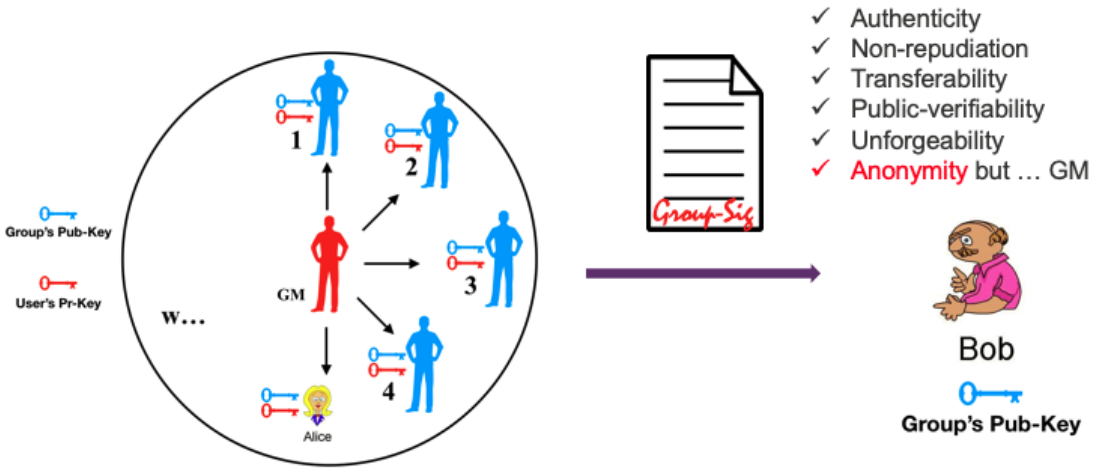
schemes, they will be completely broken if an adversary solves the factoring assumptions and the DLP, respectively. Furthermore, standard digital signatures have served as a reference point to devise new cryptographic schemes such as group, ring, and linkable ring signatures.

2.2 Group and Ring Signatures

A new type of signature, introduced as a proof of knowledge in [CVH91], aims to create a digital signature scheme based on a group of individuals. This construction, is known as the *Group Signature* scheme (see Figure 2.2), has the following properties:

- Individuals who belong to this group can create signatures.
- A group signature can be verified by public users without revealing the identity of the individual who signed the signature.
- In case of disputes, there exists a Group Manager (GM) who is capable of disclosing the identity of the signer, to handle and resolve the situation.

FIGURE 2.2: Group Signature Idea



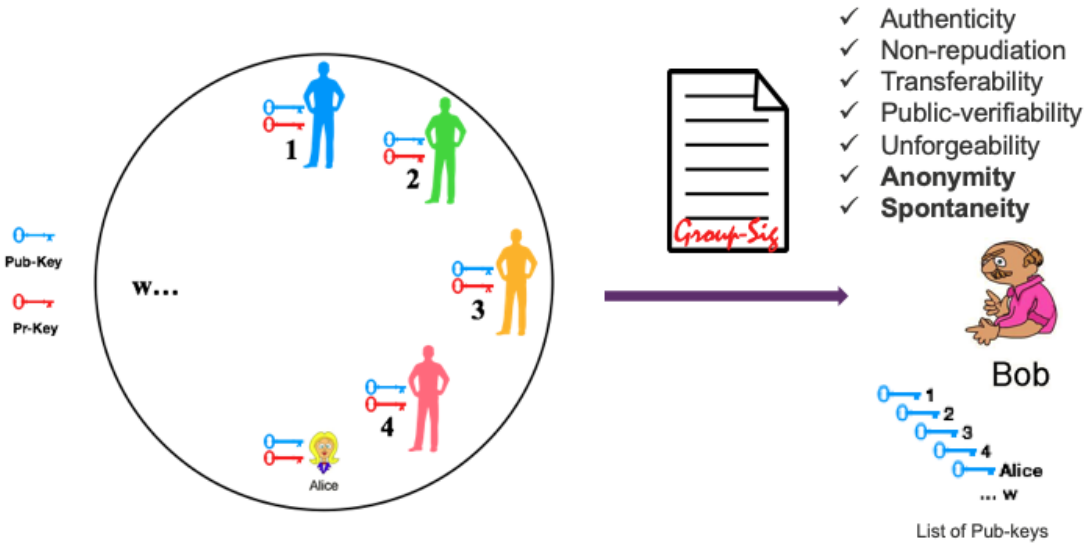
An initial setup is defined to allow the group manager to control the group's membership and generate the signing keys of the group members (w members as in

Figure 2.2) [CVH91]. Beyond the security guarantees of standard digital signature schemes, a group signature scheme provides *anonymity* and *unlinkability* of the group signer. This means that it is computationally difficult to disclose the group signer's identity (except for the group manager) and that it is computationally infeasible to decide whether two signatures were signed by the same signer. This cryptographic scheme has a wide range of applications like electronic voting, electronic auctions and electronic cash systems [CVH91]. However, as the scheme is bundled to the group manager, some issues may arise in efficiency and security [RST01]. For instance, it may require a large number of interactions between the group manager and the group members to manage the group, to create signatures and to solve disputes. Furthermore, if the group manager is targeted by an attacker, the whole scheme might be compromised since the anonymity property could be broken.

The notion of *Ring Signature* schemes was formally proposed in [RST01, RST06] as another cryptographic primitive based on group signatures, where the term *ring* defines an *ad-hoc network* and the possible set of signers. Figure 2.3 provides a brief illustration of this idea. Under the assumption that this scheme must have a standard digital signature scheme like RSA, the main differences in comparison with the standard group signatures are that

- There is no group manager, and therefore, there are no procedures for managing prearranged groups like setting, changing and deleting them. This also implies that the anonymity of the group's signer can be unconditionally or computationally preserved unless the signer reveals its identity.
- There is no key distribution among the members of the group, so groups can be formed spontaneously. This indicates that independent public key schemes with different signatures and key sizes can be used.
- The Group Signature scheme is useful when members of the group want to cooperate to generate the signature, whereas this is not a necessity for members in a *Ring Signature* scheme.

FIGURE 2.3: Ring Signature Idea



The construction of the Ring Signature scheme also has three PPT algorithms: Key Generation (KeyGen), Ring Signature Generation (RingSign) and Ring Signature Verification (RingVer).

1. $(KeyGen)$: given the security parameter λ, the pair of keys (Pub-Key, Pr-Key) are generated by each member of the group or *ring*. The (Pub-Key) of the ring signer can also be obtained from a PKI directory or a certificate.

2. $(RingSign)$: a signer π out of w members generates the signature $\sigma \leftarrow RingSign(\text{Pr-Key}_\pi, \mu, L)$, where Pr-Key_π represents the signer's private-key, the message $\mu \in \{0,1\}^*$ and the list L contains the public keys of the ring signatures' members $L = \{\text{Pub-Key}_1, \text{Pub-Key}_2, \text{Pub-Key}_3, \ldots, \text{Pub-key}_w\}$.

3. $(RingVer)$: this deterministic algorithm verifies (μ, σ) with respect to the list L and outputs: *accept* or *reject*.

Several generalisations of the *Ring Signature* scheme have been developed. These include *Threshold Ring Signature* [BSS02], *Proxy Ring Signature* [ZSNL04], *Conditionally Anonymous Ring Signature* [LLM+07] and *Ring Signcryption* [HZW06]. A primay focus of this research is on a particular extension of this ring signature scheme called *Linkable Ring Signature*, which has been applied in scenarios like cryptocurrency [Noe15] and e-voting [LWW04b].

2.3 Linkable Ring Signature

The linkability property of *Ring Signatures* allows one to detect if two signatures have been generated by the same signer (using the same private key) whilst still preserving their anonymity. The first proposal was introduced in [LWW04b] with the name *Linkable Spontaneous Anonymous Group Signatures* (LSAG), where the scheme was proved to be secure under the discrete logarithm assumptions and the Random Oracle Model (ROM). The idea is represented in Figure 2.4. In comparison with previous unlinkable ring signature schemes, this adds a PPT algorithm to verify the linkability property. In doing so, it uses a label that is attached to each signature σ and consists of hiding the identity of the signer by using a cryptographic hash function modeled as a random oracle. The label is then used by a deterministic linkable algorithm to anonymously check whether two valid signatures have identical label or not. If the label is identical, it means that a signer has created two signatures. This particular feature opens the possibility of many practical scenarios [LWW04b, LAH+14, TW05]. For example:

- **Cryptocurrency**: to prevent double spending while preserving the anonymity of the miner. In fact, this approach is being used in the Monero software application [Noe15].
- **E-voting**: to discard duplicate votes from an anonymous person (double-voting).
- **E-survey**: to detect double submissions of any pseudonymous responder.
- **Ad-hoc networks**: to provide anonymous authentication.
- **Access control**: to anonymously verify how many times a file/directory is accessed either on-premises or in cloud storage.

The security of most of the current *Linkable Ring Signatures* relies on classical cryptographic mathematical assumptions (DLP and IFP). A brief summary of these schemes is given in Table 2.4.

FIGURE 2.4: Linkable Ring Signature Idea

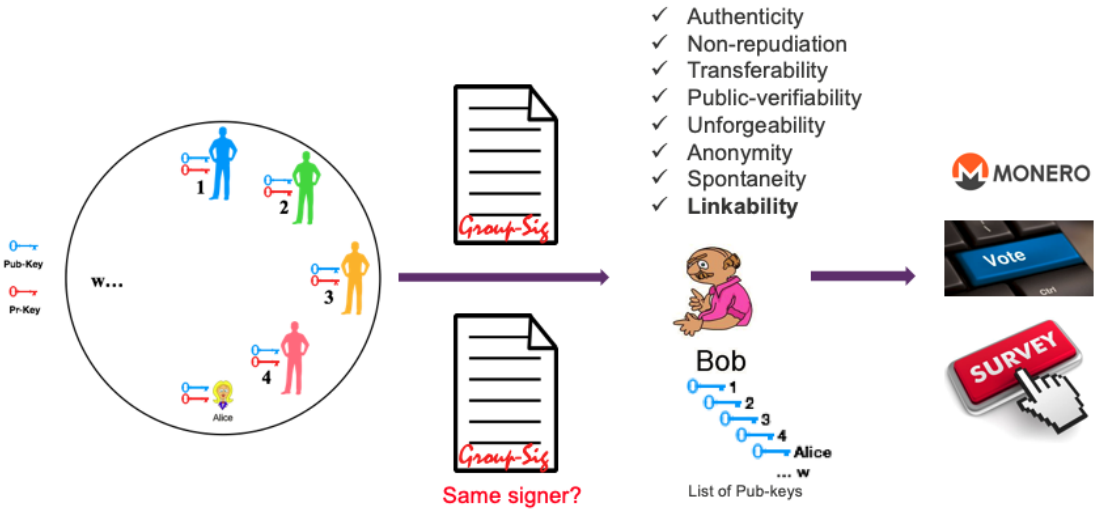


TABLE 2.4: Linkable Ring Signature Schemes – Hardness Assumptions

Linkable Ring Signature Schemes	**Hardness Assumption**
Au *et al.* [ACST06, ALSY07]	IFP
Au *et al.* [ALSY06]	DLP
Fujisaki *et al.* [FS07, Fuj11]	DLP
Liu *et al.* [LSW06, LWLW06, LWW04b, LW05a, LASZ14]	DLP
Tsang *et al.* [TW05, TAL+10]	IFP
Yuen *et al.* [YLA+13]	IFP
Zheng *et al.* [ZLCL07]	DLP

There are still some research directions on these *Linkable Ring Signatures* schemes that were initially discussed in [LWW04b], including

- To construct a scheme that provides unconditional anonymity.
- To construct a scheme using different hardness assumptions.
- To devise shorter and more efficient schemes.
- To apply the linkable ring signature in more scalable applications like e-voting and/or cryptocurrency.

This thesis investigates the design of post-quantum secure linkable ring signatures and their variants, in particular, using lattice-based cryptography, which is believed to have potential in addressing the aforementioned problems. Thus, the

next section will discuss post-quantum cryptography, which could stand against adversarial quantum technology. In addition, it turns out that some of these post-quantum algorithms provide practical performance results as well as new cryptographic applications.

2.4 The Art of Quantum Computing

This literature review has shown how classical cryptography has become relevant for computer, network and Internet security, where public-key algorithms are being used as strong protection against certain cyber-attacks. Those algorithms, which were previously discussed (see Table 2.1 and Table 2.4), rely upon computational problems (*integer factorization and discrete logarithm*) that are conjectured to be intractable or infeasible with any realistic amount of computational resources. The main security idea of these algorithms, sometimes called *one-way functions*, is that they are relatively easy to compute but extremely difficult to reverse [KL14].

However, a new physical paradigm for computing, *Quantum Computing*, is believed to pose a threat to these cryptographic algorithms. In 1994, Shor showed in *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer* [Sho99] how these computers can exploit quantum mechanics to significantly speed up certain computations, efficiently solving those mathematical problems that are associated with the security of classical public-key cryptography. As a result, it was claimed that a powerful quantum computer would utterly jeopardize the security of public-key algorithms.

In 1996, Grover devised a quantum algorithm [GK96] that sped up the search in an unordered database, showing that it also posed a threat to symmetric algorithms [CCJ+16] like Triple Data Encryption Standard (3-DES), Advanced Encryption Standard (AES) and Hash functions (SHA2 and SHA3). Symmetric algorithms can overcome this problem by doubling the key sizes and increasing the output (in the case of hash functions), but this situation will not be the same with classical public-key algorithms, where alternative foundations will be sorely needed.

Although there is not yet nowadays a practical powerful quantum computer in operation, scientists and experts predict that one will be built within 20 years [CCJ+16]. This indicates that all the schemes (including the *Linkable Ring Signatures* scheme; see Table 2.4) based on these mathematical foundations will be unusable with the onset of quantum computers. Table 2.5 illustrates, based on [CCJ+16, BL17a], how a large-scale quantum computer would impact the classical public-key cryptography.

TABLE 2.5: Impact of Quantum Computing on Classical Public-key Cryptography

Digital Signature Schemes	**Hardness Assumption**	**Impact – Shor's Algorithm**
Plain RSA – (1978) [RSA78]; RSA-FDH – (1993) [BR93, BR96]	IFP	Broken
Elgamal – (1984) [ElG84, Elg85]	DLP	Broken
DSA – (1991) [KG13]	DLP	Broken
Schnorr – (1989) [Sch89, Sch91]	DLP	Broken
DSS/ECDSA (Elliptive Curve DS Algorithm) – (2013) [KG13]	DLP	Broken

Since Shor's discovery, industry and academia have focused their research on the practicability of scalable quantum computers, mainly to exploit their undeniable capabilities. For example, this type of computer can efficiently process large amounts of information in parallel [BLM17, CCJ+16]. This situation has led to a new area in the field of cryptography called *post-quantum cryptography*, aimed at constructing new cryptographic algorithms that are intractable even in the presence of powerful quantum computers. Among the current post-quantum cryptographic proposals (see Table 2.6 based on [CCJ+16, BL17b, BL17a]), lattice-based cryptography has attracted attention within the cryptographic community. It is one of the most promising candidates to be standardized as a post-quantum cryptography solution due to its efficiency, parallelism, and believed security guarantees like resistance against quantum attacks [Lau17, MR09]. Lattices also offer a strong security assurance under the assumed *worst-case hardness* of lattice problems, which is significantly better than the assumed *average-case hardness* of

other cryptographic constructions. Besides the basic primitive schemes (Encryption, Digital Signatures and Key Exchange), lattice-based cryptography provides several new advanced and powerful constructions like Fully Homomorphic Encryption (FHE) [Go09] (with some applications [TBS14, ATBS15]), Multilinear Maps [GGH13], Identity-Base Encryption (IBE) [ABB10], Attribute-Based Encryption (ABE) [Boy13], Functional Encryption [BSW11] and Obfuscation [GGH+16]. The next section will define lattices and explain how they are applied in cryptography.

TABLE 2.6: Post-Quantum Cryptographic Algorithms

Post-Quantum Primitive	**Advantages**	**Disadvantages**
Lattice-based	New applications like Fully Homomorphic Encryption (FHE); relatively simple and efficient implementations; parallelism; strong security guarantees.	Difficulty in providing precise estimates to secure some lattice schemes; issues in performance make some schemes impractical, like FHE.
Code-based	Very short signatures; efficient verification process.	Very large public key sizes; inefficient signing process; there is not a current scheme that overcome these issues.
Multivariate polynomial	More successful in Digital Signatures.	Some proposed schemes are broken.
Hash-based signatures	Requires only hash functions to be secure; flexible as these signatures can be instantiated with any hash function. Hash-based signatures are classified as stateful and stateless. In the former, after each signature generation, the state (or the secret key) is updated, whereas in the latter, it does not require this update process, which make the implementation easier.	The signer must keep the record of previous signatures; any error will lead to insecurity; large signature size.
Isogenies on Supersingular Elliptic Curves	Computations can be parallelized.	Signing and verification processes are slow.

2.5 Lattice-based Cryptography

Lattices were initially investigated by the mathematicians Gauss, Hermite and Minkowski during the 19th century. Then, in the late 20th century, several computational aspects were investigated, resulting in certain classical cryptographic algorithms being broken [Po16]. During this time a number of important discoveries in this area were made, such as: the *Lenstra-Lenstra-Lovász* (LLL) basis-reduction algorithm in 1982 [LLL82]; and Ajtai's worst-case to average-case security reduction for lattices in 1996 [Ajt96], which yielded the first lattice-based cryptographic function. In 1997, the first public-key encryption scheme based on lattices was devised by Ajtai and Dwork [AD97]. During this year, the *Goldreich, Goldwasser and Halevi* (GGH) public-key encryption and digital signature schemes were also proposed [GGH97]; and in 1998, the NTRU public-key cryptosystem scheme [HPS98], which used polynomial rings was also constructed. Following these proposals, a few improvements in Ajtai's schemes were carried out by Micciancio and Regev in [MR09], introducing new methods of analysis over lattices, such as Gaussian measures and harmonic analysis, which were relevant for the design and analysis of lattice-based cryptographic schemes. Afterwards, lattices began to be used in designing new cryptography schemes, which to some extent were implemented to overcome the quantum computing paradigm and also to construct new types of schemes in cryptography. Table 2.7 shows a brief comparison between this post-quantum lattice primitive and classical public-key cryptography.

Table 2.7: Comparison of Lattice-Based and Classical Cryptography

Lattice-Based Cryptography	**Classical Public-key Cryptography**
Security based on a worst-case problem.	Security based on an average-case problem.
Based on hardness of lattice problems.	Based on factoring, DLPs.
Still not broken by quantum algorithms.	Broken by quantum algorithms.
Simple computation: Additions and multiplication.	Requires modular exponentiation.

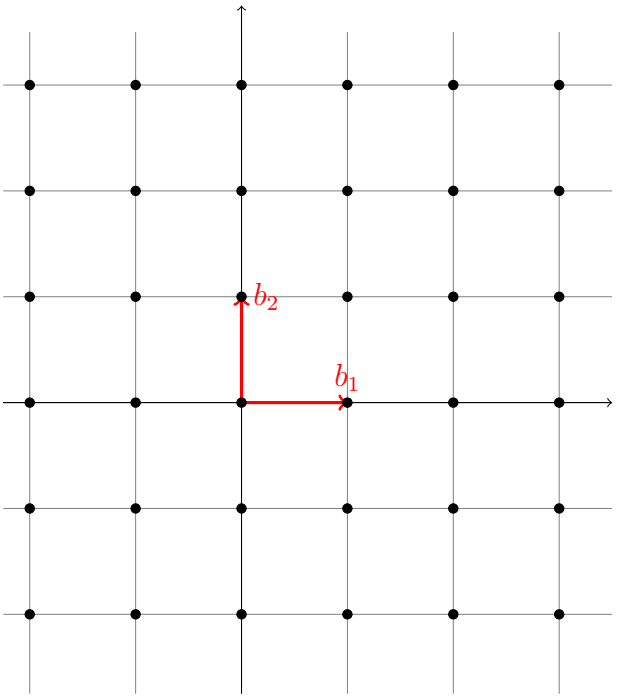


FIGURE 2.5: A *non-full-rank* lattice with basis vector $(1, 1)$ (taken from [Vai11])

2.5.1 Definition

A lattice is a geometric object that can be defined as a periodic grid of discrete points in n-dimensional real space $\mathbb{R}^m$. Mathematically, a lattice is the set of all integer combinations. Given n linearly independent vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \in \mathbb{R}^m$, where these vectors are also known as a *basis* of the lattice, a lattice is therefore generated as:

$$\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \stackrel{\text{def}}{=} \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in \mathbb{Z} \text{ for } 1 \leq i \leq n \right\}$$

In the lattice, n and m define the *rank* and the *dimension* respectively, and $n \leq m$. A lattice is called *full-rank* (see Figure 2.5) when $n = m$; otherwise the lattice is *non-full-rank* (see Figure 2.6). The basis of a lattice can be represented by a matrix $\mathbf{B} = [\mathbf{b}_1 \ldots \mathbf{b}_n] \in \mathbb{R}^{m \times n}$. Hence, using this matrix notation, where $\mathbf{Bx}$ is the matrix-vector multiplication, a lattice can also be represented as $\mathcal{L}(\mathbf{B}) \stackrel{\text{def}}{=} \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\}$.

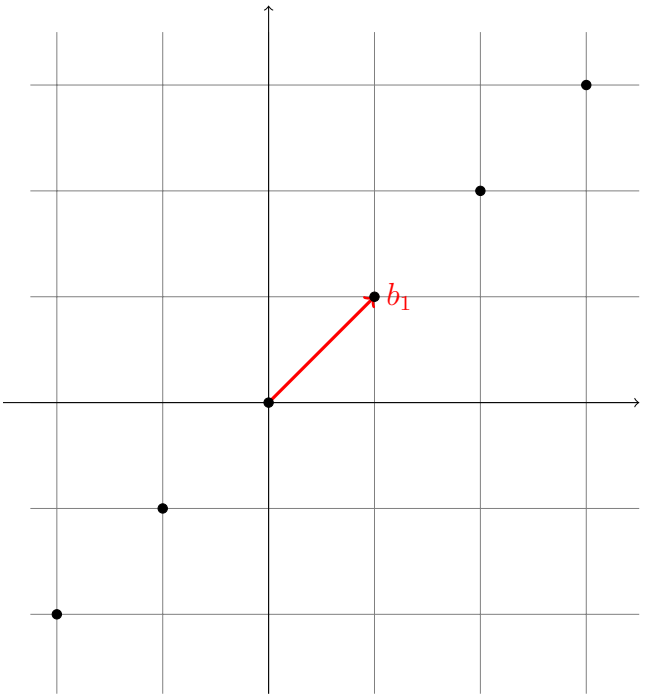


FIGURE 2.6: The lattice $\mathbb{Z}^2$ with basis vectors $(0, 1)$ and $(1, 0)$ (taken from [Vai11])

The concept of unimodular matrix $\mathbf{U} \in \mathbb{Z}^{n \times n}$ (integer square matrix with determinant ± 1) is used to generate an infinite number of different bases for a lattice. This means that if $\mathbf{B}$ is the basis of a lattice $\mathcal{L}(\mathbf{B})$ then $\mathbf{BU}$ is the basis for $\mathcal{L}(\mathbf{B})$ for any unimodular matrix $\mathbf{U}$. Using multiple bases $\mathbf{B}$s in a lattice is a relevant concept that can be applied in cryptography [MR09]. Algebraically, the determinant is used to prove that the absolute value of the determinant basis $\mathbf{B}$ is equal to the volume of the pararallelpiped generated by the basis vectors of the lattice $\mathcal{L}(\mathbf{B})$: $\det(\mathcal{L}(\mathbf{B})) = |\det(\mathbf{B})|$. Geometrically, it corresponds to the inverse of the density of the lattice points in $\mathbb{R}^m$, and to prove that the basis belongs to the lattice, the notion of the fundamental parallelepiped is used to calculate this density [Vai11]. A *dual lattice* is defined as $\mathcal{L}(\mathbf{B})^* \stackrel{\text{def}}{=} \{\mathbf{y} | \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}, \forall \mathbf{x} \in \mathcal{L}(\mathbf{B})\}$. In this case, $\mathcal{L}(\mathbf{B})^*$ is the set of vectors y satisfying $\langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}^n$ for all vectors $\mathbf{x} \in \mathcal{L}(\mathbf{B})$.

2.5.2 q-ary Lattices

The class of q-ary lattices $\mathcal{L}$ satisfies the relation $q\mathbb{Z}^n \subseteq \mathcal{L} \subseteq \mathbb{Z}^n$ for an integer q, which means that the vector $\mathbf{x} \in \mathcal{L}$ is determined by $\mathbf{x} \bmod q$. These lattices have one-to-one correspondence with linear codes in $\mathbb{Z}_q^n$. This means that, given a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, for some integers q, m and n, two m-dimensional q-ary lattices can be defined as:

$$\mathcal{L}_q(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{y} = \mathbf{A}^T\mathbf{s} \bmod q \text{ for some } \mathbf{s} \in \mathbb{Z}^n\},$$

$$\mathcal{L}_q^{\perp}(\mathbf{A}) = \{\mathbf{y} \in \mathbb{Z}^m : \mathbf{A}\mathbf{y} = 0 \bmod q\}.$$

These q-ary lattices are dual to each other:

$$\mathcal{L}_q(\mathbf{A}) = q\mathcal{L}_q^{\perp}(\mathbf{A})^*$$

$$\mathcal{L}_q^{\perp}(\mathbf{A}) = q\mathcal{L}_q(\mathbf{A})^*$$

2.5.3 Cyclic Lattices

The cyclic lattices were introduced in [Mic07] to improve the efficiency of previous lattice-based cryptographic functions by replacing the general matrices by others with a special structure. Under this technique a lattice $\mathcal{L}(\mathbf{A})$ with a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ can be replaced by a block matrix of the form $\mathbf{A} = [\mathbf{A}^{(1)}, | \ldots | \mathbf{A}^{(m/n)}]$, having each block $\mathbf{A}^{(1)} \in \mathbb{Z}_q^{n \times n}$ as a circulated matrix. This means that each block rotates the coordinates of the $\mathbf{A}^{(1)}$'s first column cyclically. Cyclic lattices brings improvements in storage and running time since the entire matrix is now replaced by this circulated matrix. In terms of storage, it now stores only the first column of the matrix (needed only m elements) rather than storing the entire matrix (with nm elements). This result is similar for the running time with $O(m)$ after employing this technique. Cyclic lattices still provide a high level of security since some studies assume that solving lattice problems on these types lattices is as hard

as the general lattices case [Mic07]. Although, this improvement was shown to be used in one-way functions, is not sufficient for other more useful cryptographic primitives, like collision resistant hash functions.

2.5.4 Ideal Lattices

Extending the cyclic lattices idea, further studies were carried out to create efficient constructions based on the worst-case assumptions. For instance, in [LM06], it demonstrated how to design collision resistant hash functions, using a standard lattice problems for a new type called *ideal lattices* of certain polynomial rings. This new variant of lattices is defined in [LM06] as:

Definition 2.1 (Ideal Lattices)**.** An ideal lattice is an integer lattice $\mathcal{L}(\mathbf{B}) \subseteq \mathbb{Z}_n$ such that $\mathcal{L}(\mathbf{B}) = \{g \mod f : g \in I\}$ for some monic polynomial f of degree n and ideal $I \subseteq \mathbb{Z}[x]/f$, where f are:

- Polynomial $f(x)$ should be monic polynomial, which means that the coefficient of the largest power of x is 1.
- Polynomial $f(x)$ should be irreducible, which means that an ideal lattice of the ring $\mathbb{Z}[x]/f$ defines a full-rank lattice in $\mathbb{Z}^n$.
- The ring norm $\|g\|_f$[1] is not much bigger than $\|g\|_\infty$ for any polynomial g. That is, the smaller the ratio $\|g\|_f/\|g\|_\infty$, the harder it is to break the function.

2.5.5 Computational Problems

The lattice computational problems are defined based on [MR09, Po16]:

1. Shortest Vector Problem (SVP): given a basis $\mathbf{B} \in \mathbb{Z}^{m \times n}$, the challenge is to find the shortest non-zero vector $\mathbf{t}$ in the lattice $\mathcal{L}(\mathbf{B})$ with $\|\mathbf{v}\| =$

[1] In the ring $\mathbb{Z}[x]/f$ with infinitive norm $\|(g+f)\|_f = \|g \mod f\|_\infty$, the $\|\cdot\|_f$ is a norm that does not depend on the choice of g [LM06].

$\lambda_1(\mathcal{L})$. Whereas the $\lambda_1(\mathcal{L})$ denotes the minimum distance of a lattice which is the length of a shortest nonzero lattice vector, $\|\cdot\|$ commonly denotes the Euclidean norm.

2. Approximate version of *SVP* (SVP_γ): given a basis a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$ of an n-dimensional lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$, find a nonzero vector $\mathbf{v} \in \mathcal{L}$ such that $\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L})$. This means that it outputs the length of a lattice vector which is at most some approximation factor $\gamma(n)$ times the length of the shortest nonzero vector with n being the dimension of the lattice.

3. Decisional Approximate *SVP* ($GapSVP_\gamma$): given the basis of a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$ of an n-dimensional lattice $\mathcal{L} = \mathcal{L}(\mathbf{B})$ and $\lambda_1(\mathcal{L})$ as the norm of the shortest non-zero vector of the lattice, it determines which is the case: (1) $\lambda_1(\mathcal{L}) \leq 1$ or (2) $\lambda_1(\mathcal{L}) > \gamma(n)$, where $\gamma(n)$ is the approximation factor $\gamma \geq 1$ and is typically taken to be a function of the lattice dimension.

4. Closest Vector Problem (*CVP*): given the basis of a lattice $\mathbf{B} \in \mathbb{Z}^{m\times n}$ and a target vector $\mathbf{t}$, which is not necessarily in the lattice, the challenge is to find the lattice point $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ closest to the target $\mathbf{t}$ with respect to a given norm.

5. Approximate version of *CVP* (CVP_γ): it is a problem that with input $\mathbf{B} \in \mathbb{Z}^{m\times n}$ and $\mathbf{t} \in \mathbb{Z}^m$, it returns $\mathbf{x} \in \mathcal{L}(\mathbf{B})$ such that for all $\mathbf{y} \in \mathcal{L}(\mathbf{B}) \|\mathbf{x} - \mathbf{t}\| \leq \gamma\|\mathbf{y} - \mathbf{t}\|$.

6. Shortest Independent Vectors Problem (*SIVP*): given the basis of a lattice $\mathbf{B} \in \mathbb{Z}^{n\times n}$, the problem consists of finding n linearly independent vectors $\mathbf{S} = [\mathbf{s}_1, \ldots \mathbf{s}_n], \forall \mathbf{s}_i \in \mathcal{L}(\mathbf{B})$, such that the maximum norm vector $\mathbf{s}_i$ is minimum: $\|\mathbf{S}\| = \max_i \|\mathbf{s}_i\|$.

In lattice-based cryptography, the approximation factor (denoted by γ), is used to treat the lattice problems *SVP*, *CVP* and *SIVP*. More precisely, SVP_γ indicates that the challenge is to find a vector whose norm is at most γ times that of the shortest nonzero vector. The most common norm used in these problems is the

Euclidean norm $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$ [Pei08]. Computationally efficient solutions might be found in these problems if a lattice is in dimension 1, 2 or 3, where basic mathematical tools such as Gaussian elimination can be used. However, it is believed the complexity increases, and it becomes computationally difficult, even for the power of quantum algorithms, when lattice-based cryptography uses higher dimensions of 100 or above [Reg04, HPSS08]. This is, in fact that LLL can solve SVP_γ in $poly(n)$ time for gamma exponential in n, but despite many years of research since the LLL algorithm was published in the 1980s, the best known algorithms for SVP_γ and CVP_γ for $\gamma = poly(n)$ have running time exponential in n, which is the lattice dimension.

2.5.6 Discrete Gaussian Distribution

Discrete Gaussian distribution is believed to be crucial in the development of lattice-based cryptographic constructions since it is used to proved certain lattice problems. This notion is defined as:

Definition 2.2 ([Po16])**.** For any positive integer n and real $s > 0$, which is taken to be $s = 1$ when omitted, define the Gaussian function $\rho_s : \mathbb{R}^n \to \mathbb{R}^+$ of parameter or width s as $\rho_s(\mathbf{x}) := \exp\left(-\pi\|\mathbf{x}\|^2/s^2\right) = \rho(\mathbf{x}/s)$.

2.5.7 Foundations

In this section, it is explained the main average-case problems, the Short Integer Solution (SIS) problem and the Learning With Errors (LWE), and their variants over mathematical rings. These problems turn out to be fundamental in the construction of lattice-based cryptosystems.

- Short Integer Solution (SIS) problem: Discovered by Ajtai [Ajt96], this serves as the foundation of schemes such as one-way functions, collision-resistant hash functions and digital signatures. A formal definition of the $\mathbf{SIS}_{n,q,\beta,m}$ problem, as stated in [Po16], is that the columns of the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ are

formed given m uniformly random vectors $\mathbf{a}_i \in \mathbb{Z}_q^n$. The challenge is to find a nonzero integer vector $\mathbf{z} \in \mathbb{Z}^m$ of norm $\|\mathbf{z}\| \leq \beta$ such that $\mathbf{Az} = \sum_i \mathbf{a}_i \mathbf{z}_i = \mathbf{0} \in \mathbb{Z}_q^n$. This problem can be represented as an average-case SVP_β on q-ary m-dimensional integer lattices: $\mathcal{L}_q^\perp(\mathbf{A}) := \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{Az} = \mathbf{0} \mod q\}$. In other words, this problem tries to find a sufficiently short nonzero vector in $\mathcal{L}_q^\perp(\mathbf{A})$, where $\mathbf{A}$ is chosen to be uniformly random [Po16, LS15]. Ajtai shows in [Ajt96] that SIS is at least as hard as approximating several worst-case lattice problems, such as GapSVP. The function $z \rightarrow \mathbf{A} * \mathbf{z}$ is typically a many-to-one and not reversible, which means that it cannot be used in encryption algorithms, as the reverse property is needed.

- Learning With Errors (LWE) problem: This scheme was introduced by Regev in [Reg10] (2005). It was shown that this problem is as hard to solve as several worst-case lattice problems and therefore, some cryptographic schemes have based their security under this assumption. The main idea of this problem is to recover a secret $\mathbf{s} \in \mathbb{Z}_q^n$, given a set of *approximate* random linear equations on $\mathbf{s}$, where each equation is corrected up to some small additive error. If there is no error, it would be trivial to solve the equations to get $\mathbf{s}$ by using mathematical techniques such as Gaussian elimination in polynomial time. However, the problem becomes complex once the error is introduced which makes Gaussian elimination techniques fail. Based on this discussion, it can be said that the LWE obtains hardness based on worst-case lattice problems. The LWE problem, in turn, is at least as hard as approximating the standard lattice problems *SIVP* and decision-LWE problem of the shortest vector problem GapSVP in the worst case. Thus, the LWE problem is also said to be difficult even for quantum computing attack [BLP+13]. In addition, LWE is attractive, as it typically leads to efficient implementations involving low-complexity operations such as additions and multiplications.

  In the LWE problem, n and q are positive integers. The $\chi \in \mathbb{Z}$ is taken to be a discrete Gaussian distribution (error distribution) with width αq for some $\alpha < 1$. For a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, the LWE distribution $A_{\mathbf{s},\chi} \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ is

sampled by choosing $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random, choosing $e \leftarrow \chi$, and the output is: $(\mathbf{a}, b = \langle \mathbf{s}, \mathbf{a} \rangle + e \bmod q)$.

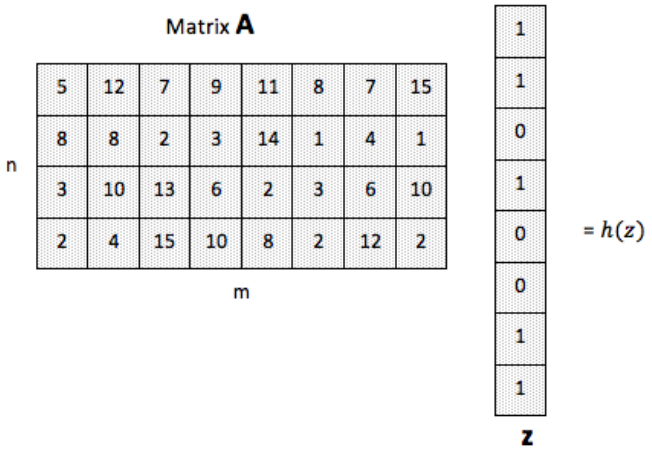
Two variants of the LWE are defined: the search-LWE (to find a secret given LWE samples) and decision-LWE (to distinguish between random and LWE samples); a precise definition is as follows:

1. Search-LWE$_{q,n,\chi,m}$ problem: For uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$, given m independent samples $(\mathbf{a}_i, b_i)$ from $A_{r \to \mathbf{s},\chi}$, find $\mathbf{s}$.
2. Decision-LWE$_{q,n,\chi,m}$ problem: Distinguish with non-negligible advantage between two cases: (1) given m independent samples $(\mathbf{a}_i, b_i)$ from $A_{r \to \mathbf{s},\chi}$ for a uniformly random $\mathbf{s} \in \mathbb{Z}_q^n$ or (2) the uniform distribution on pairs $(\mathbf{a}_i, \mathbf{b}_i)$.

- Ring SIS (R-SIS) problem: This construction was introduced by Micciancio in [Mic07] and is an analogue of the SIS problem described above. This *Ring* refers to the mathematical structure and differs from the *Ring Signature* scheme that was discussed in Section 2.2. This problem is parameterized with the ring R, which is a degree-n polynomial ring of the form $R = \mathbb{Z}[X]/(f(X))$, where $f(X)$ can be either X^n+1 or $X^{2^k}+1$. We let $R_q := R/qR$. Saying this, the **R-SIS**$_{q,\beta,m}$ is defined as follows: given m uniformly random elements $\mathbf{a}_i \in R_q$, defining a vector $\mathbf{a} \in R_q^m$, find a nonzero vector $\mathbf{z} \in R^m$ of norm $||\mathbf{z}|| \leq \beta$ such that $f_{\mathbf{a}}(\mathbf{z}) := \langle \mathbf{a}, \mathbf{z} \rangle = \mathbf{a}^t \cdot \mathbf{z} = \sum_i \mathbf{a}_i \mathbf{z}_i = 0 \in \mathbb{R}_q$ where the vector $\mathbf{z}$ is over R as $||\mathbf{z}|| = (\sum_i ||\mathbf{z}_i||^2)^{1/2}$ [Po16]. Schemes based on the SIS problem are not practical due to their inefficiency in terms of storage and computational time. This is mainly because of the random matrix $\mathbf{A}$, as illustrated in Figure 2.7, where both the storage and the computation function time require $O(nm)$.

  The main advantage of R-SIS is that it is relatively compact and efficient when compared to SIS. An ideal lattice [LM06, PR06] helps to improve the efficiency of this SIS scheme and so is commonly referred to *Ring-SIS* or *Ideal-SIS*. This improvement can be described as having a matrix $\mathbf{A}$ of size $m \times n$ that uses only two vectors (or polynomials), which

FIGURE 2.7: Short Integer Solution (SIS) – example



are rotated with a negation (see highlighted in red Figure 2.8) until entirely the size of $\mathbf{A}$. As a result, this scheme now requires $O(m)$; that is, $\mathbf{A} \times \mathbf{z}$ is computed faster and more compactly than SIS. Figure 2.9 shows an example of Ring-SIS when the polynomials are computed as: $(5 + 8x + 3x^2 + 2x^3) \times (1 + x + x^3) + (10 + 9x + x^2 + 7x^3) \times (x^2 + x^3)$ in the ring $\mathbb{Z}_p[x]/(x^n + 1)$. The hardness of R-SIS (similarly to SIS) can be proved at least as hard as certain problems on ideal lattices in the worst case, and it is claimed that, for typical choices of R, the SVP_γ and the SIVP_γ problems on ideal lattices are very hard to solve in the worst case [LM06, PR06]. As a consequence, R-SIS leads to providing constructions such as one-way functions, collision-resistance functions, identification schemes and digital signatures, but not for encryption.

- Ring LWE (R-LWE) problem: introduced in [LPR10] and a search variant in [SSTX09], this is an analogue of the standard LWE problem. As with the R-SIS problem, the LWE constructions are inefficient, in terms of storage and computational time. The ideal lattices are utilised to enhance the LWE construction, which then turns into the Ring-LWE. This is illustrated in the example (see Figure 2.10), where the left illustrates the inefficient matrix $\mathbf{A}$, whilst the right shows the Ring-LWE using ideal lattices. The ring-LWE is defined in the ring $R = \mathbb{Z}[X]/f(X)$ with the three properties of ideal lattices described in Section 2.5.4, and for an integer modulus q defining the

FIGURE 2.8: Ring-SIS - example 1

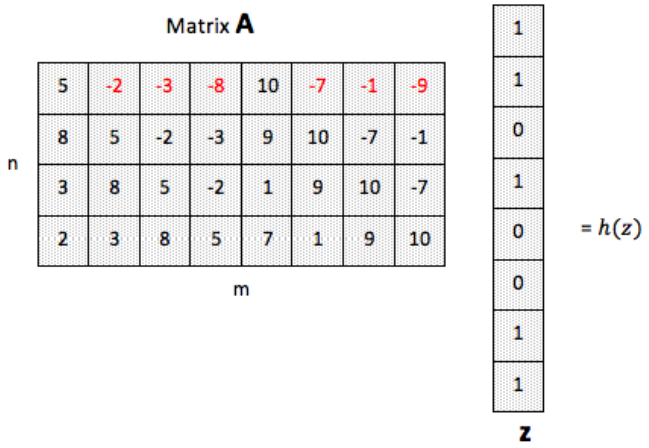
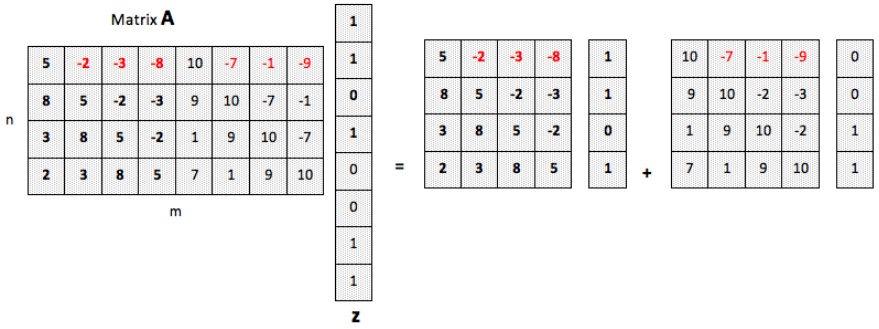


FIGURE 2.9: Ring-SIS - example 2

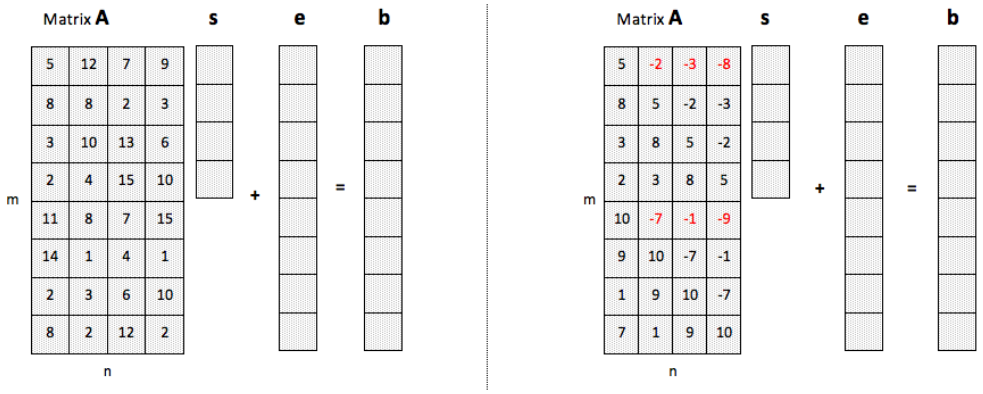


quotient ring $R_q := R/(qR) = R = Z[X]/f(X)$. The ring-LWE problem is to distinguish pairs $(a_i, b_i) \in R_q \times R_q$ from uniformly random pairs where $b_i = a_i \cdot s + e_i$. The $s \in R_q$ is a random secret, the $a_i \in R_q$ is uniformly random and independent, and the $e_i \in R$ is the noise or error, which is short and independent.

This assumption results in being secure, as stated in [LPR10]; there is quantum resistance and provable worst-case hardness, which means that breaking certain instantiations of Ring-LWE is at least as hard as quantum SVP_γ on any ideal lattice in the ring R. Two versions of R-LWE (*search* and *decision*) are explained as follows:

1. Ring-LWE *search version*: In the ring $R = \mathbb{Z}_q[x]/(x^n + 1)$, $\mathbf{A}$ is taken arbitrarily and independent from a Gaussian distribution, where $\mathbf{s}$ is random in R and $\mathbf{e}_i$ are small with a distribution symmetric around

FIGURE 2.10: LWE (left) and Ring-LWE (right) – example



zero so given: $(\mathbf{a}_1, \mathbf{a}_1 \cdot \mathbf{s} + \mathbf{e}_1)$, ..., $(\mathbf{a}_k, \mathbf{a}_k \cdot \mathbf{s} + \mathbf{e}_k)$, the challenge is to find $\mathbf{s}$ [SSTX09].

2. Ring-LWE *decision version*: In the ring $R = \mathbb{Z}_q[x]/(x^n + 1)$ and given arbitrarily many independent samples from $\mathbf{A}$ and $\mathbf{B}$: $(\mathbf{a}_1, \mathbf{b}_1)$, ..., $(\mathbf{a}_k, \mathbf{b}_k)$, the questions are

   – Does there exist an $\mathbf{s}$ and small $\mathbf{e}_1, \ldots, \mathbf{e}_k$ such that $\mathbf{b}_i = \mathbf{a}_i \cdot \mathbf{s} + \mathbf{e}_i$? or,

   – Are all $\mathbf{b}_i$ uniformly random in R?

   These questions are the foundational problems in lattice-based cryptography where the decision Ring-LWE oracle has to determine where the $\mathbf{b}$ belongs to. Most definitions in cryptography require decision problems for their constructions such as pseudorandom functions. That is, to break an encryption scheme, it needs to decide whether a value comes from the same distribution or a random one. In this case, the Ring-LWE distribution in the decision problem is pseudorandom, and therefore it is more appropriate for public-key cryptography [LPR10].

- **Ring-LWE Public-key cryptosystem**: This public-key cryptosystem with a fix ring $R_q = Z_q[x]/(x^n + 1)$ for n a power of 2, can be described as:

- *Key generator algorithm*: chooses uniformly random $a \in R_q$; two small s *and* $e \in R$. It outputs the secret key s and the public key $(a, b = a \cdot s + e) \in R_q^2$.
- *Encryption process*: to encrypt an n-bit message $\mu \in \{1, 0\}^n$, it chooses three random small $r, e_1, e_2 \in R$ from the distribution $\chi \in \mathbb{Z}$. The output of this process is the pair $(u, v) \in R_q^2$.
- *Decryption process*: to decrypt the ciphertext, it is required to compute $v - u \cdot s = (r \cdot e - s \cdot e_1 + e_2) + [q/2] \cdot m \cdot mod \cdot q$. Finally, the message μ can be recovered by removing the approximation error or noise.

The parameters *a, t, u* and *v* are pseudorandom based on the decision Ring-LWE, which implies semantic security or indistinguishability under chosen-plaintext attack (IND-CPA). This indicates that this scheme reveals nothing about the encrypted messages to a passive adversary and therefore [Po16] classifies this as passive secure encryption.

2.5.8 Lattice-Based Digital Signatures

Digital signatures underlying their security in lattice-based cryptography can be categorized as GGH/NTRUSign, Hash-and-sign and Fiat-Shamir signatures.

Signatures based on GGH [GGH97] and NTRUSign [HPS01] were the first schemes employing lattice-based cryptography, their security being based on the *CVP* problem. Both signatures, however, were completely broken in [NR09]. Hash-and-sign [GPV08] and one-time [LM08] lattice-based signatures were introduced to provide provably secure schemes which relied on the hardness of worst-case lattice problems, though they were still inefficient. For example, Hash-and-sign produced signature sizes of megabytes long. Likewise, the one-time signatures required a tree-hashing transformation techniques like Merkle trees which turned to be impractical [Po16, Lyu12]. By contrast, the Fiat-Shamir signatures [Lyu09] is the framework of Bimodal Lattice Signature Scheme (BLISS) which is one of the most practical lattice-based digital signature schemes [Po16].

2.5.8.1 Fiat-Shamir and BLISS Signatures

The construction of the Fiat-Shamir digital signature is established with a three-message identification scheme. It follows the Fiat Shamir protocol from [FS86, AABN02], which is also known as the three-move protocol, by the use of the messages: *Commitment-Challenge-Response*. The Fiat-Shamir transformation then [FS86, AABN02] converts this protocol into a signature scheme. Lattice-based signature schemes apply this transformation [Lyu12, Lyu09], typically relying their security on the SIS lattice problem. The initial step of such schemes was to construct a lattice-based identification scheme whereby the challenge is treated as a polynomial ring R. The security of the identification scheme is reduced to the hardness of the SVP problem as well as the random oracle model (ROM). The identification scheme is then transformed into a digital signature with several optimizations within the parameter settings. For instance, improving the length of the signature and making it computationally infeasible to find collisions in the hash function family H. The security of the scheme is also dependent on the hardness of finding collisions in certain hash function families. An adversary who is able to forge a signature can then use this to find a collision in a hash function chosen randomly from H. Therefore, it was said that forging a signature and finding a collision in a randomly chosen $h \leftarrow H(\cdot)$ is equivalent to finding short vectors in the ring-SIS problem.

Further improvements of these constructions were subsequently effectuated [Lyu12]. One being the hardness of the lattice problem assumption; that is, adapting ring-LWE from ring-SIS. This change resulted in optimal signature and key sizes, significantly improving the efficiency of such constructions. Another optimisation occurred in the signing procedure that involves asymptotically shorter signatures. This stage required an additional technique, denominated *rejection sampling*, so that the distribution of the signatures is independent of the secret-key. Since this distribution is sampled from the normal distribution, highly accurate computations would be needed [Lyu12]. This enhanced scheme is shown to be strongly unforgeable, based on the worst-case hardness in the SVP problem. One

state-of-the-art lattice-based signature scheme is the Bimodal Lattice Signature Scheme (BLISS) [DDLL13]. The main contribution of this work was the significant improvement in the rejection sampling stage. The BLISS is demonstrated to be practical, as Figure 2.11 shows how this scheme provides acceptable levels of efficiency and compactness. For instance, by comparing it to RSA-2048 and having an estimated security of 128 bits (BLISS-II), it provides public-key and private-key sizes of 7 and 2 kilobits, respectively.

FIGURE 2.11: Lattice-Based and Classical Assumptions Signatures [DDLL13]

Implementation	Security	Signature Size	SK Size	PK Size	Sign (ms)	Sign/s	Verify (ms)	Verify/s
BLISS-0	$\leqslant$ 60 bits	3.3 kb	1.5 kb	3.3 kb	0.241	4k	0.017	59k
BLISS-I	128 bits	5.6 kb	2 kb	7 kb	0.124	8k	0.030	33k
BLISS-II	128 bits	5 kb	2 kb	7 kb	0.480	2k	0.030	33k
BLISS-III	160 bits	6 kb	3 kb	7 kb	0.203	5k	0.031	32k
BLISS-IV	192 bits	6.5 kb	3 kb	7 kb	0.375	2.5k	0.032	31k
RSA 1024	72-80 bits	1 kb	1 kb	1 kb	0.167	6k	0.004	91k
RSA 2048	103-112 bits	2 kb	2 kb	2 kb	1.180	0.8k	0.038	27k
RSA 4096	$\geqslant$ 128 bits	4 kb	4 kb	4 kb	8.660	0.1k	0.138	7.5k
ECDSA[1] 160	80 bits	0.32 kb	0.16 kb	0.16 kb	0.058	17k	0.205	5k
ECDSA 256	128 bits	0.5 kb	0.25 kb	0.25 kb	0.106	9.5k	0.384	2.5k
ECDSA 384	192 bits	0.75 kb	0.37 kb	0.37 kb	0.195	5k	0.853	1k

Later in this thesis we show that based on this assumption digital signatures (BLISS) can be used to implement a Linkable Ring Signature scheme, which provides security guarantees against quantum attacks (or at least no known quantum algorithm) as well as offering efficiency and compactness when compared with classical public-key cryptography.

Several lattice-based ring signature schemes have been proposed [BK10, CLRS10, WW12, AMBB+13, WS11, LLNW16], where a few number of them are Linkable Ring Signatures LRS. The first of these constructions [YHAL+17] is based on the development of a lattice-based weak Pseudo Random Function (wPRF), an accumulator scheme (Acc) and a framework named as Zero-Knowledge Arguments of Knowledge (ZKAoK). These techniques are used to construct LRS schemes where the security guarantees for the LRS properties', *unforgeability*, *anonymity*, *linkability* and *non-slanderability*, rely on the lattice problems. The second lattice LRS scheme [ZZTA17] uses ideal lattices along with a lattice-based homomorphic commitment technique in its construction. The security properties are based on the hardness of lattice-based assumptions; however, there is no discussion as to

how to secure the scheme in terms of *non-slanderability*. This scheme is also shown to be used in a cryptocurrency application. The last lattice LRS proposal [BHS18] is devised using lattice-based variants named Module-SIS and Module-LWE problems and its security properties rely on the lattice-based assumptions.

With this background in place, this thesis presents the designed the *Lattice-based one-time Linkable Ring Signature* (L2RS) scheme, which was independently and concurrently constructed with [BHS18]. The L2RS scheme shares similar features, but the L2RS scheme offers unconditional anonymity. This construction is a generalisation of BLISS, a demonstrated practical lattice-based digital signature [DDLL13]. It is secure in terms of *unforgeability, linkability and non-slanderability* under the lattice hardness of the Ring-SIS problem and unlike the above lattice-based LRS schemes [YHAL$^+$17, ZZTA17, BHS18], the L2RS scheme achieves *unconditional anonymity*, meaning that this scheme will be secure even if an adversary has unlimited computational resources and time. This scheme (L2RS) will be explained in Chapter 4.

Having discussed Linkable Ring Signatures schemes along with the positive effect of using Lattice-based Cryptography, this review now examines how these cryptographic structures can be applied to cryptocurrencies.

2.6 The Age of Cryptocurrencies

Cryptocurrencies are applications that use virtual assets and cryptographic mechanisms to conduct e-commerce operations such as electronic payments or money transfers. Those payments can be carried out among accounts or wallets, independently of a central party [CELR18]. This leads to some advantages like lower transaction fees, theft resistance and anonymous payments. Bitcoin [Nak09] is by far the most widely known and decentralised cryptocurrency to date, having its three underlying building blocks: the transactions, blockchain and consensus protocol. It allows a party to perform electronic financial operations in a decentralised Peer-to-Peer (P2P) network, contrary to the traditional banking model.

Although, it was intended to achieve privacy and anonymity in Bitcoin by using pseudonyms, some analyses [RS13, KKM14] show that these security properties are compromised, thus information about the payers, payees and transactions can be revealed. This indicates that Bitcoin is only a *pseudo-anonymous* cryptocurrency.

Since its creation, Bitcoin has revolutionised the field of digital currency and thus motivated the invention of new cryptocurrencies, also known as alcoins. An example of these, CryptoNote [VS13], was proposed to address the privacy weaknesses of Bitcoin. It also offers a framework that can be extended to other cryptocurrencies such as Bytecoin [Byt15] and Monero [Mon14]. CryptoNote cryptocurrency [VS13] uses *Traceable Ring Signatures*, also referred to as *One-time Ring Signatures*, a fundamental building block to achieve *true anonymity*. With true anonymity any member of the ring (or group) can create a signature, but it is infeasible by a verifier to identify the real signer. These type of signatures hides information about the parties: (sender and receiver), and it also has a *key image* or *linking tag* to prevent double spending coins. However, evaluation of this construction in [Noe14, Mac15, NNM14] discovers serious vulnerabilities of this cryptocurrency which impacts the privacy of the involved parties in the transaction. This CryptoNote is also affected due to its weaknesses, since there is not a securely effective mechanism to hide the transferred amount.

The Ring Confidential Transactions RingCT [Noe15] protocol was devised to address these issues. This protocol extends the former CryptoNote scheme by using a new class of linkable ring signatures called *Multi-layered Linkable Spontaneous Anonymous Group Signature* (MLSAG) based on [LWW04b]. This signature is spontaneous (or *ad-hoc*), which removes the dependency of a trusted third party and group members are unaware of belonging to a determined group, thereby enhancing the *anonymity* property. It is also multilayered, meaning that it allows multiple inputs and outputs (wallets) in transactions. The security of RingCT is ameliorated by introducing Confidential Transactions [Max15], which enable amounts to be hidden by using the *Pedersen Commitment* [Ped91] technique. This cryptographic technique enables a party to commit to a chosen secret value

while keeping it hidden from other parties, allowing this commitment to be opened at a later time. Such a primitive offers homomorphic properties, allowing parties to prove the account balance by computing homomorphically computing input and output accounts/wallets to show that their difference is zero. RingCT added another verification mechanism for the committed output amounts which was called *range proof*, guaranteeing that this amount lies in a range of *non-negative* values and avoiding the creation of *free money*. Bulletproofs [BBB+18] are an efficient technique for this range preservation, but these approaches rely on number theory assumptions.

Subsequently, further improvements were proposed in RingCT 2.0 [SALY17] and RingCT 3.0 [YSL+19]. In particular, they provided formal definitions for both the cryptocurrency protocol, the security model, and a sound security analysis of the RingCT protocol. The RingCT protocol improves the size of the signature by using one-way accumulators [BdM93] along with signatures of knowledge (SoK) [CL06]. However, it requires a trusted setup for its accumulator to achieve the signature constant size and hence contradicts the philosophy of decentralised cryptocurrencies.

Nevertheless, the security of this RingCT protocol relies on classical number-theory mathematical assumptions, in particular, the hardness of DLP [ElG84, Elg85]. In Section 2.4, it was discussed how these classical assumptions would become invalid when a powerful quantum computer is fully developed, and how *Lattice-based cryptography* would overcome this threat. In the post-quantum setting, several RingCT's have been proposed. The first *post-quantum RingCT* scheme using Lattice-based cryptography was devised in [ATSS+18], called *Lattice RingCT v1.0*. This was an extended work based on both the L2RS and RingCT constructions. These schemes used the Bimodal Lattice Signature Scheme (BLISS) - a demonstrated practical and secure lattice-based digital signature [DDLL13], as a underlying building block. The schemes are secure in terms of *balance, unforgeability, linkability* and *non-slanderability* under the lattice hardness of the Module-SIS problem, and they also achieve *unconditional anonymity*. However, this proposed lattice-based RingCT protocol was limited. Firstly, it only enables transfers from a

single input wallet to a single output wallet (SISO). The RingCT model, in which signatures are *one-time*, needs to receive change after making a payment or transfer, thus a new output wallet is required. This dependent functionality points out the importance of supporting multiple input and output wallets. Secondly, introducing more than one output wallet also introduces new security problems like the negative output amount attack [BBB+18], where an adversary is capable of creating extra coins or free money. This problem was addressed in the former RingCT [Noe15] by using range proofs. Lastly, the proposed *Lattice RingCT v1.0* showed no security definitions, nor proofs.

An improved version of the Lattice-based Ring Confidential Transactions (LRCT), supporting Multiple-Input wallets and Multiple-Output (MIMO), wallets was constructed. It was called *Lattice RingCT v2.0* or MIMO.LRCT. This protocol was an extension of the SISO.LRCT scheme ([ATSS+18]) where its underlying framework (L2RS signature) was changed to be compatible to this extension. The MIMO.LRCT offers formal definitions and security proofs, and inherits the mechanisms used in former RingCT techniques, like homomorphic commitments, amount and range preservation. The proposed MIMO.LRCT scheme inherits the post-quantum security guarantees from the L2RS scheme, like the hardness of lattice-based assumptions as well as *unconditional anonymity*. Recently, a novel efficient, scalable and practical lattice-based RingCT protocol was devised in [EZS+19], where it accomplished significant improvements in the signature and key-pair sizes and running time.

Authorisation is an important feature in digital currencies which ameliorates the level of security when wallets are transferred. Since the wallets can be spent with their owners' secret-keys (SKs), then such SKs are a point of vulnerability. For instance, if they are stolen or lost, the wallets' owners will be unable to access their funds and perhaps they might lose their money forever. Therefore, multi-signature schemes enable this authorisation functionality where multiple cosigners N_{CS} cooperate to create a joint SK. The same number of cosigners N_{CS} then need to interact to sign a transaction which confirms that a wallet has been transferred.

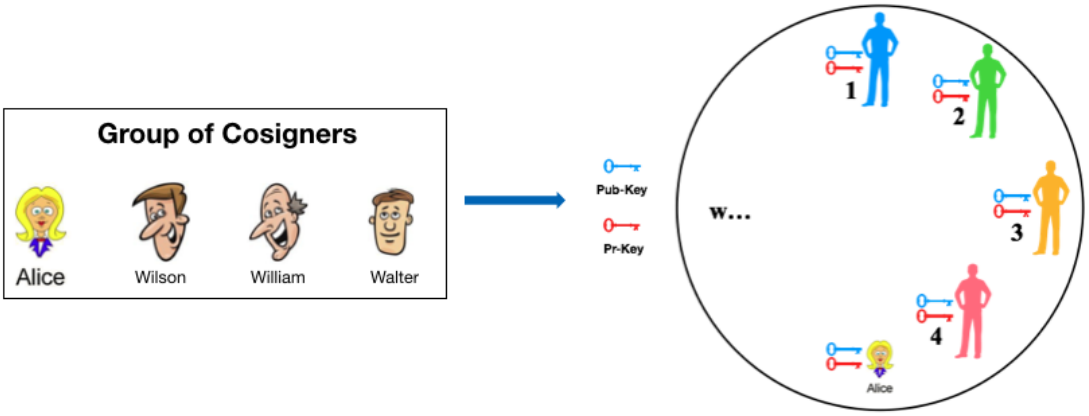
Besides multi-signatures increasing the difficulty for adversaries to mount an attack (as all N_{CS} cosigners need to be compromised), they also offer redundancy, which might protect SKs from being lost [EBS16]. The next Section discusses the definition of this cryptographic primitive.

2.7 Threshold Signature Schemes

The notion of *(t,n)-Threshold Signature* (TS) schemes was initially conceptualised by Desmedt and Frankel in [DF89]. They defined TS as a cryptographic protocol where a subset of size t out of n cosigners collaborates to jointly sign a given message m. Contrary to standard digital signatures, TS splits the secret key (**sk**) into multiple shares distributed across n participants. Later, an interactive protocol is performed with at least the threshold number of cosigners (t out of n) to produce a signature. TS constructions contain several benefits including reliability and security. For instance, TS is employed to augment the confidentiality of secret keys, increase the resilience against secret key exposure, and enable decentralisation of trust [Bra19]. Furthermore, metering applications [DHS03] utilise TS to measure the interaction between servers and clients so e-business can charge fees for advertisements. Similarly, blockchain technology, particularly cryptocurrencies [GBGN17], incorporates TS schemes to provide an extra, more restrictive layer of security. More specifically, this involves the authorisation in digital currencies where a certain number of parties collaborates to approve electronic payments. Figure 2.12 displays this model. In this case, *Alice* needs authorisation from the cosigners, *Wilson*, *William* and *Walter*, to create a (ring) signature.

Securing the cryptographic keys is always crucial to attaining a respectable level of reliability in any secure cryptocurrency application. Since the digital wallets can be spent with their **sk**'s, this would be a single point of vulnerability. For instance, if such **sk**'s are stolen or lost, the owners of the corresponding wallets would be unable to access their funds. Consequently, TS protocols enable this authorisation property to segregate the ownership of digital wallets. Besides TS

FIGURE 2.12: Threshold-Signature Idea



schemes increasing the difficulty for adversaries to mount an attack (as multiple cosigners need to be compromised), they also offer redundancy, which might protect **sk**'s from being lost [EBS16]. In addition, there are other mechanisms that help to secure the generation of cryptographic keys in digital currencies. The *Distributed Key Generation* (DKG) protocol guarantees that nobody learns about the **sk** during the execution of the protocol [GJKR07, TCZ+20]. DKG also operates in a complete decentralised distribution of the trust among the parties, so it requires no trusted party. The *public-key aggregation* is another mechanism that allows the public verifier to only see the aggregate public-key rather than the cosigners' public-keys, providing more favourable privacy and performance results [Alo18, BDN18, MPSW19]. The integration of these controls would prevent the well-known *rogue key* (or *key cancellation*) attacks where a dishonest actor is capable of signing transactions on behalf of honest cosigners [Alo18, GN18, TCZ+20].

The security concept of most cryptographic primitives and protocols is changing due to the foreseeable presence of a sufficiently powerful quantum computer. The security assumptions of *public-key cryptography* that are based on the classical number theory would be efficiently broken in the event of large-scale quantum computers becoming practical [Sho99]. Nowadays, post-quantum cryptographers are devising new algorithms in anticipation of quantum attacks. Among the several approaches proposed to address this concern, *lattice-based cryptography* appears to be a practical alternative to both classical cryptography and this quantum

computing threat. Many lattice-based schemes and protocols have shown optimal performance, simplicity, and reliable security proofs based on the *worst-case* hardness assumptions, meaning that at least one instance of the lattice problem is hard to solve. Moreover, lattice-based cryptosystems even allow powerful new classes of cryptographic mechanisms, such as fully homomorphic encryption and functional encryption [BLM17, BLM18].

The Ring Confidential Transaction (RingCT) [Noe15] is a cryptographic protocol that is widely employed by Monero, one of the most popular cryptocurrencies to date. The RingCT performs e-commerce operations in a decentralised network while maintaining *complete-anonymity* for the parties involved and also preventing double spending coins [CELR18]. These security properties are achieved by employing *Linkable Ring Signature* (LRS) schemes [LWW04b]. In the RingCT framework, *complete-anonymity* provides a remarkable advantage since other cryptocurrencies, such as Bitcoin, are only *pseudo-anonymous* [KKM14]. Further improvements were proposed in RingCT 2.0 [SALY17] and RingCT 3.0 [YSL+19]. These presented formal definitions for both the cryptocurrency protocol and the corresponding security model. Moreover, certain variants of RingCT incorporate an authorisation feature for distributed and co-signing digital wallets by adapting a combination of TS and *Ring Signature* (RS) schemes [Alo18, GN18]. However, this authorisation model has been constructed with number theory assumptions and thus it would be insecure against quantum attacks. In the *post-quantum* setting, the first Lattice-based RingCT (LRCT) was devised in [ATKS+19, ATSS+18]. The LRCT uses the Bimodal Lattice-based Signature Scheme (BLISS) [DDLL13], a demonstrated, practical and secure lattice-based digital signature [DDLL13, DLL+18], as an underlying building block. In a recent study, an efficient, scalable and practical lattice-based RingCT protocol was devised in [EZS+19]. Nonetheless, these *post-quantum* approaches did not incorporate an authorisation model in their design which, as discussed above, can be achieved by using TS schemes.

Several TS schemes have been proposed after its introduction in [DF89]. Most of

the existing TS schemes [Bol03, GGN16, GJKR96, GJKR03, Sho00] and *Threshold Ring Signature* (TRS) schemes [BSS02, LWW04a, LW05b, OTYO18, TWC+04, WFLW03, YLA+11] have been designed with the classical cryptographic assumptions, and only a few constructions are lattice-based. The first lattice-based TRS [CLRS10] was devised based on an identification scheme and the standard lattice-based Short Integer Solution (SIS) hardness problem. The signature size of this scheme was around 25 MB with $t = 50$ and $n = 100$. Later, a new study [BS13] proposed an enhanced version of [CLRS10]. The authors modified this model for the anonymity property, which brought improvements to the signature size (around 13 MB with same ring and threshold sizes as [CLRS10]). In [WDZ+14], an ID-Based TRS from lattices was designed in the random oracle model. The security properties therein relied on a non-standard lattice-based assumption that they defined as a general Graded Computational Diffie-Hellman Problem (gCDHP). Another scheme was constructed in [CHGL19] to be applied in a message block sharing application; however, its analysis disregarded the evaluation of the best known lattice attacks, and overlooked a security reduction in the standard lattice-based Ring-SIS problem seemingly used in this work. However, since DKG protocols are not utilised in their designs, in all likelihood these lattice-based proposals will find themselves vulnerable to rogue key attacks. In addition, they are incompatible with the linkability technique.

2.8 Summary

This chapter has reviewed the key topics covered in this research project. In cryptography, digital signatures enable useful mechanisms that are applied in real applications. It studied how group and ring signatures are extended from standard digital signatures and what security properties are satisfied. Linkable ring signatures, another extension, helps to preserve the privacy of the signer while inheriting the functionality and security properties from ring signatures. Some cryptocurrency applications exploit such features to offer anonymous transactions along with other security guarantees. Quantum attacks are believed to break those cryptographic

constructions based on public-key cryptography, including the linkable ring signatures that are based on number theory assumptions. Among the post-quantum cryptographic approaches, lattice-based cryptography provides certain advantages in security, performance and functionality. Some RingCT cryptocurrency protocols have been constructed using lattice-based cryptography which include the constructions of this project.

Chapter 3

Preliminaries

This chapter recalls several preliminary notations and definitions that are used throughout the entire thesis. We use a polynomial ring $\mathcal{R} = \mathbb{Z}[x]/f(x)$, where $f(x) = x^n + 1$ with n being a power of 2. The ring $\mathcal{R}_q$ is then defined to be the quotient ring $\mathcal{R}_q = \mathcal{R}/(q\mathcal{R}) = \mathbb{Z}_q[x]/f(x)$, where $\mathbb{Z}_q$ denotes the set of all positive integers modulo q (a prime number $q = 1 \bmod 2n$) in the interval $\left[\lfloor \frac{-q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor\right]$. The challenge space $\mathcal{S}_{n,\kappa}$, is the set of all binary vectors of length n and Hamming weight κ. Hash functions are modeled as Random Oracle Model (ROM), $H_0 :\rightarrow \{0,1\}^l$, H_1 with range $\mathcal{S}_{n,\kappa} \subseteq \mathcal{R}_{2q}$, and similarly H_2 with range $\mathcal{S}_{n,\kappa_2} \subseteq \mathcal{R}_{2q}$. When we write $x \hookleftarrow D$, for a distribution D, it means that if D is a set then x is chosen uniformly at random from D. The discrete Gaussian distribution over $\mathbb{Z}^m$ with standard deviation $\sigma \in \mathbb{R}$ and center at zero, is defined by $D^m_\sigma(x) = \rho_\sigma(x)/\rho_\sigma(\mathbb{Z}^m)$, where ρ_σ is the m-dimensional Gaussian function $\rho_\sigma(x) = \exp(-\|x\|^2/(2\sigma^2))$ for $x \in \mathbb{Z}^m$. We denote vectors by bold lower case (e.g. $\mathbf{t}$) whilst matrices are denoted

by upper case (e.g. $\mathbf{A}$), and the identity matrix as $\mathbf{I}$. Vector transposition is denoted by $\mathbf{v}^T$. For a vector $\mathbf{t} \in \mathbb{Z}^m$ and $p \in [1, +\infty]$, we then define the norm ℓ_p as $\|\mathbf{t}\|_p = (\sum_{i=1}^{m} |\mathbf{t}_i|^p)^{1/p}$. In the case of $p = \infty$, we define the ℓ_∞ norm of $\mathbf{t}$ as $\|\mathbf{t}\|_\infty = \max_{i=1}^{m} |\mathbf{t}_i|$. We say that a function *neg(n)* is negligible in n if *neg(n)* $< \frac{1}{2^n}$, and a function $f(n)$ is overwhelming if $1 - f(n)$ is negligible.

Definition 3.1 ($\mathcal{R}$-**SIS**$^{\mathcal{K}}_{q,n,m,\beta}$ problem)**.** (Based on [DDLL13], Def. 2.3). Let denote $\mathcal{K}$ some uniform distribution over the ring $\mathcal{R}_q^{n\times m}$. Given a random matrix $\mathbf{A} \in \mathcal{R}_q^{n\times m}$ sampled from $\mathcal{K}$ distribution, find a non-zero vector $\mathbf{v} \in \mathcal{R}_q^m$ such that $\mathbf{Av} = \mathbf{0}$ and $\|\mathbf{v}\|_2 \leq \beta$, , where $\|\cdot\|_2$ denotes the Euclidean norm.

Definition 3.2 (**MSIS**$^{\mathcal{K}}_{q,m,k,\beta}$ problem)**.** Let $\mathcal{K}$ be some uniform distribution over the ring $\mathcal{R}_q^{k\times m}$. Given a random matrix $\mathbf{A} \in \mathcal{R}_q^{k\times m}$ sampled from $\mathcal{K}$, find a non-zero vector $\mathbf{v} \in \mathcal{R}_q^{m\times 1}$ such that $\mathbf{Av} = \mathbf{0}$ and $\|\mathbf{v}\|_2 \leq \beta$, where $\|\cdot\|_2$ denotes the Euclidean norm.

Lemma 3.3. ([BCK+14]) *Let $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$ where $n > 1$ is a power of 2 and $0 < i, j < 2n - 1$. Then all the coefficients of $2(x^i - x^j)^{-1} \in \mathcal{R}$ are in $\{-1, 0, 1\}$. This implies that $\|2(x^i - x^j)^{-1}\| \leq \sqrt{n}$.*

Lemma 3.4. *For $a, b \in \mathcal{R}_q$, the following relations hold $\|a\| \leq \sqrt{n}\|a\|_\infty$, $\|a\cdot b\| \leq \sqrt{n}\|a\|_\infty \cdot \|b\|_\infty$, $\|a \cdot b\|_\infty \leq \|a\| \cdot \|b\|$.*

Lemma 3.5 (Leftover Hash Lemma (LHL))**.** (Based on [DDLL13], Lemma B.1). *Let $\mathcal{H}$ be a universal hash family of hash functions from X to Y. If $h \leftarrow \mathcal{H}$ and $x \leftarrow X$ are chosen uniformly and independently, then the statistical distance between (h,h(x)) and the uniform distribution on $\mathcal{H} \times Y$ is at most $\frac{1}{2}\sqrt{|Y|/|X|}$.*

Remark 3.6. We use this lemma for a SIS family of hash function $H(\mathbf{S}_0) = \mathbf{A}'_0 \cdot \mathbf{S}_0 \in \mathcal{R}_q$, with $\mathbf{S}_0 \in \mathsf{Dom}_{\mathbf{S}_0}$, where each function is indexed by $\mathbf{A}'_0 \in \mathcal{R}_q^{1\times(m-1)}$. The $\mathsf{Dom}_{\mathbf{S}_0} \subseteq \mathcal{R}_q^{1\times(m-1)}$ consists of a vector of $\mathcal{R}_q$ elements with coefficients in set $\Gamma \stackrel{\text{def}}{=} (-2^\gamma, 2^\gamma)$. This is a universal hash family if $s - s'$ is invertible in $\mathcal{R}_q$ for all distinct pairs s, s' in $\Gamma^n \subseteq \mathcal{R}_q$. This can be guaranteed by appropriate choice q of $\mathcal{R}_q$, e.g. as shown in ([LS18], Corollary 1.2), it is sufficient to use q such that $f(x) = x^n + 1$ factors into k irreducible factors $\mod q$ and $2^\gamma < \frac{1}{\sqrt{k}} \cdot q^{1/k}$. We assume that $\mathcal{R}_q$ is chosen to satisfy this condition.

Definition 3.7 (Gaussian Distribution)**.** The discrete Gaussian distribution over $\mathbb{Z}^m$ with standard deviation $\sigma \in \mathbb{R}$ and center at zero, is defined by $D_\sigma^m(\mathbf{x}) = \rho_\sigma(\mathbf{x})/\rho_\sigma(\mathbb{Z}^m)$, where ρ_σ is m dimensional Gaussian function $\rho_\sigma(\mathbf{x}) = \exp\left(\frac{-\|\mathbf{x}\|^2}{2\sigma^2}\right)$.

3.1 Rejection Sampling

The notion of rejection sampling was initially introduced in [VN51]. It states that given a source bound of a probability distribution g, one can sample from an arbitrary target probability distribution f. To be more specific, a sample t is selected from g and is accepted with probability $f(t)/(M \cdot g(t))$ with M being some positive real, this condition is restarted if it is not accepted.

Lemma 3.8 (Rejection Sampling)**.** (Based on [DDLL13], Lemma 2.1). *Let V be an arbitrary set, and $h : V \to \mathbb{R}$ and $f : \mathbb{Z}^m \to R$ be probability distributions. If $g_v : \mathbb{Z}^m \to R$ is a family of probability distributions indexed by $v \in V$ with the property that there exists a $M \in \mathbb{R}$ such that $\forall v \in V, \forall \mathbf{v} \in \mathbb{Z}^m, M \cdot g_v(\mathbf{z}) \geq f(\mathbf{z})$. Then the output distributions of the following two algorithms are identical:*

1. *$v \leftarrow h, z \leftarrow g_v, output(\mathbf{z}, v)$ with probability $f(\mathbf{z})/(M \cdot g_v(\mathbf{z}))$.*
2. *$v \leftarrow h, z \leftarrow f, output(\mathbf{z}, v)$ with probability $1/M$.*

3.2 Homomorphic Commitment Definition

This is a cryptographic technique that is used to provide confidential transactions, in particular cryptocurrencies [Noe15]. This primitive allows one party to commit to a chosen value while keeping it secret to other parties, then this committed value can be revealed later. The definition of such technique, which is based on [BDL+18], has three algorithms: (KeyGen, Com, Open), such that:

- Pub-Params $\leftarrow$ KeyGen(1^λ): A PPT algorithm that produces a public commitment parameter Pub-Params after receiving the security parameter (λ).

- $\mathsf{c} \leftarrow \mathsf{Com}$: A PPT algorithm that receives the $\mathsf{Pub\text{-}Params}$, the randomness r and the message m. This algorithm generates the commitment c.
- $\mathrm{m}' \leftarrow \mathsf{Open}$: A PPT algorithm that receives the commitment c along with the randomness r, and it outputs m'. A valid commitment c is opened if $(\mathrm{m}' = \mathrm{m})$.

The security properties of this non-interactive homomorphic commitment scheme are defined as:

Definition 3.9 (Hiding)**.** This property ensures that the commitment $\mathsf{Com}(\mathrm{m}, \mathsf{r})$ does not leak information on m, that is, for any PPT adversary $\mathcal{A}$, it holds that:

$$\left| \Pr \left[\mathcal{A}(\mathsf{c}_b) = b : \begin{array}{l} \mathsf{Pub\text{-}Params} \leftarrow \mathsf{KeyGen}(1^\lambda); \mathsf{r} \leftarrow \mathsf{RandGen}(\mathsf{Pub\text{-}Params}); \\ (\mathrm{m}, \mathrm{m}') \leftarrow \mathcal{A}(\mathsf{Pub\text{-}Params}); b \leftarrow \{0,1\}; \mathsf{c}_b \leftarrow \mathsf{Com}(\mathsf{r}, \mathrm{m}_b) \end{array} \right] - \frac{1}{2} \right|,$$

is $negl(\lambda)$.

Definition 3.10 ($\beta-$Binding)**.** This property ensures that the commitment $\mathsf{Com}(\mathrm{m}, \mathsf{r})$ can only be opened in one way, that is, for any PPT adversary $\mathcal{A}$, it holds that:

$$\Pr \left[\begin{array}{ll} \mathsf{r} \neq \mathsf{r}' \wedge & \mathsf{Pub\text{-}Params} \leftarrow \mathsf{KeyGen}(1^\lambda); \\ \mathrm{m} \neq \mathrm{m}' \wedge & : \ \mathsf{r} \leftarrow \mathsf{RandGen}(\mathsf{Pub\text{-}Params}); \\ \mathsf{Com}(\mathrm{m}, \mathsf{r}) = \mathsf{Com}(\mathrm{m}', \mathsf{r}') & (\mathrm{m}, \mathsf{r}, \mathrm{m}', \mathsf{r}') \leftarrow \mathcal{A}(\mathsf{r}) \end{array} \right] \leq negl(\lambda),$$

where $\|\mathsf{r}\|, \|\mathsf{r}'\| \leq \beta$.

3.3 Fiat-Shamir Non-Interactive Zero-Knowledge Proofs in the Random Oracle Model

Zero-knowledge proof of knowledge (ZKPoK) is a two party protocol between the *prover* and the *verifier*, which allows the *prover* to convince the *verifier* that he knows some information, without revealing anything about the secret apart from what the claim itself already reveals [BCK+14].

Definition 3.11. Let be $\mathcal{L} \subseteq \{0,1\}^*$ the language that has witness relationship $R \subseteq \{0,1\}^* \times \{0,1\}^*$ if $x \in \mathcal{L} \leftrightarrow \exists(x,w) \in R$. We call w a witness for $x \in \mathcal{L}$. Let $(\mathcal{P},\mathcal{V})$ be a two-party protocol where $\mathcal{P}$ (prover) and $\mathcal{V}$ (verifier) are PPT algorithms, and $\mathcal{L},\mathcal{L}'$ be languages with witness relations R, R' with $R \subset R'$. Then $(\mathcal{P},\mathcal{V})$ has a proof σ with completeness error α, public input x and private input w, if the following conditions are satisfied:

- The protocol uses a hash function H modeled as a random oracle which is called by both $\mathcal{P}$ and $\mathcal{V}$. This protocol has the following form: on input (x,w), $\mathcal{P}$ outputs a proof σ that is sent to $\mathcal{V}$. On input x, the verifier $\mathcal{V}$ accepts or rejects σ.
- Completeness: whenever $(x,w) \in R$, the honest verifier accepts the proof σ with probability at least $1-\alpha$.
- Soundness: given a dishonest prover $\mathcal{A}$ with input x, it outputs a valid proof σ with non-negligible probability, then there there exists a PPT algorithm $\mathcal{E}$ (the knowledge extractor) that extracts a witness w' satisfying $(x,w') \in R'$.
- Special honest-verifier zero-knowledgeness (HVZK): there exists two PPT algorithms $\mathcal{S}$ (the simulator) and $\mathcal{S}_H$ (random oracle simulator) that take $x \in \mathcal{L}$, and output the proofs $\sigma_{sim} = \mathcal{S}(x)$ and $\mathcal{S}_H(x,\cdot)$ such that is computationally indistinguishable from $\sigma = \mathcal{P}(x,w)$ and $H(\cdot)$ generated by a real protocol.

Chapter 4

The SISO of L2RS and LRCT

[1]The Linkable Ring Signature (LRS) primitive is receiving much attention in the cryptographic field thanks to its distinguishing capability of anonymously detecting whether two linkable ring signatures were signed by same signatory. Technically, this primitive uses a *linking tag* that has a secure relationship with the signer's secret-key to detect if such condition is satisfied. Monero [Noe15], a cryptocurrency application, employs this property to protect digital coins from double spending attacks while keeping the signer's identity unrevealed. Thus, the user's anonymity is preserved. However, as most of the proposed LRS schemes rely on number theory assumptions, then this primitive and its variants could be vulnerable to quantum attacks [Sho99, GK96, CCJ+16].

[1]This chapter was published as: Alberto Torres W.A., Steinfeld R., Sakzad A., Liu J.K., Kuchta V., Bhattacharjee, N., Au M.H., Cheng, J. (2018) *Post-Quantum One-Time Linkable Ring Signature and Application to Ring Confidential Transactions in Blockchain (Lattice RingCT v1.0)*. In: Susilo W., Yang G. (eds) Information Security and Privacy. ACISP 2018. Lecture Notes in Computer Science, vol 10946. Springer, Cham. DOI:https://doi.org/10.1007/978-3-319-93638-3_32.

In this chapter, the first post-quantum model of linkable ring signatures is introduced, which is called Lattice-based one-time Linkable Ring Signature (L2RS). The construction of the L2RS is motivated by the discrete logarithm scheme (the LRS from [LWW04a]) and the Bimodal Lattice Signature Scheme (BLISS) [DDLL13]. The security model shows that the adversary is restricted to interact with the challenger *one-time*. We prove that the distribution of the signature is independent to the secret key used to produce that signature by the leftover hash lemma, as defined in Chapter 3. Thus the L2RS provides unconditional anonymity; in contrast to [LWW04a] where this property is computationally secure. Other security properties are protected under the lattice hardness assumption, that is, the Ring Short Integer Solution (Ring-SIS) [Mic07]. Afterwards, this novel signature is applied in a post-quantum cryptocurrency protocol, denominated by Lattice Ring Confidential transaction (LRCT), which forms the foundation of a privacy-preserving protocol in any post-quantum secure cryptocurrency application. The first version of these schemes supports transfers from Single-Input to Single-Output (SISO) wallets.

This chapter begins by defining the LRS scheme, along with its algorithms and the correctness requirements, in Section 4.1. Then, in Section 4.2, the security model defines what type of attacks are considered in the security evaluation, namely, the unforgeability, anonymity, linkability, and non-slanderability attacks. Next, the construction of the L2RS is presented, in Section 4.3, where its algorithms are explained in detail. The results of the security analysis, which follow the definitions of the security model, are shown along with its proofs in Section 4.4. In the following Section 4.5, the application of the proposed scheme, the LRCT, is defined and explained. The performance analysis of the above schemes (L2RS and LRCT) is explored in Section 4.6 where the parameters and signatures sizes are provided in different versions, concretely. The chapter concludes with a summary in Section 4.7.

4.1 Definitions of LRS

The Linkable Ring Signature LRS scheme has five Probabilistic Polynomial Time (PPT) algorithms (LRS.Setup, LRS.KeyGen, LRS.SigGen, LRS.SigVer, LRS.SigLink). In addition, the correctness of this scheme is satisfied by the signature correctness (LRS.SigGen Correctness) and the linkability correctness (LRS.SigLink Correctness). These algorithms are specified as follows:

- LRS.Setup: a PPT algorithm that takes the security parameter λ and produces the Public Parameters (Pub-Params).
- LRS.KeyGen: a PPT algorithm that by taking the Pub-Params, it produces a pair of keys: the public-key pk and the private-key sk.
- LRS.SigGen: a PPT algorithm that receives the Pub-Params, the signer π's sk, a message μ and the list L of users' pk's in the ring signature, and outputs a signature $\sigma_L(\mu)$.
- LRS.SigVer: a PPT algorithm that takes Pub-Params, a signature $\sigma_L(\mu)$, a list L of pk's and the message μ, and it verifies if this signature was legitimately created. This algorithm outputs either: Accept or Reject.
- LRS.SigLink: a PPT algorithm that inputs two valid signatures $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$. It determines if these signatures were produced by same signer π, without revealing his/her identity. Thus, this algorithm has a deterministic output: Linked or Unlinked.

Correctness requirements:

- LRS.SigGen Correctness: this property guarantees that valid signatures signed by honest signers are accepted by a verifier with overwhelming probability.
- LRS.SigLink Correctness: this requirement ensures that if two signatures $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$ are signed by an honest signer π, so SigLink outputs Linked with overwhelming probability.

4.2 Security Model

4.2.1 Oracles for adversaries

The following oracles are available to any adversary who tries to break the security of the LRS scheme:

- $\mathsf{pk}_i \leftarrow \mathcal{JO}(\perp)$. The *Joining Oracle*, on request, adds new user(s) to the system. It returns the public-key(s) pk_i.
- $\mathsf{sk}_i \leftarrow \mathcal{CO}(\mathsf{pk}_i)$. The *Corruption Oracle*, on input a pk_i, it returns the corresponding sk_i.
- $\sigma'_L(\mu) \leftarrow \mathcal{SO}(w, L, \mathsf{pk}_\pi, \mu)$. The *Signing Oracle*, on input a group size w, a set L of w pk's, the signer's pk_π, and a message μ, this returns a valid signature $\sigma'_L(\mu)$.

4.2.2 Security Game Definition

- One-time Unforgeability. One time unforgeability for the LRS scheme is defined in the following game between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$ who has access to the oracles $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$ and the random oracle:
  1. $\mathcal{S}$ generates and gives the list L of pk's to $\mathcal{A}$.
  2. $\mathcal{A}$ may query the oracles according to any adaptive strategy.
  3. $\mathcal{A}$ gives $\mathcal{S}$ a ring signature size w, a set L of w pk's, a message μ and a signature $\sigma_L(\mu)$.

  $\mathcal{A}$ wins the game if:
  - $\mathsf{LRS.SigVer}(\sigma_L(\mu))$=**Accept**.
  - pk's in the L are outputs from $\mathcal{JO}$ oracle.
  - No pk in L has been input to $\mathcal{CO}$.

– $\sigma_L(\mu)$ is not an output of $\mathcal{SO}$.

– No signing key pk_π was queried more than once to $\mathcal{SO}$.

The advantage of the one-time unforgeability in the LRS scheme is denoted by

$$\textbf{Advantage}_{\mathcal{A}}^{ot-unf}(\lambda) = \Pr[\mathcal{A} \text{ wins the game }].$$

Definition 4.1 (One-Time Unforgeability)**.** The L2RS scheme is one-time unforgeable if for all PPT adversary $\mathcal{A}$, $\textbf{Advantage}_{\mathcal{A}}^{ot-unf}(\lambda)$ is negligible.

- Unconditional Anonymity. It should be infeasible for an adversary $\mathcal{A}$ to distinguish a signer's pk with probability larger than 1/2, even if the adversary has unlimited computing resources. This property for a LRS scheme is defined in the following game between a simulator $\mathcal{S}$ and an unbounded adversary $\mathcal{A}$.

  1. $\mathcal{A}$ may query $\mathcal{JO}$ according to any adaptive strategy.
  2. $\mathcal{A}$ gives $\mathcal{S}$ the $L = \{\mathsf{pk}_0, \mathsf{pk}_1\}$, which is the output of the $\mathcal{JO}$, and a message μ.
  3. $\mathcal{S}$ flips a coin $b = \{0, 1\}$, then $\mathcal{S}$ computes the signature $\sigma_b = \mathsf{LRS.SigGen}(L, \mathsf{sk}_b, \mu, \mathsf{Pub\text{-}Params})$. This signature is given to $\mathcal{A}$.
  4. $\mathcal{A}$ outputs a bit b'.
  5. The output of this experiment is defined to be 1 if $b = b'$, or 0 "zero" otherwise.

  $\mathcal{A}$ wins the game if:

  – pk_0 and pk_1cannot be used by $\mathcal{CO}$ and $\mathcal{SO}$.

  – Outputs 1, where $b = b'$, with Pr = 1/2.

  The unconditional anonymity advantage of the LRS scheme is denoted by

$$\textbf{Advantage}_{\mathcal{A}}^{Anon}(\lambda) = \Big| \Pr[b = b'] - \frac{1}{2} \Big|.$$

Definition 4.2 (Unconditional Anonymity)**.** An LRS scheme is unconditional anonymous if for any unbounded adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{Anon}(\lambda)$ is zero.

- Linkability. It should be infeasible for an adversary $\mathcal{A}$ to unlink two valid LRS signatures which were correctly generated with same sk_π. Meaning that when these two valid signatures are the input of LRS.SigLink, the algorithm outputs **Unlinked**. To describe this security property, we use the interaction between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$:

  1. The $\mathcal{A}$ queries the $\mathcal{JO}$ multiple times.
  2. The $\mathcal{A}$ outputs two signatures $\sigma_L(\mu)$, $\sigma'_{L'}(\mu')$, and two lists L, L' of pk's.

  $\mathcal{A}$ wins the game if:

  - The pk's in L and L' are outputs of $\mathcal{JO}$.
  - $\mathcal{A}$ queried $\mathcal{CO}$ only once to get the sk_π, corresponding to pk_π.
  - By calling LRS.SigVer on input $\sigma_L(\mu)$ and $\sigma'_{L'}(\mu')$, it outputs **Accept** on both inputs.
  - Finally, it gets **Unlinked**, when calling LRS.SigLink on input $\sigma_L(\mu)$ and $\sigma'_{L'}(\mu')$.

  Thus the advantage of the linkability in the LRS scheme is denoted by

  $$\mathbf{Advantage}_{\mathcal{A}}^{Link}(\lambda) = \Pr[\mathcal{A} \text{ wins the game}].$$

  Definition 4.3 (Linkability)**.** The L2RS scheme is linkable if for all PPT adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{Link}$ is negligible.

- Non-slanderability. It should be infeasible for an adversary $\mathcal{A}$ to link two valid LRS signatures which were correctly generated with different sk's. This means that an adversary can slander an honest user for signing a valid signature so the adversary can produce another valid signature such that the LRS.SigLink algorithm outputs **Linked**. To describe this, we use the interaction between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$:

1. The $\mathcal{S}$ generates and gives the list L of pk's to $\mathcal{A}$.
2. The $\mathcal{A}$ queries the $\mathcal{JO}$ and $\mathcal{CO}$ to obtain pk_π and sk_π, respectively.
3. $\mathcal{A}$ gives the generated parameters to $\mathcal{S}$.
4. $\mathcal{S}$ uses the sk_π and calls the $\mathcal{SO}$ to output a valid signature $\sigma_L(\mu)$, which is given to $\mathcal{A}$.
5. The $\mathcal{A}$ uses the remaining keys of the ring signature $(w-1)$ to create a second signature $\sigma'_L(\mu)$ by calling the $\mathcal{SO}$ algorithm.

$\mathcal{A}$ wins the game if:

- An $\mathsf{LRS.SigVer}$, on input $\sigma_L(\mu)$ and $\sigma'_L(\mu)$, outputs **Accept**.
- The keys pk_π and sk_π were not used to generate the second signature $\sigma'_L(\mu)$.
- When calling the $\mathsf{L2RS.SigLink}$ on input $\sigma_L(\mu)$ and $\sigma'_L(\mu)$, it outputs **Linked**.

Thus the advantage of the non-slanderability in the $\mathsf{L2RS}$ scheme is denoted by

$$\mathbf{Advantage}^{NS}_{\mathcal{A}}(\lambda) = \Pr[\mathcal{A} \text{ wins the game}].$$

Definition 4.4 (Non-Slanderability)**.** An LRS scheme is *non-slanderable* if for all PPT adversary $\mathcal{A}$, $\mathbf{Advantage}^{NS}_{\mathcal{A}}$ is negligible.

4.3 L2RS Scheme Description

We define the $\mathsf{L2RS}$ = ($\mathsf{L2RS.Setup}$, $\mathsf{L2RS.KeyGen}$, $\mathsf{L2RS.SigGen}$, $\mathsf{L2RS.SigVer}$, $\mathsf{L2RS.SigLink}$) as follows:

4.3.1 L2RS.Setup

By receiving the security parameter λ, this L2RS.Setup algorithm randomly chooses $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_{m-1}) \hookleftarrow \mathcal{R}_q^{1\times(m-1)}$ and $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_{m-1}) \hookleftarrow \mathcal{R}_q^{1\times(m-1)}$. This outputs the public parameters Pub-Params:$(\mathbf{A}, \mathbf{H})$.

Remark 4.5. To prevent malicious attack, the L2RS.Setup incorporates a trapdoor in $\mathbf{A}$ and $\mathbf{H}$, in practice L2RS.Setup would generate $\mathbf{A}$ and $\mathbf{H}$ based on the cryptographic Hash function H_2 evaluated at two distinct and fixed constants.

Definition 4.6 (Function L2RS.Lift)**.** This function maps $\mathcal{R}_q^{1\times m}$ to $\mathcal{R}_{2q}^{1\times m}$ with respect to a public parameter $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$. Given $\mathbf{a} \in \mathcal{R}_q$, we let $\mathsf{L2RS.Lift}(\mathbf{A}, \mathbf{a}) \triangleq (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a} + q) \in \mathcal{R}_{2q}^{1\times m}$.

4.3.2 Key Generation - L2RS.KeyGen

This algorithm receives the public parameters, consisting of: $(\mathbf{A}, \mathbf{H})$, it generates a key pair in $\mathcal{R}_q$, then we:

- Pick $(\mathbf{s}_1, \ldots, \mathbf{s}_{m-1})$ with every component chosen uniformly and independently with coefficients in $(-2^\gamma, 2^\gamma)$, where $\gamma = \log(2n\kappa)$
- Establish $\mathbf{S}^T = (\mathbf{s}_1, \ldots, \mathbf{s}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$.
- Compute $\mathbf{a} = \mathbf{A} \cdot \mathbf{S} \bmod q \in \mathcal{R}_q$. The $\mathbf{a}$ and $\mathbf{S}$ are the public-key pk and the private-key sk, respectively.

This L2RS.KeyGen algorithm is described in the following Algorithm 1:

4.3.3 Signature Generation - L2RS.SigGen

The L2RS.SigGen algorithm inputs the user's private-key $\mathbf{S}_\pi$, the message μ, the list of user's public-keys L and the public parameters Pub-Params: $\mathbf{H} \in \mathcal{R}_q^{1\times(m-1)}$ and $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$. This algorithm outputs the signature $\sigma_L(\mu)$. We call π the

Algorithm 1 L2RS.KeyGen - Key-pair Generation $(\mathbf{a}, \mathbf{S})$

Input: Pub-Param: $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$.
Output: $(\mathbf{a}, \mathbf{S})$, being the public-key and the private-key, respectively.
1: **procedure** L2RS.KeyGen($\mathbf{A}$)
2: Let $\mathbf{S}^T = (\mathbf{s}_1, \ldots, \mathbf{s}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\mathbf{s}_i \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
3: Compute $\mathbf{a} = \mathbf{A} \cdot \mathbf{S} \bmod q \in \mathcal{R}_q$.
4: **return** $(\mathbf{a}, \mathbf{S})$.

index in $\{1, \ldots, w\}$ of the user or signatory who wants to sign a message μ. For a message $\mu \in \{0,1\}^*$, the fixed list of public-keys $L = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$ and the private-key $\mathbf{S}_\pi$ which corresponds to $\mathbf{a}_\pi$ with $1 \leq \pi \leq w$; the following computations are performed:

1. We define the linkability tag as $\mathbf{H}_{2q} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h} + q\big) \in \mathcal{R}_{2q}^{1\times m}$, where $\mathbf{H}$ is the fixed public parameter for all users, and $\mathbf{h} = \mathbf{H} \cdot \mathbf{S}_\pi \in \mathcal{R}_q$. We consider $\mathbf{S}_\pi^T \in \mathcal{R}_q^{1\times(m-1)}$ as an element in $\mathcal{R}_{2q}$ and let $\mathbf{S}_{2q,\pi}^T = \big(\mathbf{S}_\pi^T, 1\big) \in \mathcal{R}_{2q}^{1\times m}$, such that $\mathbf{H}_{2q} \cdot \mathbf{S}_{2q,\pi} = q \in \mathcal{R}_{2q}$.

2. The π's public-key is lifted from $\mathcal{R}_q^{1\times m}$ to $\mathcal{R}_{2q}^{1\times m}$, so by calling the lift function $\mathsf{L2RS.Lift}(\mathbf{A}, \mathbf{a}_\pi)$, we get $\mathbf{A}_{2q,\pi} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi + q) \in \mathcal{R}_{2q}^{1\times m}$. Note that $\mathbf{A}_{2q,\pi} \cdot \mathbf{S}_{2q,\pi} = q \in \mathcal{R}_{2q}$

3. By choosing a random vector $\mathbf{u}_\pi = (u_1, \ldots, u_m)^T$, where $u_i \hookleftarrow D_\sigma^n$, for $1 \leq i \leq m$, we calculate $\mathbf{c}_{\pi+1} = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,\pi} \cdot \mathbf{u}_\pi, \mathbf{H}_{2q} \cdot \mathbf{u}_\pi\Big)$.

4. We choose random vector $\mathbf{t}_i = (t_{i,1}, \ldots, t_{i,m})^T$, where $t_{i,j} \hookleftarrow D_\sigma^n$, for $1 \leq j \leq m$, then for $(i = \pi+1, \ldots, w, 1, 2, \ldots, \pi-1)$, after lifting from $\mathcal{R}_q^{1\times m}$ to $\mathcal{R}_{2q}^{1\times m}$, using $\mathsf{L2RS.Lift}(\mathbf{A}, \mathbf{a}_i)$, we obtain $\mathbf{A}_{2q,i} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i + q) \in \mathcal{R}_{2q}^{1\times m}$. Then, we compute $\mathbf{c}_{i+1} = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i\Big)$.

5. A random bit $b \in \{0,1\}$ is selected and then it computes $\mathbf{t}_\pi = \mathbf{u} + \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \cdot (-1)^b$ by using rejection sampling (Definition 3.8).

6. Finally, it outputs the signature $\sigma_L(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$.

A formal description of this algorithm is shown in Algorithm 2.

Algorithm 2 L2RS.SigGen - Signature Generation $\sigma_L(\mu)$

Input: $\mathbf{S}_\pi, \mu, L = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$, Pub-Params: $\mathbf{H} \in \mathcal{R}_q^{1\times(m-1)}$ and $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$.
Output: $\sigma_L(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \text{Pub-Params}\Big)$

procedure L2RS.SigGen($\mathbf{S}_\pi, \mu, L$, Pub-Params)
  Set $\mathbf{H}_{2q} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h} + q\big) \in \mathcal{R}_{2q}^{1\times m}$, where $\mathbf{h} = \mathbf{H} \cdot \mathbf{S}_\pi \in \mathcal{R}_q$.
  Call L2RS.Lift$(\mathbf{A}, \mathbf{a}_\pi)$ to obtain $\mathbf{A}_{2q,\pi} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi + q) \in \mathcal{R}_{2q}^{1\times m}$.
  Let $\mathbf{u} = (u_1, \ldots, u_m)^T$, where $u_i \hookleftarrow D_\sigma^n$, for $1 \le i \le m$.
  Compute $\mathbf{c}_{\pi+1} = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,\pi} \cdot \mathbf{u}, \mathbf{H}_{2q} \cdot \mathbf{u}\Big)$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    Call L2RS.Lift$(\mathbf{A}, \mathbf{a}_i)$ to obtain $\mathbf{A}_{2q,i} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i + q) \in \mathcal{R}_{2q}^{1\times m}$
    Let $\mathbf{t}_i = (t_{i,1}, \ldots, t_{i,m})^T$, where $t_{i,j} \hookleftarrow D_\sigma$, for $1 \le j \le m$.
    Compute $\mathbf{c}_{i+1} = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i\Big)$.
  Choose $b \hookleftarrow \{0, 1\}$.
  Let $\mathbf{t}_\pi \hookleftarrow \mathbf{u} + \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \cdot (-1)^b$.
  Continue with probability $\dfrac{1}{\left(M \exp\left(-\dfrac{\|\mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi\|^2}{2\sigma^2}\right) \cosh\left(\dfrac{\langle \mathbf{t}_\pi, \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \rangle}{\sigma^2}\right)\right)}$
  otherwise **Restart**.
  return $\sigma_L(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$.

4.3.4 Signature Verification - L2RS.SigVer

The L2RS.SigVer algorithm receives the signature $\sigma_L(\mu)$ along with the message μ, the fixed list $L = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$ and the Pub-Params: $(\mathbf{A}, \mathbf{H}) \in \mathcal{R}_q^{1\times(m-1)} \times \mathcal{R}_q^{1\times(m-1)}$, and it outputs a decisional verification answer: Accept or Reject (see Algorithm 3). The signature $\sigma_L(\mu)$ can be publicly validated by computing $\mathbf{H}_{2q} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h} + q\big) \in \mathcal{R}_{2q}^{1\times m}$ in $\mathbf{c}_{i+1}$ for $(i = 1, \ldots, w)$, and it is verified and only accepted under the following conditions: $\|\mathbf{t}_i\|_2 \le B_2$ and $\|\mathbf{t}_i\|_\infty < q/4$ for $1 \le i \le w$, where B_2 is the acceptance bound [DDLL13], $\mathbf{c}_1 = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,w} \cdot \mathbf{t}_w + q \cdot \mathbf{c}_w, \mathbf{H}_{2q} \cdot \mathbf{t}_w + q \cdot \mathbf{c}_w\Big)$.

Theorem 4.7. *Let* $B_2 = \eta\sigma\sqrt{nm}$ *and* $q/4 > \big(\sqrt{2(\lambda+1)\ln 2 + 2\ln(nm)}\big)\sigma$ *and* $\sigma_L(\mu) = \Big(\boldsymbol{c}_1, \boldsymbol{t}_1, \ldots, \boldsymbol{t}_w, \boldsymbol{h}\Big)$ *be generated based on Algorithm 2. Then the output of Algorithm 3 on input* $\sigma_L(\mu)$ *is* ***Accept*** *with probability* $1 - 2^{-\lambda}$.

Proof. In this proof, we use [lemma 4.4, parts 1 and 3, in [Lyu12]]. The part 3 of this lemma shows that the bound on Euclidean norm $B_2 = \eta\sigma\sqrt{nm}$, for a given $\eta > 1$, has a probability $\Pr[\|\mathbf{t}_i\|_2 > \eta\sigma\sqrt{nm}] \ge 1 - 2^\lambda$. In addition,

Algorithm 3 L2RS.SigVer - Signature Verification

Input: $\sigma_L(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$, $L = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$, μ, Pub-Params
Output: Accept or Reject

procedure L2RS.SigVer($\sigma_L(\mu)$, Pub-Params)
  if $\mathbf{H}_{2q} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h} + q\big)$ **then** Continue
    for $(i = 1, \ldots, w)$ **do**
      Call L2RS.Lift$(\mathbf{A}, \mathbf{a}_i)$ to obtain $\mathbf{A}_{2q,i} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i + q) \in \mathcal{R}_{2q}^{1\times m}$.
      if $\mathbf{c}_{i+1} = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i\Big)$ **then** Continue
      else if $\|\mathbf{t}_i\|_2 \leq B_2$ **then** Continue
      else if $\|\mathbf{t}_i\|_\infty < q/4$ **then** Continue
  else if $\mathbf{c}_1 = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,w} \cdot \mathbf{t}_w + q \cdot \mathbf{c}_w, \mathbf{H}_{2q} \cdot \mathbf{t}_w + q \cdot \mathbf{c}_w\Big)$ **then** Accept
  else Reject
  return Accept or Reject

the bound on infinity norm ($\|\mathbf{t}_i\|_\infty < q/4$) is analysed in part 1 of this lemma where its union bound is also considered. It turns out that η is required such $q/4 > \eta\sigma > \big(\sqrt{2(\lambda+1)\ln 2 + 2\ln(nm)}\big)\sigma$, except with probability of $2^{-\lambda}$. □

4.3.5 Signature Linkability - L2RS.SigLink

The L2RS.SigLink algorithm, illustrated in Algorithm 4, takes two signatures as input: $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$, and it outputs either Linked if these signatures were generated by same signatory, or Unlinked, otherwise. For a fixed list of public-keys L and given two signatures: $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$, with the list L which can be described as: $\sigma_L(\mu_1) = \Big(\mathbf{c}_{1,\mu_1}, \mathbf{t}_{1,\mu_1}, \ldots, \mathbf{t}_{w,\mu_1}, \mathbf{h}_{\mu_1}\Big)$ and $\sigma_L(\mu_2) = \Big(\mathbf{c}_{1,\mu_2}, \mathbf{t}_{1,\mu_2}, \ldots, \mathbf{t}_{w,\mu_2}, \mathbf{h}_{\mu_2}\Big)$.

These two signatures must be successfully accepted by the L2RS.SigVer algorithm, then one can verify that the linkability property is achieved if the linkability tags ($\mathbf{h}_{\mu_1}$ and $\mathbf{h}_{\mu_2}$) of the above signatures $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$ are equal.

Algorithm 4 L2RS.SigLink - Signature Linkability

Input: $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$
Output: Linked or Unlinked
1: **procedure** L2RS.SigLink($\sigma_L(\mu_1), \sigma_L(\mu_2)$)
2: **if** $\Big($L2RS.SigVer($\sigma_L(\mu_1)$) = Accept **and** L2RS.SigVer($\sigma_L(\mu_2)$) = Accept$\Big)$ **then** Continue [
3: **else if** $\mathbf{h}_{\mu_1} = \mathbf{h}_{\mu_2}$ **then** Linked
4: **else** Unlinked]
5: **return** Linked or Unlinked

4.3.6 Correctness of SigGen

Beyond the required conditions of L2RS.SigVer, we claim that if $\sigma_L(\mu_1) = (\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h})$ is the output of the L2RS.SigGen algorithm on input $(\mu, L, \mathbf{S}_\pi, \text{Pub-Params})$, then the output of L2RS.SigVer on input $(\mu, L, \sigma_L(\mu_1))$ should be accepted. We need to show that when L2RS.SigVer computes $H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,w} \cdot \mathbf{t}_w + q \cdot \mathbf{c}_w, \mathbf{H}_{2q} \cdot \mathbf{t}_w + q \cdot \mathbf{c}_w)$, the result is equal to $\mathbf{c}_1$. We also show that $H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i) = \mathbf{c}_{i+1}$ for $1 \leq i \leq w-1$ in L2RS.SigVer. In this evaluation, we consider two scenarios, one when $i \neq \pi$ and $i = \pi$:

- For $i \neq \pi$, in L2RS.SigGen we have $\mathbf{c}_{i+1} = H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i)$, while in L2RS.SigVer we compute $\mathbf{c}_{i+1} = H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i)$. These are equal since $\mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i$ (in L2RS.SigGen) $= \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i$ (in L2RS.SigVer); and $\mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i$ (in L2RS.SigGen) $= \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i$ (in L2RS.SigVer).

- For $i = \pi$, in L2RS.SigGen we have $\mathbf{c}_{\pi+1} = H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,\pi} \cdot \mathbf{u}, \mathbf{H}_{2q} \cdot \mathbf{u})$, whereas in L2RS.SigVer we calculate $\mathbf{c}_{\pi+1} = H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,\pi} \cdot \mathbf{t}_\pi + q \cdot \mathbf{c}_\pi, \mathbf{H}_{2q} \cdot \mathbf{t}_\pi + q \cdot \mathbf{c}_\pi)$. In this case, we need to show that $\mathbf{c}_{\pi+1}$ (in L2RS.SigGen) $= \mathbf{c}_{\pi+1}$ (in L2RS.SigVer). In doing so, the following equalities need to be proved:

1. $\mathbf{A}_{2q,\pi} \cdot \mathbf{u} = \mathbf{A}_{2q,\pi} \cdot \mathbf{t}_\pi + q \cdot \mathbf{c}_\pi$, which is equivalent to $\mathbf{A}_{2q,\pi} \cdot (\mathbf{u} - \mathbf{t}_\pi) = q \cdot \mathbf{c}_\pi$. Here, we replace $\mathbf{t}_\pi$ as defined in Algorithm 2, to obtain:

$$\mathbf{A}_{2q,\pi} \cdot (\mathbf{u} - \mathbf{u} - \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \cdot (-1)^b) = q \cdot \mathbf{c}_\pi \iff$$
$$-\mathbf{A}_{2q,\pi} \cdot \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \cdot (-1)^b = q \cdot \mathbf{c}_\pi \iff$$
$$-q \cdot \mathbf{c}_\pi \cdot (-1)^b = q \cdot \mathbf{c}_\pi$$

   We distinguish two cases for b:

   – When b = 0, we verify that -$q \cdot \mathbf{c}_\pi = q \cdot \mathbf{c}_\pi \bmod 2q$.

   – When b = 1, we have $q \cdot \mathbf{c}_\pi = q \cdot \mathbf{c}_\pi \bmod 2q$.

2. $\mathbf{H}_{2q} \cdot \mathbf{u} = \mathbf{H}_{2q} \cdot \mathbf{t}_\pi + q \cdot \mathbf{c}_\pi$, which means that:

$$\mathbf{H}_{2q} \cdot (\mathbf{u} - \mathbf{t}_\pi) = q \cdot \mathbf{c}_\pi \iff$$
$$\mathbf{H}_{2q} \cdot (\mathbf{u} - \mathbf{u} - \mathbf{S}_\pi \cdot \mathbf{c}_\pi \cdot (-1)^b) = q \cdot \mathbf{c}_\pi \iff$$
$$-\mathbf{H}_{2q} \cdot \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \cdot (-1)^b = q \cdot \mathbf{c}_\pi \iff$$
$$-q \cdot \mathbf{c}_\pi \cdot (-1)^b = q \cdot \mathbf{c}_\pi$$

   We distinguish between two cases:

   – When b = 0, it is verified that $-q \cdot \mathbf{c}_\pi = q \cdot \mathbf{c}_\pi \bmod 2q$.

   – When b = 1, we have $q \cdot \mathbf{c}_\pi = q \cdot \mathbf{c}_\pi \bmod 2q$.

4.3.7 Correctness of SigLink

We show that an honest user π who signs two messages μ_1 and μ_2 in the L2RS scheme with the list of public-keys L, obtains a **Linked** output from L2RS.SigLink algorithm with overwhelming probability. As shown in Algorithm 4, two signatures $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$ were created, and then successfully verified by L2RS.SigVer. Therefore, the linkability tags $\mathbf{h}_{\mu_1}$ and $\mathbf{h}_{\mu_2}$ must be equal. To prove this, we show

that:

$$\mathbf{H}_{2q,\mu_1} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\mu_1} + q\big) \in \mathcal{R}_{2q}, \text{where}$$
$$\mathbf{H} = \mathsf{Pub\text{-}Param} \text{ and } \mathbf{h}_{\mu_1} = (\mathbf{H} \cdot \mathbf{S}_\pi + q) \in \mathcal{R}_q$$
$$\mathbf{H}_{2q,\mu_2} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\mu_2} + q\big) \in \mathcal{R}_{2q}, \text{where}$$
$$\mathbf{H} = \mathsf{Pub\text{-}Param} \text{ and } \mathbf{h}_{\mu_2} = (\mathbf{H} \cdot \mathbf{S}_\pi + q) \in \mathcal{R}_q$$

The first parts of the linkability tag in both L2RS signatures have same equality with following probability:

$$Pr\big[2 \cdot \mathbf{H} = 2 \cdot \mathbf{H}\big] = 1.$$

Ultimately, the second part uses the honest user's private-key $\mathbf{S}_\pi$ is used, so we conclude that:

$$Pr\big[-2 \cdot \mathbf{h}_{\mu_1} + q + 2 \cdot \mathbf{h}_{\mu_2} - q = 0\big] = 1.$$

4.4 Security Analysis

Theorem 4.8 (One-Time Unforgeability)**.** *Suppose* $\sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ *is negligible in* n *and* $\frac{1}{|\mathcal{S}_{n,\kappa}|}$ *is negligible and* $y = h$ *is polynomial in* n*, where* h *denotes the number of queries to the random oracle* H_1*. If there is a PPT algorithm against one-time unforgeability of L2RS with non-negligible probability* δ*, then there exist a PPT algorithm that can extract a solution to the* $\mathcal{R}\text{-}\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ *problem (for* $\beta = 2B_2$*) with non-negligible probability* $\left(\delta - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \cdot \left(\frac{\delta - \frac{1}{|\mathcal{S}_{n,\kappa}|}}{y} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) - \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$.

Proof. As stated in [DDLL13], this L2RS scheme relies on the $\mathcal{R}\text{-}\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ problem to be secure against any existential forger. This means that a forgery algorithm succeeds with a negligible probability and so we conclude that under this probability, the attacker will also find a solution to the $\mathcal{R}\text{-}\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ problem. To

prove this, we start replacing the L2RS.SigGen algorithm with L2RS.Hybrid-1 and L2RS.Hybrid-2 algorithms that are used to simulate the creation of the signatures, until we obtain an algorithm that breaks the $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ problem. These Hybrid algorithms are illustrated in Algorithm 5 and Algorithm 6, respectively.

In L2RS.Hybrid-1, the output of the random oracle H_1 is chosen at random from $\mathcal{S}_{n,\kappa} \subseteq \mathcal{R}_{2q}$ and then it is programmed, without checking the value of $\mathbf{A}_{2q,\pi} \cdot \mathbf{u}$ and $\mathbf{H}_{2q} \cdot \mathbf{u}$ being already set. This equality can be described as:

$$H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,w} \cdot \mathbf{t}_w + q \cdot \mathbf{c}_w, \mathbf{H}_{2q} \cdot \mathbf{t}_w + q \cdot \mathbf{c}_w) =$$
$$H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,\pi} \cdot \mathbf{u}, \mathbf{H}_{2q} \cdot \mathbf{u})$$

Every time the L2RS.Hybrid-1 is called, the probability of generating $\mathbf{u}$ such that $\mathbf{A}_{2q,\pi} \cdot \mathbf{u}$ and $\mathbf{H}_{2q} \cdot \mathbf{u}$ are equal to one of the previous output that was queried is at most 2^{-n+1}. We define that the probability of getting a collusion each time is at most $h \cdot 2^{-n+1}$, where "h" is the number of calls to the random oracle H_1, whereas the probability of occurring a collision after "o" queries to the L2RS.Hybrid-1 is at most $o \cdot h \cdot 2^{-n+1}$, which is negligible (Based on [DDLL13], Lemma 3.4).

After analyzing how $\mathbf{c}_1$ can be forged, we evaluate the $(\mathbf{t}_1, \ldots, \mathbf{t}_w)$ of the L2RS scheme. We claim that these are forgeable when an attacker finds a PPT algorithm $\mathcal{F}$ to solve the $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ problem. This attack can be simulated using the L2RS.Hydrid-2 shown in Algorithm 6, where $\mathbf{t}_\pi$ is directly chosen from the distribution D^n_σ (Based on [DDLL13], Lemma 3.5).

The public-key $\mathbf{A}_{2q} \in \mathcal{R}^{1\times m}_{2q}$ is generated such $\mathbf{A}_{2q} \cdot \mathbf{S}_{2q} = q \in \mathcal{R}_{2q}$, so finding a vector $\mathbf{v}$ such that $\mathbf{A}_{2q} \cdot \mathbf{v} = 0 \bmod q$. We denote $y = h$ where y is the number of times the random oracle H_1 is programmed during this attack. Then this attack is performed as follows:

1. Random coins are selected for the forger ϕ and signer ψ.
2. The random oracle H_1 is called to generate the responses of the users in the L2RS scheme, $(\mathbf{c}_1, \ldots, \mathbf{c}_w) \hookleftarrow \mathcal{S}_{n,\kappa}$.

Algorithm 5 One-Time Unforgeability - Signature algorithm of L2RS Hybrid 1

Input: $\mathbf{S}_\pi, \mu$, $L = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$, Pub-Params: $\mathbf{H}$ and $\mathbf{A}$.

Output: $\sigma_L(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$

1: **procedure** L2RS.HYBRID-1($\mathbf{S}_\pi, \mu, L$, Pub-Params)
2: Set $\mathbf{H}_{2q} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h} + q\big) \in \mathcal{R}_{2q}^{1\times m}$, where $\mathbf{h} = (\mathbf{H} \cdot \mathbf{S}_\pi + q) \in \mathcal{R}_q$.
3: Call L2RS.Lift($\mathbf{A}, \mathbf{a}_\pi$) to obtain $\mathbf{A}_{2q,\pi} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi + q) \in \mathcal{R}_{2q}^{1\times m}$.
4: Let $\mathbf{u} = (u_1, \ldots, u_m)^T$, where $u_i \hookleftarrow D_\sigma^n$, for $1 \le i \le m$.
5: Choose at random $\mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}$
6: **for** $(i = \pi + 1, \pi + 2, \ldots, w, 1, 2, \ldots, \pi - 1)$ **do**
7: Call L2RS.Lift($\mathbf{A}, \mathbf{a}_i$) to obtain $\mathbf{A}_{2q,i} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i + q) \in \mathcal{R}_{2q}^{1\times m}$.
8: Let $\mathbf{t}_i = (t_{i,1}, \ldots, t_{i,m})^T$, where $t_{i,j} \hookleftarrow D_\sigma^n$, for $1 \le j \le m$.
9: Compute $\mathbf{c}_{i+1} = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i\Big)$.
10: Choose $b \hookleftarrow \{0, 1\}$.
11: Let $\mathbf{t}_\pi \hookleftarrow \mathbf{u} + \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \cdot (-1)^b$.
12: **Continue** with probability $\dfrac{1}{\left(M \exp\left(-\dfrac{\|\mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi\}\|^2}{2\sigma^2}\right) \cosh\left(\dfrac{\langle \mathbf{t}_\pi, \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \rangle}{\sigma^2}\right)\right)}$
otherwise **Restart**.
13: **return** $\sigma_L(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$.

3. These create a SubRoutine that takes as input $(\mathbf{A}_{2q}, \phi, \psi, \mathbf{c}_1, \ldots, \mathbf{c}_w)$.

4. $\mathcal{F}$ is initialized and run by providing the $\mathbf{A}_{2q}$ and forger's random coins ϕ.

5. The SubRoutine signs the message μ using the signer's coins ψ in the L2RS.Hydrid-2, this produces a signature $\sigma_L(\mu)$.

6. During the signing process, $\mathcal{F}$ calls the oracle H_1 and answers are placed in the list $(\mathbf{c}_1, \ldots, \mathbf{c}_w)$, the queries are kept in a table in the event that same queries are used in this oracle.

7. $\mathcal{F}$ is stopped and it outputs a forgery that is the SubRoutine's result $(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h})$, with negligible probability δ. This output has to be successfully accepted by the L2RS.SigVer algorithm.

If the random oracle was not called using some input $\mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i$, then $\mathcal{F}$ has $1/|\mathcal{S}_{n,\kappa}|$ chances of producing a $\mathbf{c}$ such that $\mathbf{c} = H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q} \cdot \mathbf{t} + q \cdot \mathbf{c}, \mathbf{H}_{2q} \cdot \mathbf{t} + q \cdot \mathbf{c})$. This turns out that $\delta - 1/|\mathcal{S}_{n,\kappa}|$ be the probability that $\mathbf{c} = \mathbf{c}_j$ for some j.

Algorithm 6 One-Time Unforgeability - Signature algorithm of L2RS Hybrid 2 $\sigma_L(\mu)$

Input: $\mathbf{S}_\pi, \mu$, $L = \{\mathbf{a}_1, \ldots, \mathbf{a}_w\}$, Pub-Params: $\mathbf{H}$ and $\mathbf{A}$.
Output: $\sigma_L(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$

1: **procedure** L2RS.HYBRID-2($\mathbf{S}_\pi, \mu, L$, Pub-Params)
2: Set $\mathbf{H}_{2q} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h} + q\big) \in \mathcal{R}_{2q}^{1\times m}$, where $\mathbf{h} = (\mathbf{H} \cdot \mathbf{S}_\pi + q) \in \mathcal{R}_q$.
3: Call L2RS.Lift($\mathbf{A}, \mathbf{a}_\pi$) to obtain $\mathbf{A}_{2q,\pi} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi + q) \in \mathcal{R}_{2q}^{1\times m}$.
4: Let $\mathbf{u} = (u_1, \ldots, u_m)^T$, where $u_i \hookleftarrow D_\sigma^n$, for $1 \le i \le m$.
5: Choose at random $\mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}$
6: **for** $(i = \pi + 1, \pi + 2, \ldots, w, 1, 2, \ldots, \pi - 1)$ **do**
7: Call L2RS.Lift($\mathbf{A}, \mathbf{a}_i$) to obtain $\mathbf{A}_{2q,i} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i + q) \in \mathcal{R}_{2q}^{1\times m}$.
8: Let $\mathbf{t}_i = (t_{i,1}, \ldots, t_{i,m})^T$, where $t_{i,j} \hookleftarrow D_\sigma^n$, for $1 \le j \le m$.
9: Compute $\mathbf{c}_{i+1} = H_1\Big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i, \mathbf{H}_{2q} \cdot \mathbf{t}_i + q \cdot \mathbf{c}_i\Big)$.
10: Choose $b \hookleftarrow \{0, 1\}$.
11: Choose $\mathbf{t}_\pi \hookleftarrow D_\sigma^m$
12: **Continue** with probability $\dfrac{1}{M}$ otherwise **Restart**.
13: **return** $\sigma_L(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$.

FORGERY 1. Let's consider the situation that $\mathbf{c}_{j+1}$ is the result after using $\mathcal{F}$ which is $\mathbf{c}_{j+1} = H_1(L, \mathbf{H}_{2q}, \mu', \mathbf{A}_{2q} \cdot \mathbf{t}' + q \cdot \mathbf{c}_j, \mathbf{H}_{2q} \cdot \mathbf{t}' + q \cdot \mathbf{c}_j)$. Then by comparing this with a legitimate signature, we have:

$$H_1\big(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q}\cdot\mathbf{t}+q\cdot\mathbf{c}_j, \mathbf{H}_{2q}\cdot\mathbf{t}+q\cdot\mathbf{c}_j\big) = H_1\big(L, \mathbf{H}_{2q}, \mu', \mathbf{A}_{2q}\cdot\mathbf{t}'+q\cdot\mathbf{c}_j, \mathbf{H}_{2q}\cdot\mathbf{t}'+q\cdot\mathbf{c}_j\big)$$

$\mathcal{F}$ will find a preimage of $\mathbf{c}_j$ if $\mu \neq \mu'$ or $\mathbf{A}_{2q} \cdot \mathbf{t} + q \cdot \mathbf{c}_j \neq \mathbf{A}_{2q} \cdot \mathbf{t}' + q \cdot \mathbf{c}_j$ or $\mathbf{H}_{2q} \cdot \mathbf{t} + q \cdot \mathbf{c}_j \neq \mathbf{H}_{2q} \cdot \mathbf{t}' + q \cdot \mathbf{c}_j$. Then, we have with overwhelming probability that $\mu = \mu'$ and $\mathbf{A}_{2q} \cdot \mathbf{t} + q \cdot \mathbf{c}_j = \mathbf{A}_{2q} \cdot \mathbf{t}' + q \cdot \mathbf{c}_j$ and $\mathbf{H}_{2q} \cdot \mathbf{t} + q \cdot \mathbf{c}_j = \mathbf{H}_{2q} \cdot \mathbf{t}' + q \cdot \mathbf{c}_j$. These equalities will result in: $\mathbf{A}_{2q}(\mathbf{t} - \mathbf{t}') = 0 \bmod 2q$ and $\mathbf{H}_{2q}(\mathbf{t} - \mathbf{t}') = 0 \bmod 2q$. We assume that both $\mathbf{t}$ and $\mathbf{t}'$ are different and they met the L2RS.SigVer conditions, so it yields $\mathbf{t} - \mathbf{t}' \neq 0 \bmod q$, and $\|\mathbf{t} - \mathbf{t}'\| \le 2B_2$.

FORGERY 2. In this scenario, we assume that the L2RS scheme can be forged by an attacker $\mathcal{F}$ as it was presented in the FORGERY 1 and obtain $\mathbf{c}_j$, then another attacker can generate $(\mathbf{c}'_j, \ldots, \mathbf{c}'_w) \hookleftarrow \mathcal{S}_{n,\kappa}$ by replaying the first attack and using same message μ. We use the forking lemma [BN06] to show the probability of

$\mathbf{c}_j = \mathbf{c}'_j$ and the forger uses an oracle response $\mathbf{c}'_j$ is at least:

$$\left(\delta - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \cdot \left(\frac{\delta - \frac{1}{|\mathcal{S}_{n,\kappa}|}}{y} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \tag{4.1}$$

Therefore, with the probability (4.1), $\mathcal{F}$ creates a signature $\sigma_L(\mu) = (\mathbf{c}'_1, \mathbf{t}'_1, \ldots, \mathbf{t}'_w, \mathbf{h})$ where $\mathbf{A}_{2q} \cdot \mathbf{t} + q \cdot \mathbf{c}_j = \mathbf{A}_{2q} \cdot \mathbf{t}' + q \cdot \mathbf{c}'_j$ and $\mathbf{H}_{2q} \cdot \mathbf{t} + q \cdot \mathbf{c}_j = \mathbf{H}_{2q} \cdot \mathbf{t}' + q \cdot \mathbf{c}'_j$. We now obtained: $\mathbf{A}_{2q} \cdot (\mathbf{t} - \mathbf{t}') = \mathbf{q}(\mathbf{c}_j - \mathbf{c}'_j) \bmod 2q$ and $\mathbf{H}_{2q} \cdot (\mathbf{t} - \mathbf{t}') = \mathbf{q}(\mathbf{c}_j - \mathbf{c}'_j) \bmod 2q$. Since $\mathbf{c}_j - \mathbf{c}'_j \neq 0 \bmod 2$, so in both equations, we have $\mathbf{t} - \mathbf{t}' \neq 0 \bmod 2q$ where $\|\mathbf{t} - \mathbf{t}'\|_\infty < q/2$. By applying this reduction, we find a small non-zero vector $\mathbf{v} = \mathbf{t} - \mathbf{t}' \neq 0 \bmod q$. This $\mathbf{v}$ will compute $\mathbf{A}_{2q} \cdot \mathbf{v} = 0 \bmod q$ with $\|\mathbf{v}\| \leq 2B_2$. Since $\mathbf{A}_{2q} \bmod q = 2(\mathbf{A}, -\mathbf{a}) \bmod q$, we have $2(\mathbf{A}, -\mathbf{a})\mathbf{v} = 0 \bmod q$, this implies that $(\mathbf{A}, -\mathbf{a})\mathbf{v} = 0 \bmod q$, since q is odd. Notice that L2RS.Hydrid-2 shown in Algorithm 6 no longer uses the private-key $\mathbf{S}_\pi$, except for generating $\mathbf{A}_{2q,\pi}$ to obtain the final $\mathcal{R}\text{-}\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ solution. We modified the L2RS.KeyGen algorithm with the L2RS.Hydrid-3 game shown in Algorithm 7, where the public-key $\mathbf{a}$ is uniformly and randomly taken: $\mathbf{a} \hookleftarrow \mathcal{R}_q$. By the argument of the Leftover Hash Lemma (LHL) - **Lemma 3.5** and our assumption that $\sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ is negligible in n. The probability of success of an attacker in L2RS.Hydrid-3 differs by a negligible amount from the success probability in L2RS.KeyGen and is thus non-negligible. Therefore, this vector $\mathbf{v}$ will be a solution to the $\mathcal{R}\text{-}\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ problem, where $\beta = 2B_2$, with non-negligible probability and with respect to $(\mathbf{A}, -\mathbf{a})$ over $\mathcal{R}_q$.

Algorithm 7 Key pair generation of L2RS Hybrid 3 $(\mathbf{a}, \mathbf{S})$

Input: Pub-Param: $\mathbf{A}$.
Output: $(\mathbf{a}, \mathbf{S})$, being the public-key and the private-key, respectively.

1: **procedure** L2RS.HYBRID-3($\mathbf{A}$)
2: Let $\mathbf{S}^T = (\mathbf{s}_1, \ldots, \mathbf{s}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\mathbf{s}_i \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
3: Choose $\mathbf{a} \hookleftarrow \mathcal{R}_q$
4: **return** $(\mathbf{a}, \mathbf{S})$.

□

Theorem 4.9 (Anonymity)**.** *Suppose* $\sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ *is negligible in* n *with an attack against the unconditional anonymity that makes* h *queries to the random oracle* H_1*, where* h*,* w *are polynomial in* n*, then the* L2RS *scheme is unconditionally secure for anonymity as defined in Def. 4.2.*

Proof. We prove the anonymity of this scheme using the sequence-of-games approach [Sho04] where we make changes between successive games. In doing so, we use the "*transition based on indistinguishability*". We can start this analysis by:

Game 0: Suppose that an attacker $\mathcal{A}$ is given the list of pk's $L = \{\mathbf{a}_0, \mathbf{a}_1\}$, the signature $\sigma_L(\mu)$, message μ, and the random oracle models (H_1 and H_2). The key generation algorithm creates the pair of users' keys in this ring signature: Private-Keys $\hookleftarrow (\mathbf{S}_0, \mathbf{S}_1)$ and the Public-Keys $\hookleftarrow (\mathbf{a}_0, \mathbf{a}_1)$; a user b is chosen uniformly at random from the list $L = \{\mathbf{a}_0, \mathbf{a}_1\}$, then the signature $\sigma_L(\mu) = \mathsf{L2RS.SigGen}(\mathbf{S}_b, \mu, L, \mathsf{Pub\text{-}Param})$ is generated. So in **Game 0**, a PPT adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$; thus in the event **Game 0**, $\mathcal{A}$ succeeds in breaking ambiguity **Game 0**$(b = b')$ if $\Pr[\textbf{Game 0}] \leq \frac{1}{2} + non - negl(\lambda)$.

Game 1: Changes in this game are made to the user π in the second part of the linkability tag $\mathbf{h} = (\mathbf{H} \cdot \mathbf{S}) \in \mathcal{R}_q$, in signature of user π, and public-key $\mathbf{a} = (\mathbf{A} \cdot \mathbf{S}) \in \mathcal{R}_q$ in the L2RS.KeyGen algorithm. The $\mathbf{h}$ and $\mathbf{a}_1$ are now randomly chosen from $\mathcal{R}_q$. We claim that $|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq \epsilon_{LHL_{G1}}$.

Where $\epsilon_{LHL_{G1}}$ is the advantage of some efficient algorithm which is negligible. In both cases $\mathbf{h} = (\mathbf{H} \cdot \mathbf{S}) \in \mathcal{R}_q$ and $\mathbf{a} = (\mathbf{A} \cdot \mathbf{S}) \in \mathcal{R}_q$, we know that $\mathbf{H}$ and $\mathbf{A}$ are uniform and $\mathbf{S}$ is chosen small and with coefficients in $(-2^\gamma, 2^\gamma)$. When $\mathbf{S}$ is multiplied by $\mathbf{H}$ and $\mathbf{A}$ respectively, it gives $\mathbf{h}$ and $\mathbf{a}$ that are close to uniform over $\mathcal{R}_q$. By applying the Leftover Hash Lemma (LHL) - **Lemma 3.5**, the statistical distance between the distribution of ($\mathbf{h}$ mod q and $\mathbf{a}$ mod q) and the uniform distribution on $\mathcal{R}_q \times \mathcal{R}_q$ is at most $n \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$. We conclude that in **Game 1**:

$$|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq n \cdot \tfrac{1}{2} \cdot \sqrt{\tfrac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}. \tag{4.2}$$

Game 2: This time a change is made in the second part of the remaining public-keys $\mathbf{a}_i$ $(1 \leq i \leq w,\ i \neq \pi)$ which are in the ring signature list L. They are now randomly chosen as $\mathbf{a}_i \hookleftarrow \mathcal{R}_q$. It turns out that $|\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 2}]| \leq \epsilon_{LHL_{G2}}$.

Where $\epsilon_{LHL_{G2}}$ is the advantage of some efficient algorithm which is negligible. We consider that for $(i = 1$ to w where $i \neq \pi)$, we know that $\mathbf{a}_i = (\mathbf{A} \cdot \mathbf{S}_i \bmod q)$ are uniform and all $\mathbf{S}_i$'s are chosen small with coefficients in $(-2^\gamma, 2^\gamma)$. When the $\mathbf{S}_i$'s are multiplied by $\mathbf{A}_i$'s, it gives $(\mathbf{a}_i \bmod q)$'s that are close to uniform over $\mathcal{R}_q$. By applying the Leftover Hash Lemma (LHL) - **Lemma 3.5**, the statistical distance between the distribution of the $(\mathbf{A} \cdot \mathbf{S}_i \bmod q)$'s and the uniform distribution on $\mathcal{R}_q \times \mathcal{R}_q$ is at most $n \cdot \frac{1}{2} \cdot \sqrt{\frac{q^n}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \cdot (w-1)$. So in **Game 2**, we conclude that:

$$|\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 2}]| \leq n \cdot \frac{1}{2} \cdot \sqrt{\frac{q^n}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \cdot (w-1). \tag{4.3}$$

Game 3: At this time, we make a change in $\mathbf{c}_{\pi+1}$. Instead of programming the oracle as $H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,i} \cdot \mathbf{u}, \mathbf{H}_{2q} \cdot \mathbf{u})$, it is now randomly chosen $\mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}$. We have that $|\Pr[\textbf{Game 2}] - \Pr[\textbf{Game 3}]| \leq \epsilon_{G3}$ where ϵ_{G3} is the advantage of some efficient algorithm which is negligible. This scenario outputs a signature $\sigma_L(\mu_1) = (\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h})$ and programs the oracle as $H_1(L, \mathbf{H}_{2q}, \mu, \mathbf{A}_{2q,\pi} \cdot \mathbf{t}_\pi + q \cdot \mathbf{c}_\pi, \mathbf{H}_{2q} \cdot \mathbf{t}_\pi + q \cdot \mathbf{c}_\pi) = \mathbf{c}_{\pi+1}$. Then, the adversary $\mathcal{A}$ makes h queries to H_1; so the distinguishing advantage of the signing algorithm and the one in **Game 2** is at most $h \cdot 2^{-n+1}$. We conclude that in **Game 3**:

$$|\Pr[\textbf{Game 2}] - \Pr[\textbf{Game 3}]| \leq h \cdot 2^{-n+1}. \tag{4.4}$$

Game 4: In this game a change is made in $\mathbf{t}_\pi$. Namely, instead of computing it as $\mathbf{u} + \mathbf{S}_{2q,\pi} \cdot \mathbf{c}_\pi \cdot (-1)^{bit}$, it is now directly chosen from the Gaussian distribution D_σ^n. It is argued that $|\Pr[\textbf{Game 3}] - \Pr[\textbf{Game 4}]| \leq \epsilon_{RS_{G4}}$.

Where $\epsilon_{RS_{G4}}$ is the advantage of some efficient algorithm which is negligible. In previous Games, $\mathbf{t}_\pi$ is computed using rejection sampling - **Lemma 3.8**, thus it is always sample from the Gaussian distribution D_σ^n. In this Game, however, $\mathbf{t}_\pi$

is directly chosen from D_{σ}^{n}, this means that the advantage $\epsilon_{RS_{G4}}$ will be zero as in both **Game 3** and **Game 4**, $\mathbf{t}_{\pi}$ is having same distribution. In **Game 4**, we have:

$$|\Pr[\textbf{Game 3}] - \Pr[\textbf{Game 4}]| = 0. \tag{4.5}$$

Game 5: Finally, in the **Game 5**, a change is made in the index π. Namely, instead of choosing $\pi + 1$, it will be randomly chosen $(1, \ldots, w)$. We claim that $|\Pr[\textbf{Game 4}] - \Pr[\textbf{Game 5}]| \leq \epsilon_{G5}$ where ϵ_{G5} is the advantage of some efficient algorithm which is negligible. In this **Game 5**, we consider that when π is replaced by a fixed d, it might produce some collisions with previous queries to the oracle H_1; saying this, the adversary $\mathcal{A}$ may make h queries to H_1; therefore, the distinguishing advantage of the signing algorithm between **Game 4** and this **Game 5** is at most $h \cdot 2^{-n+1} \cdot w$. Finally, in **Game 5** we have:

$$|\Pr[\textbf{Game 4}] - \Pr[\textbf{Game 5}]| \leq h \cdot 2^{-n+1} \cdot w. \tag{4.6}$$

We also conclude that in **Game 5**, the adversary's view is statistical independent of π, thus $\Pr[\textbf{Game 5}] = \frac{1}{w}$.

Combining the probabilities of the above games (4.2), (4.3), (4.4), (4.5), and (4.6), we obtain:

$$|\Pr[\textbf{Game 5}] - \Pr[\textbf{Game 0}]| \leq |\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 0}]| + |\Pr[\textbf{Game 2}] - \Pr[\textbf{Game 1}]| + |\Pr[\textbf{Game 3}] - \Pr[\textbf{Game 2}]| + |\Pr[\textbf{Game 4}] - \Pr[\textbf{Game 3}]| + |\Pr[\textbf{Game 5}] - \Pr[\textbf{Game 4}]|.$$

By replacing the resulting probabilities, we have:

$$|\Pr[\textbf{Game 5}] - \Pr[\textbf{Game 0}]| \leq \frac{1}{w} - \frac{1}{2} + \epsilon, \tag{4.7}$$

which means that $|\Pr[\textbf{Game 5}] - \Pr[\textbf{Game 0}]| \leq \epsilon$, which itself is smaller than

$$\frac{n \cdot (w-1)}{2} \cdot \left(\sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} + \sqrt{\frac{q^{n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \right) + h \cdot 2^{-n+1} \cdot (1+w).$$

We notice that since h and w are polynomial in n, we get $h \cdot 2^{-n+1} \cdot (1+w)$ is negligible in n. In addition, we can say that $\left(\sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} + \sqrt{\frac{q^{n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}\right) \leq 2 \cdot \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$, which is negligible by the assumption that $\sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ is also negligible. Hence we conclude that ϵ is negligible, meaning that $\Pr[\textbf{Game 0}] \leq \frac{1}{2} + \epsilon$. □

Theorem 4.10 (Linkability)**.** *The L2RS scheme is linkable in the random oracle model if the* $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ *problem is hard.*

Proof. We construct the algorithm $\mathcal{B}$ for the $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ problem. This algorithm runs the linkability attack game (Def. 4.3) as follows:

1. $\mathcal{B}$ generates using the L2RS.KeyGen algorithm all private-keys $\mathbf{S}_i$'s with the corresponding public-keys $\mathbf{a}_i$'s, then $\mathcal{B}$ gives $\mathbf{S}_\pi$ to the attacker $\mathcal{A}$ as a response to the attacker's $\mathcal{CO}$ query.

2. $\mathcal{A}$ outputs two signatures $\sigma_L(\mu_1)$ and $\sigma'_{L'}(\mu')$ along with their corresponding lists L and L' such that both signatures are successfully verified by L2RS.SigVer, but the linkability tags are different $\mathbf{h}_{\mu_1} \neq \mathbf{h}_{\mu'}$.

3. $\mathcal{B}$ computes $\mathbf{h}_{\mu_\pi} = \mathbf{H} \cdot \mathbf{S}_\pi \bmod q$, where π is the true signer's π linkability tag. This $\mathbf{h}_{\mu_\pi}$ tag can then be compared with the linkability tags $\mathbf{h}_{\mu_1}$and $\mathbf{h}_{\mu'}$, output by $\mathcal{A}$, in step 2, and one of them will be different.

4. Without loss of generality, suppose $\mathbf{h}_{\mu_1} \neq \mathbf{h}_{\mu_\pi} \bmod q$. Using the forking lemma [BN06], $\mathcal{B}$ rewinds the attacker $\mathcal{A}$ to the H_1 query corresponding to the L2RS.SigVer of the signature $\sigma_L(\mu_1)$. $\mathcal{B}$ reruns $\mathcal{A}$ with a different response of H_1 and ultimately gets another signature: $\sigma_L(\mu_2) = (\mathbf{c}_{1,\mu_2}, \mathbf{t}_{1,\mu_2}, \ldots, \mathbf{t}_{w,\mu_2}, \mathbf{h}_{\mu_2})$. This second signature is used to extract a solution to the $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ problem, in case the $\mathcal{A}$ finds an efficient way to unlink these signatures, as shown in step 7.

5. The adversary $\mathcal{A}$ matches the challenge message of both signatures where $\mathbf{H}_{2q,\mu_1}$ and $\mathbf{A}_{2q,w,\mu_1}$ are kept. Thus we have:

(a) $\mathbf{A}_{2q,w,\mu_1} \cdot \mathbf{t}_{w,\mu_1} + q \cdot \mathbf{c}_{w,\mu_1} = \mathbf{A}_{2q,w,\mu_1} \cdot \mathbf{t}_{w,\mu_2} + q \cdot \mathbf{c}_{w,\mu_2}$,

(b) $\mathbf{H}_{2q,\mu_1} \cdot \mathbf{t}_{w,\mu_1} + q \cdot \mathbf{c}_{w,\mu_1} = \mathbf{H}_{2q,\mu_1} \cdot \mathbf{t}_{w,\mu_2} + q \cdot \mathbf{c}_{w,\mu_2}$.

These expressions can be represented as:

(a) $\mathbf{A}_{2q,w,\mu_1} \cdot (\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}) = q \cdot (\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1})$,

(b) $\mathbf{H}_{2q,\mu_1} \cdot (\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}) = q \cdot (\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1})$.

Reducing them mod$q$ we have (if $(\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1}) \neq 0 \bmod 2$):

(a) $\mathbf{A}_{2q,w,\mu_1} \cdot (\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}) = 0 \bmod q$,

(b) $\mathbf{H}_{2q,\mu_1} \cdot (\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}) = 0 \bmod q$.

We denote by $\mathbf{t}'_{w,\mu_1}$, the first $(m-1)$ ring elements in $\mathbf{t}_{w,\mu_1}$ and by $\mathbf{t}''_{w,\mu_1}$the m-th ring element in $\mathbf{t}_{w,\mu_1}$, i.e. $\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2} = \begin{pmatrix} \mathbf{t}'_{w,\mu_1} - \mathbf{t}'_{w,\mu_2} \\ \mathbf{t}''_{w,\mu_1} - \mathbf{t}''_{w,\mu_2} \end{pmatrix} \in \mathcal{R}_q^m$, and using the public-key and linkability parts, we have:

(a) $2 \cdot \mathbf{A} \cdot (\mathbf{t}'_{w,\mu_1} - \mathbf{t}'_{w,\mu_2}) = -2 \cdot \mathbf{a} \cdot (\mathbf{t}''_{w,\mu_1} - \mathbf{t}''_{w,\mu_2})$,

(b) $2 \cdot \mathbf{H} \cdot (\mathbf{t}'_{w,\mu_1} - \mathbf{t}'_{w,\mu_2}) = -2 \cdot \mathbf{h}_{\mu_1} \cdot (\mathbf{t}''_{w,\mu_1} - \mathbf{t}''_{w,\mu_2})$, where $\mathbf{h}_{\mu_1} \triangleq \mathbf{H} \cdot \mathbf{S}_\pi \in \mathcal{R}_q$.

6. We let $\bar{\mathbf{S}} = \frac{(\mathbf{t}'_{w,\mu_1} - \mathbf{t}'_{w,\mu_2})}{(\mathbf{t}''_{w,\mu_1} - \mathbf{t}''_{w,\mu_2})} \bmod q$ where $(\mathbf{t}''_{w,\mu_1} - \mathbf{t}''_{w,\mu_2}) \neq 0 \bmod q$. We distinguish two cases:

   (a) If $\bar{\mathbf{S}} \neq \mathbf{S}_\pi \bmod q$, since we have $\mathbf{A} \cdot \bar{\mathbf{S}} = \mathbf{A} \cdot \mathbf{S}_\pi = \mathbf{a} \bmod q$, then $(\bar{\mathbf{S}} - \mathbf{S})$ is a small non-zero vector $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ solution for $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$.

   (b) If $\bar{\mathbf{S}} = \mathbf{S}_\pi \bmod q$, then $\mathbf{h}_{\mu_1} = \mathbf{H} \cdot \bar{\mathbf{S}} \bmod q = \mathbf{H} \cdot \mathbf{S}_\pi \bmod q$. The target is to show that $\mathbf{h}_{\mu_1} = \mathbf{h}_{\mu_\pi} \bmod 2$ and $\mathbf{h}_{\mu_1} = \mathbf{h}_{\mu_\pi} \bmod q$. If so, then we have $\mathbf{h}_{\mu_1} = \mathbf{h}_{\mu_\pi} \bmod 2q$, which is a contradiction with our assumption at step 4 of this proof. We now prove the first target:

$$\mathbf{h}_{\mu_1} = -2 \cdot \mathbf{h}'_{\mu_1} + q = 1 \bmod 2 = -2 \cdot \mathbf{H} \cdot \mathbf{S}_\pi + q = \mathbf{h}_{\mu_\pi},$$

   where the first and the last equalities follow from definition of $\mathbf{h}$ in second line of Algorithm 2. To show the second target, we have

$$\mathbf{h}_{\mu_1} = -2 \cdot \mathbf{h}_{\mu_1} + q = -2 \cdot \mathbf{h}_{\mu_1} \bmod q$$

$$= -2 \cdot \mathbf{H} \cdot \bar{\mathbf{S}} \bmod q = -2 \cdot \mathbf{H} \cdot \mathbf{S}_\pi \bmod q = \mathbf{h}_{\mu_\pi},$$

where the first and the last equalities follow from definition of $\mathbf{h}$ in second line of Algorithm 2 and the middle equality is true based on the argument at the beginning of step (6.b).

7. Since $(\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1}) \neq 0 \bmod 2$, we have $(\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}) \neq 0 \bmod 2q$. In addition, we know that $\|\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}\|_\infty < q/2$, which implies that $(\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}) \neq 0 \bmod q$. Ultimately, we have $\mathbf{A} \cdot (\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}) = 0 \bmod q$ and $\|(\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2}) \bmod q\| \leq 2B_2$. Therefore, this small non-zero vector $(\mathbf{t}_{w,\mu_1} - \mathbf{t}_{w,\mu_2})$ is the output of the algorithm $\mathcal{B}$, and this vector is a solution to the $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ problem with $\beta = 2B_2$ for $\mathbf{a} \in \mathcal{R}_q$.

□

Theorem 4.11 (Non-Slanderability)**.** *For any linkable ring signature, if it satisfies unforgeability and linkability, then it satisfies non-slanderability.*

Proof. Let's suppose there is a non-slanderability adversary $\mathcal{A}_{Sland}$ who is given $\mathbf{pk}_i, \mathbf{sk}_i, i \neq \pi,$ and $i \in \{1, \dots w\}$, and he produces a valid signature $\sigma'_L(\mu)$ with linkability tag $\mathbf{h}_{\sigma'_L(\mu)}$ which is equal to $\mathbf{h}_{\sigma_L(\mu)}$, $\sigma_L(\mu)$ being the legitimate signature generated with respect to $\mathbf{sk}_\pi$. This means that $\mathcal{A}_{Sland}$ can create a signature with the linkability tag $\mathbf{h}_{\sigma_L(\mu)}$ without knowing $\mathbf{sk}_\pi$. The adversary can also compute a valid $\sigma''_L(\mu)$ with $\mathbf{sk}_i$, $i \neq \pi,$ and $i \in \{1, \dots w\}$ for which $\mathbf{h}_{\sigma''_L(\mu)} \neq \mathbf{h}_{\sigma'_L(\mu)}$. We give $(\sigma''_L(\mu), \sigma'_L(\mu))$ to the forger, which can turn it to an $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ solution. In particular, it will be computationally secure when two valid signatures created by different users are unlinked using the L2RS algorithms. An adversary $\mathcal{A}$ will break these properties with negligible probability as demonstrated in Theorems (4.8 and 4.10), and with these probabilities the $\mathcal{A}$ will find a $\mathcal{R}$-$\mathbf{SIS}^{\mathcal{K}}_{q,m,\beta}$ solution. Therefore, non-slanderability is implied by the definitions of the unforgeability (Def. 4.1) and linkability (Def. 4.3). □

Corollary 4.12 (Non-Slanderability)**.** *The L2RS scheme is non-slanderable under the assumptions of* ***Theorem 4.8*** *and* ***Theorem 4.10****.*

4.5 Lattice RingCT v1.0 Protocol (LRCT)

This RCT protocol described in [SALY17] can be constructed based on the LRS scheme. Its algorithms are defined as follows:

- RCT.Setup: this PPT algorithm uses L2RS.Setup where it takes the security parameter λ and outputs the public parameters Pub-Params.
- RCT.KeyGen: this PPT algorithm uses the L2RS.KeyGen to produce a pair of keys, the public-key pk and the private-key sk.
- RCT.Mint: a PPT algorithm that generates new coins. This algorithm receives the public-key pk and the amount \$, it outputs a coin **cn** along with its associated coin-key **ck**.
- RCT.Spend: a PPT algorithm that receives the Pub-Params, a set of input wallets IW_i with $1 \leq i \leq w$, a user π's input wallet IW_π along with its set of secret keys K_π, a set of output addresses OA, some transaction string $\mu \in \{0,1\}^*$ and the set of output wallets OW. Then, this algorithm outputs the transaction $TX = (\mu, IW, OW)$, uses L2RS.SigGen to generate and output the signature $\sigma(\mu)$, and finally output a set of transaction/serial numbers TN, which is used to prevent the double spending.
- RCT.Verify: a deterministic PPT algorithm that takes as input the Pub-Params, the signature $\sigma(\mu)$, the TX, and the TN, it then uses L2RS.SigVer and outputs either: Accept (1) or Reject (0).

4.5.1 LRCT construction

The Lattice RingCT LRCT scheme requires a homomorphic commitment (Com) as an additional primitive. It is a cryptographic technique used to provide confidential transactions, in particular cryptocurrencies [Noe15]. This primitive allows one party to commit to a chosen value while keeping it secret to other parties, then this committed value can be revealed later. This model is restricted to have a

Single-Input Single-Output (SISO) wallets, meaning that an Input Wallet will be spent into an Output Wallet (OW) only. We use the structure of the $\mathsf{L2RS.KeyGen}$ scheme Algorithm 1, where the public parameter $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$ is used to commit to a scalar message $\mathrm{m} \in \mathsf{Dom_m} \subseteq \mathcal{R}_q$ with $\mathsf{Dom_m} = [0, \ldots, 2^{\ell-1}] \subseteq \mathbb{Z}$. This property is defined as $\mathsf{Com}_{\mathbf{A}}(\mathrm{m}, \mathsf{sk}) = \mathbf{A} \cdot \mathsf{sk} + \mathrm{m} \in \mathcal{R}_q$, where the randomness $\mathsf{sk} \in \mathsf{Dom_{sk}} \subseteq \mathcal{R}_q^{(m-1)\times 1}$. The properties of the homomorphic operations are also defined as:

$$\begin{aligned}\mathsf{Com}_{\mathbf{A}}(\mathrm{m}_1, \mathsf{sk}) \boxed{\pm} \mathsf{Com}_{\mathbf{A}}(\mathrm{m}_2, \mathsf{sk}') &\triangleq \mathsf{Com}_{\mathbf{A}}(\mathrm{m}_1, \mathsf{sk}) \pm \mathsf{Com}_{\mathbf{A}}(\mathrm{m}_2, \mathsf{sk}') \bmod q \\ &\triangleq \mathsf{Com}_{\mathbf{A}}(\mathrm{m}_1 \pm \mathrm{m}_2, \mathsf{sk} \pm \mathsf{sk}') \bmod q, \quad (4.8)\end{aligned}$$

where $\mathrm{m}_1, \mathrm{m}_2 \in \mathcal{R}_q$; and $\mathsf{sk}, \mathsf{sk}' \in \mathcal{R}_q^{(m-1)\times 1}$. The integers $\mathrm{m}_1, \mathrm{m}_2 \in \mathbb{Z}$ are encoded in binary as coefficient vectors $\mathbf{m}_1 = (\mathrm{m}_{1,0}, \ldots, \mathrm{m}_{1,\ell-1}, 0, \ldots, 0) \in \{0,1\}^n$ and $\mathbf{m}_2 = (\mathrm{m}_{2,0}, \ldots, \mathrm{m}_{2,\ell-1}, 0, \ldots, 0) \in \{0,1\}^n$ where $\mathbf{m}_j = \sum_{i=0}^{\ell-1}(\mathrm{m}_{j,i} \cdot 2^i)$, with $\mathrm{m}_{j,i} \in \{0,1\}$ and $j \in \{0,1\}$, and $\mathbf{m} = \mathbf{m}_1 - \mathbf{m}_2 = (\mathrm{m}_{1,0} - \mathrm{m}_{2,0}, \ldots, \mathrm{m}_{1,\ell-1} - \mathrm{m}_{2,\ell-1}, 0, \ldots, 0) \in \{-1, 0, 1\}^n$. The difference between these vectors is zero $\in \mathcal{R}_q$ if $\mathbf{m}_1 = \mathbf{m}_2$, non-zero otherwise. This means that the commitment is performed to each bit.

The SISO scheme using the protocol Lattice RingCT v1.0, LRCT = ($\mathsf{LRCT.Setup}$, $\mathsf{LRCT.KeyGen}$, $\mathsf{LRCT.Mint}$, $\mathsf{LRCT.Spend}$, $\mathsf{LRCT.Verify}$) works as follows.

1. $(\mathsf{Pub\text{-}Params}) \leftarrow \mathsf{LRCT.Setup}(\lambda)$: On input security parameter λ, this algorithm calls $\mathsf{L2RS.Setup}$ and outputs the public parameters, $(\mathbf{A}, \mathbf{H}) \in \mathcal{R}_q^{1\times(m-1)} \times \mathcal{R}_q^{1\times(m-1)}$.

2. $(\mathbf{a}, \mathbf{S}) \leftarrow \mathsf{LRCT.KeyGen}(\mathbf{A})$: Given the public parameter $\mathbf{A}$, it outputs a pair of keys, the public-key pk: $\mathbf{a} \in \mathcal{R}_q$ and the private-key sk: $\mathbf{S} \in \mathcal{R}_q^{(m-1)\times 1}$. Then we define the commitment of the $\mathsf{LRCT.KeyGen}$ as $\mathbf{a} = \mathbf{A}\cdot\mathbf{S}+0 \bmod q \in \mathcal{R}_q = \mathsf{Com}_{\mathbf{A}}\big(0, \mathbf{S}\big)$.

3. $(\mathbf{cn}, \mathbf{ck}) \leftarrow \mathsf{LRCT.Mint}(\mathbf{a}, \$)$: This is illustrated in Algorithm 8. It receives a valid one-time address $\mathbf{a}$ as well as an input amount $\$ \in \left[0, \ldots, 2^{\ell_\$} - 1\right]$. Then, to create a coin $\mathbf{cn}$, this algorithm chooses a coin-key $\mathbf{ck} \in \mathsf{Dom}_{\mathbf{S}}$. Then, the commitment of Mint is computed as $\mathbf{cn} = \mathbf{A} \cdot \mathbf{ck} + \$ \bmod q \in \mathcal{R}_q = \mathsf{Com}_{\mathbf{A}}(\$, \mathbf{ck})$. This algorithm returns $(\mathbf{cn}, \mathbf{ck})$.

Algorithm 8 LRCT.Mint

Input: $\left(\mathbf{a} \in \mathcal{R}_q, \$ \in \mathbb{B}_w^n\right)$, being the Public-key, the amount and the public parameter, respectively.

Output: $(\mathbf{cn}, \mathbf{ck})$, where they are the coin and the coin key, respectively.

1: **procedure** LRCT.MINT($\mathbf{a}, \$$)
2: Let $\mathbf{ck}^T = (\mathbf{ck}_1, \ldots, \mathbf{ck}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$with $\mathbf{ck}_i \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
3: $\mathbf{cn} = \mathbf{A} \cdot \mathbf{ck} + \$ \bmod q \in \mathcal{R}_q = \mathsf{Com}_{\mathbf{A}}(\$, \mathbf{ck})$, where $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$ is the public parameter and a component of $\mathbf{a}$.
4: **return** $(\mathbf{cn}, \mathbf{ck})$

4. $(TX, \sigma_{L'}(\mu), TN) \leftarrow \mathsf{LRCT.Spend}(\mu, IW, IW_\pi, K_\pi, OA, \mathsf{Pub\text{-}Params})$: Described in Algorithm 9, this follows the steps:

   (a) The IW and IW_π were properly constructed. In this SISO protocol a user π spends one IW into one OW, this means that the π's number of wallets to be spent $N_{in} = 1$.

   (b) We denote the π's input wallet to be spent as $IW_\pi^{(1)} = \left\{\mathbf{a}_{(in),\pi}^{(1)}, \mathbf{cn}_{(in),\pi}^{(1)}\right\} \in \mathcal{R}_q \times \mathcal{R}_q$, with the corresponding private part $K_\pi^{(1)} = \left\{\mathbf{S}_{(in),\pi}^{(1)}, \mathbf{ck}_{(in),\pi}^{(1)}\right\} \in \mathcal{R}_q^{1\times(m-1)} \times \mathcal{R}_q^{1\times(m-1)}$, and the one $(N_{out} = 1)$ output valid address $OA = \mathbf{a}_{(out)}^{(1)}$ where π intends to spend his money. Then, π selects $\$_{(out)}^{(1)} \in \left[0, \ldots, 2^{\ell_\$} - 1\right]$, such balances satisfy: $\$_{(in),\pi}^{(1)} = \$_{(out)}^{(1)}$. The $\mathsf{LRCT.Mint}\left(\mathbf{a}_{(out)}^{(1)}, \$_{(out)}^{(1)}\right)$ is called to obtain $\left(\mathbf{cn}_{(out)}^{(1)}, \mathbf{ck}_{(out)}^{(1)}\right)$, this defines an output wallet as $OW = OW^{(1)} = \left\{\mathbf{a}_{(out)}^{(1)}, \mathbf{cn}_{(out)}^{(1)}\right\}$. Then, the coin-key $\mathbf{ck}_{(out)}^{(1)}$ and $\$_{(out)}$ are securely sent to the user holding the output valid address $OA = \mathbf{a}_{(out)}^{(1)}$.

   (c) π selects $w - 1$ (or the $\mathsf{L2RS}$ list L) of input wallets $IW = IW_i^{(1)} = \left\{\mathbf{a}_{(in),i}^{(1)}, \mathbf{cn}_{(in),i}^{(1)}\right\}_{i\in[w]}$, to anonymously spend $IW_\pi^{(1)}$, with w being the ring signature size.

(d) A new list is constructed as $L' = \{\widehat{\mathbf{a}}^{(1)}_{(in),i}\}_{i\in[w]} \in \mathcal{R}_q$, where $\widehat{\mathbf{a}}^{(1)}_{(in),i}$ is the homomorphic commitment with randomness $\widehat{\mathbf{S}}^{(1)}_{(in),i}$ that we define as follows:

- $\widehat{\mathbf{a}}^{(1)}_{(in),i} = \mathbf{a}^{(1)}_{(in),i} + \mathbf{cn}^{(1)}_{(in),i} - \mathbf{cn}^{(1)}_{(out)} = \mathsf{Com}_{\mathbf{A}}\big(\$^{(1)}_{(in),i} - \$^{(1)}_{(out)}, \widehat{\mathbf{S}}^{(1)}_{(in),i}\big)$, such that for the user's π this is a zero commitment: $\mathsf{Com}_{\mathbf{A}}\big(0, \widehat{\mathbf{S}}^{(1)}_{(in),\pi}\big)$.
- $\widehat{\mathbf{S}}^{(1)}_{(in),i} = \big(\mathbf{S}^{(1)}_{(in),i} + \mathbf{ck}^{(1)}_{(in),i} - \mathbf{ck}^{(1)}_{(out)}\big) \in \mathcal{R}_q$.

(e) To create the proof of knowledge, we use the π's private-key: $\widehat{\mathbf{S}}^{(1)}_{(in),\pi}$, the list L' and a transaction string $\mu \in \{0,1\}^*$. Then, the signature of knowledge is generated by calling the $\mathsf{L2RS.SigGen}\big(\widehat{\mathbf{S}}^{(1)}_{(in),\pi}, L', \mu, \mathsf{Pub\text{-}Params}\big)$, Algorithm 2, which outputs $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$.

(f) We set the transaction TX as (μ, IW, OW) and $TN = \mathbf{h}$.

(g) This algorithm ultimately outputs TX, TN, and $\sigma_{L'}(\mu)$.

5. $(\mathbf{Accept}/\mathbf{Reject}) \leftarrow \mathsf{LRCT.Verify}\big(TX, \sigma_{L'}(\mu), TN\big)$: This algorithm calls $\mathsf{L2RS.SigVer}$ (Algorithm 3) with $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$, $TN = \mathbf{h}$, $L' = \big\{\widehat{\mathbf{a}}_{(in),i}\big\}_{i\in[w]} = \big\{\mathbf{a}_{(in),i} + \mathbf{cn}_{(in),i} - \mathbf{cn}_{(out)}\big\} \in \mathcal{R}_q$ and $\mathsf{Pub\text{-}Params}$, this ultimately outputs either **Accept** or **Reject**.

4.6 Performance Analysis

Remark 4.13. This research project did not consider the implementation of the schemes $\mathsf{SISO.L2RS}$ and $\mathsf{SISO.LRCT}$, as a result there is not run time analysis. The project only evaluates the signature and key sizes of the proposed constructions.

We proposed a set of parameters (Table 4.1) to implement the $\mathsf{L2RS}$ and $\mathsf{SISO.LRCT}$ schemes. They are secure against direct lattice attacks in terms of the BKZ algorithm Hermite factor δ, using the value of $\delta = 1.007$, based on the BKZ 2.0 complexity estimates with pruning enumeration-based Shortest Vector Problem (SVP) [CN11], this might give $90-100$ bits of security. We use the conditions stated in the $\mathsf{L2RS.SigVer}$ algorithm and in the security analysis (Section

Algorithm 9 LRCT.Spend - SISO

Input: $(\mu, IW, IW_\pi, OA, \mathsf{Pub\text{-}Params})$, being the message, the Input Wallets, π's Input Wallet, the Output Address and the public parameters, respectively.

Output: $\big(TX, \sigma_{L'}(\mu), TN\big)$

1: **procedure** LRCT.SPEND$(\mu, IW, IW_\pi OA, \mathsf{Pub\text{-}Params})$

2: Define π's $IW_\pi^{(1)} = \big\{\mathbf{a}_{(in),\pi}^{(1)}, \mathbf{cn}_{(in),\pi}^{(1)}\big\} \in \mathcal{R}_q \times \mathcal{R}_q$and $K_\pi^{(1)} = \big\{\mathbf{S}_{(in),\pi}^{(1)}, \mathbf{ck}_{(in),\pi}^{(1)}\big\} \in \mathcal{R}_q^{1\times(m-1)} \times \mathcal{R}_q^{1\times(m-1)}$.

3: Define a valid output address $OA = \mathbf{a}_{(out)}^{(1)}$ and $\$_{(out)}^{(1)} \in \big[0, \ldots, 2^{\ell_\$} - 1\big]$ such $\$_{(in),\pi}^{(1)} = \$_{(out)}^{(1)}$, then compute $\big(\mathbf{cn}_{(out)}^{(1)}, \mathbf{ck}_{(out)}^{(1)}\big) \leftarrow \mathsf{LRCT.Mint}\big(\mathbf{a}_{(out)}^{(1)}, \$_{(out)}^{(1)}\big)$.

4: Define $OW^1 = \big\{\mathbf{a}_{(out)}^{(1)}, \mathbf{cn}_{(out)}^{(1)}\big\} \in \mathcal{R}_q \times \mathcal{R}_q$.

5: Send securely coin-key $\mathbf{ck}_{(out)}^{(1)}$ to user's $\mathbf{a}_{(out)}^{(1)}$.

6: Create the list of input wallets $IW_i^{(1)}\big\{\mathbf{a}_{(in),i}^{(1)}, \mathbf{cn}_{(in),i}^{(1)}\big\}_{i\in[w-1]}$ (Ring Confidential Transaction).

7: Set $L' = \big\{\widehat{\mathbf{a}}_{(in),i}\big\}_{i\in[w]} \in \mathcal{R}_q$, where $\widehat{\mathbf{a}}_{(in),i}$ is the homomorphic commitment with randomness $\widehat{\mathbf{S}}_{(in),i}$.

8: Define $\widehat{\mathbf{a}}_{(in),i}^{(1)} = \mathbf{a}_{(in),i}^{(1)} + \mathbf{cn}_{(in),i}^{(1)} - \mathbf{cn}_{(out)}^{(1)} = \mathsf{Com}_{\mathbf{A}}\big(\$_{(in),i}^{(1)} - \$_{(out)}^{(1)}, \widehat{\mathbf{S}}_{(in),i}^{(1)}\big)$.

9: Define $\widehat{\mathbf{S}}_{(in),i} = \big(\mathbf{S}_{(in),i} + \mathbf{ck}_{(in),i} - \mathbf{ck}_{(out)}\big) \in \mathcal{R}_q$.

10: Call $\mathsf{L2RS.SignGen}\big(\widehat{\mathbf{S}}_{(in),\pi}, L', \mu, \mathsf{Pub\text{-}Params}\big)$ and retrieve $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \mathbf{t}_1, \ldots, \mathbf{t}_w, \mathbf{h}\Big)$.

11: Set $TX = (\mu, IW, OW)$, $TN = \mathbf{h}$.

12: **return** $\big(TX, \sigma_{L'}(\mu), TN\big)$

4.4). Table 4.1 illustrates this information for five different versions of both L2RS and SISO.LRCT: I, II, III, IV and V, where these versions vary with the polynomial ring degree n. The figures of this table infer that the signature size grows linear with the number of users in the Ring Signature.

TABLE 4.1: Concrete parameters and sizes for L2RS and SISO.LRCT

Parameter	**Description**	I	II	III	IV	V
n	Polynomial ring degree	128	256	512	1024	2048
m	Polynomial ring size	18	10	6	5	5
λ	Security parameter	100	100	100	100	100
δ	Hermite factor	1.007	1.007	1.007	1.007	1.007
$\log(q)$	Modulus q - quotient	123	61	31	26	27
κ	Random Oracle weight	32	21	17	14	12
η	Correctness	1.1	1.1	1.1	1.1	1.1
α	Rejection sampling	0.1	0.1	0.1	0.1	0.1
M	Rejection sampling	1.0027	1.0027	1.0027	1.0027	1.0027
$\gamma \approx \log(2 \cdot n \cdot \kappa)$	Private-key density	13.6	13.6	13.6	13.6	13.6
σ	Gaussian standard deviation	337151	287898	283754	332435	435260
private-key		1.95 KB	1.93 KB	1.96 KB	3.28 KB	6.78 KB
public-key		1.92 KB	1.90 KB	1.93 KB	3.24 KB	6.74 KB
$w = 1$[2]		7.2 KB	7.7 KB	8.9 KB	15 KB	30.9 KB
$w = 5$		28.4 KB	30.9 KB	36.7 KB	62 KB	126 KB
$w = 8$		44.3 KB	48.4 KB	57.6 KB	97.2 KB	198.7 KB
$w = 16$		86.6 KB	94.8 KB	113.2 KB	191.1 KB	390.5 KB
$w = 32$		171.2 KB	187.6 KB	224.5 KB	379 KB	774.1 KB
$w = 64$		340.4 KB	373.3 KB	447.1 KB	754.6 KB	1541.4 KB
$w = 128$		678.9 KB	744.7 KB	892.3 KB	1505.9 KB	3075.9 KB

[1] w is the Ring Signature size

4.7 Summary

As our society becomes increasingly reliant on technology, cybersecurity is becoming an essential ingredient to protect our information assets. Cryptographic techniques, an important tool of cyber-security, currently rely on computational assumptions to offer security guarantees. However, these assumptions could easily be broken by a large quantum computer. The cryptographic community is beginning to understand the importance of constructing post-quantum cryptographic techniques to withstand such attacks. Therefore, this research aimed to devise post-quantum constructions that can be applied in cryptocurrencies.

This chapter showed the first contribution, which was designing and constructing a privacy preserving linkable ring signature (the L2RS) that is protected against quantum attacks by using lattice-based cryptography. The scheme provides unconditional anonymity, meaning that even an adversary with unlimited computational resources and time, would be unable to break into this property. The remaining properties, the unforgeability, linkability and non-slanderability are computationally secure under the lattice hardness assumptions. Moreover, based on the L2RS scheme, a novel cryptocurrency protocol (LRCT) was devised and constructed which inherited the post-quantum security guarantees of the L2RS. The performance results illustrated that the signature size grows linearly with the number of users in the Ring Signature.

Nevertheless, these proposals have some limitations. To begin with, they only enable transfers from a Single Output wallet to a Single Output wallet (SISO). In the RingCT model, signatures are *one-time*. If one then needs to receive change after making a payment or transfer, a new output wallet is required, so this points out the importance of supporting multiple input and output wallets. Secondly, having more than one output wallet also introduces a new security problem like the negative output amount (or out-of-range) attack [BBB+18], where an adversary is capable of creating extra coins. This attack is addressed in the previous RingCT [Noe15] by using a range proof technique; however, this technique is not post-quantum secure.

These constraints were, in part, the motivation for continuing this research project. In the next chapter, the second version of both schemes, the L2RS and the LRCT, will be presented. These constructions are migrated to support transfers from Multiple-Input to Multiple-Output MIMO wallets.

Chapter 5

The MIMO of L2RS and LRCT

[1]CryptoNote [VS13], a cryptocurrency protocol, was proposed to address privacy weaknesses in Bitcoin [RS13, KKM14]. It also offers a framework that can be extended by other cryptocurrencies such as Bytecoin [Byt15] and Monero [Mon14]. CryptoNote uses *traceable ring signatures* [FS07] as a fundamental component to achieve true anonymity, where any member of the ring (or group) can create a signature, but it is infeasible by a verifier to identify the real signer. This type of signature hides information about both the sender and receiver, and it also has a *linking tag* to prevent the double spending of coins. Further enhancements to this framework have resulted in an extended protocol called Ring Confidential Transactions "RingCT" [Noe15]. The RingCT protocol uses three techniques: a

[1]This chapter was published as: Alberto Torres W., Kuchta V., Steinfeld R., Sakzad A., Liu J.K., Cheng J. (2019) *Lattice RingCT V2.0 with Multiple Input and Multiple Output Wallets*. In: Jang-Jaccard J., Guo F. (eds) Information Security and Privacy. ACISP 2019. Lecture Notes in Computer Science, vol 11547. Springer, Cham. DOI:https://doi.org/10.1007/978-3-030-21548-4_9.

new type of ring signature *Linkable Ring Signatures* [LWW04b], a *homomorphic commitment* and a *range proof*, to preserve the privacy of the sender and the receiver as well as the transaction amounts.

This chapter describes the construction of the Lattice-based Ring Confidential Transactions (LRCT), which supports Multiple-Input and Multiple-Output wallets (MIMO). This construction is a generalisation of the SISO.LRCT scheme (from Chapter 4) where its underlying structure - the L2RS signature - was modified for improved compatibility with this new version. The MIMO.LRCT inherits the post-quantum security guarantees from SISO.LRCT, such as the hardness of lattice mathematical assumptions, as well as unconditional anonymity.

At the outset, we enhanced the MIMO.LRCT's security model, particularly the anonymity and balance properties. In the case of anonymity, the chapter includes the analysis of both user and amount privacy; in contrast to another similar work [SALY17] which only considered user anonymity. More specifically, the proposed construction is reduced to demonstrate that the anonymity property relies on the left over hash lemma (defined in Chapter 3), meaning that the distribution of the signature is independent to the secret key (including the secret part of the wallets) used to produce that signature.

The balance property now includes the out-of-range attacks [BBB+18] and the security proofs which previous RingCT's proposals such as [SALY17] and the SISO.LRCT (in Chapter 4) did not address. The RingCT model, where signatures are *one-time*, needs to receive change after making digital wallet transactions, so the SISO.LRCT is incompatible with this requirement since it only supports single transfers. As a result, a new output wallet is required. This leads to the importance of constructions that allow multiple input and output wallets. In addition, introducing more than one output wallet also demands for another security concern; that is, the negative output amount attack [BBB+18], where an adversary is capable of creating extra coins (also known as free money). The security analysis illustrates how to incorporate a lattice-based range proof in the MIMO.LRCT protocol to overcome such attacks. This protocol deals with the difficulties stemming

from the imperfection of lattice-based zero-knowledge proofs. To be more specific, the range proofs follow the approach based on 1-of-2 OR-proofs, but our analysis shows that directly applying lattice-based OR-proofs from [dPLNS17] does not provide soundness for the range proof. Although these challenges are smaller -in norm- than the ones used in the OR-proofs, they are still larger than the challenges in [LLNW18]. In this framework, we achieve lower soundness error than the previous lattice-based range proof as in [LLNW18].

Moreover, a thorough concrete performance analysis of the MIMO.LRCT protocol is provided by including this range proof analysis. Concrete bounds are applied to derive preliminary scheme parameters for regular RingCT transactions that support operations of 64-bit amounts along with fewer Multiple Input and Output wallets. Therefore, these analyses serve as a benchmark and motivation for future studies.

The organisation of this chapter is as follows. The definitions and security model of the upgraded MIMO.LRS are illustrated in Section 5.1, which is followed by its construction (the MIMO.L2RS) and the security analysis in Section 5.2 and Section 5.3, respectively. The definition and security analysis of the MIMO.LRCT are presented in Section 5.4. Whereas Section 5.5 introduces the definitions of the building blocks used by the cryptocurrency protocol MIMO.LRCT scheme, whereas Section 5.6 describes its construction. The techniques utilised to handle the out-of-range attacks (the range preservation) are reported in Section 5.7. Section 5.8 and Section 5.9 display the security and performance analyses of the MIMO.LRCT scheme, respectively. Finally, Section 5.10 presents the summary of this chapter.

5.1 MIMO.LRS - Definitions and Security Model

This section introduces the definition and security model of a Multiple Input Multiple Output Linkable Ring Signature (MIMO.LRS).

5.1.1 MIMO.LRS Definitions

A MIMO.LRS scheme has five Probabilistic Polynomial Time (PPT) algorithms (MIMO.LRS.Setup, MIMO.LRS.KeyGen, MIMO.LRS.SigGen, MIMO.LRS.SigVer, MIMO.LRS.SigLink). In addition, the correctness of this scheme is satisfied by the signature correctness MIMO.LRS.SigGen Correctness and the linkability correctness MIMO.LRS.SigLink Correctness. Table 5.1 illustrates a summary of these algorithms.

TABLE 5.1: MIMO.LRS Algorithms

Algorithm	**Input**	**Output**	**Description**
MIMO.LRS.Setup	The security parameter	The public parameters	Public parameters creation
MIMO.LRS.KeyGen	The public parameters	The pair keys	Public and private keys creation
MIMO.LRS.SigGen	Private-key, message, list of public-keys in the Ring Signature and the public parameters	Signature	Ring signature generation
MIMO.LRS.SigVer	Signature, message, list of public-keys in the Ring Signature and the public parameters	Accept or Reject	Verify whether or not a signature was successfully generated
MIMO.LRS.SigLink	Two verified signatures	Linked or Unlinked	Verify whether or not two successfully generated signatures are linked

- MIMO.LRS.Setup: a PPT algorithm that takes the security parameter λ and produces the Public Parameters (Pub-Params).

- MIMO.LRS.KeyGen: a PPT algorithm that by taking the Pub-Params, it produces a pair of keys: the public-key pk and the private-key sk.

- MIMO.LRS.SigGen: a PPT algorithm that receives the Pub-Params, a signer's private-keys $\mathsf{sk}_\pi^{(k)}$ where $\mathsf{sk}_\pi^{(k)}$ denotes the k'th input private key of signer π, for $k = 1, ..., N_{in}$, a message μ, and the list L of pk's in the ring (as in 5.1). We defined w as the size of the ring and N_{in} as the number of input wallets (useful in the MIMO.LRCT protocol). This algorithm outputs a signature $\sigma_L(\mu)$.

$$L \triangleq \left\{ \mathsf{pk}_i^{(k)} \right\}_{i \in [w], k \in [N_{in}]} \tag{5.1}$$

- MIMO.LRS.SigVer: a PPT algorithm that takes Pub-Params, a signature $\sigma_L(\mu)$, a list L of pk's and the message μ, and it verifies if this signature was legitimately created, this algorithm outputs either: *Accept* or *Reject*.

- MIMO.LRS.SigLink: a PPT algorithm that inputs two valid signatures $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$ and it anonymously determines if these signatures were produced by same signer π. Thus, this algorithm has a deterministic output: *Linked* or *Unlinked*.

Correctness Requirements:

- MIMO.LRS.SigGen Correctness: this guarantees that valid signatures signed by honest signers will be accepted by a verifier with overwhelming probability.

- MIMO.LRS.SigLink Correctness: this ensures that if two signatures $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$ are signed by an honest signer π, SigLink will output *Linked* with overwhelming probability.

5.1.2 Oracles for adversaries

The following oracles are available to any adversary who tries to break the security of an MIMO.LRS scheme:

- $\mathsf{pk}_i^{(k)} \hookleftarrow \mathcal{JO}(\perp)$. The *Joining Oracle*, on request, adds new user(s) to the system. It returns the public-key(s) $\mathsf{pk}_i^{(k)}$.
- $\mathsf{sk}_i^{(k)} \leftarrow \mathcal{CO}(\mathsf{pk}_i^{(k)})$. The *Corruption Oracle*, on input a $\mathsf{pk}_i^{(k)}$ that is a query output of $\mathcal{JO}$, returns the corresponding $\mathsf{sk}_i^{(k)}$.
- $\sigma'_L(\mu) \leftarrow \mathcal{SO}(w, L, \mathsf{pk}_\pi^{(k)}, \mu)$. The *Signing Oracle*, on input a group size w, a set L of w $\mathsf{pk}^{(k)}$'s, the signer's $\mathsf{pk}_\pi^{(k)}$, and a message μ, this oracle returns a valid signature $\sigma'_L(\mu)$.

5.1.3 Security Game Definition for MIMO.LRS

- One-time Unforgeability. One time unforgeability for the MIMO.LRS scheme is defined in the following game between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$ who has access to the oracles $\mathcal{JO}$, $\mathcal{CO}$, $\mathcal{SO}$ and the random oracle:
  - $\mathcal{S}$ generates and gives the list L of $\mathsf{pk}^{(k)}$'s to $\mathcal{A}$.
  - $\mathcal{A}$ may query the oracles according to any adaptive strategy.
  - $\mathcal{A}$ gives $\mathcal{S}$ a ring signature size w, a set L of w $\mathsf{pk}^{(k)}$'s, a message μ and a signature $\sigma_L(\mu)$.

  $\mathcal{A}$ wins the game if:

  1. MIMO.LRS.SigVer$(\sigma_L(\mu))$=**Accept**.
  2. $\mathsf{pk}^{(k)}$'s in the L are outputs from $\mathcal{JO}$ oracle.
  3. $\sigma_L(\mu)$ is not an output of $\mathcal{SO}$.
  4. No signing key $\mathsf{pk}_\pi^{(k)}$ was queried more than once to $\mathcal{SO}$.
  5. $\forall i \in [w]\ \exists k \in [N_{in}]\ s.t.\ \mathsf{pk}_i^{(k)}$ is not corrupted.

The advantage of the one-time unforgeability in the MIMO.LRS scheme is denoted by

$$\mathbf{Advantage}_{\mathcal{A}}^{ot-unf}(\lambda) = \Pr[\mathcal{A} \text{ wins the game }]$$

Definition 5.1 (One-Time Unforgeability)**.** The MIMO.LRS scheme is one-time unforgeable if for all PPT adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{ot-unf}(\lambda)$ is negligible.

- Unconditional Anonymity. It should be infeasible for an adversary $\mathcal{A}$ to distinguish a signer's $\mathsf{pk}^{(k)}$ with probability larger than 1/2, even if the adversary has unlimited computing resources. This property for MIMO.LRS schemes is defined in the following game between a simulator $\mathcal{S}$ and an unbounded adversary $\mathcal{A}$.
  - $\mathcal{A}$ may query $\mathcal{JO}$ according to any adaptive strategy.
  - $\mathcal{A}$ gives $\mathcal{S}$ the $L = \{\mathsf{pk}_0^{(k)}, \mathsf{pk}_1^{(k)}\}_{k\in[N_{in}]}$, which is the output of the $\mathcal{JO}$, and a message μ.
  - $\mathcal{S}$ flips a coin $b = \{0, 1\}$, then $\mathcal{S}$ computes the signature $\sigma_b = \mathsf{MIMO.LRS.SigGen}(L, \mathsf{sk}_b^{(k)}, \mu, \mathsf{Pub\text{-}Params})$. This signature is given to $\mathcal{A}$.
  - $\mathcal{A}$ outputs a bit b'.
  - The output of this experiment is defined to be 1 if $b = b'$, or 0 "zero" otherwise.

  $\mathcal{A}$ wins the game if:

  1. $\mathsf{pk}_0^{(k)}$ and $\mathsf{pk}_1^{(k)}$ cannot be used by $\mathcal{CO}$ and $\mathcal{SO}$.
  2. $\mathcal{A}$ outputs b' such $b = b'$.

  The unconditional anonymity advantage of the MIMO.LRS scheme is denoted by

$$\mathbf{Advantage}_{\mathcal{A}}^{Anon}(\lambda) = \Big|\Pr[b = b'] - \frac{1}{2}\Big|.$$

Definition 5.2 (Unconditional Anonymity)**.** The MIMO.LRS scheme is unconditional anonymous if for any unbounded adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{Anon}(\lambda)$ is zero.

- LINKABILITY. It should be infeasible for an adversary $\mathcal{A}$ to *unlinked* two valid MIMO.LRS signatures which were correctly generated with same $\mathsf{sk}_{\pi}^{(k)}$. To describe this, we use the interaction between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$:

  – The $\mathcal{A}$ queries the $\mathcal{JO}$ multiple times.

  – The $\mathcal{A}$ outputs two signatures $\sigma_L(\mu)$ and $\sigma'_{L'}(\mu')$ and two lists L and L' of $\mathsf{pk}^{(k)}$'s.

$$L' \triangleq \left\{\mathsf{pk}_i'^{(k)}\right\}_{i\in[w],k\in[N_{in}]} \tag{5.2}$$

  $\mathcal{A}$ wins the game if:

  1. By calling MIMO.LRS.SigVer on input $\sigma_L(\mu)$ and $\sigma'_{L'}(\mu')$, it outputs **Accept** on both inputs.
  2. The $\mathsf{pk}^{(k)}$'s in L and L' are outputs of $\mathcal{JO}$.
  3. Finally, it gets *unlinked*, when calling MIMO.LRS.SigLink on input $\sigma_L(\mu)$ and $\sigma'_{L'}(\mu')$.

  Thus the advantage of the linkability in the MIMO.LRS scheme is denoted by

$$\mathbf{Advantage}_{\mathcal{A}}^{Link}(\lambda) = \Pr[\mathcal{A} \text{ wins the game}].$$

  Definition 5.3 (Linkability)**.** The MIMO.LRS scheme is linkable if for all PPT adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{Link}$ is negligible.

- NON-SLANDERABILITY. It should be infeasible for an adversary $\mathcal{A}$ to *linked* two valid MIMO.LRS signatures which were correctly generated with different $\mathsf{sk}^{(k)}$'s. This means that an adversary can frame an honest user for signing a valid signature so the adversary can produce another valid signature such

that the MIMO.LRS.SigLink algorithm outputs *linked*. To describe this, we use the interaction between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$:

- The $\mathcal{S}$ generates and gives the list L of $\mathsf{pk}^{(k)}$'s to $\mathcal{A}$.
- The $\mathcal{A}$ queries the $\mathcal{JO}$ and $\mathcal{CO}$ to obtain $\mathsf{pk}_{\pi}^{(k)}$ and $\mathsf{sk}_{\pi}^{(k)}$, respectively.
- $\mathcal{A}$ gives the generated parameters to $\mathcal{S}$.
- $\mathcal{S}$ uses the $\mathsf{sk}_{\pi}^{(k)}$ and calls the $\mathcal{SO}$ to output a valid signature $\sigma_L(\mu)$, which is given to $\mathcal{A}$.
- The $\mathcal{A}$ uses the remaining keys of the ring signature $(w-1)$ to create a second signature $\sigma'_L(\mu)$ by calling the $\mathcal{SO}$ algorithm.

$\mathcal{A}$ wins the game if:

1. The MIMO.LRS.SigVer, on input $\sigma_L(\mu)$ and $\sigma'_L(\mu)$, outputs **Accept**.
2. The keys $\mathsf{pk}_{\pi}^{(k)}$ and $\mathsf{sk}_{\pi}^{(k)}$ were not used to generate the second signature $\sigma'_L(\mu)$.
3. When calling the MIMO.LRS.SigLink on input $\sigma_L(\mu)$ and $\sigma'_L(\mu)$, it outputs *linked*.

Thus the advantage of the non-slanderability in the MIMO.LRS scheme is denoted by

$$\mathbf{Advantage}_{\mathcal{A}}^{NS}(\lambda) = \Pr[\mathcal{A} \text{ wins the game}].$$

Definition 5.4 (Non-Slanderability)**.** The MIMO.LRS scheme is *non-slanderable* if for all PPT adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{NS}$ is negligible.

5.2 MIMO.L2RS Scheme construction

In this section, we construct a lattice-based version of the MIMO.LRS. The scheme MIMO.L2RS = (MIMO.L2RS.Setup, MIMO.L2RS.KeyGen, MIMO.L2RS.SigGen, MIMO.L2RS.SigVer, MIMO.L2RS.SigLink) works as follows.

5.2.1 MIMO.L2RS.Setup

By receiving the security parameter λ, this MIMO.L2RS.Setup algorithm randomly chooses $\mathbf{A} \hookleftarrow \mathcal{R}_q^{2\times(m-1)}$ and $\mathbf{H} \hookleftarrow \mathcal{R}_q^{2\times(m-1)}$. This outputs the public parameters (Pub-Params): $\mathbf{A}$ and $\mathbf{H}$.

Remark 5.5. To prevent malicious attack, MIMO.L2RS.Setup incorporates a trapdoor in $\mathbf{A}$ or $\mathbf{H}$, in practice MIMO.L2RS.Setup would generate $\mathbf{A}$ and $\mathbf{H}$ based on the cryptographic Hash function H_2 evaluated at two distinct and fixed constants.

Definition 5.6 (Function MIMO.L2RS.Lift)**.** This function maps $\mathcal{R}_q^2$ to $\mathcal{R}_{2q}$ with respect to a public parameter $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$. Given $\mathbf{a} \in \mathcal{R}_q^2$, we let $\mathsf{MIMO.L2RS.Lift}(\mathbf{A}, \mathbf{a}) \triangleq (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$ with $\mathbf{q} = q \cdot (1,1)^T$.

5.2.2 Key Generation - MIMO.L2RS.KeyGen

This algorithm receives the public parameter Pub-Param: $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$, then it generates a key pair in $\mathcal{R}_q^2$, we:

- Pick $(\mathbf{s}_1, \ldots, \mathbf{s}_{m-1})$ with every component chosen uniformly and independently with coefficients in $(-2^\gamma, 2^\gamma)$.
- Define $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_{m-1})^T \in \mathcal{R}_q^{1\times(m-1)}$.
- Compute $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2)^T = \mathbf{A} \cdot \mathbf{S} \bmod q \in \mathcal{R}_q^2$. The $\mathbf{a}$ and $\mathbf{S}$ are the public-key pk and the private-key sk, respectively.

This MIMO.L2RS.KeyGen algorithm is described in the following Algorithm 10.

Algorithm 10 MIMO.L2RS.KeyGen - Key-pair Generation $(\mathbf{a}, \mathbf{S})$

Input: Pub-Param: $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$.
Output: $(\mathbf{a}, \mathbf{S})$, being the public-key and the private-key, respectively.
1: **procedure** MIMO.L2RS.KeyGen($\mathbf{A}$)
2: Let $\mathbf{S}^T = (\mathbf{s}_1, \ldots, \mathbf{s}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\mathbf{s}_i \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \le i \le m-1$
3: Compute $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2)^T = \mathbf{A} \cdot \mathbf{S} \bmod q \in \mathcal{R}_q^2$.
4: **return** $(\mathbf{a}, \mathbf{S})$.

5.2.3 Signature Generation - MIMO.L2RS.SigGen

The MIMO.L2RS.SigGen algorithm inputs the user's private-key $\mathbf{S}_{(in),\pi}^{(k)}$, the message μ, the list of user's public-keys L' and the public parameters Pub-Params: $\mathbf{H} \in \mathcal{R}_q^{1\times(m-1)}$ and $\mathbf{A} \in \mathcal{R}_q^{1\times(m-1)}$. This algorithm outputs the signature $\sigma_{L'}(\mu)$. We call π the index in $\{1,\ldots,w\}$ of the user or signatory who wants to sign a message μ. For a message $\mu \in \{0,1\}^*$, the fixed list of public-keys $L = \{\mathbf{a}_{(in),1}^{(k)},\ldots,\mathbf{a}_{(in),w}^{(k)}\}$ and the private-key $\mathbf{S}_{(in),\pi}^{(k)}$ which corresponds to $\mathbf{a}_{(in),\pi}^{(k)}$ with $1 \leq \pi \leq w$ and $k \in [1, N_{in}+1]$; the following computations are performed:

1. We define the linkability tag as $\mathbf{H}_{2q}^{(k)} = \big(2\cdot\mathbf{H}, -2\cdot\mathbf{h}^{(k)} + \mathbf{q}\big) \in \mathcal{R}_{2q}^{2\times m}$, where $\mathbf{H}$ is the fixed public parameter for all users, and $\mathbf{h}^{(k)} = \mathbf{H}\cdot\mathbf{S}_{(in),\pi}^{(k)} \in \mathcal{R}_q^2$. We consider $\mathbf{S}_{(in),\pi}^{(k),T} \in \mathcal{R}_q^{1\times(m-1)}$ as an element in $\mathcal{R}_{2q}$ and let $\mathbf{S}_{(in),2q,\pi}^{(k),T} = \big(\mathbf{S}_{(in),\pi}^{(k),T}, 1\big) \in \mathcal{R}_{2q}^{1\times m}$, such that $\mathbf{H}_{2q}^{(k)}\cdot\mathbf{S}_{(in),\pi}^{(k),T} = q \in \mathcal{R}_{2q}$.

2. The π's public-key is lifted from $\mathcal{R}_q^{1\times m}$ to $\mathcal{R}_{2q}^{1\times m}$, so by calling the lift function MIMO.L2RS.Lift$(\mathbf{A}, \mathbf{a}_{(in),\pi}^{(k)})$, we get $\mathbf{A}_{2q,\pi}^{(k)} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}_{(in),\pi}^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.

3. Note that $\mathbf{A}_{2q,\pi}^{(k)}\cdot\mathbf{S}_{(in),\pi}^{(k),T} = q \in \mathcal{R}_{2q}$

4. By choosing a random vector $\mathbf{u}^{(k)} = (u_1,\ldots,u_m)^T$, where $u_i \hookleftarrow D_\sigma^n$, for $1 \leq i \leq m$, we calculate $\mathbf{c}_{\pi+1} = H_1\Big(L', \big\{\mathbf{H}_{2q}^{(k)}\big\}_{k\in[N_{in}+1]}, \mu, \big\{\mathbf{A}_{2q,\pi}^{(k)}\cdot\mathbf{u}^{(k)}\big\}_{k\in[N_{in}+1]}, \big\{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{u}^{(k)}\big\}_{k\in[N_{in}+1]}\Big)$.

5. We choose random vector $\mathbf{t}_i^{(k)} = (t_{i,1},\ldots,t_{i,m})^T$, where $t_{i,j} \hookleftarrow D_\sigma^n$, for $1 \leq j \leq m$, then for $(i = \pi+1,\ldots,w,1,2,\ldots,\pi-1)$, after lifting from $\mathcal{R}_q^{1\times m}$ to $\mathcal{R}_{2q}^{1\times m}$, using MIMO.L2RS.Lift$(\mathbf{A}, \mathbf{a}_{(in),i}^{(k)})$, we obtain $\mathbf{A}_{2q,i}^{(k)} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}_{(in),i}^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$. Then, we compute $\mathbf{c}_{i+1} = H_1\Big(L', \big\{\mathbf{H}_{2q}^{(k)}\big\}_{k\in[N_{in}+1]}, \mu, \big\{\mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\big\}_{k\in[N_{in}+1]}, \big\{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\big\}_{k\in[N_{in}+1]}\Big)$.

6. Select a random bit $b \in \{0,1\}$ and finally compute $\mathbf{t}_\pi^{(k)} \leftarrow \mathbf{u}^{(k)} + \mathbf{S}_{2q,\pi}^{(k)}\cdot\mathbf{c}_\pi\cdot(-1)^{b^{(k)}}$ using rejection sampling (Definition 3.8).

7. Output the signature $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \big\{\mathbf{t}_1^{(k)},\ldots,\mathbf{t}_w^{(k)}\big\}_{k\in[N_{in}+1]}, \big\{\mathbf{h}^{(k)}\big\}_{k\in[N_{in}]}\Big)$.

Remark 5.7. π adds a record to homomorphically compute and verify the *amount preservation* property; this uses the homomorphic commitment scheme (defined in Section 5.5.1). The result of this computation is a commitment to zero. This new record is placed in the position $(N_{in}+1)$ and then a list L' is defined as (5.13) (Section 5.6 provides further explanation of this record within the proposed cryptocurrency protocol). Contrary, the linking tags (or $\mathbf{h}^{(k)}$) only needs N_{in} since $\mathbf{h}^{(k)}$ are tags generated based on the number of input secret keys.

A formal description of this algorithm is shown in Algorithm 11.

Algorithm 11 MIMO.L2RS.SigGen - MIMO Signature Generation $\sigma_{L'}(\mu)$

Input: $\{\mathbf{S}^{(k)}_{(in),\pi}\}_{k\in[N_{in}+1]}, \mu$, L' as in (5.13), and Pub-Params.
Output: $\sigma_{L'}(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\right)$

procedure MIMO.L2RS.SigGen($\mathbf{S}^{(k)}_{(in),\pi}, \mu, L'$, Pub-Params)
  for $(1 \le k \le N_{in}+1)$ **do**
    Set $\mathbf{H}^{(k)}_{2q} = (2\cdot\mathbf{H}, -2\cdot\mathbf{h}^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$, where $\mathbf{h}^{(k)} = \mathbf{H}\cdot\mathbf{S}^{(k)}_{(in),\pi} \in \mathcal{R}_q^2$.
    Call MIMO.L2RS.Lift$(\mathbf{A}, \mathbf{a}^{(k)}_{(in),\pi})$ to obtain $\mathbf{A}^{(k)}_{2q,\pi} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}^{(k)}_{(in),\pi} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
    Let $\mathbf{u}^{(k)} = (u_1, \ldots, u_m)^T$, where $u_i \hookleftarrow D^n_\sigma$, for $1 \le i \le m$.
  Compute $\mathbf{c}_{\pi+1} = H_1\Big(L', \{\mathbf{H}^{(k)}_{2q}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}^{(k)}_{2q,\pi}\cdot\mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{H}^{(k)}_{2q}\cdot\mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}\Big)$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \le k \le N_{in}+1)$ **do**
      Call MIMO.L2RS.Lift$(\mathbf{A}, \mathbf{a}^{(k)}_{(in),i})$ to obtain $\mathbf{A}^{(k)}_{2q,i} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}^{(k)}_{(in),i} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
      Let $\mathbf{t}_i^{(k)} = (t_{i,1}, \ldots, t_{i,m})^T$, where $t_{i,j} \hookleftarrow D^n_\sigma$, for $1 \le j \le m$.
    Compute $\mathbf{c}_{i+1} = H_1\Big(L', \{\mathbf{H}^{(k)}_{2q}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}^{(k)}_{2q,i}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\}_{k\in[N_{in}+1]}, \{\mathbf{H}^{(k)}_{2q}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\}_{k\in[N_{in}+1]}\Big)$.
  for $(1 \le k \le N_{in}+1)$ **do**
    Choose $b^{(k)} \hookleftarrow \{0,1\}$.
    Let $\mathbf{t}_\pi^{(k)} \leftarrow \mathbf{u}^{(k)} + \mathbf{S}^{(k)}_{2q,\pi}\cdot\mathbf{c}_\pi\cdot(-1)^{b^{(k)}}$, where $\mathbf{S}^{(k)}_{2q,\pi} = [(\mathbf{S}^{(k)}_\pi)^T, 1]^T$.
    Continue with prob. $\left(M\exp\left(-\frac{\|\mathbf{S}^{(k)}_{2q,\pi}\cdot\mathbf{c}_\pi\|^2}{2\sigma^2}\right)\cosh\left(\frac{\langle\mathbf{t}^{(k)}_\pi, \mathbf{S}^{(k)}_{2q,\pi}\cdot\mathbf{c}_\pi\rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
  return $\sigma_{L'}(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\right)$.

5.2.3.1 Correctness of MIMO.L2RS.SigGen

Proof. Beyond the required conditions of MIMO.L2RS.SigVer, we claim that if $\sigma_{L'}(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\right)$ is the output of the MIMO.L2RS.SigGen algorithm on input $(\mu, L, \mathbf{S}_\pi$, Pub-Params$)$, then the output of MIMO.L2RS.SigVer on input $(\mu, L, \sigma_L(\mu))$ should be accepted. We need to show that when MIMO.L2RS.SigVer computes $H_1\Big(L', \{\mathbf{H}^{(k)}_{2q}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}^{(k)}_{2q,i}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\}_{k\in[N_{in}+1]}, \{\mathbf{H}^{(k)}_{2q}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\}_{k\in[N_{in}+1]}\Big)$, the result is equal to $\mathbf{c}_1$. We also show

that this $H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}\Big) = \mathbf{c}_{i+1}$ for $1 \leq i \leq w-1$ in MIMO.L2RS.SigVer. In this evaluation, we consider two scenarios, one when $i \neq \pi$ and $i = \pi$:

- For $i \neq \pi$, in MIMO.L2RS.SigGen we have $\mathbf{c}_{i+1} = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}\Big)$, while in MIMO.L2RS.SigVer we compute $\mathbf{c}_{i+1} = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}\Big)$. These are equal since $\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i$ (in MIMO.L2RS.SigGen) $= \mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i$ (in MIMO.L2RS.SigVer); and $\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}$ (in MIMO.L2RS.SigGen) $= \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}$ (in MIMO.L2RS.SigVer).

- For $i = \pi$, in MIMO.L2RS.SigGen we have $\mathbf{c}_{\pi+1} = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}\Big)$, whereas in MIMO.L2RS.SigVer we calculate $\mathbf{c}_{\pi+1} = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{t}_\pi^{(k)} + \mathbf{q} \cdot \mathbf{c}_\pi\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_\pi^{(k)} + \mathbf{q} \cdot \mathbf{c}_\pi\}_{k\in[N_{in}+1]}\Big)$. In this case, we need to show that $\mathbf{c}_{\pi+1}$ (in MIMO.L2RS.SigGen) $= \mathbf{c}_{\pi+1}$ (in MIMO.L2RS.SigVer). In doing so, the following equalities need to be proved:

  1. $\mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{t}_\pi^{(k)} + \mathbf{q} \cdot \mathbf{c}_\pi$, which is equivalent to $\mathbf{A}_{2q,\pi}^{(k)} \cdot (\mathbf{u}^{(k)} - \mathbf{t}_\pi^{(k)}) = \mathbf{q} \cdot \mathbf{c}_\pi$. Here, we replace $\mathbf{t}_\pi^{(k)}$ as defined in Algorithm 11, to obtain:

$$\mathbf{A}_{2q,\pi}^{(k)} \cdot (\mathbf{u}^{(k)} - \mathbf{u}^{(k)} + \mathbf{S}_{2q,\pi}^{(k)} \cdot \mathbf{c}_\pi \cdot (-1)^{b^{(k)}}) = \mathbf{q} \cdot \mathbf{c}_\pi \iff$$
$$-\mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{S}_{2q,\pi}^{(k)} \cdot \mathbf{c}_\pi \cdot (-1)^{b^{(k)}} = \mathbf{q} \cdot \mathbf{c}_\pi \iff$$
$$-\mathbf{q} \cdot \mathbf{c}_\pi \cdot (-1)^b = \mathbf{q} \cdot \mathbf{c}_\pi$$

     We distinguish two cases for b:

     - When $b = 0$, we verify that -$\mathbf{q} \cdot \mathbf{c}_\pi = \mathbf{q} \cdot \mathbf{c}_\pi \bmod 2q$.
     - When $b = 1$, we have $\mathbf{q} \cdot \mathbf{c}_\pi = \mathbf{q} \cdot \mathbf{c}_\pi \bmod 2q$.

2. $\mathbf{H}_{2q}^{(k)} \cdot \mathbf{u}^{(k)} = \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_{\pi}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{\pi}$, which means that:

$$\begin{gathered}
\mathbf{H}_{2q}^{(k)} \cdot (\mathbf{u}^{(k)} - \mathbf{t}_{\pi}^{(k)}) = \mathbf{q} \cdot \mathbf{c}_{\pi} \iff \\
\mathbf{H}_{2q}^{(k)} \cdot (\mathbf{u}^{(k)} - \mathbf{u}^{(k)} + \mathbf{S}_{2q,\pi}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}}) = \mathbf{q} \cdot \mathbf{c}_{\pi} \iff \\
-\mathbf{H}_{2q}^{(k)} \cdot \mathbf{S}_{2q,\pi}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b} = \mathbf{q} \cdot \mathbf{c}_{\pi} \iff \\
-\mathbf{q} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b} = \mathbf{q} \cdot \mathbf{c}_{\pi}
\end{gathered}$$

We distinguish between two cases:

- When $b = 0$, it is verified that $-\mathbf{q} \cdot \mathbf{c}_{\pi} = \mathbf{q} \cdot \mathbf{c}_{\pi} \bmod 2q$.
- When $b = 1$, we have $\mathbf{q} \cdot \mathbf{c}_{\pi} = \mathbf{q} \cdot \mathbf{c}_{\pi} \bmod 2q$.

□

5.2.4 Signature Verification - MIMO.L2RS.SigVer

This is described in Algorithm 12. Furthermore, in the following theorem, we show the bound of β_v which is used in this verification algorithm (MIMO.L2RS.SigVer).

Algorithm 12 MIMO.L2RS.SigVer - MIMO Signature Verification

Input: $\sigma_{L'}(\mu)$ as in (5.16), L' as in (5.13), μ, and Pub-Params.
Output: Accept or Reject

procedure MIMO.L2RS.SigVer($\sigma_{L'}(\mu)$, L', Pub-Params)
  for $(1 \leq k \leq N_{in} + 1)$ **do**
    if $\mathbf{H}_{2q}^{(k)} = (2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$ **then** Continue
  for $(i = 1, \ldots, w)$ **do**
    for $(1 \leq k \leq N_{in} + 1)$ **do**
      Call MIMO.L2RS.Lift($\mathbf{A}, \mathbf{a}_{(in),i}^{(k)}$) to obtain $\mathbf{A}_{2q,i}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
      if $\mathbf{c}_{i+1} = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k \in [N_{in}+1]}, \mu, \{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k \in [N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k \in [N_{in}+1]}\Big)$ **then** Continue
      else if $\|\mathbf{t}_i^{(k)}\|_2 \leq \beta_v$ (the acceptance bound based on [DDLL13]) **then** Continue
      else if $\|\mathbf{t}_i^{(k)}\|_{\infty} < q/4$ **then** Continue
  if $\mathbf{c}_1 = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k \in [N_{in}+1]}, \mu, \{\mathbf{A}_{2q,w}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_w\}_{k \in [N_{in}+1]}, \{\mathbf{h}_{2q}^{(k)} \cdot \mathbf{t}_w^{(k)} + \mathbf{q} \cdot \mathbf{c}_w\}_{k \in [N_{in}+1]}\Big)$ **then** Accept
  else Reject
  return Accept or Reject

Theorem 5.8. *Let* $\beta_v = \eta\sigma\sqrt{nm}$ *and* $q/4 > \left(\sqrt{2(\lambda+1)\ln 2 + 2\ln(nm)}\right)\sigma$ *and* $\sigma_{L'}(\mu) = \left(\boldsymbol{c}_1, \left\{\boldsymbol{t}_1^{(k)}, \ldots, \boldsymbol{t}_w^{(k)}\right\}_{k \in [N_{in}+1]}, \left\{\boldsymbol{h}^{(k)}\right\}_{k \in [N_{in}]}\right)$ *be generated based on Algorithm 11. Then the output of Algorithm 12 on input* $\sigma_{L'}(\mu)$ *is accepted with probability* $1 - 2^{-\lambda}$.

Proof. In this proof, we start mentioning that in BLISS [DDLL13], for a desired expected rejection and repetition M, if we take the definition of α where $M = e^{\frac{1}{2\alpha^2}}$, then $\mathbf{t}_{\pi}^{(k)}$ will be indistinguishable from D_σ if $\sigma \geq \alpha \cdot \|\mathbf{S}_{2q,\pi}^{(k)} \cdot \mathbf{c}_\pi\|$ [Section 3.2 in [DDLL13]]. We also use [lemma 4.4, parts 1 and 3, in [Lyu12]]. The part 3 of this lemma shows that the bound on Euclidean norm $\beta_v = \eta\sigma\sqrt{nm}$, for a given $\eta > 1$, has a probability $\Pr[\|\mathbf{t}_i^{(k)}\|_2 > \eta\sigma\sqrt{nm}] \geq 1 - 2^{\lambda}$. In addition, the bound on infinity norm ($\|\mathbf{t}_i\|_\infty < q/4$) is analysed in part 1 of this lemma where its union bound is also considered. It turns out that η is required such $q/4 > \eta\sigma > \left(\sqrt{2(\lambda+1)\ln 2 + 2\ln(nm)}\right)\sigma$, except with probability of $2^{-\lambda}$. □

5.2.5 Signature Linkability - MIMO.L2RS.SigLink

The MIMO.L2RS.SigLink algorithm, illustrated in Algorithm 13, takes two signatures as input: $\sigma_L(\mu_1)$ and $\sigma'_{L'}(\mu_2)$, and it outputs either *Linked* if these signatures were generated by same signatory, or *Unlinked*, otherwise. Given public-keys' lists L and L', and two signatures: $\sigma_L(\mu_1)$ and $\sigma'_{L'}(\mu_2)$, which can be described as: $\sigma_L(\mu_1) = \left(\mathbf{c}_{1,\mu_1}, \{\mathbf{t}_{1,\mu_1}^{(k)}, \ldots, \mathbf{t}_{w,\mu_1}^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}_{\mu_1}^{(k)}\}_{k\in[N_{in}]}\right)$ and $\sigma'_{L'}(\mu_2) = \left(\mathbf{c}_{1,\mu_2}, \{\mathbf{t}_{1,\mu_2}^{(k)}, \ldots, \mathbf{t}_{w,\mu_2}^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}_{\mu_2}^{(k)}\}_{k\in[N_{in}]}\right)$.

These two signatures must be successfully accepted by the MIMO.L2RS.SigVer algorithm, then one can verify that the linkability property is achieved if the linkability tags ($\mathbf{h}_{\mu_1}^{(k)}$ and $\mathbf{h}_{\mu_2}^{(k)}$) of the above signatures $\sigma_L(\mu_1)$ and $\sigma'_{L'}(\mu_2)$ are equal.

Algorithm 13 L2RS.SigLink - Signature Linkability

Input: $\sigma_L(\mu_1)$ and $\sigma'_{L'}(\mu_2)$
Output: Linked or Unlinked

1: **procedure** MIMO.L2RS.SigLink($\sigma_L(\mu_1), \sigma'_{L'}(\mu_2)$)
2: **if** $\Big($MIMO.L2RS.SigVer($\sigma_L(\mu_1)$) = Accept **and** MIMO.L2RS.SigVer($\sigma'_{L'}(\mu_2)$) = Accept$\Big)$ **then** Continue [
3: **else if** $\mathbf{h}_{\mu_1}^{(k)} = \mathbf{h}_{\mu_2}^{(k)}$ **then** Linked
4: **else** Unlinked]
5: **return** Linked or Unlinked

5.2.5.1 Correctness of MIMO.L2RS.SigLink

Proof. We show that an honest user π who signs two messages μ_1 and μ_2 in the MIMO.L2RS scheme with the list of public-keys L, obtains a *Linked* output from MIMO.L2RS.SigLink algorithm with overwhelming probability. As shown in Algorithm 13, two signatures $\sigma_L(\mu_1)$ and $\sigma_L(\mu_2)$ were created, and then successfully verified by MIMO.L2RS.SigVer. Therefore, the linkability tags $\mathbf{h}_{\mu_1}^{(k)}$ and $\mathbf{h}_{\mu_2}^{(k)}$ must be equal. To prove this, we show that:

$$\begin{gathered}
\mathbf{H}_{2q,\mu_1}^{(k)} = \left(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\mu_1}^{(k)} + \mathbf{q}\right) \in \mathcal{R}_{2q}^{2\times m}, \text{where} \\
\mathbf{H} = \mathsf{Pub\text{-}Param} \text{ and } \mathbf{h}_{\mu_1}^{(k)} = (\mathbf{H} \cdot \mathbf{S}_{(in),\pi}^{(k)} + \mathbf{q}) \in \mathcal{R}_q^2 \\
\mathbf{H}_{2q,\mu_2}^{(k)} = \left(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\mu_2}^{(k)} + \mathbf{q}\right) \in \mathcal{R}_{2q}^{2\times m}, \text{where} \\
\mathbf{H} = \mathsf{Pub\text{-}Param} \text{ and } \mathbf{h}_{\mu_2}^{(k)} = (\mathbf{H} \cdot \mathbf{S}_{(in),\pi}^{(k)} + \mathbf{q}) \in \mathcal{R}_q^2
\end{gathered}$$

The first parts of the linkability tag in both MIMO.L2RS signatures have same equality with following probability:

$$Pr\left[2 \cdot \mathbf{H} = 2 \cdot \mathbf{H}\right] = 1.$$

Ultimately, the second part uses the honest user's private-key $\mathbf{S}_{(in),\pi}^{(k)}$ is used, so we conclude that:

$$Pr\left[-2 \cdot \mathbf{h}_{\mu_1}^{(k)} + \mathbf{q} + 2 \cdot \mathbf{h}_{\mu_2}^{(k)} - \mathbf{q} = 0\right] = 1.$$

□

5.3 MIMO.L2RS - Security Analysis

Theorem 5.9 (One-Time Unforgeability)**.** *Suppose* $\sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ *is negligible in* n, $\frac{1}{|\mathcal{S}_{n,\kappa}|}$ *is negligible and* $y = h$ *is polynomial in* n*, where* h *denotes the number*

of queries to the random oracle H_1. If there is a PPT algorithm against one-time unforgeability of MIMO.L2RS with non-negligible probability δ, then there exist a PPT algorithm that can extract a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ *problem (with $\beta = 2\beta_v$) with non-negligible probability* $\left(\delta - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \cdot \left(\frac{\delta - \frac{1}{|\mathcal{S}_{n,\kappa}|}}{y} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) - \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$.

Proof. As stated in [DDLL13], this MIMO.L2RS scheme relies on the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem to be secure against any existential forger. This means that a forgery algorithm succeeds with a negligible probability and so we conclude that under this probability, the attacker will also find a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem. To prove this, we start replacing the MIMO.L2RS.SigGen algorithm with MIMO.L2RS.Hybrid-1 and MIMO.L2RS.Hybrid-2 algorithms that are used to simulate the creation of the signatures, until we obtain an algorithm that breaks the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem. These Hybrid algorithms are illustrated in Algorithm 14 and Algorithm 15, respectively.

In MIMO.L2RS.Hybrid-1, the output of the random oracle H_1 is chosen at random from $\mathcal{S}_{n,\kappa} \subseteq \mathcal{R}_{2q}$ and then it is programmed, without checking the value of $\mathbf{A}^{(k)}_{2q,\pi} \cdot \mathbf{u}^{(k)}$ and $\mathbf{H}^{(k)}_{2q} \cdot \mathbf{u}^{(k)}$ being already set. This equality can be described as:

$$H_1\Big(L', \{\mathbf{H}^{(k)}_{2q}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}^{(k)}_{2q,w} \cdot \mathbf{t}^{(k)}_i + \mathbf{q}\cdot\mathbf{c}_w\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}_{2q} \cdot \mathbf{t}^{(k)}_w + \mathbf{q}\cdot\mathbf{c}_w\}_{k\in[N_{in}+1]}\Big) =$$
$$H_1\Big(L', \{\mathbf{H}^{(k)}_{2q}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}^{(k)}_{2q,\pi} \cdot \mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{H}^{(k)}_{2q} \cdot \mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}\Big)$$

Every time the MIMO.L2RS.Hybrid-1 is called, the probability of generating $\mathbf{u}$, (such that $\mathbf{A}^{(k)}_{2q,\pi} \cdot \mathbf{u}^{(k)}$ and $\mathbf{H}^{(k)}_{2q} \cdot \mathbf{u}^{(k)}$ are equal to one of the previous output that was queried), is at most 2^{-n+1}. We define that the probability of getting a collusion each time is at most $h \cdot 2^{-n+1}$, where "h" is the number of calls to the random oracle H_1, whereas the probability of occurring a collision after "o" queries to the MIMO.L2RS.Hybrid-1 is at most $o \cdot h \cdot 2^{-n+1}$, which is negligible (Based on [DDLL13], Lemma 3.4).

After analyzing how $\mathbf{c}_1$ can be forged, we evaluate the $\mathbf{t}^{(k)}_1, \ldots, \mathbf{t}^{(k)}_w$ of the MIMO.L2RS scheme. We claim that these are forgeable when an attacker finds

Algorithm 14 MIMO.L2RS.Hybrid-1

Input: $\{\mathbf{S}^{(k)}_{(in),\pi}\}_{k\in[N_{in}+1]}, \mu$, L' as in (5.13), and Pub-Params.
Output: $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \{\mathbf{t}^{(k)}_1, \ldots, \mathbf{t}^{(k)}_w\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\Big)$

1: **procedure** MIMO.L2RS.HYBRID-1($\mathbf{S}^{(k)}_{(in),\pi}, \mu, L'$, Pub-Params)
2: **for** $(1 \le k \le N_{in}+1)$ **do**
3: Set $\mathbf{H}^{(k)}_{2q} = (2\cdot\mathbf{H}, -2\cdot\mathbf{h}^{(k)} + \mathbf{q}) \in \mathcal{R}^{2\times m}_{2q}$, where $\mathbf{h}^{(k)} = \mathbf{H}\cdot\mathbf{S}^{(k)}_{(in),\pi} \in \mathcal{R}^2_q$.
4: Call L2RS.Lift$(\mathbf{A}, \mathbf{a}^{(k)}_{(in),\pi})$ to obtain $\mathbf{A}^{(k)}_{2q,\pi} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}^{(k)}_{(in),\pi} + \mathbf{q}) \in \mathcal{R}^{2\times m}_{2q}$.
5: Let $\mathbf{u}^{(k)} = (u_1, \ldots, u_m)^T$, where $u_i \hookleftarrow D^n_\sigma$, for $1 \le i \le m$.
6: Choose at random $\mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}$
.
7: **for** $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
8: **for** $(1 \le k \le N_{in}+1)$ **do**
9: Call L2RS.Lift$(\mathbf{A}, \mathbf{a}^{(k)}_{(in),i})$ to obtain $\mathbf{A}^{(k)}_{2q,i} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}^{(k)}_{(in),i} + \mathbf{q}) \in \mathcal{R}^{2\times m}_{2q}$.
10: Let $\mathbf{t}^{(k)}_i = (t_{i,1}, \ldots, t_{i,m})^T$, where $t_{i,j} \hookleftarrow D^n_\sigma$, for $1 \le j \le m$.
11: Compute $\mathbf{c}_{i+1} = H_1\Big(L', \{\mathbf{H}^{(k)}_{2q}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}^{(k)}_{2q,i}\cdot\mathbf{t}^{(k)}_i + \mathbf{q}\cdot\mathbf{c}_i\}_{k\in[N_{in}+1]}, \{\mathbf{H}^{(k)}_{2q}\cdot\mathbf{t}^{(k)}_i + \mathbf{q}\cdot\mathbf{c}_i\}_{k\in[N_{in}+1]}\Big)$.
12: **for** $(1 \le k \le N_{in}+1)$ **do**
13: Choose $b^{(k)} \hookleftarrow \{0,1\}$.
14: Let $\mathbf{t}^{(k)}_\pi \leftarrow \mathbf{u}^{(k)} + \mathbf{S}^{(k)}_{2q,\pi}\cdot\mathbf{c}_\pi\cdot(-1)^{b^{(k)}}$, where $\mathbf{S}^{(k)}_{2q,\pi} = [(\mathbf{S}^{(k)}_\pi)^T, 1]^T$.
15: **Continue** with prob. $\left(M\exp\left(-\frac{\|\mathbf{S}^{(k)}_{2q,\pi}\cdot\mathbf{c}_\pi\|^2}{2\sigma^2}\right)\cosh\left(\frac{\langle\mathbf{t}^{(k)}_\pi, \mathbf{S}^{(k)}_{2q,\pi}\cdot\mathbf{c}_\pi\rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
16: **return** $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \{\mathbf{t}^{(k)}_1, \ldots, \mathbf{t}^{(k)}_w\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\Big)$.

a PPT algorithm $\mathcal{F}$ to solve the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem. This attack can be simulated using the MIMO.L2RS.Hybrid-2 shown in Algorithm 15, where $\mathbf{t}_\pi$ is directly chosen from the distribution D^n_σ (Based on [DDLL13], Lemma 3.5).

The public-key $\mathbf{A}^{(k)}_{2q,\pi} \in \mathcal{R}^{2\times m}_{2q}$ is generated such $\mathbf{A}^{(k)}_{2q,\pi}\cdot\mathbf{S}^{(k),T}_{(in),\pi} = q \in \mathcal{R}^2_{2q}$, so finding a vector $\mathbf{v}$ such that $\mathbf{A}^{(k)}_{2q,\pi}\cdot\mathbf{v} = \mathbf{0} \bmod q$ with $\mathbf{0} = (0,0)^T$. We denote $y = h$ where y is the number of times the random oracle H_1 is programmed during this attack. Then this attack is performed as follows:

1. Random coins are selected for the forger ϕ and signer ψ.
2. The random oracle H_1 is called to generate the responses of the users in the L2RS scheme, $(\mathbf{c}_1, \ldots, \mathbf{c}_w) \hookleftarrow \mathcal{S}_{n,\kappa}$.
3. These create a SubRoutine that takes as input $(\mathbf{A}^{(k)}_{2q,\pi}, \phi, \psi, \mathbf{c}_1, \ldots, \mathbf{c}_w)$.
4. $\mathcal{F}$ is initialized and run by providing the $\mathbf{A}^{(k)}_{2q,\pi}$ and forger's random coins ϕ.
5. The SubRoutine signs the message μ using the signer's coins ψ in the MIMO.L2RS.Hybrid-2, this produces a signature $\sigma_L(\mu)$.

6. During the signing process, $\mathcal{F}$ calls the oracle H_1 and answers are placed in the list $(\mathbf{c}_1, \ldots, \mathbf{c}_w)$, the queries are kept in a table in the event that same queries are used in this oracle.

7. $\mathcal{F}$ is stopped and it outputs a forgery that is the SubRoutine's result $(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]})$, with negligible probability δ. This output has to be successfully accepted by the MIMO.L2RS.SigVer algorithm.

If the random oracle was not called using some input $\{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}$, then $\mathcal{F}$ has $1/|\mathcal{S}_{n,\kappa}|$ chances of producing a $\mathbf{c}$ such that $\mathbf{c} = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q}^{(k)} \cdot \mathbf{t}^{(k)} + \mathbf{q} \cdot \mathbf{c}\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}^{(k)} + \mathbf{q} \cdot \mathbf{c}\}_{k\in[N_{in}+1]}\Big)$. This turns out that $\delta - 1/|\mathcal{S}_{n,\kappa}|$ be the probability that $\mathbf{c} = \mathbf{c}_j$ for some j.

Algorithm 15 MIMO.L2RS.Hybrid-2

Input: $\{\mathbf{S}_{(in),\pi}^{(k)}\}_{k\in[N_{in}+1]}, \mu$, L' as in (5.13), and Pub-Params.

Output: $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\Big)$

procedure MIMO.L2RS.Hybrid-2($\mathbf{S}_{(in),\pi}^{(k)}, \mu, L'$, Pub-Params)
  for $(1 \le k \le N_{in}+1)$ **do**
    Set $\mathbf{H}_{2q}^{(k)} = \big(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}^{(k)} + \mathbf{q}\big) \in \mathcal{R}_{2q}^{2\times m}$, where $\mathbf{h}^{(k)} = \mathbf{H} \cdot \mathbf{S}_{(in),\pi}^{(k)} \in \mathcal{R}_q^2$.
    Call L2RS.Lift$(\mathbf{A}, \mathbf{a}_{(in),\pi}^{(k)})$ to obtain $\mathbf{A}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_{(in),\pi}^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
    Let $\mathbf{u}^{(k)} = (u_1, \ldots, u_m)^T$, where $u_i \hookleftarrow D_\sigma^n$, for $1 \le i \le m$.
  Choose at random $\mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \le k \le N_{in}+1)$ **do**
      Call L2RS.Lift$(\mathbf{A}, \mathbf{a}_{(in),i}^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_{(in),i}^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
      Let $\mathbf{t}_i^{(k)} = (t_{i,1}, \ldots, t_{i,m})^T$, where $t_{i,j} \hookleftarrow D_\sigma^n$, for $1 \le j \le m$.
    Compute $\mathbf{c}_{i+1} = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\}_{k\in[N_{in}+1]}\Big)$.
  for $(1 \le k \le N_{in}+1)$ **do**
    Choose $b^{(k)} \hookleftarrow \{0, 1\}$.
    Choose $\mathbf{t}_\pi^{(k)} \hookleftarrow D_\sigma^{n\times m}$
    Continue with probability $\frac{1}{M}$ otherwise **Restart**.
  return $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\Big)$.

Forgery 1. Let's consider the situation that $\mathbf{c}_{j+1}$ is the result after using $\mathcal{F}$ which is $\mathbf{c}_{j+1} = H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu', \{\mathbf{A}_{2q}^{(k)} \cdot \mathbf{t}'^{(k)} + \mathbf{q} \cdot \mathbf{c}_j\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}'^{(k)} + \mathbf{q} \cdot \mathbf{c}_j\}_{k\in[N_{in}+1]}\Big)$. Then by comparing this with a legitimate signature, we have:

$$H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q}^{(k)}\cdot\mathbf{t}^{(k)}+\mathbf{q}\cdot\mathbf{c}_j\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}^{(k)}+\mathbf{q}\cdot\mathbf{c}_j\}_{k\in[N_{in}+1]}\Big) =$$
$$H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu', \{\mathbf{A}_{2q}^{(k)}\cdot\mathbf{t}'^{(k)}+\mathbf{q}\cdot\mathbf{c}_j\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}'^{(k)}+\mathbf{q}\cdot\mathbf{c}_j\}_{k\in[N_{in}+1]}\Big)$$

$\mathcal{F}$ will find a preimage of $\mathbf{c}_j$ if $\mu \neq \mu'$ or $\mathbf{A}_{2q}^{(k)} \cdot \mathbf{t}^{(k)} + \mathbf{q} \cdot \mathbf{c}_j \neq \mathbf{A}_{2q}^{(k)} \cdot \mathbf{t}'^{(k)} + \mathbf{q} \cdot \mathbf{c}_j$ or $\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}^{(k)} + \mathbf{q} \cdot \mathbf{c}_j \neq \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}'^{(k)} + \mathbf{q} \cdot \mathbf{c}_j$. Then, we have with overwhelming probability that $\mu = \mu'$ and $\mathbf{A}_{2q}^{(k)} \cdot \mathbf{t}^{(k)} + \mathbf{q} \cdot \mathbf{c}_j = \mathbf{A}_{2q}^{(k)} \cdot \mathbf{t}'^{(k)} + \mathbf{q} \cdot \mathbf{c}_j$ and $\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}^{(k)} + \mathbf{q} \cdot \mathbf{c}_j = \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}'^{(k)} + \mathbf{q} \cdot \mathbf{c}_j$. These equalities will result in: $\mathbf{A}_{2q}^{(k)}(\mathbf{t}^{(k)} - \mathbf{t}'^{(k)}) = 0 \bmod q$ and $\mathbf{H}_{2q}^{(k)}(\mathbf{t}^{(k)} - \mathbf{t}'^{(k)}) = 0 \bmod q$. We assume that both $\mathbf{t}$ and $\mathbf{t}'$ are different and they met the MIMO.L2RS.SigVer conditions, so it yields $\mathbf{t} - \mathbf{t}' \neq 0 \bmod q$, and $\|\mathbf{t} - \mathbf{t}'\| \leq 2\beta_v$.

Forgery 2. In this scenario, we assume that the MIMO.L2RS scheme can be forged by an attacker $\mathcal{F}$ as it was presented in the Forgery 1 and obtain $\mathbf{c}_j$, then another attacker can generate $(\mathbf{c}'_j, \ldots, \mathbf{c}'_w) \hookleftarrow \mathcal{S}_{n,\kappa}$ by replaying the first attack and using same message μ. We use the forking lemma [BN06] to show the probability of $\mathbf{c}_j = \mathbf{c}'_j$ and the forger uses an oracle response $\mathbf{c}'_j$ is at least:

$$\left(\delta - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \cdot \left(\frac{\delta - \frac{1}{|\mathcal{S}_{n,\kappa}|}}{y} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \tag{5.3}$$

Therefore, with the probability (5.3), $\mathcal{F}$ creates a signature $\sigma_L(\mu) = \Big(\mathbf{c}'_1, \{\mathbf{t}_1'^{(k)}, \ldots, \mathbf{t}_w'^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\Big)$ where $\mathbf{A}_{2q}^{(k)}\cdot\mathbf{t}^{(k)}+\mathbf{q}\cdot\mathbf{c}_j = \mathbf{A}_{2q}^{(k)}\cdot\mathbf{t}'^{(k)}+\mathbf{q}\cdot\mathbf{c}_j$ and $\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}^{(k)} + \mathbf{q} \cdot \mathbf{c}_j = \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}'^{(k)} + \mathbf{q} \cdot \mathbf{c}_j$. We now obtained: $\mathbf{A}_{2q}^{(k)} \cdot (\mathbf{t}^{(k)} - \mathbf{t}'^{(k)}) = \mathbf{q}(\mathbf{c}_j - \mathbf{c}'_j) \bmod 2q$ and $\mathbf{H}_{2q}^{(k)} \cdot (\mathbf{t}^{(k)} - \mathbf{t}'^{(k)}) = \mathbf{q}(\mathbf{c}_j - \mathbf{c}'_j) \bmod 2q$. Since $\mathbf{c}_j - \mathbf{c}'_j \neq 0 \bmod 2$, so in both equations, we have $\mathbf{t}^{(k)} - \mathbf{t}'^{(k)} \neq 0 \bmod 2q$ where $\|\mathbf{t}^{(k)} - \mathbf{t}'^{(k)}\|_\infty < q/2$. By applying $\bmod q$ reduction, we find a small non-zero vector $\mathbf{v}^{(k)} = \mathbf{t}^{(k)} - \mathbf{t}'^{(k)} \neq 0 \bmod q$. This $\mathbf{v}^{(k)}$ will compute $\mathbf{A}_{2q}^{(k)} \cdot \mathbf{v}^{(k)} = \mathbf{0} \bmod q$ and $\mathbf{H}_{2q}^{(k)} \cdot \mathbf{v}^{(k)} = \mathbf{0} \bmod q$ with $\|\mathbf{v}^{(k)}\| \leq 2\beta_v$. Since $\mathbf{v}^{(k)}$ is same for both $\mathbf{A}_{2q}^{(k)}$ and $\mathbf{H}_{2q}^{(k)}$, we only use the former to continue this analysis. We say that $\mathbf{A}_{2q}^{(k)} \bmod q = 2(\mathbf{A}, -\mathbf{a}^{(k)}) \bmod q$, then $2(\mathbf{A}, -\mathbf{a}^{(k)})\mathbf{v}^{(k)} = \mathbf{0} \bmod q$, this implies that $(\mathbf{A}, -\mathbf{a}^{(k)})\mathbf{v}^{(k)} = \mathbf{0} \bmod q$, since q is odd. The probability of success of an attacker in MIMO.L2RS.Hydrid-3 differs by a negligible amount from the success

probability in MIMO.L2RS.KeyGen and is thus non-negligible. Therefore, this vector $\mathbf{v}$ will be a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem, where $\beta = 2\beta_v$, with non-negligible probability and with respect to $(\mathbf{A}, -\mathbf{a}^{(k)})$ over $\mathcal{R}_q^2$. Furthermore, notice that MIMO.L2RS.Hybrid-2 shown in Algorithm 15 no longer uses the private-key $\mathbf{S}_{\pi}^{(k)}$, except for generating $\mathbf{A}_{2q,\pi}^{(k)}$ and $\mathbf{H}_{2q}^{(k)}$ to obtain the final $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ solution. For $\mathbf{A}_{2q,\pi}^{(k)}$, we modified the MIMO.L2RS.KeyGen algorithm with the MIMO.L2RS.Hydrid-3 game shown in Algorithm 16, where the public-key $\mathbf{a}^{(k)}$ is uniformly and randomly taken as $\mathbf{a}^{(k)} \hookleftarrow \mathcal{R}_q^2$. On the other hand, for $\mathbf{H}_{2q}^{(k)}$, we chose the linking taq uniformly and randomly as $\mathbf{h}^{(k)} \hookleftarrow \mathcal{R}_q^2$. By the argument of the Leftover Hash Lemma (LHL) - Lemma 3.5 and our assumption that $\sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ is negligible in n.

Algorithm 16 MIMO.L2RS.Hybrid-3 $(\mathbf{a}, \mathbf{S})$

Input: Pub-Param: $\mathbf{A}$.

Output: $(\mathbf{a}, \mathbf{S})$, being the public-key and the private-key, respectively.

1: **procedure** MIMO.L2RS.HYBRID-3($\mathbf{A}$)
2: Let $\mathbf{S}^T = (\mathbf{s}_1, \ldots, \mathbf{s}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\mathbf{s}_i \hookleftarrow (-2^{\gamma}, 2^{\gamma})^n$, for $1 \leq i \leq m-1$
3: Choose $\mathbf{a} \hookleftarrow \mathcal{R}_q^2$
4: **return** $(\mathbf{a}, \mathbf{S})$.

□

Theorem 5.10 (Anonymity)**.** *Suppose* $\sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ *is negligible in* n *with an attack against the unconditional anonymity that makes* h *queries to the random oracle* H_1*, where* h*,* w *are polynomial in* n*, then the* MIMO.L2RS *scheme is unconditionally secure for anonymity as defined in Definition 5.2.*

Proof. We prove the anonymity of this scheme using the sequence-of-games approach [Sho04] where we make changes between successive games. In doing so, we use the "*transition based on indistinguishability*". We can start this analysis by:

Game 0: Suppose that an attacker $\mathcal{A}$ is given the list of pk's $L = \{\mathbf{a}_0^{(k)}, \mathbf{a}_1^{(k)}\}_{k\in[N_{in}+1]}$, the signature $\sigma_L(\mu)$, message μ, and the random oracle model

(H_1). The key generation algorithm creates the pair of users' keys in this ring signature: Private-Keys $\hookleftarrow \{\mathbf{S}_0^{(k)}, \mathbf{S}_1^{(k)}\}_{k\in[N_{in}+1]}$ and the Public-Keys $\hookleftarrow (\mathbf{a}_0^{(k)}, \mathbf{a}_1^{(k)})$; a user b is chosen uniformly at random from the list $L = \{\mathbf{a}_0^{(k)}, \mathbf{a}_1^{(k)}\}$, then the signature $\sigma_L(\mu) = \mathsf{MIMO.L2RS.SigGen}(\mathbf{S}_b^{(k)}, \mu, L, \mathsf{Pub\text{-}Param})$ is generated. So in **Game 0**, a PPT adversary $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$; thus in the event **Game 0**, $\mathcal{A}$ succeeds in breaking ambiguity **Game 0**$(b = b')$ if $\Pr[\textbf{Game 0}] \leq \frac{1}{2} + \textit{non-negl}(\lambda)$.

Game 1: Changes in this game are made to the user π in the second part of the linkability tag $\mathbf{h}^{(k)} = (\mathbf{H}\cdot\mathbf{S}^{(k)}) \in \mathcal{R}_q^2$, in signature of user π, and public-key $\mathbf{a}^{(k)} = (\mathbf{A}\cdot\mathbf{S}^{(k)}) \in \mathcal{R}_q^2$ in the MIMO.L2RS.KeyGen algorithm. The $\mathbf{h}^{(k)}$ and $\mathbf{a}^{(k)}$ are now randomly chosen from $\mathcal{R}_q^2$. We claim that $|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq \epsilon_{LHL_{G1}}$.

Where $\epsilon_{LHL_{G1}}$ is the advantage of some efficient algorithm which is negligible. In both cases $\mathbf{h}^{(k)} = (\mathbf{H}\cdot\mathbf{S}^{(k)}) \in \mathcal{R}_q^2$ and $\mathbf{a}^{(k)} = (\mathbf{A}\cdot\mathbf{S}^{(k)}) \in \mathcal{R}_q^2$, we know that $\mathbf{H}$ and $\mathbf{A}$ are uniform and $\mathbf{S}^{(k)}$ is chosen small and with coefficients in $(-2^\gamma, 2^\gamma)$. When $\mathbf{S}^{(k)}$ is multiplied by $\mathbf{H}$ and $\mathbf{A}$ respectively, it gives $\mathbf{h}^{(k)}$ and $\mathbf{a}^{(k)}$ that are close to uniform over $\mathcal{R}_q^2$. By applying the Leftover Hash Lemma (LHL) - **Lemma 3.5**, the statistical distance between the distribution of ($\mathbf{h}^{(k)} \bmod q$ and $\mathbf{a}^{(k)} \bmod q$) and the uniform distribution on $\mathcal{R}_q^2 \times \mathcal{R}_q^2$ is at most $n \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$. We conclude that in **Game 1**:

$$|\Pr[\textbf{Game 0}] - \Pr[\textbf{Game 1}]| \leq n \cdot \tfrac{1}{2} \cdot \sqrt{\tfrac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}. \tag{5.4}$$

Game 2: This time a change is made in the second part of the remaining public-keys $\mathbf{a}_i$ $(1 \leq i \leq w,\ i \neq \pi)$ which are in the ring signature list L. They are now randomly chosen as $\mathbf{a}_i^{(k)} \hookleftarrow \mathcal{R}_q^2$. It turns out that $|\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 2}]| \leq \epsilon_{LHL_{G2}}$.

Where $\epsilon_{LHL_{G2}}$ is the advantage of some efficient algorithm which is negligible. We consider that for ($i = 1$ to w where $i \neq \pi$), we know that $\mathbf{a}_i^{(k)} = (\mathbf{A}\cdot\mathbf{S}_i^{(k)} \bmod q)$ are uniform and all $\mathbf{S}_i^{(k)}$'s are chosen small with coefficients in $(-2^\gamma, 2^\gamma)$. When the $\mathbf{S}_i^{(k)}$'s are multiplied by $\mathbf{A}_i$'s, it gives ($\mathbf{a}_i^{(k)} \bmod q$)'s that are close to uniform

over $\mathcal{R}_q^2$. By applying the Leftover Hash Lemma (LHL) - **Lemma 3.5**, the statistical distance between the distribution of the $(\mathbf{A} \cdot \mathbf{S}_i^{(k)} \bmod q)$'s and the uniform distribution on $\mathcal{R}_q^2 \times \mathcal{R}_q^2$ is at most $n \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \cdot (w-1)$. So in **Game 2**, we conclude that:

$$|\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 2}]| \leq n \cdot \tfrac{1}{2} \cdot \sqrt{\tfrac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \cdot (w-1). \tag{5.5}$$

Game 3: At this time, we make a change in $\mathbf{c}_{\pi+1}$. Instead of programming the oracle as $H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}\Big)$, it is now randomly chosen $\mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}$. We have that $|\Pr[\textbf{Game 2}] - \Pr[\textbf{Game 3}]| \leq \epsilon_{G3}$ where ϵ_{G3} is the advantage of some efficient algorithm which is negligible. This scenario outputs a signature $\sigma_{L'}(\mu) = \Big(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\Big)$ and programs the oracle as $H_1\Big(L', \{\mathbf{H}_{2q}^{(k)}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}, \{\mathbf{H}_{2q}^{(k)} \cdot \mathbf{u}^{(k)}\}_{k\in[N_{in}+1]}\Big) = \mathbf{c}_{\pi+1}$. Then, the adversary $\mathcal{A}$ makes h queries to H_1; so the distinguishing advantage of the signing algorithm and the one in **Game 2** is at most $h \cdot 2^{-n+1}$. We conclude that in **Game 3**:

$$|\Pr[\textbf{Game 2}] - \Pr[\textbf{Game 3}]| \leq h \cdot 2^{-n+1}. \tag{5.6}$$

Game 4: In this game a change is made in $\mathbf{t}_\pi^{(k)}$. Namely, instead of computing it as $\mathbf{u}^{(k)} + \mathbf{S}_{2q,\pi}^{(k)} \cdot \mathbf{c}_\pi \cdot (-1)^{bit}$, it is now directly chosen from the Gaussian distribution D_σ^n. It is argued that $|\Pr[\textbf{Game 3}] - \Pr[\textbf{Game 4}]| \leq \epsilon_{RS_{G4}}$.

Where $\epsilon_{RS_{G4}}$ is the advantage of some efficient algorithm which is negligible. In previous Games, $\mathbf{t}_\pi^{(k)}$ is computed using rejection sampling - **Lemma 3.8**, thus it is always sample from the Gaussian distribution D_σ^n. In this Game, however, $\mathbf{t}_\pi^{(k)}$ is directly chosen from D_σ^n, this means that the advantage $\epsilon_{RS_{G4}}$ will be zero as in both **Game 3** and **Game 4**, $\mathbf{t}_\pi^{(k)}$ is having same distribution. In **Game 4**, we have:

$$|\Pr[\textbf{Game 3}] - \Pr[\textbf{Game 4}]| = 0. \tag{5.7}$$

Game 5: Finally, in the **Game 5**, a change is made in the index π. Namely, instead of choosing $\pi + 1$, it will be randomly chosen $(1, \ldots, w)$. We claim that

$|\Pr[\textbf{Game 4}] - \Pr[\textbf{Game 5}]| \leq \epsilon_{G5}$ where ϵ_{G5} is the advantage of some efficient algorithm which is negligible. In this **Game 5**, we consider that when π is replaced by a fixed d, it might produce some collisions with previous queries to the oracle H_1; saying this, the adversary $\mathcal{A}$ may make h queries to H_1; therefore, the distinguishing advantage of the signing algorithm between **Game 4** and this **Game 5** is at most $h \cdot 2^{-n+1} \cdot w$. Finally, in **Game 5** we have:

$$|\Pr[\textbf{Game 4}] - \Pr[\textbf{Game 5}]| \leq h \cdot 2^{-n+1} \cdot w. \tag{5.8}$$

We also conclude that in **Game 5**, the adversary's view is statistical independent of π, thus $\Pr[\textbf{Game 5}] = \frac{1}{w}$.

Combining the probabilities of the above games (5.4), (5.5), (5.6), (5.7) and (5.8) we obtain:

$$|\Pr[\textbf{Game 5}] - \Pr[\textbf{Game 0}]| \leq |\Pr[\textbf{Game 1}] - \Pr[\textbf{Game 0}]| + |\Pr[\textbf{Game 2}] - \Pr[\textbf{Game 1}]| + |\Pr[\textbf{Game 3}] - \Pr[\textbf{Game 2}]| + |\Pr[\textbf{Game 4}] - \Pr[\textbf{Game 3}]| + |\Pr[\textbf{Game 5}] - \Pr[\textbf{Game 4}]|.$$

By replacing the resulting probabilities, we have:

$$|\Pr[\textbf{Game 5}] - \Pr[\textbf{Game 0}]| \leq \frac{1}{w} - \frac{1}{2} + \epsilon, \tag{5.9}$$

which means that $|\Pr[\textbf{Game 5}] - \Pr[\textbf{Game 0}]| \leq \epsilon$, which itself is smaller than

$$\frac{n \cdot (w-1)}{2} \cdot \left(\sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} + \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \right) + h \cdot 2^{-n+1} \cdot (1+w).$$

We notice that since h and w are polynomial in n, we get $h \cdot 2^{-n+1} \cdot (1+w)$ is negligible in n. In addition, we can say that $\left(\sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} + \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \right) \leq 2 \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$, which is negligible by the assumption that $\sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ is also

negligible. Hence we conclude that ϵ is negligible, meaning that $\Pr[\textbf{Game 0}] \leq \frac{1}{2} + \epsilon$.

□

Theorem 5.11 (Linkability)**.** *The MIMO.L2RS scheme with parameter β_v is linkable in the random oracle model if the* $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ *problem (with $\beta = 2\beta_v$) is hard.*

Proof. We construct the algorithm $\mathcal{B}$ for the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem. This algorithm runs the linkability attack game (Definition 5.3) as follows:

1. $\mathcal{B}$ generates using the MIMO.L2RS.KeyGen algorithm all private-keys $\mathbf{S}_i^{(k)}$'s with the corresponding public-keys $\mathbf{a}_i^{(k)}$'s, then $\mathcal{B}$ gives $\mathbf{S}_\pi^{(k)}$ to the attacker $\mathcal{A}$ as a response to the attacker's $\mathcal{CO}$ query.

2. $\mathcal{A}$ outputs two signatures $\sigma_L(\mu_1)$ and $\sigma'_{L'}(\mu')$ along with their corresponding lists L and L' such that both signatures are successfully verified by MIMO.L2RS.SigVer, but the linkability tags are different $\mathbf{h}_{\mu_1}^{(k)} \neq \mathbf{h}_{\mu'}^{(k)}$ with $k \in [N_{in}]$.

3. $\mathcal{B}$ computes $\mathbf{h}_{\mu_\pi}^{(k)} = \mathbf{H} \cdot \mathbf{S}_\pi^{(k)} \bmod q$, where π is the true signer's π linkability tag. This $\mathbf{h}_{\mu_\pi}^{(k)}$ tag can then be compared with the linkability tags $\mathbf{h}_{\mu_1}^{(k)}$ and $\mathbf{h}_{\mu'}^{(k)}$, output by $\mathcal{A}$, in step 2, and one of them will be different.

4. Without loss of generality, suppose $\mathbf{h}_{\mu_1}^{(k)} \neq \mathbf{h}_{\mu_\pi}^{(k)} \bmod q$. Using the forking lemma [BN06], $\mathcal{B}$ rewinds the attacker $\mathcal{A}$ to the H_1 query corresponding to the MIMO.L2RS.SigVer of the signature $\sigma_L(\mu_1)$. $\mathcal{B}$ reruns $\mathcal{A}$ with a different response of H_1 and ultimately gets another signature: $\sigma_L(\mu_2) = \left(\mathbf{c}_{1,\mu_2}, \left\{\mathbf{t}_{1,\mu_2}^{(k)}, \ldots, \mathbf{t}_{w,\mu_2}^{(k)}\right\}_{k\in[N_{in}+1]}, \left\{\mathbf{h}_{\mu_2}^{(k)}\right\}_{k\in[N_{in}]}\right)$. This second signature is used to extract a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem, in case the $\mathcal{A}$ finds an efficient way to unlink these signatures, as shown in step 7.

5. The adversary $\mathcal{A}$ matches the challenge message of both signatures where $\mathbf{H}_{2q,\mu_1}^{(k)}$ and $\mathbf{A}_{2q,w,\mu_1}^{(k)}$ are kept. Thus we have:

(a) $\mathbf{A}^{(k)}_{2q,w,\mu_1} \cdot \mathbf{t}^{(k)}_{w,\mu_1} + \mathbf{q} \cdot \mathbf{c}_{w,\mu_1} = \mathbf{A}^{(k)}_{2q,w,\mu_1} \cdot \mathbf{t}^{(k)}_{w,\mu_2} + \mathbf{q} \cdot \mathbf{c}_{w,\mu_2}$,

(b) $\mathbf{H}^{(k)}_{2q,\mu_1} \cdot \mathbf{t}^{(k)}_{w,\mu_1} + \mathbf{q} \cdot \mathbf{c}_{w,\mu_1} = \mathbf{H}^{(k)}_{2q,\mu_1} \cdot \mathbf{t}^{(k)}_{w,\mu_2} + \mathbf{q} \cdot \mathbf{c}_{w,\mu_2}$.

These expressions can be represented as:

(a) $\mathbf{A}^{(k)}_{2q,w,\mu_1} \cdot (\mathbf{t}^{(k)}_{w,\mu_1} - \mathbf{t}^{(k)}_{w,\mu_2}) = \mathbf{q} \cdot (\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1})$,

(b) $\mathbf{H}^{(k)}_{2q,\mu_1} \cdot (\mathbf{t}^{(k)}_{w,\mu_1} - \mathbf{t}^{(k)}_{w,\mu_2}) = \mathbf{q} \cdot (\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1})$.

Reducing them mod$q$ we have (if $(\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1}) \neq \mathbf{0} \bmod 2$):

(a) $\mathbf{A}^{(k)}_{2q,w,\mu_1} \cdot (\mathbf{t}^{(k)}_{w,\mu_1} - \mathbf{t}^{(k)}_{w,\mu_2}) = \mathbf{0} \bmod q$,

(b) $\mathbf{H}^{(k)}_{2q,\mu_1} \cdot (\mathbf{t}^{(k)}_{w,\mu_1} - \mathbf{t}^{(k)}_{w,\mu_2}) = \mathbf{0} \bmod q$.

We denote by $\mathbf{t}'^{(k)}_{w,\mu_1}$, the first $(m-1)$ ring elements in $\mathbf{t}^{(k)}_{w,\mu_1}$ and by $\mathbf{t}''^{(k)}_{w,\mu_1}$the m-th ring element in $\mathbf{t}^{(k)}_{w,\mu_1}$, i.e. $\mathbf{t}^{(k)}_{w,\mu_1} - \mathbf{t}^{(k)}_{w,\mu_2} = \begin{pmatrix} \mathbf{t}'^{(k)}_{w,\mu_1} - \mathbf{t}'^{(k)}_{w,\mu_2} \\ \mathbf{t}''^{(k)}_{w,\mu_1} - \mathbf{t}''^{(k)}_{w,\mu_2} \end{pmatrix} \in \mathcal{R}^m_q$, and using the public-key and linkability parts, we have:

(a) $2 \cdot \mathbf{A} \cdot (\mathbf{t}'^{(k)}_{w,\mu_1} - \mathbf{t}'^{(k)}_{w,\mu_2}) = -2 \cdot \mathbf{a}^{(k)} \cdot (\mathbf{t}''^{(k)}_{w,\mu_1} - \mathbf{t}''^{(k)}_{w,\mu_2})$,

(b) $2 \cdot \mathbf{H} \cdot (\mathbf{t}'^{(k)}_{w,\mu_1} - \mathbf{t}'^{(k)}_{w,\mu_2}) = -2 \cdot \mathbf{h}^{(k)}_{\mu_1} \cdot (\mathbf{t}''^{(k)}_{w,\mu_1} - \mathbf{t}''^{(k)}_{w,\mu_2})$, where $\mathbf{h}^{(k)}_{\mu_1} \triangleq \mathbf{H} \cdot \mathbf{S}^{(k)}_{\pi} \in \mathcal{R}^2_q$.

6. We let $\bar{\mathbf{S}}^{(k)} = \frac{(\mathbf{t}'^{(k)}_{w,\mu_1} - \mathbf{t}'^{(k)}_{w,\mu_2})}{(\mathbf{t}''^{(k)}_{w,\mu_1} - \mathbf{t}''^{(k)}_{w,\mu_2})} \bmod q$ where $(\mathbf{t}''^{(k)}_{w,\mu_1} - \mathbf{t}''^{(k)}_{w,\mu_2}) \neq \mathbf{0} \bmod q$. We distinguish two cases:

   (a) If $\bar{\mathbf{S}}^{(k)} \neq \mathbf{S}^{(k)}_{\pi} \bmod q$, since we have $\mathbf{A} \cdot \bar{\mathbf{S}}^{(k)} = \mathbf{A} \cdot \mathbf{S}^{(k)}_{\pi} = \mathbf{a}^{(k)} \bmod q$, then $(\bar{\mathbf{S}}^{(k)} - \mathbf{S}^{(k)})$ is a small non-zero vector $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ solution for $\mathbf{A} \in \mathcal{R}^{2\times(m-1)}_q$.

   (b) If $\bar{\mathbf{S}}^{(k)} = \mathbf{S}^{(k)}_{\pi} \bmod q$, then $\mathbf{h}^{(k)}_{\mu_1} = \mathbf{H} \cdot \bar{\mathbf{S}}^{(k)} \bmod q = \mathbf{H} \cdot \mathbf{S}^{(k)}_{\pi} \bmod q$. The target is to show that $\mathbf{h}^{(k)}_{\mu_1} = \mathbf{h}^{(k)}_{\mu_\pi} \bmod 2$ and $\mathbf{h}^{(k)}_{\mu_1} = \mathbf{h}^{(k)}_{\mu_\pi} \bmod q$. If so, then we have $\mathbf{h}^{(k)}_{\mu_1} = \mathbf{h}^{(k)}_{\mu_\pi} \bmod 2q$, which is a contradiction with our assumption at step 4 of this proof. We now prove the first target:

$$\mathbf{h}^{(k)}_{\mu_1} = -2 \cdot \mathbf{h}'^{(k)}_{\mu_1} + \mathbf{q} = \mathbf{1} \bmod 2 = -2 \cdot \mathbf{H} \cdot \mathbf{S}^{(k)}_{\pi} + \mathbf{q} = \mathbf{h}^{(k)}_{\mu_\pi},$$

where the first and the last equalities follow from definition of $\mathbf{h}^{(k)}$ in second line of Algorithm 11. To show the second target, we have

$$\mathbf{h}_{\mu_1}^{(k)} = -2 \cdot \mathbf{h}_{\mu_1}^{(k)} + \mathbf{q} = -2 \cdot \mathbf{h}_{\mu_1}^{(k)} \bmod q$$

$$= -2 \cdot \mathbf{H} \cdot \bar{\mathbf{S}}^{(k)} \bmod q = -2 \cdot \mathbf{H} \cdot \mathbf{S}_{\pi}^{(k)} \bmod q = \mathbf{h}_{\mu_\pi}^{(k)},$$

where the first and the last equalities follow from definition of $\mathbf{h}^{(k)}$ in second line of Algorithm 11 and the middle equality is true based on the argument at the beginning of step (6.b).

7. Since $(\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1}) \neq \mathbf{0} \bmod 2$, we have $(\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) \neq \mathbf{0} \bmod 2q$. In addition, we know that $\|\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}\|_\infty < q/2$, which implies that $(\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) \neq \mathbf{0} \bmod q$. Ultimately, we have $\mathbf{A} \cdot (\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) = \mathbf{0} \bmod q$ and $\|(\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) \bmod q\| \leq 2\beta_v$. Therefore, this small non-zero vector $(\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)})$ is the output of the algorithm $\mathcal{B}$, and this vector is a solution to the $\mathbf{MSIS}_{q,m,k,\beta}^{\mathcal{K}}$ problem with $\beta = 2\beta_v$ for $\mathbf{a}^{(k)} \in \mathcal{R}_q^2$.

□

Theorem 5.12 (Non-Slanderability)**.** *For any linkable ring signature, if it satisfies unforgeability and linkability, then it satisfies non-slanderability.*

Proof. Let's suppose there is a non-slanderability adversary $\mathcal{A}_{Sland}$ who is given $\mathbf{pk}_i, \mathbf{sk}_i, i \neq \pi,$ and $i \in \{1, \dots w\}$, and he produces a valid signature $\sigma'_L(\mu)$ with linkability tag $\mathbf{h}_{\sigma'_L(\mu)}$ which is equal to $\mathbf{h}_{\sigma_L(\mu)}$, $\sigma_L(\mu)$ being the legitimate signature generated with respect to $\mathbf{sk}_\pi$. This means that $\mathcal{A}_{Sland}$ can create a signature with the linkability tag $\mathbf{h}_{\sigma_L(\mu)}$ without knowing $\mathbf{sk}_\pi$. The adversary can also compute a valid $\sigma''_L(\mu)$ with $\mathbf{sk}_i$, $i \neq \pi,$ and $i \in \{1, \dots w\}$ for which $\mathbf{h}_{\sigma''_L(\mu)} \neq \mathbf{h}_{\sigma'_L(\mu)}$. We give $(\sigma''_L(\mu), \sigma'_L(\mu))$ to the forger, which can turn it to an $\mathbf{MSIS}_{q,m,k,\beta}^{\mathcal{K}}$ solution. In particular, it will be computationally secure when two valid signatures created by different users are unlinked using the L2RS algorithms. An adversary $\mathcal{A}$ will break these properties with negligible probability as demonstrated in Theorems (5.9 and 5.11), and with these probabilities the $\mathcal{A}$ will find a $\mathbf{MSIS}_{q,m,k,\beta}^{\mathcal{K}}$ solution.

Therefore, non-slanderability is implied by the definitions of the unforgeability (Definition 5.1) and linkability (Definition 5.3). □

Corollary 5.13 (Non-Slanderability)**.** *The MIMO.L2RS scheme is non-slanderable under the assumptions of Theorem 5.9 and Theorem 5.11.*

5.4 Ring Confidential Transaction Protocol (RCT)

In this section we formally define the RCT protocol which was initially established in the former RingCT 2.0 protocol [SALY17].

Definition 5.14 (Account or wallet)**.** A wallet has a public component "*act*" and a private component "*ask*". The *act* is composed of the user's pk (or a valid address) and the coin **cn**, while the *ask* is formed of the user's sk along with the coin-key **ck**.

The RCT protocol has five PPT algorithms (RCT.Setup, RCT.KeyGen, RCT.Mint, RCT.Spend, RCT.Verify). It also satisfies correctness (RCT.Correctness). These algorithms are summarised in Table 5.2.

TABLE 5.2: RCT Algorithms

Algorithm	**Input**	**Output**	**Description**
RCT.Setup	The security parameter	The public parameters	Public parameters creation. This calls the L2RS.Setup
RCT.KeyGen	The public parameters	The pair keys	Public and private keys creation. This calls the L2RS.KeyGen
RCT.Mint	The public-key and the amount ($)	Coins	Coins generation
RCT.Spend	The message, the input wallets, *user*'s input wallet, *user*'s private keys, the output addresses and the public parameter	The transaction, the signature and the group of linking tags	Amount transfer from one or more input wallets to one or more output wallets. This also checks double spending, amount and range preservation
RCT.Verify	RCT.Spend's output	Accept/Reject	Verify whether or not the transactions were successfully generated

- RCT.Setup: this PPT algorithm takes the security parameter λ and outputs the public parameters Pub-Params.
- RCT.KeyGen: this PPT algorithm uses the Pub-Params to produce a pair of keys, the public-key pk and the private-key sk.
- RCT.Mint: a PPT algorithm generating new coins by receiving Pub-Params and the amount $. This algorithm outputs a coin **cn** and a coin-key **ck**.
- RCT.Spend: a PPT algorithm that receives the Pub-Params, a set of input wallets $\{IW_i\}_{i\in[w]}$ with w being the size of the ring, a user π's input wallets

IW_π along with its set of secret keys K_π, a set of output addresses OA, some transaction string $\mu \in \{0,1\}^*$, the output amount \$ and the set of output wallets OW. Then, this algorithm outputs: the transaction $TX = (\mu, IW, OW)$, the signature sig and a set of transaction/serial numbers TN, which is used to prevent the double spending coins.

- $\mathsf{RCT.Verify}$: a deterministic PPT algorithm that takes as input the $\mathsf{Pub\text{-}Params}$, the signature sig, the TX, and the TN and verifies if the transaction was legitimately generated and outputs either: Accept or Reject.

Transaction Correctness requirements: $\mathsf{RCT.Correctness}$ ensures that an honest user (payer) is able to spend or transfer any of his accounts (wallets) into a group of destination accounts (payee), where this transaction is accepted with overwhelming probability by a verifier. Thus the correctness of RCT is guaranteed if for all PPT adversaries $\mathcal{A}$, it holds that:

$$\Pr\left[\mathsf{RCT.Verify}\big(TX, \mathsf{sig}, TN\big) = 1 : \begin{array}{l} \mathsf{Pub\text{-}Params} \hookleftarrow \mathsf{RCT.Setup}(1^\lambda); \\ (\mu, IW, OA) \hookleftarrow \mathcal{A}(\mathsf{Pub\text{-}Params}, IW_\pi, K_\pi) \\ \text{with } (IW_\pi, K_\pi); \\ (\mathsf{pk}, \mathsf{sk}) \hookleftarrow \mathsf{RCT.KeyGen}(\mathsf{Pub\text{-}Params}); \\ (\mathbf{cn}, \mathbf{ck}) \hookleftarrow \mathsf{RCT.Mint}(\mathsf{Pub\text{-}Params}, \$); \\ (TX, \mathsf{sig}, TN) \hookleftarrow \mathsf{RCT.Spend}(\mu, \\ \mathsf{Pub\text{-}Params}, IW_\pi, K_\pi, IW, OA, \$_{(out)}). \end{array}\right] = 1 - neg(\lambda).$$

5.4.1 Oracles for adversaries

We now list all the adversarial oracles used in RCT, and we define them as:

- $\mathsf{AddGen}(i)$: on input a query number i, this oracle picks randomness τ_i, runs algorithm $\big(\mathsf{pk}_i, \mathsf{sk}_i\big) \leftarrow \mathsf{RCT.KeyGen}(\mathsf{Pub\text{-}Params}, \tau_i)$, and returns the public-key or one-time address pk_i.

- $\mathsf{ActGen}(i, \$_i)$: on input a query number i and an amount $\$_i$, it runs $(\mathbf{cn}_i, \mathbf{ck}_i) \leftarrow \mathsf{RCT.Mint}\big(\mathsf{Pub\text{-}Params}, \$_i\big)$. Then, ActGen adds i and the account $act_i = \big(\mathsf{pk}_i, \mathbf{cn}_i\big)$ to empty lists $\mathcal{I}$ and IW, respectively. ActGen outputs $(act_i, \mathbf{ck}_i)$ for the one-time address pk_i, where these addresses are added to a list $\mathcal{PK}$. The associated secret key with account act_i is defined as $ask_i \triangleq (\mathsf{sk}_i, \mathbf{ck}_i)$. With this ask_i, the challenger calls $\mathsf{MIMO.L2RS.SigGen}(\mathsf{sk}_i, \cdot, \cdot, \cdot)$ to determine the transaction number TN_i of act_i and adds it to a list $\mathcal{TN}$.

- $\mathsf{O\text{-}Spend}(\mu, IW, IW_\pi, OA, \$_{(out)}, \mathsf{Pub\text{-}Params})$: on input the transaction string μ, input accounts (wallets) IW containing IW_π and output addresses OA, it runs $(TX, \mathsf{sig}, TN) \leftarrow \mathsf{RCT.Spend}(\mu, K_\pi, IW, IW_\pi, OA, \$_{(out)}, \mathsf{Pub\text{-}Params})$ and adds the outputs to $\mathcal{T}$, where $IW_\pi \in IW$. We assume that at least one account/address in IW_π has not been corrupted. We define the set of transaction numbers in the $\mathsf{RCT.Spend}$ queries as $\mathcal{TN}^*$.

- $\mathsf{Corrupt}(i)$: on input query number $i \in \mathcal{I}$, uses account key ask_i to determine the transaction/serial number TN_i of account act_i with address pk_i, then adds TN_i and $(TN_i, \$_i)$ to lists $\mathcal{C}$ and $\mathcal{B}$ respectively and finally returns τ_i.

5.4.2 Security Game Definition

The protocol RCT is modeled in terms of *balance*, *anonymity* and *non-slanderability* for security analysis purposes, which are defined as follows.

Definition 5.15 (Balance)**.** This property requires that any adversary cannot spend any account without her control and cannot spend her own accounts with a larger output amount. This security property is guaranteed if for all PPT adversaries $\mathcal{A}$, it holds that:

$$\Pr\left[\mathcal{A}\text{ wins}: \begin{array}{l} \mathsf{Pub\text{-}Params} \hookleftarrow \mathsf{LRCT.Setup}(1^\lambda); \\ (\{IW_i^{(k)}\}_{i\in[w],k\in[N_{in}]}, \mathcal{T}) \hookleftarrow \mathcal{A}^{\mathsf{AddGen},\mathsf{ActGen},\mathsf{O\text{-}Spend},\mathsf{Corrupt}}(\mathsf{Pub\text{-}Params}) \end{array}\right],$$

is $negl(\lambda)$, where adversaries' oracles are defined in Section 5.4.1. We have that $IW_i^{(k)} = \{\mathsf{pk}_{(in),i}^{(k)}, \mathbf{cn}_{(in),i}^{(k)}\}_{i\in[w],k\in[N_{in}]}$ and $\mathcal{T} = (TX, \mathsf{sig}, TN)$. These spends can be transferred to the challenger with the account address $\mathsf{pk}_{(out)} = \{\mathsf{pk}_{(out)}^{(j)}\}_{j\in[N_{out}]}$, where we assume not all of them are corrupted, and at least one of them is honest. This $\mathsf{pk}_{(out)}$ has been created by the AddGen oracle, so the challenger knows all balances of the spent accounts and output accounts involved in the adversarial spends $\mathcal{T}$. This means that $TX = (\mu, IW, OW)$ with $OW = \{OW^{(j)}\}_{j\in[N_{out}]} = \{\mathsf{pk}_{(out)}^{(j)}, \mathbf{cn}_{(out)}^{(j)}\}_{j\in[N_{out}]}$ being the output wallet corresponding to output account $\mathsf{pk}_{(out)}$. The adversary $\mathcal{A}$ wins this experiment if her outputs satisfy the following conditions:

1. $\mathsf{RCT.Verify}(TX, \mathsf{sig}, TN) = 1$.
2. $\sum_{k\in E_{(in)}} \$_{(in),\pi}^{(k)} < \sum_{j\in G_{(out)}} \$_{(out)}^{(j)}$, where we let $\pi \in [w]$ $s.t.$ $\pi's$ row $\{\mathsf{pk}_{(in),\pi}^{(1)}, \ldots, \mathsf{pk}_{(in),\pi}^{(N_{in})}\}$ are the ones that have $\{TN_\pi^{(1)}, \ldots, TN_\pi^{(N_{in})}\}$ which are found in ActGen, $E_{(in)}$ are the corrupted inputs, and $G_{(out)}$ are the not corrupted outputs in $\mathcal{T}$. For each $TN^{(k)}$ let $\$_{(in)}^{(k)}$ be the amount queried to ActGen at the index query i such $TN \subseteq \mathcal{TN}$. $\$_{(in)}^{(k)}$ is also defined as equal to zero if $IW_i^{(k)}$ is equal to some input wallet IW queried to $\mathsf{O\text{-}Spend}$, using same TN, which means that $IW_i^{(k)}$ has been spent.
3. TN cannot be the output of previous queries to the $\mathsf{O\text{-}Spend}(\cdot)$ (i.e. $TN \cap \mathcal{TN}^* = \emptyset$).
4. pk_π is queried to $\mathsf{O\text{-}Spend}$ oracle only once.
5. $PK \subseteq \mathcal{PK}$, where $PK \triangleq \{\mathsf{pk}_{(in),i}^{(k)}\}_{i\in[w],k\in[N_{in}]}$.

Our extended *anonymity* property captures two types of attacks (compared to one type in [SALY17]) that depend on the adversary's choices for users $\pi_0, \pi_1 \in [w]$ and output amounts $\$_{(out),0}, \$_{(out),1}$. It starts with the user anonymity attack where the adversary selects $\pi_0 \neq \pi_1$ with $\$_{(out),0} = \$_{(out),1}$, while in the amount privacy attack this adversary chooses $\pi_0 = \pi_1$ with $\$_{(out),0} \neq \$_{(out),1}$. We formally define this property as:

Definition 5.16 (Anonymity)**.** This property requires that two proofs of knowledge with the same transaction string μ, input accounts IW, output addresses OA, distinct both output amounts $(\$_{(out),0}, \$_{(out),1})$ and spent accounts $IW_{\pi_0}, IW_{\pi_1} \in IW$ are indistinguishable, meaning that the spender's accounts and amounts are successfully hidden among all the honestly generated accounts. The protocol RCT is called anonymous if for all PPT adversaries $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, it holds that:

$$\left| \Pr \left[b' = b : \begin{array}{l} \mathsf{Pub\text{-}Params} \hookleftarrow \mathsf{Setup}(1^\lambda); \\ (\mu, IW_{\pi_0}, IW_{\pi_1}, IW, OA, \$_{(out),0}, \\ \$_{(out),1}) \hookleftarrow \mathcal{A}_1^{\mathsf{AddGen,ActGen,O\text{-}Spend,Corrupt}}(\mathsf{Pub\text{-}Params}); \\ b \hookleftarrow \{0,1\}; \\ (TX^*, \mathsf{sig}_b^*, TN^*) \hookleftarrow \mathsf{RCT.Spend}(\mu, K_{\pi_b}, IW_{\pi_b}, IW, \\ OA, \$_{(out),b}, \mathsf{Pub\text{-}Params}); \\ b' \hookleftarrow \mathcal{A}_2^{\mathsf{O\text{-}Spend,Corrupt}}(\mathsf{Pub\text{-}Params}, (TX^*, \mathsf{sig}_b^*, TN^*)) \end{array} \right] - \frac{1}{2} \right|,$$

is $negl(\lambda)$, where adversaries' oracles are defined in Section 5.4.1. In addition, the following restrictions should be satisfied:

1. For all $b \in \{0,1\}$, any account in IW_{π_i} has not been corrupted.
2. Any query in the form of $(\cdot, IW_\pi, \cdot, \cdot)$, such that $IW_\pi \cap IW_{\pi_i} \neq \emptyset$ has not been issued to O-Spend oracle.

Definition 5.17 (Non-Slanderability)**.** This property requires that a malicious user cannot slander any honest user after observing an honestly generated spending. That is, it is infeasible for any malicious user to produce a valid spending that shares at least one transaction/serial number with a previously generated honest spending. The protocol RCT is non-slanderable if for all PPT adversaries $\mathcal{A}$, it holds that:

$$\Pr\left[\begin{array}{l|l} \mathcal{A} \text{ wins}: & \begin{array}{l} \mathsf{Pub\text{-}Params} \hookleftarrow \mathsf{RCT.Setup}(1^\lambda); \\ \big((TX, \mathsf{sig}, TN), (TX^*, \mathsf{sig}^*, TN^*)\big) \\ \hookleftarrow \mathcal{A}^{\mathsf{AddGen},\mathsf{ActGen},\mathsf{O\text{-}Spend},\mathsf{Corrupt}}(\mathsf{Pub\text{-}Params}) \end{array} \end{array}\right],$$

is $negl(\lambda)$, where adversaries' oracles are defined in Section 5.4.1, and (TX, sig, TN) is one output of the oracle O-Spend for some (μ, IW_π, IW, OA). We say $\mathcal{A}$ succeeds if the output satisfies:

1. $\mathsf{RCT.Verify}(TX^*, \mathsf{sig}^*, TN^*) = 1$,
2. $(TX^*, \mathsf{sig}^*, TN^*) \notin \mathcal{T}$, and
3. $TN \cap \mathcal{C} = \emptyset$ but $TN \cap TN^* \neq \emptyset$.

5.5 Building Blocks Construction

In this section, we describe the construction of the underlying lattice-based primitives that are used in the construction of a MIMO.RCT. This includes a lattice-based homomorphic commitment (COM) scheme and a MIMO version of L2RS signatures (specified in Section 5.1) that is used as a Proof of Knowledge (PoK). The COM and PoK are formally defined in the preliminaries within Chapter 3.

5.5.1 Lattice-based Commitment Construction

The MIMO.LRCT protocol requires a non-interactive homomorphic commitment (Com) as an essential primitive. We construct the three algorithms: (KeyGen, Com, Open), using the MIMO.L2RS construction (Section 5.2):

- $\mathbf{A} \leftarrow \mathsf{KeyGen}(1^\lambda)$: A PPT algorithm that produces a public commitment parameter $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$ after receiving the security parameter (λ). In doing so, we call the MIMO.L2RS.Setup (Section 5.2) to generate $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$.

- $\mathbf{c} \leftarrow \mathsf{Com}_{\mathbf{A}}(\mathrm{m}, \mathsf{sk})$: A PPT algorithm that receives the public parameter $\mathbf{A}$ (from KeyGen), the randomness sk and the message formed as $\overline{\mathrm{m}} = (0, \mathrm{m})^T \in \mathcal{R}_q^{1\times 2}$. This algorithm generates the commitment $\mathbf{c} \in \mathcal{R}_q^2$. The randomness $\mathsf{sk} \in \mathsf{Dom}_{\mathsf{sk}} \subseteq \mathcal{R}_q^{(m-1)\times 1}$ with every component chosen uniformly and independently with coefficients in $(-2^\gamma, 2^\gamma)$, is produced by calling the MIMO.L2RS.KeyGen (Algorithm 10) and the message $\mathrm{m} \in \mathsf{Dom}_{\mathrm{m}} = \mathcal{R}_q$, then the commitment $\mathbf{c} = \mathsf{Com}_{\mathbf{A}}(\mathrm{m}, \mathsf{sk}) = \mathbf{A} \cdot \mathsf{sk} + \overline{\mathrm{m}} \in \mathcal{R}_q^2$.
- $\mathrm{m}' \leftarrow \mathsf{Open}_{\mathbf{A}}(\mathbf{c}, \mathsf{sk})$: A PPT algorithm receiving commitment $\mathbf{c}$ and randomness sk, and it outputs m'. A valid $\mathbf{c}$ is opened if $(\mathrm{m}' = \mathrm{m})$. This algorithm computes $\overline{\mathrm{m}}' = (0, \mathrm{m}')^T = \mathsf{Open}_{\mathbf{A}}(\mathbf{c}, \mathsf{sk}) = \mathbf{c} - \mathbf{A} \cdot \mathsf{sk}$.

Remark 5.18. $\mathsf{Dom}_{\mathrm{m}}$ is full and not a small subset $\mathcal{R}_q$, whereas $\mathsf{Dom}_{\mathsf{sk}}$ is only a small domain versus q. These adjustments help us to obtain better parameters than SISO.LRCT and security against out-of-range attacks.

This homomorphic commitment scheme performs the following operations:

$$\begin{aligned}\mathsf{Com}_{\mathbf{A}}(\mathrm{m}, \mathsf{sk}) \boxed{\pm} \mathsf{Com}_{\mathbf{A}}(\mathrm{m}', \mathsf{sk}') &\triangleq \mathsf{Com}_{\mathbf{A}}(\mathrm{m}, \mathsf{sk}) \pm \mathsf{Com}_{\mathbf{A}}(\mathrm{m}', \mathsf{sk}') \bmod q \\ &\triangleq \mathsf{Com}_{\mathbf{A}}(\mathrm{m} \pm \mathrm{m}', \mathsf{sk} \pm \mathsf{sk}') \bmod q. \quad (5.10)\end{aligned}$$

Theorem 5.19 (Hiding)**.** *If* $\frac{1}{2}\sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ *is negligible in security parameter* λ*, then the above* Com *is information theoretically hiding.*

Proof. Suppose that a PPT adversary $\mathcal{A}$ is given two messages $(\mathrm{m}, \mathrm{m}')$, the public parameter $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$ and the randomness sk. A bit b is chosen uniformly at random from $b = \{0, 1\}$, and the commitment is generated as $\mathbf{c}_b \leftarrow \mathsf{Com}_{\mathbf{A}}(\mathrm{m}_b, \mathsf{sk}) = \mathbf{A} \cdot \mathsf{sk} + \overline{\mathrm{m}}_b$. This adversary $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$, where $\mathcal{A}$ succeeds in breaking the hiding property when $(b = b')$. We now analyze the generated commitment $\mathbf{c}_b$ with a uniformly random element from $\mathcal{R}_q^2$. We know that sk is chosen small with coefficients in $(-2^\gamma, 2^\gamma)$. By applying the Left-over Hash Lemma (Lemma 3.5), we argue that the statistical distance between the distribution of $\mathbf{c}$ and the uniform distribution on $\mathcal{R}_q^2$ is at most $\left(\frac{1}{2} \cdot \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}\right)$, which is negligible in (λ). □

Theorem 5.20 ($\beta-$Binding)**.** *The described Commitment Scheme is computationally $\beta-$binding if the* $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,2\beta}$ *problem is hard.*

Proof. Suppose that an adversary $\mathcal{A}$ generates $(\mathbf{c}, \mathsf{sk}, \mathsf{sk}')$ such that $\overline{\mathrm{m}} = \mathsf{Open}_{\mathbf{A}}(\mathbf{c}, \mathsf{sk})$ and $\overline{\mathrm{m}}' = \mathsf{Open}_{\mathbf{A}}(\mathbf{c}, \mathsf{sk}')$ with $\overline{\mathrm{m}} = (0, \mathrm{m})^T \in \mathcal{R}_q^{1\times 2}$ and $\overline{\mathrm{m}}' = (0, \mathrm{m}')^T \in \mathcal{R}_q^{1\times 2}$ being valid messages and $\mathrm{m} \neq \mathrm{m}'$. Using the Open algorithm, we have $\mathbf{A} \cdot (\mathsf{sk} - \mathsf{sk}') = (\overline{\mathrm{m}} - \overline{\mathrm{m}}') = (0, \mathrm{m} - \mathrm{m}')^T \neq 0$, where we find a small non-zero vector $\mathbf{v} = \begin{pmatrix} \mathsf{sk} - \mathsf{sk}' \end{pmatrix}^T$ with respect to the first row $\mathbf{A}_1$ of the public commitment parameter $\mathbf{A}$, such that $\mathbf{A}_1 \cdot \mathbf{v} = 0 \bmod q$, with $\|\mathbf{v}\| \leq 2\beta$. Therefore, this vector $\mathbf{v}$ gives a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,2\beta}$ problem. □

5.5.2 (MIMO.L2RS) as a Proof of Knowledge

We adapt all the notations from [ATSS+18] into our MIMO.L2RS that signs a signature for multiple wallets, which means that it signs N_{in} L2RS signatures in parallel. This MIMO.L2RS is an extension of the single-input and single-output proposal from [ATSS+18]. In such extension, we needed to modify the Lattice-based one-time Linkable Ring Signature (L2RS) to be capable of signing multiple wallets. Precisely, we adjusted the key generation, the signature generation and the verification algorithms to sign the total number of input wallets that a user wants to transfer to some output wallets. We call these algorithms: MIMO.L2RS.KeyGen, MIMO.L2RS.SigGen and MIMO.L2RS.SigVer, and we describe them in Algorithms (10, 11 and 12), respectively.

The security properties of the MIMO.L2RS are inherited from the L2RS' security analysis. By appropriately modifying these analysis, we can obtain the same results for unforgeability, anonymity, linkability and non-slanderability, which are shown in Theorems (5.9, 5.10, 5.11, 5.12) , respectively.

The MIMO.L2RS signature scheme is also used as a Proof of Knowledge (PoK) to accomplish, in part, the MIMO.LRCT's balance property. This PoK is formalised, namely as:

Proposition 5.21. *The MIMO.L2RS.SigGen and MIMO.L2RS.SigVer which are described in Algorithms 11 and 12, respectively, are a Fiat-Shamir Non-Interactive Proof of Knowledge in the Random Oracle Model (Section 3.3) for the relations R_{PoK} and R'_{PoK} that we represent as:*

$$R_{PoK} \triangleq \left\{ \begin{array}{c} \{\mathbf{a}^{(k)}_{(in),i}, \mathbf{cn}^{(k)}_{(in),i}, \mathbf{cn}^{(j)}_{(out)}, \mu\}; \{\mathbf{S}^{(k)}_{(in),i}, \mathbf{ck}^{(k)}_{(in),i}, \mathbf{ck}^{(j)}_{(out)}, \$_{in}, \$_{out}\} : \\ \exists i \in [w] \ s.t. \ \mathbf{a}^{(N_{in}+1)}_{(in),i} = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{S}^{(N_{in}+1)}_{(in),i}); \|\mathbf{S}^{(N_{in}+1)}_{(in),i}\| \leq \beta_{wit} \end{array} \right\}$$

$$R'_{PoK} \triangleq \left\{ \begin{array}{c} \{\mathbf{a}^{(k)}_{(in),i}, \mathbf{cn}^{(k)}_{(in),i}, \mathbf{cn}^{(j)}_{(out)}, \mu'\}; \{\mathbf{S}^{(k)}_{(in),i}, \mathbf{ck}^{(k)}_{(in),i}, \mathbf{ck}^{(j)}_{(out)}, \$_{in}, \$_{out}\} : \\ \exists z \in [w] \ s.t. \ \mathbf{v}^{(N_{in}+1)}_{z} = (\mathbf{v}^{(N_{in}+1)}_{z,(1)}, \mathbf{v}^{(N_{in}+1)}_{z,(2)})^T; \\ \mathbf{a}^{(N_{in}+1)}_{(in),z} \cdot \mathbf{v}^{N_{in}+1}_{z,(2)} = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{v}^{(N_{in}+1)}_{z,(1)}); \left\|\mathbf{v}^{(N_{in}+1)}_{z}\right\| \leq \beta'_{wit} \end{array} \right\}$$

where $\beta_{wit} = 3 \cdot 2^{\gamma}$ is said to be the honest prover's witness norm and $\beta'_{wit} = 2 \cdot \beta_v$ being the extracted malicious prover's witness norm. β_v is the acceptance bound of $\boldsymbol{t}$ from Algorithm 12 and $\mathbf{a}^{(N_{in}+1)}_{(in),i}$ is defined in (5.14).

Proof.

Completeness: Since MIMO.L2RS runs parallel L2RS signatures, we said that the MIMO.L2RS's correctness (Subsection 5.2.3.1) allows to achieve completeness in the MIMO.L2RS signature scheme.

Soundness: We show that for all PPT adversaries $\mathcal{A}$ of MIMO.L2RS, there is a PPT algorithm Ext, which extracts a valid witness of MIMO.L2RS. We perform a first run $(\mathbf{c}_i, \ldots, \mathbf{c}_w) \hookleftarrow \mathcal{S}_{n,\kappa}$ where we assume that $\mathbf{c}_i = H_1\Big(L', \{\mathbf{H}^{(k)}_{2q}\}_{k\in[N_{in}+1]}, \mu, \{\mathbf{A}^{(k)}_{2q,i-1} \cdot \mathbf{t}^{(k)}_{i-1} + \mathbf{q} \cdot \mathbf{c}_{i-1}\}_{k\in[N_{in}+1]}, \{\mathbf{H}^{(k)}_{2q} \cdot \mathbf{t}^{(k)}_{i-1} + \mathbf{q} \cdot \mathbf{c}_{i-1}\}_{k\in[N_{in}+1]}\Big)$ was a response to a random oracle H_1 (collision resistance) query made by $\mathcal{A}$. When $\mathcal{A}$ rewinds (second run) by responding with $\mathbf{c}_i \neq \mathbf{c}'_i$, we obtain another proof $(\mathbf{t}'^{(k)}_1, \ldots, \mathbf{t}'^{(k)}_w)$ and the corresponding hash values $(\mathbf{c}'_i, \ldots, \mathbf{c}'_w)$. Then, we verify around the ring signature loop (going backwards) to find a collision in the input of H_1, so for $k = N_{in}+1$, such that $\mathbf{A}^{(N_{in}+1)}_{2q,i-1} \cdot \mathbf{t}^{(N_{in}+1)}_{i-1} + \mathbf{q} \cdot \mathbf{c}_{i-1} = \mathbf{A}^{(N_{in}+1)}_{2q,i-1} \cdot \mathbf{t}'^{(N_{in}+1)}_{i-1} + \mathbf{q} \cdot \mathbf{c}'_{i-1} \bmod 2q$. In each stage, we analyze two cases. If $\mathbf{c}_{i-1} \neq \mathbf{c}'_{i-1}$ (case 1), then we use this collision to extract a witness; otherwise, if $\mathbf{c}_{i-1} = \mathbf{c}'_{i-1}$ (case 2) then we move backwards (-1) until the first case holds. Once this condition is met, we set the index $z = i - x$ where x is the number that decreases if case

2 holds. Subsequently, the following equality is built based on this collision, we said that $\mathbf{A}_{2q,z}^{(N_{in}+1)} \cdot \mathbf{t}_z^{(N_{in}+1)} + \mathbf{q} \cdot \mathbf{c}_z = \mathbf{A}_{2q,z}^{(N_{in}+1)} \cdot \mathbf{t}_z'^{(N_{in}+1)} + \mathbf{q} \cdot \mathbf{c}'_z \bmod 2q$ with $\mathbf{c}_{z+1} = \mathbf{c}'_{z+1}$. We reorganise this equality as $\mathbf{A}_{2q,z}^{(N_{in}+1)} \cdot (\mathbf{t}_z^{(N_{in}+1)} - \mathbf{t}_z'^{(N_{in}+1)}) = \mathbf{q} \cdot (\mathbf{c}_z - \mathbf{c}'_z) \bmod 2q$, when this is reduced mod$q$, we have $\mathbf{A}_{2q,z}^{(N_{in}+1)} \cdot (\mathbf{t}_z^{(N_{in}+1)} - \mathbf{t}_z'^{(N_{in}+1)}) = 0 \bmod q$. Since $\mathbf{c}_z - \mathbf{c}'_z \neq 0 \bmod 2$, so we have $\mathbf{t}_z^{(N_{in}+1)} - \mathbf{t}_z'^{(N_{in}+1)} \neq 0 \bmod 2q$ where $\|\mathbf{t}_z^{(N_{in}+1)} - \mathbf{t}_z'^{(N_{in}+1)}\|_\infty < q/2$. By reducing mod$q$, we find a small non-zero vector $\mathbf{v}_z^{(N_{in}+1)} \triangleq \mathbf{t}_z^{(N_{in}+1)} - \mathbf{t}_z'^{(N_{in}+1)} \neq 0 \bmod q$ with $\|\mathbf{v}_z^{(N_{in}+1)}\| \leq 2 \cdot \beta_{wit}$. This $\mathbf{v}_z^{(N_{in}+1)}$ will compute $\mathbf{A}_{2q,z}^{(N_{in}+1)} \cdot \mathbf{v}_z^{(N_{in}+1)} = 0 \bmod q$. Since $\mathbf{A}_{2q,z}^{(N_{in}+1)} \bmod q = 2 \cdot (\mathbf{A}, -\mathbf{a}) \bmod q$, we have $2 \cdot (\mathbf{A}, -\mathbf{a}_{(in),z}^{(N_{in}+1)}) \cdot \mathbf{v}_z^{(N_{in}+1)} = 0 \bmod q$, this implies that $(\mathbf{A}, -\mathbf{a}_{(in),z}^{(N_{in}+1)}) \cdot \mathbf{v}_z^{(N_{in}+1)} = 0 \bmod q$, since q is odd. Then, we consider $\mathbf{v}_z^{(N_{in}+1)} = (\mathbf{v}_{z,(1)}^{(N_{in}+1)}, \mathbf{v}_{z,(2)}^{(N_{in}+1)})^T$ where $\mathbf{a}_{(in),z}^{(N_{in}+1)} \cdot \mathbf{v}_{z,(2)}^{(N_{in}+1)} = \mathbf{A} \cdot \mathbf{v}_{z,(1)}^{(N_{in}+1)} \bmod q$. Finally, we extract the witness as $\mathbf{a}_{(in),z}^{(N_{in}+1)} \cdot \mathbf{v}_{z,(2)}^{(N_{in}+1)} = \mathsf{Com}_{\mathbf{A}}\big(0, \mathbf{v}_{z,(1)}^{(N_{in}+1)}\big)$ with $(\mathbf{v}_{z,(1)}^{(N_{in}+1)}, \mathbf{v}_{z,(2)}^{(N_{in}+1)})^T \neq 0 \bmod q$.

HVZK: This property is guaranteed by the MIMO.L2RS's anonymity (in Theorem 5.10) as well as the hiding property of the homomorphic commitment scheme which was proved in Theorem 5.19. □

5.6 MIMO Lattice-based RingCT Construction

In this section, we construct the MIMO Lattice-based RingCT (MIMO.LRCT) protocol (Table 5.3 shows the MIMO.LRCT's notations), where one is allowed to have multiple (IW) and to spend them into multiple (OW). Furthermore, two sub-protocols are needed to support the MIMO.LRCT's threat model, which are: MIMO.L2RS security properties (subsection 5.2) and range preservation (subsection 5.7).

The MIMO scheme works using a set of algorithms MIMO.LRCT = (MIMO.LRCT.Setup, MIMO.LRCT.KeyGen, MIMO.LRCT.Mint, MIMO.LRCT.Spend, MIMO.LRCT.Verify) and they are listed as:

[1]In this work, we consider that all users have a fixed number of input wallets N_{in}.

TABLE 5.3: Notation of the Lattice RingCT v2.0

Notation	Description
act	Account or Wallet "Public part" $= \left(\mathsf{pk}, \mathbf{cn}\right) \in \mathcal{R}_q^2 \times \mathcal{R}_q^2$.
ask	Account or Wallet "Private part" $= \left(\mathsf{sk}, \mathbf{ck}\right) \in \mathcal{R}_q^2 \times \mathcal{R}_q^2$.
$\mathcal{S}_{n,\kappa}$	Binary vectors of length n of weight κ.
$\$$	Amount $\in \mathcal{S}_{n,\kappa}$.
$\$_{(in)}$	Group of input amounts $\$_{(in)}^{(k)}$ for $k \in [N_{in}]$.
$\$_{(out)}$	Group of output amounts $\$_{(out)}^{(j)}$ for $j \in [N_{out}]$.
$\ell_{\$}$	The bit-length of $\$$.
w	Number of users in the ring.
N_{in}	Number of input wallets of a user[1].
IW_i	Input wallet of the i-th user $act_i = \left\{\mathsf{pk}_{(in),i}^{(k)}, \mathbf{cn}_{(in),i}^{(k)}\right\}_{k\in[N_{in}]}$.
IW	Set of input wallet $= \{IW_i\}_{i\in[w]}$.
IW_π	Input wallet of user $\pi = \left\{\mathsf{pk}_{(in),\pi}^{(k)}, \mathbf{cn}_{(in),\pi}^{(k)}\right\}_{k\in[N_{in}]}$.
K_π	User π's private-keys $= ask_\pi = \left\{\mathsf{sk}_{(in),\pi}^{(k)}, \mathbf{ck}_{(in),\pi}^{(k)}\right\}_{k\in[N_{in}]}$.
N_{out}	Number of output wallets.
OW	Set of output wallet $= \{OW^{(j)}\}_{j\in[N_{out}]} = \{\mathsf{pk}_{(out)}^{(j)}, \mathbf{cn}_{(out)}^{(j)}\}_{j\in[N_{out}]}$.
OA	Set of output addresses $= \left\{\mathsf{pk}_{(out)}^{(j)}\right\}_{j\in[N_{out}]}$.
TX	Transaction $= (\mu, IW, OW)$.
TN	Set of serial/transaction numbers (linking tag).

1. $(\mathsf{Pub\text{-}Params}) \leftarrow \mathsf{MIMO.LRCT.Setup}(\lambda)$: On input the security parameter λ, this algorithm calls $\mathsf{MIMO.L2RS.Setup}$ (Section 6.3.1) and outputs the public parameters $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$ and $\mathbf{H} \in \mathcal{R}_q^{2\times(m-1)}$.

2. $(\mathbf{a}, \mathbf{S}) \leftarrow \mathsf{MIMO.LRCT.KeyGen}(\mathbf{A})$: Given the public parameter $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$, it runs $\mathsf{MIMO.L2RS.KeyGen}$ (Algorithm 10) and outputs a pair of keys, the public-key or one-time address pk as $\mathbf{a} \in \mathcal{R}_q^2$ and the private-key sk as $\mathbf{S} \in \mathcal{R}_q^{(m-1)\times 1}$. A homomorphic commitment is generated as $\mathbf{a} = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{S}) = \mathbf{A} \cdot \mathbf{S} + \overline{0} \bmod q \in \mathcal{R}_q^2$.

3. $(\mathbf{cn}, \mathbf{ck}) \leftarrow \mathsf{MIMO.LRCT.Mint}(\mathbf{A}, \$)$: It receives the public parameter $\mathbf{A}$ and input amount $\$ \in [0, 2^{\ell_{\$}-1}]$. It computes a coin $\mathbf{cn}$, by choosing a coin-key $\mathbf{ck} \in \mathsf{Dom}_{\mathbf{S}}$, where every component of $\mathbf{ck}$ is chosen uniformly and independently, then compute $\mathbf{cn}$ (as Algorithm 17) and this algorithm returns $(\mathbf{cn}, \mathbf{ck})$.

Algorithm 17 MIMO.LRCT.Mint

Input: $\big(\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}, \$ \in [0, 2^{\ell_\$ - 1}]\big)$, being the public parameter $\mathbf{A}$ and the amount $\$$.

Output: $(\mathbf{cn}, \mathbf{ck})$, where they are the coin and the coin key, respectively.

1: **procedure** MIMO.LRCT.Mint($\mathbf{A}, \$$)
2: Let $\mathbf{ck}^T = (\mathbf{ck}_1, \ldots, \mathbf{ck}_{m-1}) \in \mathcal{R}_q^{1\times(m-1)}$ with $\mathbf{ck}_i \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
3: $\mathbf{cn} = \mathsf{Com}_{\mathbf{A}}(\$, \mathbf{ck}) = \mathbf{A} \cdot \mathbf{ck} + \overline{\$} \bmod q \in \mathcal{R}_q^2$ with $\overline{\$} = (0, \$)^T \in \mathcal{R}_q^{1\times 2}$
4: **return** $(\mathbf{cn}, \mathbf{ck})$

4. $(TX, \mathsf{sig}, TN) \leftarrow \mathsf{MIMO.LRCT.Spend}(\mu, IW, IW_\pi, K_\pi, OA, \$_{(out)}^{(j)}, \mathsf{Pub\text{-}Params})$: This algorithm spends/transfers amounts from the user π's input wallets to some output wallets. We denote the user π who successfully created its input wallets IW_π, based on determine amounts $\$_{(in)}$. Note that notation of these parameters are defined in Table 5.3, and this spend algorithm is briefly described in Algorithm 18. Then, π selects the recipients' valid public keys or output addresses OA where π wants to spend his/her amount. To do so π performs the following steps:

   (a) π receives $\{\$_{(out)}^{(j)}\}_{j\in[N_{out}]}$, with $\$_{(out)}^{(j)} \in [0, \ldots, 2^{\ell_\$ - 1}]$, such balance satisfies, we call this condition *amount preservation*. This checks that input amounts are equal to output amounts, by checking if the following equality holds:

$$\sum_{k=1}^{N_{in}} \$_{(in),\pi}^{(k)} = \sum_{j=1}^{N_{out}} \$_{(out)}^{(j)}. \tag{5.11}$$

   π then runs $\mathsf{MIMO.LRCT.Mint}(\mathbf{A}, \$_{(out)}^{(j)})$ for $j \in [N_{out}]$ and obtain $(\mathbf{cn}_{(out)}^{(j)}, \mathbf{ck}_{(out)}^{(j)})_{j\in[N_{out}]}$, which define the output wallets as

$$OW = \{OW^{(j)}\}_{j\in[N_{out}]} = \{\mathbf{a}_{(out)}^{(j)}, \mathbf{cn}_{(out)}^{(j)}\}_{j\in[N_{out}]}. \tag{5.12}$$

   Then, the output coin-keys and amounts $\{\mathbf{ck}_{(out)}^{(j)}, \$_{(out)}^{(j)}\}_{j\in[N_{out}]}$ are securely sent to users with valid $OA^j = \{\mathbf{a}_{(out)}^{(j)}\}_{j\in[N_{out}]}$.

   (b) User π selects $(w-1)$ input wallets from the blockchain which he/she uses to anonymously transfer her/his input wallets $\{IW_\pi^{(k)}\}_{k\in[N_{in}]}$.

Then, a preliminary ring signature list is built as $IW = \{IW_i\}_{i\in[w]} = \{\mathbf{a}^{(k)}_{(in),i}, \mathbf{cn}^{(k)}_{(in),i}\}_{i\in[w],k\in[N_{in}]}$.

(c) π adds a record to IW_i in order to homomorphically compute and verify the *amount preservation*; this uses the homomorphic commitment scheme (defined in Section 5.5.1). The result of this computation is a commitment to zero, where the user π is only able to obtain since he/she knows both IW_π and OW. This new record is placed in the position $(N_{in}+1)$ and then a list L' is defined as:

$$L' = \left\{\mathbf{a}^{(k)}_{(in),i}\right\}_{i\in[w],k\in[N_{in}+1]}, \tag{5.13}$$

with $\mathbf{a}^{(N_{in}+1)}_{(in),i} \triangleq \mathsf{Com}_{\mathbf{A}}\left(\sum_{k=1}^{N_{in}} \$^{(k)}_{(in),i} - \sum_{j=1}^{N_{out}} \$^{(j)}_{(out)}, \mathbf{S}^{(N_{in}+1)}_{(in),i}\right)$, where $\mathbf{S}^{(N_{in}+1)}_{(in),i} \triangleq \sum_{k=1}^{N_{in}} \mathbf{S}^{(k)}_{(in),i} + \mathbf{ck}^{(k)}_{(in),i} - \sum_{j=1}^{N_{out}} \mathbf{ck}^{(j)}_{(out)} \in \mathcal{R}_q^{(m-1)\times 1}$. This implies that

$$\mathbf{a}^{(N_{in}+1)}_{(in),i} = \sum_{k=1}^{N_{in}} \mathbf{a}^{(k)}_{(in),i} + \mathbf{cn}^{(k)}_{(in),i} - \sum_{j=1}^{N_{out}} \mathbf{cn}^{(j)}_{(out)}. \tag{5.14}$$

Note that if the *amount preservation* conditions (5.11) and (5.15) (for every $k \in [N_{in}]$) are achieved, then $\mathbf{a}^{(N_{in}+1)}_{(in),i} = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{S}^{(N_{in}+1)}_{(in),i})$.

$$\mathbf{a}^{(k)}_{(in),i} = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{S}^{(k)}_{(in),i}) = \mathbf{A}\cdot\mathbf{S}^{(k)}_{(in),i} + \overline{0} \mod q \in \mathcal{R}_q^2. \tag{5.15}$$

(d) To sign the transaction, we use the π's private-keys: $\{\mathbf{S}^{(k)}_{(in),\pi}\}_{k\in[N_{in}+1]}$, the list L' and a transaction string $\mu \in \{0,1\}^*$. Then, we run MIMO.L2RS.SigGen (Algorithm 11) which outputs:

$$\sigma_{L'}(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}^{(k)}_1, \ldots, \mathbf{t}^{(k)}_w\}_{k\in[N_{in}+1]}, \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}\right). \tag{5.16}$$

(e) We show that the output amount $\$^{(j)}_{(out)}$ lies in a non-zero range value, by running our range proof (Algorithm 21)

$\Pi_{Range}.\mathcal{P}_{Range}\left(\left\{\mathbf{cn}^{(j)}_{(out)}, \mathbf{ck}^{(j)}_{(out)}, \$^{(j)}_{(out)}, \mathbf{A}\right\}_{j\in[N_{out}]}\right)$. This proof outputs: $\left\{PoK^{(j)}_{Range}\right\}_{j\in[N_{out}]}$.

(f) We set the transaction TX as (μ, IW, OW) and $TN = \{\mathbf{h}^{(k)}\}_{k\in[N_{in}]}$. This algorithm outputs TX, TN, $\mathsf{sig}_\pi = \left(\sigma_{L'}(\mu), \left\{PoK^{(j)}_{Range}\right\}_{j\in[N_{out}]}\right)$.

5. (Accept/Reject) $\leftarrow$ MIMO.LRCT.Verify$(TX, \mathsf{sig}_\pi, TN)$: This algorithm calls MIMO.L2RS.SigVer$(\mathsf{sig}_{\pi,1}, L', \text{Pub-Params})$ (Algorithm 12) with $\mathsf{sig}_{\pi,1} = \sigma_{L'}(\mu)$, and on input $\mathsf{sig}_{\pi,2} = \left\{PoK^{(j)}_{Range}\right\}_{j\in[N_{out}]}$, it runs (Algorithm 22) $\Pi_{Range}.\mathcal{V}_{Range}\left(\left\{PoK^{(j)}_{Range}, \mathbf{cn}^{(j)}_{(out)}, \mathbf{A}\right\}_{j\in[N_{out}]}\right)$. This MIMO.LRCT.Verify outputs *Accept* if both MIMO.L2RS.SigVer$(\cdot)$ and $\Pi_{Range}.\mathcal{V}_{Range}(\cdot)$ output *Accept*, else it outputs *Reject*.

Algorithm 18 MIMO.LRCT.Spend

Input: $(\mu, IW, IW_\pi, K_\pi, OA, \$^{(j)}_{(out)}, \text{Pub-Params})$, being the message, the input wallets, π's input wallet, π's private keys, the output addresses, the output amount and the public parameter, respectively.
Output: $(TX, \sigma_{L'}(\mu), TN)$

procedure MIMO.LRCT.Spend$(\mu, IW, IW_\pi, K_\pi, OA, \$^{(j)}_{(out)}, \text{Pub-Params})$
User π selects $\{\$^{(j)}_{(out)}\}_{j\in[N_{out}]}$ such that (5.11) is satisfied.
User π runs MIMO.LRCT.Mint$(\mathbf{A}, \$^{(j)}_{(out)})$ for $j \in [N_{out}]$ to generate $(\mathbf{cn}^{(j)}_{(out)}, \mathbf{ck}^{(j)}_{(out)})$ and sets OW as in (5.12).
User π sends securely coin-keys and amounts $\{\mathbf{ck}^{(j)}_{(out)}, \$^{(j)}_{(out)}\}_{j\in[N_{out}]}$ to user's $OA^j = \mathbf{a}^{(j)}_{(out)}$ for $j \in [N_{out}]$.
Create the list of input wallets $IW = \{IW_i\}_{i\in[w]} = \{\mathbf{a}^{(k)}_{(in),i}, \mathbf{cn}^{(k)}_{(in),i}\}_{i\in[w],k\in[N_{in}]}$.
Let $L' = \{\mathbf{a}^{(k)}_{(in),i}\}_{i\in[w],k\in[N_{in}+1]}$, where $\mathbf{a}^{(k)}_{(in),i}$ are defined in (5.15) and (5.14) for $1 \le k \le N_{in}$ and $k = N_{in}+1$, respectively.
Call MIMO.L2RS.SigGen$(\left\{\mathbf{S}^{(k)}_{(in),\pi}\right\}_{k\in[N_{in}+1]}, L', \mu, \text{Pub-Params})$ and obtain $\sigma_{L'}(\mu)$ as in (5.16).
Run $\Pi_{Range}.\mathcal{P}_{Range}\left(\left\{\mathbf{cn}^{(j)}_{(out)}, \mathbf{ck}^{(j)}_{(out)}, \$^{(j)}_{(out)}, \mathbf{A}\right\}_{j\in[N_{out}]}\right)$ for $j \in [N_{out}]$, it outputs $\left\{PoK^{(j)}_{Range}\right\}_{j\in[N_{out}]}$.
Set $\mathsf{sig}_\pi = (\sigma_{L'}(\mu), \left\{PoK^{(j)}_{Range}\right\}_{j\in[N_{out}]})$.
Let $TX = (\mu, IW, OW)$ and $TN = \{\mathbf{h}^{(k)}\}_{k\in[N_{in}+1]}$.
return $(TX, \mathsf{sig}_\pi, TN)$

5.7 Range Preservation of the MIMO.LRCT

This section presents the techniques that we use to preserve the range preservation; that is, these techniques prevent the negative output amount $\$_{(out)}$ attack, also known as out-of-range attack. Since $\$_{(out)}$ is decomposed into binary representation

such $\$_{(out)} = \sum_{i=0}^{\ell_\$ - 1} 2^i b_i$, where $b_i \in \{0, 1\}$, we start proving that each bit b_i is binary, using the OR-Proof technique from [dPLNS17]. We adjust this OR-Proof technique to our commitment scheme, which is constructed in Section 5.5.1. Thereafter, we use a range-proof technique to show that this amount lies in a range of *non-negative* values, so we prove that each committed output amount is within a range which cannot overflow (e.g. $[0, 2^{64})$).

5.7.1 Binary Proof.

We define a protocol for the OR-Proof for our homomorphic commitments as Π_{OR} with the prover $\mathcal{P}_{OR}$ and the verifier $\mathcal{V}_{OR}$. This protocol is a zero knowledge proof where the commitment of the bit $\mathbf{cb}$ opens to have a value $b \in \{0, 1\}$. The proof is approximate *or relaxed* by using a relaxation factor f. Meaning that this proof only proves knowledge of a randomness of the bit $\mathbf{rb}$ such that $f \cdot \mathbf{cb}$ opens to a message $f \cdot b$ with $f \in \mathcal{R}_q$ with respect to $\mathbf{rb}$. We use the public parameter $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$ to define our two binary relations such $R_{OR} = R_0 \vee R_1$:

$$R_0 \triangleq \{(\mathbf{cb}, \mathbf{rb}) \in \mathcal{R}_q^2 \times \mathcal{R}_q^{(m-1)\times 1}, \mathbf{cb} = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{rb}) = \mathbf{A} \cdot \mathbf{rb} + \overline{0}, \|\mathbf{rb}\| \leq B_{OR}\}$$
$$R_1 \triangleq \{(\mathbf{cb}, \mathbf{rb}) \in \mathcal{R}_q^2 \times \mathcal{R}_q^{(m-1)\times 1}, \mathbf{cb} = \mathsf{Com}_{\mathbf{A}}(1, \mathbf{rb}) = \mathbf{A} \cdot \mathbf{rb} + \overline{1}, \|\mathbf{rb}\| \leq B_{OR}\}$$

Two *relaxed* binary relations (i.e. $R'_{OR} = R'_0 \vee R'_1$) with $B'_{OR} > B_{OR}$ are also defined where a witness will be recovered by the soundness extractor:

$$R'_0 \triangleq \{(\mathbf{cb}, \mathbf{rb}), 2 \cdot \mathbf{cb} = \mathsf{Com}_{\mathbf{A}}(2 \cdot 0, \mathbf{rb}) = \mathbf{A} \cdot \mathbf{rb} + 2 \cdot \overline{0}, \|\mathbf{rb}\| \leq B'_{OR}\}$$
$$R'_1 \triangleq \{(\mathbf{cb}, \mathbf{rb}), 2 \cdot \mathbf{cb} = \mathsf{Com}_{\mathbf{A}}(2 \cdot 1, \mathbf{rb}) = \mathbf{A} \cdot \mathbf{rb} + 2 \cdot \overline{1}, \|\mathbf{rb}\| \leq B'_{OR}\}$$

We now define below a challenge set of monomials such that for any distinct pair of monomials $X^i, X^j \in \mathcal{C}_0$ with all the coefficients of $2 \cdot (X^i - X^j)^{-1}$ are in $\{-1, 0, 1\}$ according to Lemma 3.3. Since we are using monomials, the relaxation factor is defined as $f = 2$.

$$\mathcal{C}_0 \triangleq \{X^i \in \mathcal{R}_q, i = 0, \ldots, 2n - 1\}$$

Remark 5.22. The main difference between our OR-proof and the OR-proof from [dPLNS17] is the size of the challenges. As we cannot achieve soundness of our range proof using the same challenge space as in [dPLNS17], we adapt their protocol to another challenge space which we call $\mathcal{C}_0$ (this space was introduced in [BCK+14]), which consists of monomials in $\mathcal{R}_q$. With this new space $\mathcal{C}_0$ we are now able to prove the soundness of our relaxed proof to the relaxed relation R'_{OR} with relaxation factor $f = 2$. However, because the relatively small size of the challenge space $\mathcal{C}_0$ these relatively small challenges, Π_{OR} needs to repeat the basic protocol θ times in parallel, where the rejection sampling (defined in Lemma 3.8) returns something after $\theta - 1$ repetitions. In practice, we only need a relatively small $\theta < 20$, whereas previous lattice based range proofs (i.e. [LLNW18]) need much larger $\theta > 100$ for the same soundness level.

The challenge space $\mathfrak{P}$ consists of the set of all permutations of dimension n, $\texttt{Perm}(n)$, and a bit $c \in \{0, 1\}$, i.e. $\mathfrak{P} \triangleq \{p = (s, c) \in \texttt{Perm}(n) \times \{0, 1\}\}$. Each $p \in \mathfrak{P}$ permutes the exponent of a polynomial in $\mathcal{C}_0$ according to the permutation s. Let $f, g \in \mathcal{C}_0$ be two monomials, if $f = X^{i_f}, g = (-1)^c \cdot X^{i_g}$ and $s(i_f) = i_g$, then we define $p(f) = g$. It holds that $\Pr[p(f) = g] = 1/|\mathcal{C}_0|$ for a uniformly random $p \in \mathfrak{P}$, for any two fixed elements $f, g \in \mathcal{C}_0$. σ_{OR} is defined to be a positive real parameter, whereas B_{OR} is a positive real bound. We also need a cryptographic hash function H modeled as random oracle, which maps arbitrary inputs to the uniform distribution over the challenge space $\mathfrak{P}^\theta$. Our OR-Proof protocol Π_{OR} is defined in Algorithms (19 and 20), for proving if one bit is binary.

5.7.2 Range Proof.

We define a range proof Π_{range}, having two algorithms, one for the prover $\mathcal{P}_{range}$, and one for the verifier $\mathcal{V}_{range}$. $\mathcal{P}_{range}$ receives from $\mathsf{MIMO.LRCT.Spend}(\cdot)$, Algorithm 18, the parameters $\{\$^{(j)}_{(out)}, \mathbf{cn}^{(j)}_{(out)}, \mathbf{ck}^{(j)}_{(out)}\}_{j \in [N_{out}]} \in [0, \ldots, 2^{\ell_\$ - 1}] \times \mathcal{R}_q^2 \times \mathcal{R}_q^{(m-1)}$. We define the first relation R_{range} for this protocol:

Algorithm 19 OR-Proof Protocol Π_{OR}, prover's algorithm $\mathcal{P}_{OR}$

Input: $(\mathbf{cb} = \mathbf{A} \cdot \mathbf{rb} + \overline{b}, \mathbf{A}, \mathbf{rb}, b \in \{0,1\})$
Output: $\left\{f_0^{(t)}, f_1^{(t)}, \mathbf{r}_0^{(t)}, \mathbf{r}_1^{(t)}\right\}_{t=1}^{\theta}$

procedure $\Pi_{OR}.\mathcal{P}_{OR}(\mathbf{cb}, \mathbf{A}, \mathbf{rb}, b)$
  for $(1 \leq t \leq \theta)$ **do**
    Let $f_{1-b}^{(t)} \hookleftarrow \mathcal{C}_0$
    Let $\mathbf{u}^{(t)} \hookleftarrow D_{\sigma_{OR}}^{n(m-1)}$
    $\mathbf{r}_{1-b}^{(t)} \hookleftarrow D_{\sigma_{OR}}^{n(m-1)}$
    $\mathbf{a}_{1-b}^{(t)} \coloneqq \mathbf{A} \cdot \mathbf{r}_{1-b}^{(t)} + f_{1-b}^{(t)} \cdot \overline{(1-b)} - f_{1-b}^{(t)} \cdot \mathbf{cb}$
    $\mathbf{a}_{b}^{(t)} \coloneqq \mathbf{A} \cdot \mathbf{u}^{(t)}$
  Concatenate $(\mathbf{a}_{1-b})_{||} \coloneqq \left(\mathbf{a}_{1-b}^{(1)}, \ldots, \mathbf{a}_{1-b}^{(\theta)}\right)$
  Concatenate $(\mathbf{a}_{b})_{||} \coloneqq \left(\mathbf{a}_{b}^{(1)}, \ldots, \mathbf{a}_{b}^{(\theta)}\right)$
  $(p^{(1)}, \ldots, p^{(\theta)}) \coloneqq H\left(\mathbf{cb}, (\mathbf{a}_{1-b}^{(t)})_{||}, (\mathbf{a}_{b}^{(t)})_{||}\right) \hookleftarrow \mathfrak{P}^{\theta}$
  for $(1 \leq t \leq \theta)$ **do**
    $f_b^{(t)} = (p^{(t)})^{2b-1}(f_{1-b}^{(t)})$
    $\mathbf{r}_b^{(t)} = \mathbf{u}^{(t)} + f_b^{(t)} \cdot \mathbf{rb}$
  $\mathbf{r}_{b||} \coloneqq \left(\mathbf{r}_b^{(1)}, \ldots, \mathbf{r}_b^{(\theta)}\right)$
  $(f_b \cdot \mathbf{rb})_{||} \coloneqq \left(f_b^{(1)} \cdot \mathbf{rb}, \ldots, f_b^{(\theta)} \cdot \mathbf{rb}\right)$
  Continue with probability $1 - \min\left\{\frac{D_{\sigma_{OR}}^{n(m-1)\theta}(\mathbf{r}_{b||})}{3 \cdot D_{(f_b \cdot \mathbf{rb})_{||}, \sigma_{OR}}^{n(m-1)\theta}(\mathbf{r}_{b||})}, 1\right\}$ otherwise **Restart**
  return $PoK_{OR} = \left\{f_0^{(t)}, f_1^{(t)}, \mathbf{r}_0^{(t)}, \mathbf{r}_1^{(t)}\right\}_{t=1}^{\theta}$

Algorithm 20 OR-Proof Protocol Π_{OR}, verifier's algorithm $\mathcal{V}_{OR}$

Input: $(\mathbf{cb}, \mathbf{A}, PoK_{OR})$ with $PoK_{OR} = \left\{f_0^{(t)}, f_1^{(t)}, \mathbf{r}_0^{(t)}, \mathbf{r}_1^{(t)}\right\}_{t=1}^{\theta}$
Output: Accept or Reject

procedure $\Pi_{OR}.\mathcal{V}_{OR}(\mathbf{cb}, \mathbf{A}, PoK_{OR})$
  for $(1 \leq t \leq \theta)$ **do**
    Let $\mathbf{a}_0^{(t)} \coloneqq \mathbf{A} \cdot \mathbf{r}_0^{(t)} - f_0^{(t)} \cdot \mathbf{cb}$
    Let $\mathbf{a}_1^{(t)} \coloneqq \mathbf{A} \cdot \mathbf{r}_1^{(t)} + f_1^{(t)} \cdot \overline{1} - f_1^{(t)} \cdot \mathbf{cb}$
  Concatenate $(\mathbf{a}_0)_{||} \coloneqq \left(\mathbf{a}_0^{(1)}, \ldots, \mathbf{a}_0^{(\theta)}\right)$
  Concatenate $(\mathbf{a}_1)_{||} \coloneqq \left(\mathbf{a}_1^{(1)}, \ldots, \mathbf{a}_1^{(\theta)}\right)$
  $(p^{(1)}, \ldots, p^{(\theta)}) \coloneqq H\left(\mathbf{cb}, (\mathbf{a}_0)_{||}, (\mathbf{a}_1)_{||}\right) \hookleftarrow \mathfrak{P}^{\theta}$
  for $(1 \leq t \leq \theta)$ **do**
    if $\|\mathbf{r}_0^{(t)}\| \leq B'_{OR} \wedge \|\mathbf{r}_1^{(t)}\| \leq B'_{OR}$ **then** Continue; otherwise Reject
    else if $f_0^{(t)} \in \mathcal{C}_0 \wedge f_1^{(t)} = p^{(t)}(f_0^{(t)})$ **then** Continue; otherwise Reject
  return Accept

$$R_{range} \triangleq \left\{ \begin{array}{c} \{\mathbf{cn}_{(out)}^{(j)}\}, \{\$_{(out)}^{(j)}, \mathbf{ck}_{(out)}^{(j)}\} : \forall j \ , \ \mathbf{cn}_{(out)}^{(j)} = \mathsf{Com}_{\mathbf{A}}(\$_{(out)}^{(j)}, \mathbf{ck}_{(out)}^{(j)}) = \\ \mathbf{A} \cdot \mathbf{ck}_{(out)}^{(j)} + \overline{\$}_{(out)}^{(j)}, \|\mathbf{ck}_{(out)}^{(j)}\| \leq 2\beta, \$_{(out)}^{(j)} \in [0, \ldots, 2^{\ell_{\$}-1}] \end{array} \right\}$$

A *relaxed* relation (R'_{range}) is also defined by using f where the corresponding witness will be recovered by its soundness extractor:

$$R'_{range} \triangleq \left\{ \begin{array}{c} \{\mathbf{cn}^{(j)}_{(out)}\}, \{\$^{(j)}_{(out)}, \mathbf{ck}^{(j)}_{(out)}\} : \forall j \ s.t. \ f \cdot \mathbf{cn}^{(j)}_{(out)} = \mathsf{Com}_{\mathbf{A}}(f \cdot \$^{(j)}_{(out)}, \mathbf{ck}^{(j)}_{(out)}) = \\ \mathbf{A} \cdot \mathbf{ck}^{(j)}_{(out)} + f \cdot \overline{\$}^{(j)}_{(out)}, \|\mathbf{ck}^{(j)}_{(out)}\| \leq 2\beta, \$^{(j)}_{(out)} \in [0, \ldots, 2^{\ell_\$ - 1}], f = 4 \end{array} \right\}$$

$\mathcal{P}_{range}$ and $\mathcal{V}_{range}$ are described below in algorithms (21 and 22):

Algorithm 21 Range-Proof Protocol Π_{Range}, prover's algorithm $\mathcal{P}_{Range}$

Input: $\left(\left\{\mathbf{cn}^{(j)}_{(out)}, \mathbf{ck}^{(j)}_{(out)}, \$^{(j)}_{(out)}, \mathbf{A}\right\}_{j\in[N_{out}]}\right)$

Output: $\left\{PoK^{(j)}_{Range}\right\}_{j\in[N_{out}]}$

procedure $\Pi_{Range}.\mathcal{P}_{Range}(\left\{\mathbf{cn}^{(j)}_{(out)}, \mathbf{ck}^{(j)}_{(out)}, \$^{(j)}_{(out)}, \mathbf{A}\right\}_{j\in[N_{out}]})$
  for $(1 \leq j \leq N_{out})$ **do**
    Decompose in binary $\$^{(j)}_{(out)} = \left\{b^{(j)}_i\right\}_{i=0}^{\ell_\$ - 1}$
    Compute commitment to each bit $\mathbf{cb}^{(j)}_i = \mathsf{Com}_{\mathbf{A}}(b^{(j)}_i, \mathbf{rb}^{(j)}_i) = \mathbf{A} \cdot \mathbf{rb}^{(j)}_i + \overline{b}^{(j)}_i$, as defined in Section 5.5.1
    for $(0 \leq i \leq \ell_\$ - 1)$ **do**
      Calls binary proof $PoK^{(j)}_{OR,i} = \Pi_{OR}.\mathcal{P}_{OR}(\mathbf{cb}^{(j)}_i, \mathbf{A}, \mathbf{rb}^{(j)}_i, b^{(j)}_i)$
    Compute $\mathbf{prck}^{(j)} = \sum_{i=0}^{\ell_\$ - 1} 2^i \mathbf{rb}^{(j)}_i - \mathbf{ck}^{(j)}_{(out)}$
    Compute $\mathbf{prcn}^{(j)} = \sum_{i=0}^{\ell_\$ - 1} 2^i \mathbf{cb}^{(j)}_i - \mathbf{cn}^{(j)}_{(out)}$, it should be equal to $\mathsf{Com}_{\mathbf{A}}(0, \mathbf{prck}^{(j)})$
    $\mathbf{y}^{(j)} \hookleftarrow D^{n(m-1)}_{\sigma_{Range}}$
    $\mathbf{c}^{(j)} \leftarrow H(\mathbf{prcn}^{(j)} \cdot \mathbf{y}^{(j)})$
    $\mathbf{z}^{(j)} \leftarrow \mathbf{prck}^{(j)} \cdot \mathbf{c}^{(j)} + \mathbf{y}^{(j)}$, this is a PoK of $\mathsf{Com}_{\mathbf{A}}(0, \mathbf{prck}^{(j)})$
  $\mathbf{z}_{||} = (\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N_{out})})$
  $(\mathbf{prck} \cdot \mathbf{c})_{||} = \left(\mathbf{prck}^{(1)} \cdot \mathbf{c}^{(1)}, \ldots, \mathbf{prck}^{(N_{out})} \cdot \mathbf{c}^{(N_{out})}\right)$
  Continue with probability $1 - \min\left\{\frac{D^{n(m-1)}_{\sigma_{Range}}(\mathbf{z}_{||})}{M \cdot D^{n(m-1)}_{(\mathbf{prck}\cdot\mathbf{c})_{||}, \sigma_{Range}}(\mathbf{z}_{||})}, 1\right\}$ otherwise **Restart**
  return $PoK^{(j)}_{Range} = \left\{PoK^{(j)}_{OR,i}, \mathbf{cb}^{(j)}_i, \mathbf{z}^{(j)}, \mathbf{c}^{(j)}\right\}_{i\in[0,\ell_\$-1], j\in[N_{out}]}$

Algorithm 22 Range-Proof Protocol Π_{Range}, verifier's algorithm $\mathcal{V}_{Range}$

Input: $\left(\left\{PoK^{(j)}_{Range}\right\}_{j\in[N_{out}]}\right)$, where $PoK^{(j)}_{Range} = \left\{PoK^{(j)}_{OR,i}, \mathbf{cb}^{(j)}_i, \mathbf{z}^{(j)}, \mathbf{c}^{(j)}\right\}_{i\in[0,\ell_\$-1], j\in[N_{out}]}$.

Output: Accept or Reject

procedure $\Pi_{Range}.\mathcal{V}_{Range}(\left\{PoK^{(j)}_{Range}\right\}_{j\in[N_{out}]})$
  for $(1 \leq j \leq N_{out})$ **do**
    for $(0 \leq i \leq \ell_\$ - 1)$ **do**
      Accept $\stackrel{?}{=} \Pi_{OR}.\mathcal{V}_{OR}(\mathbf{cb}^{(j)}_i, \mathbf{A}, PoK^{(j)}_{OR,i})$, this checks the binary proof for $\mathbf{cb}^{(j)}_i$
    Compute $\mathbf{vrcn}^{(j)} = \sum_{i=0}^{\ell_\$ - 1} 2^i \mathbf{cb}^{(j)}_i - \mathbf{cn}^{(j)}_{(out)}$
    Check $\mathbf{c}^{(j)} \stackrel{?}{=} H(\mathbf{A} \cdot \mathbf{z}^{(j)} - \mathbf{vrcn}^{(j)} \cdot \mathbf{c}^{(j)})$, otherwise Reject; this checks the range proof of zero commitment
  return Accept or Reject

Theorem 5.23 (Binary Proof)**.** *If $\sigma_{OR} \geq 22\sqrt{\kappa} B_{OR}$ and $B'_{OR} \geq 2\sqrt{n}\sigma_{OR}$, then the protocol $\Pi_{OR}(\mathcal{P}_{OR}, \mathcal{V}_{OR})$ is a R'_b-Protocol complete for relation R_{OR} and sound for relation R'_{OR}.*

Proof.

Correctness We show that $\mathbf{A} \cdot \mathbf{r}_0^{(t)} - f_0^{(t)} \cdot \mathbf{cb} = \mathbf{A} \cdot \mathbf{u}^{(t)}$ so:

$$\begin{aligned}
\mathbf{a}_0^{(t)} &= \mathbf{A} \cdot \mathbf{r}_0^{(t)} - f_0^{(t)} \cdot \mathbf{cb} \\
&= \mathbf{A} \cdot (\mathbf{u}^{(t)} + f_b^{(t)} \cdot \mathbf{rb}) - f_0^{(t)} \cdot \mathbf{cb} \\
&= \mathbf{A} \cdot \mathbf{u}^{(t)} + f_b^{(t)} \cdot (\mathbf{A} \cdot \mathbf{rb} - \mathbf{cb}) \\
&= \mathbf{A} \cdot \mathbf{u}^{(t)} + f_b^{(t)} \cdot (\bar{b})
\end{aligned}$$

Since $\bar{b} = 0$, then this condition holds.

In the case when $\bar{b} = 1$, we refer to the line 12 of $\mathcal{P}_{OR}$ (Algorithm 19), we have $f_1^{(t)} = (p^{(t)})^1(f_0^{(t)})$, then line 9 of $\mathcal{V}_{OR}$ (Algorithm 20) is satisfied. When $\bar{b} = 0$, we have $f_0^{(t)} = (p^{(t)})^{-1}(f_1^{(t)}) \iff (p^{(t)})f_0^{(t)} = (f_1^{(t)})$, this also shown that the condition holds. We also check the bound of $(\mathbf{r}_0^{(t)}, \mathbf{r}_1^{(t)})$, where the rejection sampling lemma (to include the new lemma in the definition) is used to show that the distribution of $\mathbf{r}_{b||}$ (from Algorithm 19, line 14) is statistically close to $D_{\sigma_{OR}}^{n(m-1)\theta}$. Therefore, and each component $\mathbf{r}_b^{(t)}$ is statistically close to $D_{\sigma_{OR}}^{n(m-1)}$. A condition is needed $\sigma_{OR} \geq \|f_b \cdot \mathbf{rb})_{||}\|$. By the tail bound lemma (to be included) $\|\mathbf{r}_b^{(t)}\|_2 \leq \sqrt{n(m-1)} \cdot \sigma_{OR} = B'_{OR}$, except with probability $2^{n(m-1)}$.

Soundness Let $(\mathbf{cn}, \mathbf{r}_b) \in R_0 \vee R_1$. Let $\mathcal{P}_{OR}$ be a deterministic prover, who queries H on the same input. Therefore, her success probability depends on the output of H only. Let $p_0 = 1/|\mathcal{C}_0| + \epsilon$ be the success probability of the prover $\mathcal{P}_{OR}$. We need to construct an extractor $\mathcal{E}$ to extract the values $\mathbf{r}''_b$ and f''_b while making $poly(|(\mathbf{cn}, \mathbf{r}_b)|)/\epsilon$ times queries to H. It holds that $(\mathbf{cn}, \mathbf{r}''_b, f''_b) \in R'_0 \vee R'_1$. Extractor $\mathcal{E}$ runs $\mathcal{P}_{OR}(\mathbf{cn})$ on a challenge $p \hookleftarrow \mathfrak{P}$ and outputs a valid proof $(\mathbf{cn}, (\mathbf{r}_0, \mathbf{r}_1, f_0, f_1))$. Then, $\mathcal{E}$ runs $\mathcal{P}_{OR}(\mathbf{cn})$ on random challenges and outputs a proof $(\mathbf{cn}, (\mathbf{r}'_0, \mathbf{r}'_1, f'_0, f'_1))$ such that $f_0 \neq f'_0$ or $f_1 \neq f'_1$. Let $\alpha \in \{0, 1\}$ be a bit such that $f_\alpha \neq f'_\alpha$. Let $(\mathbf{cn}, \mathbf{a}_0, \mathbf{a}_1)$ be the hash query by $\mathcal{P}_{OR}(\mathbf{cn})$. Since both proofs verify, we have $\mathbf{a}_\alpha = \mathbf{A} \cdot \mathbf{r}_\alpha + f_\alpha \alpha - f_\alpha \cdot \mathbf{cn}$ and $\mathbf{a}_\alpha = \mathbf{A} \cdot \mathbf{r}'_\alpha + f'_\alpha \alpha - f'_\alpha \cdot \mathbf{cn}$.

Subtracting these two equations results into:

$$(f_\alpha - f'_\alpha) \cdot \mathbf{cn} = \mathbf{A} \cdot (\mathbf{r}_\alpha - \mathbf{r}'_\alpha) + \alpha(f_\alpha - f'_\alpha)$$

Set $\mathbf{r}''_\alpha = \mathbf{r}_\alpha - \mathbf{r}'_\alpha$ and $f''_\alpha = f_\alpha - f'_\alpha$. It follows that $(\mathbf{cn}, \mathbf{r}''_\alpha, f''_\alpha) \in \mathcal{R}'_0 \vee \mathcal{R}'_1$. Finally, we show that $\mathcal{P}_{OR}(\mathbf{cn})$ outputs a proof such that $f_\alpha \neq f'_\alpha$ with at least negligible probability ϵ.

$$\begin{aligned}
\Pr[\mathcal{P}_{OR} \text{ succ.} \quad & \wedge (f_0 = f'_0 \vee f_1 = f'_1)] \\
= \; & \Pr[\mathcal{P}_{OR} \text{ succ.}] - \Pr[\mathcal{P}_{OR} \text{ succ.} \wedge (f_0 = f'_0 \wedge f_1 = f'_1)] \\
= \; & p_0 - \Pr[\mathcal{P}_{OR} \text{ succ.} \wedge (f_0 = f'_0 \wedge p(f_0) = p(f'_0))] \\
\geq \; & p_0 - \Pr[p(f_0) = p(f'_0)] = \epsilon. \qquad (5.17)
\end{aligned}$$

HVZK: To prove special honest-verifier zero-knowledgeness, we have to show that the honest-verifier distribution and simulated distribution are identical. We show how to construct a simulator $\mathcal{S}$. For $(\mathbf{cn}, \mathbf{r}_b) \in R_0 \vee R_1$ and a challenge $p \in \mathfrak{P}$ the simulator $\mathcal{S}$ does the following:

1. $f_0 \hookleftarrow \mathcal{C}_0$
2. $f_1 = p(f_0)$
3. For $\alpha \in \{0, 1\}$, sample $\mathbf{r}_\alpha \hookleftarrow D^{n(m-1)}_{\sigma_{OR}}$
4. For $\alpha \in \{0, 1\}$, compute $\mathbf{a}_\alpha = \mathbf{A} \cdot \mathbf{r}_\alpha + f_\alpha \cdot \alpha - f_\alpha \cdot \mathbf{cn}$
5. Abort with probability $1 - 1/M$
6. Output $(\mathbf{r}_0, \mathbf{r}_1, f_0, f_1)$

Using the rejection sampling bounds, the distribution of the output of $\mathcal{S}$ is identical to the honest one. □

Theorem 5.24 (Range Proof)**.** *The protocol described in Step 2 of the range proof is a proof of knowledge (from [Lyu12]) complete for relation R_{range} and sound for relation R'_{range} with $\beta_{range} = 2^{\ell_\$ + 2} n\sqrt{\kappa n(m-1)}\sigma_{OR} + 2^2\sqrt{n}\beta_v$.*

Proof. We prove the three security of a zero-knowledge proof of knowledge as defined in Definition 3.3.

Completeness: If the prover follows the protocol, then the following equation holds:

$$H(\mathbf{A}\bar{\mathbf{r}} - f'D, \mu) = H\Big(\mathbf{A} \cdot (f'\mathbf{r} + \mathbf{r}_0) - f'D, \mu\Big) \tag{5.18}$$

From (5.21) follows that $f'D = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{r}) = \mathbf{A} \cdot \mathbf{r}$. Then, (5.18) is equivalent to:

$$\begin{aligned} H(\mathbf{A}\bar{\mathbf{r}} - f'D, \mu) &= H\Big(\mathbf{A} \cdot (f'\mathbf{r} + \mathbf{r}_0) - f'D, \mu\Big) \\ &= H\Big(f'\mathbf{A}\mathbf{r} + \mathbf{A}\mathbf{r}_0 - f'\mathbf{A} \cdot \mathbf{r}, \mu\Big) \\ &= H(\mathbf{A}\mathbf{r}_0, \mu) = f', \end{aligned}$$

where the last equation satisfies the verification. By the rejection sampling (Definition 3.8 the prover responds with probability $1/M^2$. The distribution of $\bar{\mathbf{r}}$ is statistically close to $D^{n(m-1)}_{12n\sqrt{n(m-1)}}$ since $\|f'\mathbf{r}\| \leq n\sqrt{n(m-1)}$ within the statistical distance 2^{-100}.

Soundness: To prove the soundness, we need to extract a witness $(f, \$, \mathbf{r})$ s.t. $f \cdot \mathbf{cn} = \mathsf{Com}_{\mathbf{A}}(f \cdot \$, \mathbf{r})$ with $\$ \in [0, 2^{\ell_\$} - 1]$.

From the OR proof witness extraction in Theorem 5.23, we first extract $(f''_i, b_i, \mathbf{r}_i)$ with $b_i \in \{0, 1\}$ such that for all $i \in [\ell_\$]$ the following relation holds:

$$f''_i \cdot \mathbf{cn}_i = \mathsf{Com}_{\mathbf{A}}(f''_i \cdot b_i, \mathbf{r}_i). \tag{5.19}$$

Let $f''_i = f_i - f'_i$ bet the difference between two challenges f_i and f'_i. According to Lemma 3.3 it holds that f'' is invertible in $\mathcal{R}_q$. Consequently we can multiply

TABLE 5.4: Π_{PoK^*} protocol [Lyu12]

$\mathcal{P}_{PoK}(\mathbf{r}, \mu, D)$		$\mathcal{V}_{PoK}(D)$
Pick $\mathbf{r}_0 \in D_{\sigma_0}^{m(n-1)}$		
Compute $U := \mathbf{A} \cdot \mathbf{r}_0$		
Set $f' := H(\mathbf{A}\mathbf{r}_0, \mu)$		
Compute $\overline{\mathbf{r}} := f'\mathbf{r} + \mathbf{r}_0$		
Abort with prob. ρ_0 from (5.22)		
	$\xrightarrow{f', \overline{\mathbf{r}}}$	
		Check $f' := H(\mathbf{A}\overline{\mathbf{r}} - f'D, \mu)$

(5.19) by $(f_i'')^{-1}$ and get:

$$\mathbf{cn}_i = \mathsf{Com}_{\mathbf{A}}\left(b_i, (f_i'')^{-1} \cdot \mathbf{r}_i\right), \tag{5.20}$$

for all $i \in [\ell_\$]$. We now extract an opening $(f', \tilde{\mathbf{r}})$ of a commitment to 0 in the last step of the range proof protocol such that:

$$\begin{aligned} & f' \cdot \left(\sum_{i=0}^{\ell_\$ - 1} 2^i \cdot \mathbf{cn}_i - \mathbf{cn}\right) = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{r}) \\ \iff \quad & f' \cdot \sum_{i=0}^{\ell_\$ - 1} 2^i \cdot \mathbf{cn}_i - f' \cdot \mathbf{cn} = \mathsf{Com}_{\mathbf{A}}(0, \mathbf{r}) \end{aligned} \tag{5.21}$$

holds. Note that we use the PoK^* protocol from [Lyu12] which we adapt to our setting using $D := \left(\sum_{i=0}^{\ell_\$ - 1} 2^i \cdot \mathbf{cn}_i - \mathbf{cn}\right)$ and present in the Table 5.4:
where

$$\rho_0 := 1 - \min\left\{\frac{D_{\sigma_0}^{n(m-1)}(\overline{\mathbf{r}})}{M \cdot D_{(f' \cdot \mathbf{r}), \sigma_0}^{n(m-1)}(\overline{\mathbf{r}})}, 1\right\} \tag{5.22}$$

and $\sigma_0 = 12n\sqrt{n(m-1)}$. Using (5.20), it follows that:

$$f' \cdot \sum_{i=0}^{\ell_\$ - 1} 2^i \cdot \mathbf{cn}_i = f' \cdot \mathsf{Com}_{\mathbf{A}}\left(\sum_{i=0}^{\ell_\$ - 1} 2^i \cdot b_i, \sum_{i=0}^{\ell_\$ - 1} 2^i (f_i'')^{-1} \cdot \mathbf{r}_i\right). \tag{5.23}$$

After inserting the definition of $\mathbf{cn}$ into (5.21), we obtain:

$$f' \cdot \mathbf{cn} = f' \cdot \sum_{i=0}^{\ell_\$ - 1} 2^i \cdot \mathbf{cn}_i - \mathsf{Com}_{\mathbf{A}}(0, \mathbf{r}) \tag{5.24}$$

Next, we insert (5.23) into (5.24) to get:

$$\begin{aligned} f' \cdot \mathbf{cn} &= f' \cdot \mathsf{Com}_{\mathbf{A}} \left(\sum_{i=0}^{\ell_{\$}-1} 2^i \cdot b_i, \sum_{i=0}^{\ell_{\$}-1} 2^i (f_i'')^{-1} \cdot \mathbf{r}_i \right) - \mathsf{Com}_{\mathbf{A}}(0, \mathbf{r}) \\ &= \mathsf{Com}_{\mathbf{A}} \left(f' \cdot \$, f' \cdot \sum_{i=0}^{\ell_{\$}-1} 2^i (f_i'')^{-1} \cdot \mathbf{r}_i - \mathbf{r} \right), \end{aligned} \tag{5.25}$$

where we set $\$ = \sum_i 2^i b_i$ (note $\$ \in [0, 2^{\ell_{\$}} - 1]$). Now, we would have liked to show that $f \cdot (f_i'')^{-1} \cdot \mathbf{r}_i$ is 'small', but it is not. Instead, assume, there is a small and invertible $g \in \mathcal{R}_q$, such that $g \cdot (f_i'')^{-1} = h_i$ is small. Since f_i'' is a non-zero difference of monomials from $\mathcal{C}_0$, by Lemma 3.3, we can take $g = 2$ as it is small and invertible in $\mathcal{R}_q$. Multiplying the right hand-side of (5.25) by g yields:

$$(g \cdot f') \cdot \mathbf{cn} = \mathsf{Com}_{\mathbf{A}} \left(g \cdot f' \cdot \$, f' \cdot \sum_{i=0}^{\ell_{\$}-1} 2^i h_i \cdot \mathbf{r}_i - g \cdot \mathbf{r} \right). \tag{5.26}$$

We get the desired 'small' range proof witness $\left(f = g \cdot f', \$, \mathbf{r}' = f' \cdot \sum_{i=0}^{\ell_{\$}-1} 2^i h_i \cdot \mathbf{r}_i - g \cdot \mathbf{r} \right)$, where $\|\mathbf{r}'\| \leq \beta_{range}$ and using estimations from Lemma 3.4,

$$\begin{aligned} \beta_{range} &= \left\| f \cdot \sum_{i=0}^{\ell_{\$}-1} h_i \cdot \mathbf{r}_i - g \cdot \mathbf{r} \right\|_2 \leq \left\| f \cdot \sum_{i=0}^{\ell_{\$}-1} h_i \cdot \mathbf{r}_i \right\|_2 + \|g \cdot \mathbf{r}\|_2 \\ &\leq \sqrt{n} \cdot \|f\|_\infty \cdot \left\| \sum_{i=0}^{\ell_{\$}-1} 2^i h_i \cdot \mathbf{r}_i \right\|_\infty + \sqrt{n} \cdot \|g\|_\infty \cdot \|\mathbf{r}\|_\infty \\ &\leq \sqrt{n} \cdot 2\sqrt{\kappa} \cdot \sqrt{n} \cdot 2^{\ell_{\$}} \cdot 2\sqrt{n(m-1)}\sigma_{OR} + \sqrt{n} \cdot 2\sqrt{\kappa} \cdot 2\beta_v \\ &\leq 2^{\ell_{\$}+2} n \sqrt{\kappa n (m-1)}\sigma_{OR} + 2^2 \sqrt{n\kappa}\beta_v. \end{aligned} \tag{5.27}$$

SHVZK: Here we have to show that our range proof satisfies the requirement of perfect simulation. Since the underlying OR proof is perfectly simulatable as showed in the last proof of Theorem 5.23, we only need to show that the underlying proof of knowledge from Table 5.4 is simulatable too. Given a challenge f', the simulator aborts with probability $1 - 1/M^2$. Otherwise, we have to show the PoK

is zero-knowledge. To do so the simulator picks $\bar{\mathbf{r}} \hookleftarrow D^{n(m-1)}_{12n\sqrt{n(m-1)}}$ and computes $\mathbf{A}\cdot\bar{\mathbf{r}} - f'D$ to guarantee that the verification equation $f' = H(\mathbf{A}\cdot\bar{\mathbf{r}} - f'D)$ is satisfied. The simulator outputs simulated transcript $\bar{\mathbf{r}}, f'$, which is indistinguishable by rejection sampling (3.8) and by hiding property declared in Theorem 5.19 of our commitment scheme. □

5.8 Security Analysis of the MIMO.LRCT

Theorem 5.25 (Balance)**.** *If* MIMO.L2RS *with parameter* β_v *is unforgeable, linkable and* $\mathsf{Com}_{\boldsymbol{A}}$ *is* $\beta-$*binding with* $\beta = 4\sqrt{\kappa(2\beta_v)^2 + \kappa(2\beta_v)^2 n(m-1)((2N_{in}+N_{out})2^{\gamma})^2} + 2\beta_v N_{out}(2^{\ell_{\$}+2} n\sqrt{\kappa n(m-1)}\sigma_{OR} + 2^2\sqrt{n}\beta_v)$*, then* MIMO.LRCT *satisfies balance.*

Proof. By definition of successful balance attack (Definition 5.15), $\exists i \in [w]$ such that $\sum_{k\in E^{i^*}_{(in)}} \$^{(k)}_{(in),i^*} < \sum_{j\in G^{i^*}_{(out)}} \$^{(j)}_{(out),i^*}$, being i^* a dishonest transaction. For this analysis we consider three cases, case 1.1, case 1.2 and case 2:

- **Case 1 -** $TN \in \mathcal{TN}$ **from ActGen**: we consider two sub-cases, the outsider and insider attacks which are described as follows:
  - **Case 1.1 - The outsider attack**: $\forall i \in [w]$ $\exists k^* \in [N_{in}]$ such that $IW_i^{(k^*)}$ is not corrupted, this means that not all inputs to $\mathcal{T}_{i^*}$ are corrupted. We show that given any PPT MIMO.LRCT adversary, we can construct a MIMO.L2RS adversary, which has equal advantage. In doing this reduction, we firstly define the entities interacting to prove LRCT-Unforgeability. We use a challenger, MIMO.L2RS.Challenger, and two adversaries MIMO.L2RS($\mathcal{B}$) and MIMO.LRCT($\mathcal{A}$). This experiment begins with the challenger who generates the Pub-Params $\leftarrow$ MIMO.L2RS.Setup(λ), and these Pub-Params are given to the adversary $\mathcal{B}$. This adversary then runs $\mathcal{A}$, by simulating $\mathcal{A}$'s oracle answers

(Definition 5.15). We assume that $\mathcal{A}$ makes at most q_{ad}, q_{ac}, q_{co} and q_{spend} queries to AddGen, ActGen, Corrupt and O-Spend respectively. This simulation runs as follows:

* AddGen(i): on input a query number i, $\mathcal{B}$ forwards the query to its own $\mathcal{JO}$ and obtains the public-key(s) $\mathsf{pk}_i^{(k)}$. $\mathcal{B}$ returns these to $\mathcal{A}$.
* ActGen$(i, \$_i)$: on input address index i and an amount $\$_i$, $\mathcal{B}$ runs algorithm MIMO.LRCT.Mint(Pub-Params, $\$_i$) and returns the account $IW_i = \left(\mathsf{pk}_i^{(k)}, \mathbf{cn}_i^{(k)}\right)$ and its corresponding $\mathbf{ck}_i^{(k)}$ to $\mathcal{A}$.
* O-Spend$(\mu, IW, IW_i, OA, \$_{(out)}^{(j)}, \text{Pub-Params})$: on input the transaction strings μ, input wallet IW containing IW_i, output addresses OA, and Pub-Params, $\mathcal{B}$ creates a signature by calling its signing oracle as: $\sigma_{L'}(\mu)_i \leftarrow \mathcal{SO}(w, IW, \mathsf{pk}_i^{(k)}, \mu)$, then $\mathcal{B}$ builds the MIMO.LRCT.Spend output as $(TX_i, \mathsf{sig}_i, TN_i)$, where $TX = (\mu, IW, OA)$, TN_i is the linking tag, and $\mathsf{sig}_i = (\sigma_{L'}(\mu)_i, \{\sigma_{range}^{(j)}\}_{j\in[N_{out}]})$. These outputs are returned to $\mathcal{A}$.
* Corrupt(i): on input query number i, $\mathcal{B}$ calls its corruption oracle to obtain the private key $\mathsf{sk}_i^{(k)} \leftarrow \mathcal{CO}(\mathsf{pk}_i^{(k)})$. This private-key is returned to $\mathcal{A}$.

$\mathcal{A}$ outputs a forgery transaction $(TX^*, \mathsf{sig}_\pi^*, TN^*)$ such MIMO.LRCT.Verify$(TX^*, \mathsf{sig}_\pi^*, TN^*) = 1$ where $\mathsf{sig}_\pi^* = (\sigma_{L'}(\mu)_\pi^*, \{\sigma_{range}^{(j)}\}_{j\in[N_{out}]}^*)$. $\mathcal{B}$ also outputs its forgery $\sigma_{L'}(\mu)_\pi^*$ and IW^*, where IW^* is the input list in TX^*. We show that the advantage of MIMO.L2RS$(\mathcal{B})$ adversary is equal as the advantage of MIMO.LRCT$(\mathcal{A})$ to break the unforgeability property. In this simulation, $\mathcal{A}$'s view is perfectly simulated by $\mathcal{B}$ as in the real balance game. Moreover, in the event where $\mathcal{A}$ wins the game and case 1.1 occurs, then $\mathcal{B}$ also wins its game. This forgery meets the conditions of both definitions, the MIMO.L2RS one-time unforgeability (Definition 5.1) and balance (Definition 5.15), which we summarise below:

1. In both views $\mathsf{MIMO.LRCT.Verify}(\cdot) = 1$ (Definition 5.15, Condition 1) and $\mathsf{MIMO.L2RS.SigVer}(\cdot) = 1$ (Definition 5.1, Condition 1), transaction signatures must be valid.
2. The $\mathsf{pk}_i^{(k)}$ of the list accounts were generated during the simulation by AddGen oracle (Definition 5.15, Condition 2) and this oracle forwarded queries to the $\mathsf{MIMO.L2RS}$'s oracle $\mathcal{JO}(\cdot)$ (Definition 5.1, Condition 2).
3. The forgeries sig_π^* and $\sigma_{L'}(\mu)_\pi^*$ are not the output of the $\mathsf{O\text{-}Spend}(\cdot)$ (Definition 5.15, Condition 3) and $\mathcal{SO}(\cdot)$ (Definition 5.1, Condition 3) oracles, respectively.
4. $\mathsf{pk}_\pi^{(k)}$ was only queried to $\mathsf{O\text{-}Spend}(\cdot)$ oracle once (Definition 5.15, Condition 4), and thus only a query was forwarded to $\mathcal{SO}(\cdot)$ (Definition 5.1, Condition 4).
5. The condition of this case 1.1 ($\forall i \in [w]\ \exists k^* \in [N_{in}]$ such that $IW_i^{(k^*)}$) implies that $\exists k^*\ s.t.\ \mathsf{pk}_i^{(k^*)}$ is not corrupted (Definition 5.1, Condition 5). Therefore, this also meets the condition of the $\mathsf{MIMO.L2RS}$.

To sum up, if the outsider adversary breaks this case 1.1 attack, then we refer to the Theorem 5.9 (Unforgeability) where the security analysis reduces to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem with $\beta = 2\beta_v$. Thus we said that $\beta_{case1.1} = 2\beta_v$.

– **Case 1.2 - The insider attack**: $\exists i \in [w]\ \forall k^* \in [N_{in}]$ such that $IW_i^{(k^*)}$ is corrupted, meaning that all inputs to $\mathcal{T}_{i^*}$ are corrupted. We start this case by running the extractor of the $\mathsf{MIMO.L2RS}$'s proof of knowledge (in Proposition 5.21) so we can extract the witness of this signature relation as $\mathbf{a}_{(in),i^*}^{(N_{in}+1)} \cdot \mathbf{v}_{i^*,(2)}^{(N_{in}+1)} = \mathsf{Com}_{\mathbf{A}}\big(0, \mathbf{v}_{i^*,(1)}^{(N_{in}+1)}\big)$ with $(\mathbf{v}_{i^*,(1)}^{(N_{in}+1)}, \mathbf{v}_{i^*,(2)}^{(N_{in}+1)})^T \neq \mathbf{0} \bmod q$. For simplicity, we define $\mathbf{g}_{L2RS} \triangleq \mathbf{v}_{i^*,(2)}^{(N_{in}+1)}$ and $\mathbf{r} \triangleq \mathbf{v}_{i^*,(1)}^{(N_{in}+1)}$. Then, we have $\mathbf{a}_{(in),i^*}^{(N_{in}+1)} = \mathbf{g}_{L2RS}^{-1} \cdot \mathsf{Com}_{\mathbf{A}}\big(0, \mathbf{r}\big) = \sum_{k=1}^{N_{in}} \mathbf{a}_{(in),i^*}^{(k)} + \mathbf{cn}_{(in),i^*}^{(k)} - \sum_{j=1}^{N_{out}} \mathbf{cn}_{(out),i^*}^{(j)}$ from

definition of $\mathbf{a}_{(in)}^{(N_{in}+1)}$ in (5.14) [from Section 5.6]. We said that $\sum_{j=1}^{N_{out}} \mathbf{cn}_{(out),i^*}^{(j)} \triangleq \sum_{j\in G_{(out)}^{i^*}} \mathbf{cn}_{(out),i^*}^{(j)} + \sum_{j\in E_{(out)}^{i^*}} \mathbf{cn}_{(out),i^*}^{(j)}$ where G_{out} and E_{out} are $\mathcal{T}_{i^*}$'s not corrupted and corrupted outputs, respectively. Then, replacing with this definition, we have $\mathbf{g}_{L2RS}^{-1} \cdot \mathsf{Com}_{\mathbf{A}}\big(0, \mathbf{r}\big) = \mathsf{Com}_{\mathbf{A}}\left(\sum_{k=1}^{N_{in}} \$_{(in),i^*}^{(k)} - \sum_{j\in G_{(out)}^{i^*}} \$_{(out),i^*}^{(j)}, \mathbf{S}_{(in),i^*}^{(N_{in}+1)}\right) + \sum_{j\in E_{(out)}^{i^*}} \mathbf{cn}_{(out),i^*}^{(j)}$. The latter equation is equivalent to $\sum_{j\in E_{(out)}^{i^*}} \mathbf{cn}_{(out),i^*}^{(j)} = \mathbf{g}_{L2RS}^{-1} \cdot \mathsf{Com}_{\mathbf{A}}\left(\sum_{k=1}^{N_{in}} \$_{(in),i^*}^{(k)} - \sum_{j\in G_{(out)}^{i^*}} \$_{(out),i^*}^{(j)}, \mathbf{r} - \mathbf{S}_{(in),i^*}^{(N_{in}+1)}\right)$. Afterwards, we multiply both sides by $\mathbf{g}_{L2RS}$ and it results in:

$$\mathbf{g}_{L2RS} \cdot \sum_{j\in E_{(out)}^{i^*}} \mathbf{cn}_{(out),i^*}^{(j)} = \mathsf{Com}_{\mathbf{A}}\left(\mathbf{g}_{L2RS} \cdot \Delta, \overline{\mathbf{r}}\right), \tag{5.28}$$

where $\overline{\mathbf{r}} \triangleq \mathbf{g}_{L2RS} \cdot (\mathbf{r} - \mathbf{S}_{(in),i^*}^{(N_{in}+1)})$ and $\Delta \triangleq \sum_{k=1}^{N_{in}} \$_{(in),i^*}^{(k)} - \sum_{j\in G_{(out)}^{i^*}} \$_{(out),i^*}^{(j)}$. Since $\$_{(in),i^*}^{(k)} \in [0, 2^{\ell_\$} - 1]$ and $\max(N_{in}, N_{out}) \leq N$, then $\sum_{k=1}^{N_{in}} \$_{(in),i^*}^{(k)} \in [0, N \cdot 2^{\ell_\$} - 1]$ and $\sum_{j\in G_{(out)}^{i^*}} \$_{(out),i^*}^{(j)} \in [0, N \cdot 2^{\ell_\$} - 1]$ where $N_{in} \leq N$ and $N_{out} \leq N$, respectively. We have, $\left|\$_{(in),i^*}^{(k)} - \$_{(out),i^*}^{(j)}\right| \leq N \cdot (2^{\ell_\$} - 1)$ and $\$_{(in),i^*}^{(k)} - \$_{(out),i^*}^{(j)} < 0$, which is less than $q/2$ by the choice of q. Therefore, $\Delta \bmod q = \Delta \in [-N \cdot (2^{\ell_\$} - 1), -1]$. We now run the proof of knowledge extractor of parallel range proofs from Theorem 5.24, $\forall j \in [E_{out}]$ $\mathbf{cn}_{(out),i^*}^{(j)}$. We then obtain:

$$\mathbf{g}_{Range} \cdot \sum_{j\in E_{(out)}^{i^*}} \mathbf{cn}_{(out),i^*}^{(j)} = \mathsf{Com}_{\mathbf{A}}\left(\mathbf{g}_{Range} \cdot \overline{\$}, \overline{\mathbf{ck}}\right), \tag{5.29}$$

where $\mathbf{g}_{Range} = f, \overline{\$} \triangleq \sum_{j\in E_{(out)}^{i^*}} \overline{\$}_{(out)}^{(j)}$ and the randomness $\overline{\mathbf{ck}} \triangleq \sum_{j\in E_{(out)}^{i^*}} \overline{\mathbf{ck}}_{(out)}^{(j)}$, as per in (5.26). If we multiply and subtract both equations (5.28) and (5.29) by $\mathbf{g}_{Range}$ and $\mathbf{g}_{L2RS}$, respectively, it results to $0 = \mathsf{Com}_{\mathbf{A}}\left(\mathbf{g}_{Range} \cdot \mathbf{g}_{L2RS} \cdot (\Delta - \overline{\$}), \mathbf{g}_{Range} \cdot \overline{\mathbf{r}} - \mathbf{g}_{L2RS} \cdot \overline{\mathbf{ck}}\right)$. Assuming that $\|\mathbf{g}_{Range} \cdot \mathbf{g}_{L2RS}\| < \frac{1}{\sqrt{k}} \cdot q^{1/k}$ where k denotes the number of irreducible factors mod$q$ of $x^n + 1$, then by [Corollary 1.2 from 2.[LS18]], $\mathbf{g}_{Range} \cdot \mathbf{g}_{L2RS}$ is invertible in $\mathcal{R}_q$. This implies that $\mathbf{g}_{Range} \cdot \mathbf{g}_{L2RS} \cdot (\Delta - \overline{\$}) \neq 0 \bmod q$, using the fact that $\Delta - \overline{\$} \neq 0 \bmod q$. Therefore, we obtain a $\beta-$binding collision for $\mathsf{Com}_{\mathbf{A}}$

with $\beta-$binding $= \left\|\mathbf{g}_{Range} \cdot \bar{\mathbf{r}} - \mathbf{g}_{L2RS} \cdot \overline{\mathbf{ck}}\right\| \leq \beta_{case1.2}$. By replacing this $\beta-$binding with the results of the witness extraction, it turns out that $\beta_{case2.2} \geq 4\sqrt{\kappa(2\beta_v)^2 + \kappa(2\beta_v)^2 n(m-1)((2N_{in}+N_{out})2^{\gamma})^2} + 2\beta_v N_{out}(2^{\ell_\$+2} n\sqrt{\kappa n(m-1)}\sigma_{OR} + 2^2\sqrt{n}\beta_v)$.

- **Case 2 - $TN \notin \mathcal{TN}$ from ActGen (Linkability Attack)**: $\exists k^* \in [N_{in}]$ *s.t.* IW_i^{k*} with $i \in [w]$ was queried to O-Spend, where k^*'th is the real input account in the forgery transaction with TN, and $TN \not\subseteq \mathcal{TN}$.

  In this proof, we show that any PPT MIMO.LRCT adversary has equal advantage to the corresponding MIMO.L2RS adversary. In doing this reduction, we firstly define the entities interacting to prove the LRCT-Linkability. We use a challenger, MIMO.L2RS.Challenger, and two adversaries MIMO.L2RS($\mathcal{B}$) and MIMO.LRCT($\mathcal{A}$). This experiment begins with the challenger who generates the Pub-Params $\leftarrow$ MIMO.L2RS.Setup(λ), and these are given to the adversary $\mathcal{B}$. This adversary then runs $\mathcal{A}$, by simulating $\mathcal{A}$'s oracle answers (see Section 5.4.1). We assume that $\mathcal{A}$ makes at most q_{ad}, q_{ac} queries to AddGen and ActGen respectively, then by querying the oracle O-Spend, it will generate a signature or PoK. This simulation runs as follows:

  - AddGen(i): on input a query number i, $\mathcal{B}$ forwards the query to its own $\mathcal{JO}$ and obtains the public-key(s) $\mathsf{pk}_i^{(k)}$. $\mathcal{B}$ returns these to $\mathcal{A}$.
  - ActGen($i, \$_i$): on input address index i and an amount $\$_i$, $\mathcal{B}$ runs algorithm LRCT.Mint(Pub-Params, $\$_i$) and returns the account $IW_i = \left(\mathsf{pk}_i^{(k)}, \mathbf{cn}_i^{(k)}\right)$ and its corresponding $\mathbf{ck}_i^{(k)}$ to $\mathcal{A}$.
  - O-Spend($\mu, IW, IW_i, OA, \$_{(out)}^{(j)}$, Pub-Params): on input the transaction strings μ, input wallet IW containing IW_i, output addresses OA, and Pub-Params, $\mathcal{B}$ creates a signature by calling its signing oracle as: $\sigma_{L'}(\mu)_i \leftarrow \mathcal{SO}(w, IW, \mathsf{pk}_i^{(k)}, \mu)$, then $\mathcal{B}$ builds the MIMO.LRCT.Spend output as $(TX_i, \mathsf{sig}_i, TN_i)$, where $TX = (\mu, IW, OA)$, TN_i is the linking

tag, and $\mathsf{sig}_i = (\sigma_{L'}(\mu)_i, \{\sigma^{(j)}_{range}\}_{j\in[N_{out}]})$. These outputs are returned to $\mathcal{A}$.

– $\mathsf{Corrupt}(i)$: on input query number i, $\mathcal{B}$ calls its corruption oracle to obtain the private key $\mathsf{sk}^{(k)}_i \leftarrow \mathcal{CO}(\mathsf{pk}^{(k)}_i)$, and this private-key is returned to $\mathcal{A}$.

$\mathcal{A}$ outputs two transaction forgeries $(TX^*, \mathsf{sig}^*_\pi, TN^*)$ and $(TX'^{,*}, \mathsf{sig}'^{,*}_\pi, TN'^{,*})$, whereas $\mathcal{B}$ outputs two signature forgeries $\sigma_{L^*}(\mu)^*_\pi$ and $\sigma_{L^*}(\mu)'^{,*}_\pi$ with their corresponding IW^* and $IW'^{,*}$ which were taken from TX^* and $TX'^{,*}$, respectively. These forgeries meet the conditions of the balance MIMO.LRCT (Definition 5.15) and the MIMO.L2RS linkability definition (Definition 5.3), and we summarise these as:

1. In both views $\mathsf{MIMO.LRCT.Verify}(\cdot) = 1$ (Definition 5.15, Condition 1) and $\mathsf{MIMO.L2RS.SigVer}(\cdot) = 1$ (Definition 5.3, Condition 1), transaction signatures must be valid.
2. The $\mathsf{pk}^{(k)}_i$ of the list accounts were generated during the simulation by AddGen oracle (Definition 5.15, Condition 2) and this oracle forwarded queries to the MIMO.L2RS's oracle $\mathcal{JO}(\cdot)$ (Definition 5.3, Condition 2).
3. Condition 3 of (Definition 5.15) implies $\mathsf{MIMO.L2RS.SigLink}(\sigma_{L^*}(\mu)^*_\pi, \sigma_{L^*}(\mu)'^{,*}_\pi) = \mathit{Unlinked}$ (Definition 5.3, Condition 3).

We showed that the advantage of $\mathsf{MIMO.L2RS}(\mathcal{B})$ adversary is equal as $\mathsf{MIMO.LRCT}(\mathcal{A})$ to break the linkability property. Then, we refer to the Theorem 5.11 (Linkability) where the security analysis reduces to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem with $\beta = 2\beta_v$. Thus we said that $\beta_{case2} = 2\beta_v$.

□

Corollary 5.26 (Balance)**.** *The Balance of MIMO.LRCT is satisfied if* $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta_{Balance}}$ *is hard where* $\beta_{balance} = \max(\beta_{case1.1}, \beta_{case1.2}, \beta_{case2})$.

Proof. By combining Theorem 5.25 (Balance), along with Theorem 5.20 ($\beta-$Binding), Theorem 5.9 (MIMO.L2RS Unforgeability) and Theorem 5.11 (Linkability), this analysis concludes that the $\beta_{balance} = \max(\beta_{case1.1}, \beta_{case1.2}, \beta_{case2})$. $\beta_{case1.2}$ is seen as the maximum, then we said that $\beta_{balance} = 4\sqrt{\kappa(2\beta_v)^2 + \kappa(2\beta_v)^2 n(m-1)((2N_{in}+N_{out})2^{\gamma})^2} + 2\beta_v N_{out}(2^{\ell_\$+2} n\sqrt{\kappa n(m-1)}\sigma_{OR} + 2^2\sqrt{n}\beta_v)$. □

Remark 5.27. In the balance proof, we only need zero-time unforgeability, meaning that in the reduction the attacker produces a forgery without seeing any signatures. Secondly, we do not need the message part of the signature, and thus this is treated as a Proof of Knowledge.

Theorem 5.28 (LRCT-Anonymity)**.** *Suppose* $n \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ *and* $o \cdot h \cdot 2^{-n+1}$ *are negligible in* n *with an attack against the unconditional anonymity that makes* h *queries to the random oracle* H_1*, then the MIMO.LRCT scheme is unconditionally secure for anonymity and amount privacy as defined in Definition 5.16.*

Proof. We prove the anonymity of this scheme using the sequence-of-games approach. We begin our analysis by:

Game 0 - Real Game: We firstly define the entities interacting to prove this LRCT-Anonymity property. We use a challenger, MIMO.LTCR.Challenger, and two adversaries, MIMO.LRCT($\mathcal{A}_1$) and MIMO.LRCT($\mathcal{A}_2$). This experiment begins with the challenger who generates the Pub-Params $\leftarrow$ MIMO.L2RS.Setup(λ), and this output is given to the adversary $\mathcal{A}_1$. Then, $\mathcal{A}_1$ runs the oracles, which were defined in Definition 5.16. We assume that $\mathcal{A}_1$ makes at most q_{ad}, q_{ac}, and q_{co} queries to AddGen, ActGen, and Corrupt, respectively. This simulation runs as follows:

- AddGen(i): on input a query number i, it returns the public-key(s) $\mathsf{pk}_i^{(k)}$.
- ActGen($i, \$_i$): on input address index i and an amount $\$_i$, $\mathcal{A}_1$ runs algorithm LRCT.Mint(Pub-Params, $\$_i$) and returns the account $IW_i = \left(\mathsf{pk}_i^{(k)}, \mathbf{cn}_i^{(k)}\right)$ and its corresponding $\mathbf{ck}_i^{(k)}$.

- $\mathsf{O\text{-}Spend}(\mu, IW, IW_i, OA, \{\$^{(j)}_{(out),i}\}_{j\in[N_{out}]}, \mathsf{Pub\text{-}Params})$, and it outputs $(TX, \mathsf{sig}_i, TN_i)$.
- $\mathsf{Corrupt}(i)$: on input query number i, it outputs $(\mathsf{sk}_i^{(k)}, \mathsf{ck}_i^{(k)})$.

Now $\mathcal{A}_1$ construct IW with w accounts from the ActGen's queries q_{ac}, then it selects two elements π_0 and π_1 from IW, such $IW_{\pi_0} = \{\mathsf{pk}^{(k)}_{\pi,0}, \mathbf{cn}^{(k)}_{\pi,0}\}_{k\in[N_{in}]}$ and $IW_{\pi_1} = \{\mathsf{pk}^{(k)}_{\pi,1}, \mathbf{cn}^{(k)}_{\pi,1}\}_{k\in[N_{in}]}$, with $\mathsf{pk}^{(k)}_{\pi,0} = \mathsf{Com}_{\mathbf{A}}(\mathbf{0}, \mathbf{sk}^{(k)}_{\pi,0})$, $\mathsf{pk}^{(k)}_{\pi,1} = \mathsf{Com}_{\mathbf{A}}(\mathbf{0}, \mathbf{sk}^{(k)}_{\pi,1})$, $\mathbf{cn}^{(k)}_{\pi,0} = \mathsf{Com}_{\mathbf{A}}(\$^{(k)}_{(in),0}, \mathbf{ck}^{(k)}_{\pi,0})$ and $\mathbf{cn}^{(k)}_{\pi,1} = \mathsf{Com}_{\mathbf{A}}(\$^{(k)}_{(in),1}, \mathbf{ck}^{(k)}_{\pi,1})$. After this $\mathcal{A}_1$ outputs $(\mu, IW_{\pi_0}, IW_{\pi_1}, IW, OA, \$^{(j)}_{(out),0}, \$^{(j)}_{(out),1})$, where $\sum_{k=1}^{N_{in}} \$^{(k)}_{(in),0} = \sum_{j=1}^{N_{out}} \$^{(j)}_{(out),0}$ and $\sum_{k=1}^{N_{in}} \$^{(k)}_{(in),1} = \sum_{j=1}^{N_{out}} \$^{(j)}_{(out),1}$. The MIMO.LRCT.Challenger picks at random $b = \{0, 1\}$ and returns $(TX^*, \mathsf{sig}^*_b, TN^*_b) \leftarrow$ $hookleftarrow$RCT.Spend$(\mu, K_{\pi_b}, IW_{\pi_b}, IW, OA, \$_{(out)_b}\mathsf{Pub\text{-}Params})$ where $IW_{\pi_b} = \{\mathbf{pk}^{(k)}_{\pi_b}, \mathbf{cn}^{(k)}_{\pi_b}\}$ and $\mathbf{cn}^{(k)}_{\pi_b} = \mathsf{Com}_{\mathbf{A}}(\$^{(k)}_{(in),\pi_b}, \mathbf{ck}^{(k)}_{\pi_b})$ to $\mathcal{A}_2$. The adversary $\mathcal{A}_2$ runs the oracles as (Definition 5.16):

- $\mathsf{O\text{-}Spend}(\mu, IW', IW'_\pi, OA, \$^{(j)}_{(out)}, \mathsf{Pub\text{-}Params})$ with $IW' \neq IW$ and $IW'_\pi \neq W_{\pi_0}, W_{\pi_1}$. This outputs $(TX^{*\prime}, \mathsf{sig}^{*\prime}_{b'}, TN^{*\prime}_{b'})$, with $TN^{*\prime}_{b'} = \mathsf{Com}_{\mathbf{H}}(\mathbf{0}, \mathbf{sk}^{(k)\prime}_{\pi,b'})$
- $\mathsf{Corrupt}(i)$: on input query number i, it returns $(\mathsf{sk}_i^{(k)}, \mathsf{ck}_i^{(k)})$.

The adversary $\mathcal{A}_2$ outputs b'. If we define the S_0 to be the event that $b = b'$, then the $\mathcal{A}_2$'s advantage is $|\Pr[S_0] - \frac{1}{2}|$.

Game 1 - Signature: In this game, we perform changes in the signature $\mathsf{sig}^*_b = (\sigma_{L'}(\mu)_b, \{\sigma_{range}{}^{(j)}_b\}_{j\in[N_{out}]})$, in particular $\sigma_{L'}(\mu)_b$. Instead of generating this real signature with MIMO.L2RS.SigGen (Algorithm 11), we use the hybrids MIMO.L2RS.Hybrid-1 and MIMO.L2RS.Hybrid-2, Algorithms 14 and 15, respectively; based on our security analysis in Section 5.3 (MIMO.L2RS unforgeability). In the transition from the real signature to hybrid 1, the $(\mathbf{c}_{\pi+1})_b$ is chosen at random. This transition concluded that the statistical distance between $\mathbf{c}_{\pi+1}$ and $\mathcal{S}_{n,\kappa} \subseteq \mathcal{R}_{2q}$ will be at most $\epsilon_{Game1} = o \cdot h \cdot 2^{-n+1}$, which is negligible (Based on [DDLL13], Lemma 3.4), where h and o are the number of queries to H_1 and the hybrid 1, respectively. We now consider the transition from hybrid 1 to hybrid

2. The output of both hybrids follows the same distribution due to the rejection sampling (Lemma 3.8). This means that choosing $\mathbf{t}_\pi$ at random, will not have any effect in the output of both hybrids. Let S_0 be the event that $b = b'$ in Game 1. We claim that the view of the adversary in Game 0 and Game 1 is:

$$|\Pr[S_0] - \Pr[S_1]| \leq \epsilon_{Game1}. \tag{5.30}$$

Game 2 - User Anonymity ($\pi_0 \neq \pi_1$ with $\$_{(out),0} = \$_{(out),1}$): Changes in this game are made to TN_b^* and $\mathbf{pk}_{\pi_b}$. TN_b^* is now randomly chosen from $\mathcal{R}_q^2$. When $b' = 0$, then $\mathbf{pk}_{\pi_0} \hookleftarrow \mathcal{R}_q^2$ whereas $\mathsf{pk}_{\pi,1} = \mathsf{Com}_{\mathbf{A}}(\mathbf{0}, \mathbf{sk}_{\pi,1})$. When $b' = 1$ then $\mathbf{pk}_{\pi_1} \hookleftarrow \mathcal{R}_q^2$ whereas $\mathsf{pk}_{\pi,0} = \mathsf{Com}_{\mathbf{A}}(\mathbf{0}, \mathbf{sk}_{\pi,0})$. When $\mathbf{sk}_{\pi,b}$ is multiplied by $\mathbf{H}$ and $\mathbf{A}$ respectively, it gives TN_b^* and $\mathbf{pk}_{\pi_b}$ that are close to uniform over $\mathcal{R}_q^2$. By applying the Leftover Hash Lemma (LHL) - **Lemma 3.5**, the statistical distance between the distribution of (TN_b^* mod q and $\mathbf{pk}_{\pi_b}$ mod q) and the uniform distribution on $\mathcal{R}_q^2 \times \mathcal{R}_q^2$ is at most $\left(n \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}\right)$, which is negligible. Let S_2 be the event that $b = b'$ in Game 2. We claim that

$$|\Pr[S_1] - \Pr[S_2]| \leq \epsilon_{Game2}. \tag{5.31}$$

Game 3 - User Anonymity ($\pi_0 \neq \pi_1$ with $\$_{(out),0} = \$_{(out),1}$): We now transform Game 1 into Game 2, where we choose $\mathbf{pk}_{\pi_{1-b}}$ at random. This means that when $b' = 0$, then $\mathbf{pk}_{\pi_1} \hookleftarrow \mathcal{R}_q^2$ and when $b' = 1$ then $\mathbf{pk}_{\pi_0} \hookleftarrow \mathcal{R}_q^2$. We conclude that by applying the Leftover Hash Lemma (LHL) - **Lemma 3.5**, the statistical distance between the distribution of ($\mathbf{pk}_{\pi_b}$ mod q) and the uniform distribution on $\mathcal{R}_q^2$ is at most $\left(n \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}\right)$ which is negligible. Let S_3 be the event that $b = b'$ in Game 3. We claim that

$$|\Pr[S_2] - \Pr[S_3]| \leq \epsilon_{Game3}. \tag{5.32}$$

Game 4 - Amount Privacy ($\pi_0 = \pi_1$ with $\$_{(out),0} \neq \$_{(out),1}$): In this transitional Game, we choose $\mathbf{cn}_{\pi_b}$ at random, instead of computing $\mathbf{cn}_{\pi_b} =$

$\mathsf{Com}_{\mathbf{A}}(\$_{(out),\pi_b}, \mathbf{ck}_{\pi_b})$. We use the result of the Theorem 5.19 (Homomorphic Commitment Hiding), to show that by applying the Leftover Hash Lemma (Lemma 3.5), we argue that the statistical distance between the distribution of $\mathbf{cn}_{\pi_b}$ and the uniform distribution on $\mathcal{R}_q^2$ is at most $\left(\frac{1}{2} \cdot \sqrt{\frac{q^{2n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}\right)$ which is negligible. Let S_4 be the event that $b = b'$ in Game 4, then we claim that

$$|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_{Game4}. \tag{5.33}$$

Combining (5.30), (5.31), (5.32), and (5.33), we obtain

$$\left|\Pr[S_0] - \frac{1}{2}\right| \leq \epsilon_{Game1} + \epsilon_{Game2} + \epsilon_{Game3} + \epsilon_{Game4},$$

and this is negligible. □

Theorem 5.29 (LRCT-Non-Slanderability)**.** *If MIMO.LRCT satisfies balance, then it satisfies non-slanderability as in Definition 5.17. In addition, the non-slanderability of MIMO.LRCT can be reduced to the non-slanderability of MIMO.L2RS.*

5.9 Performance Analysis

Remark 5.30. This research project did not consider the implementation of the schemes MIMO.L2RS and MIMO.LRCT, as a result there is not run time analysis. The project only evaluates the signature and key sizes of the proposed constructions.

In this section, we propose a set of parameters for the MIMO.LRCT scheme. This construction is secure against direct lattice attacks in terms of the BKZ algorithm Hermite factor δ, using the value of $\delta = 1.007$, based on the BKZ 2.0 complexity estimates with pruning enumeration-based Shortest Vector Problem (SVP) [CN11]. We let $n = 1024$, $m = 132$, $\log q = 196$, $\kappa = 14$, $\eta = 1.1$, $\alpha = 0.5$, $\sigma = 22010$, $\sigma_{OR} = 277350$ and $\ell_{\$} = 64$ to achieve the security parameter $\lambda = 100$, with α

being the rejection sampling parameter determined in ([DDLL13] Section 3.2). Signature sizes of this analysis are illustrated in Table 5.5, where regular numbers for N_{in} and N_{out} were taken from Monero blockchain network[2].

TABLE 5.5: Size estimation for MIMO.LRCT

MIMO.LRCT	$(N_{in}, N_{out}) = (1, 2)$	$(N_{in}, N_{out}) = (2, 2)$	$(N_{in}, N_{out}) = (3, 2)$
$\log(\beta)$ (Theorem 5.25)	≈ 126.3	≈ 126.3	≈ 126.3
Signature size ($w = 1$)	≈ 4.8 MB	≈ 5.1 MB	≈ 5.4 MB
Signature size ($w = 5$)	≈ 6.7 MB	≈ 8 MB	≈ 9.2 MB
Private-key size	≈ 49 KB	≈ 73 KB	≈ 98 KB
Public-key size	≈ 97 KB	≈ 146 KB	≈ 195 KB

5.10 Summary

This chapter has presented the upgraded version of the Lattice-based Ring Confidential Transactions protocol. This new protocol supports Multiple-Input and Multiple-Output (MIMO) wallets in transactions, and it is a fully functional protocol construction for cryptocurrency applications such as *Hcash*. This version of the protocol has been implemented [3]; however, this implementation is out of the scope of this research project. Since the MIMO cryptocurrency setting introduces new balance security requirements (i.e. those against *out-of-range* amount attacks), we have provided a refined balance security model to capture such attacks, as well as several improvements in the anonymity model to capture amount privacy attacks. This protocol extends a previously proposed ring signature scheme in the LRCT v1.0 protocol (described in Chapter 4), to support the MIMO requirements while preserving the post-quantum security guarantees, and uses a lattice-based zero-knowledge range proof to achieve security against *out-of-range* attacks.

As a point of reference for future research, we propose preliminary parameter evaluation and signature sizes. The result of the performance analysis shows that the signature size grows linearly with the number of users in the ring, as well as the number of wallets to be transferred. This results in large signature sizes.

[2] https://moneroblocks.info/
[3] https://github.com/chainchip/Lattice-RingCT-v2.0

Several improvements could be applied to this scheme which include, splitting the transferred amount or using amortisation techniques to reduce the signature size. This work has also already served as a motivation for further study; for instance, the recent work of [EZS+19] produced a significant improvement in a lattice-based RingCT which provided practical signatures sizes of around 100 KB.

In the next chapter, this research project continues to devise an authorisation model to control the wallets that are transferred in the post-quantum cryptographic world. In doing so, a new cryptographic technique is explored, which is a combination of threshold-signature and ring signature schemes. Then, we adjusted this new proposal in our construction, the MIMO.L2RS.

Chapter 6

Lattice-based Linkable Ring Signature with Co-Signing

[1]When electronic wallets are transferred by more than one party, the level of security can be enhanced by decentralising the distribution of authorisation amongst those parties. Threshold signature schemes enable this functionality by allowing multiple cosigners to cooperate in order to create a joint signature. These cosigners interact to sign a transaction which then confirms that a wallet has been transferred. However, in the event of a post-quantum attack, existing threshold signature schemes that support such an authorisation technique in privacy-preserving cryptocurrency protocols - like Ring Confidential Transaction (RingCT) - would not provide adequate security.

[1]This chapter was stored in the public repository https://eprint.iacr.org/2020/1121: Alberto Torres W.A., Steinfeld R., Sakzad A., Kuchta V. (2020) *Post-Quantum Linkable Ring Signature Enabling Distributed Authorised Ring Confidential Transactions in Blockchain [ATSSK20]*

In this chapter, we present a new post-quantum cryptographic mechanism, called Lattice-based Linkable Ring Signature with Co-Signing (L2RS-CS), which offers a distributed authorisation feature to protect electronic wallets. A novel security model for L2RS-CS is also formalised to capture the security and privacy requirements to protect transactions in applications to blockchain cryptocurrency protocols, such as the RingCT. To address key-generation security concerns, and to support compression of keys and signatures, the L2RS-CS incorporates a distributed key generation along with a solid public-key aggregation. Finally, we prove the security of our constructed L2RS-CS in the random oracle model and the standard lattice-based Module-SIS hardness assumption.

We construct the first *post-quantum* Lattice-based Linkable Ring Signature with Co-Signing (L2RS-CS) scheme which can be adapted to a *post-quantum* cryptocurrency protocol such as the LRCT [ATKS+19]. The L2RS-CS offers *complete-anonymity*, and can support Multiple Input wallets to be transferred to Multiple Output wallets (MIMO). The L2RS-CS is built on top of the *post-quantum* LRS from [ATSS+18] and integrates a DKG together with a solid public-key aggregation (in the *post-quantum* settings) which bring a high level of security and compression for the cosigners' keys.

Additionally, we formalise another new security model, called Linkable Ring Signature with Co-Signing (LRS-CS), having a special combination of two constructions, the (N_{CS}-out-of-N_{CS})-TS and (1-out-of-w)-LRS schemes (which are used in Monero [Alo18, GN18]). Although TRS can be seen as a combination of TS and RS schemes, it is a *different* type of primitive to our proposed LRS-CS. Namely, in TRS any subset of t out of n signers can cooperate to generate a signature while hiding the signers' subset. In contrast, under our LRS-CS model, there are w groups of N_{CS} cosigners, so that all the N_{CS} signing keys within the signing group cooperate to produce the signature while hiding the signers among the w groups. Furthermore, in LRS-CS the N_{CS} cosigners interactively generate and share a *single* public-key, whereas in TRS each cosigner has an individual public-key generated with a non-interactive key-generation algorithm. Therefore, LRS-CS can be viewed as a more specialised primitive than TRS; however, one that suffices for RingCT

authorisation and can also be implemented with much shorter signatures than existing lattice-based TRS schemes, as we demonstrate in the evaluation of our scheme.

The security of our proposed L2RS-CS scheme is proven in the classical random oracle model where the properties of *unforgeability*, *linkability* and *non-slanderability* are demonstrated to be computationally secure from the standard lattice-based Module-SIS hardness assumption. In terms of *anonymity*, we show that this construction is unconditionally secure under the Leftover Hash Lemma (LHL) [DDLL13]. Table 6.1 illustrates a comparison of the existing lattice-based TRS schemes, including our L2RS-CS construction.

Finally, the chapter illustrates the selection of concrete and secure parameters as a result of the security analysis to protect the scheme against adversaries and known lattice attacks. These parameters are then use to estimate signature and key-pair sizes that are also compare with similar constructions.

TABLE 6.1: Lattice-based Threshold Ring Signatures with $N_{CS} = 50$ and $w = 100$.

Proposals	**Linkability**	**DKG**	**Aggregate pk**	**Lattice-based Assumption**	**Signature Size**
Cayrel et al. [CLRS10]	✗	✗	✗	SIS	25 MB
Bettaieb et al. [BS13]	✗	✗	✗	ISIS	13 MB
Wei et al. [WDZ+14]	✗	✗	✗	gGCDHP[1]	NP[2]
This work (L2RS-CS)	✓	✓	✓	Module-SIS	1.2 MB

[1] general Graded Computational Diffie-Hellman Problem
[2] Parameter values and signature sizes were not provided

The remaining chapter is structured as follows. In Section 6.1 our multi-signatures scheme is defined and its security model is explained 6.2. Next, the construction of this proposed scheme is described in Section 6.3. The security and performance analyses are shown in Section 6.4 and Section 6.5, respectively. This chapters concludes with a summary in Section 6.6.

6.1 Definition of a Linkable Ring Signature with Co-Signing

In this section, we present the definition of our proposed model, the Linkable Ring Signature with Co-Signing (LRS-CS), which offers an authorisation feature. Under this model, any group of N_{CS} cosigners among w groups has the ability to participate in a protocol that produces the signature, whilst hiding the identity of the signing group. The model also includes a linking tag, making it capable of detecting whether two signatures have been signed by same group of cosigners. Despite this authorisation functionality being implicitly used by Monero [Alo18, GN18], it was not formalised; therefore, we have proposed this new model. The LRS-CS consists of a five-tuple scheme, with (Setup, KeyGen, SigGen, SigVer, SigLink), which we define as follows:

- $\mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda)$: a Probabilistic Polynomial Time (PPT) algorithm that takes the security parameter λ and produces the Public Parameters (PP).
- $(\mathbf{pk}, \mathcal{SK}) \leftarrow \mathsf{KeyGen}(\mathsf{PP})$: a PPT interactive protocol among a number of cosigners (N_{CS}) that by taking the PP, it produces a pair of keys: the aggregate shared public-key $\mathbf{pk}$ and the set of cosigner' secret-keys $\mathcal{SK} = \{\mathbf{sk}_1, \ldots, \mathbf{sk}_{N_{CS}}\}$.
- $\sigma(\mu) \leftarrow \mathsf{SigGen}(\mathcal{SK}, \mu, L, \mathsf{PP})$: a PPT interactive protocol that receives the PP, a message μ, the list L as in (6.1) to be the list of public keys with w users in the ring, and N_{in} inputs (i.e this represents the number of input wallets of each user in a cryptocurrency application). The cosigners owning the secret keys in the set $\mathcal{SK} = \{\mathbf{sk}_{i,1}^{(k)}, \ldots, \mathbf{sk}_{i,N_{CS}}^{(k)}\}$ interact to produce the signature $\sigma(\mu)$.

$$L = \left\{\mathbf{pk}_i^{(k)}\right\}_{i\in[w],k\in[N_{in}]} \tag{6.1}$$

- $(\mathsf{Accept}/\mathsf{Reject}) \leftarrow \mathsf{SigVer}(\sigma(\mu), \mu, L, \mathsf{PP})$: a deterministic algorithm that takes PP, a signature $\sigma(\mu)$, the list L, and the message μ and checks $\sigma(\mu)$

is a correct signature. If the signature is valid, it outputs Accept, otherwise Reject.

- (Linked/Unlinked) $\leftarrow$ SigLink$(\sigma(\mu)_1, \sigma(\mu)_2)$: a deterministic algorithm that verifies if two signatures $\sigma(\mu)_1$ and $\sigma(\mu)_2$ were produced by the same signer while hiding the identity of such signer. Thus, this algorithm outputs Linked if such condition is met, otherwise outputs Unlinked.

The LRS-CS scheme satisfies the SigGen Correctness where valid signatures are produced by honest signers, and it is then accepted by a public verifier with overwhelming probability. We said that the LRS-CS scheme is correct if for any PP $\leftarrow$ Setup(1^λ), a honest user π runs the protocols $(\mathbf{pk}_\pi, \mathcal{SK}_\pi) \leftarrow$ KeyGen(PP), and $\sigma(\mu) \leftarrow$ SigGen$(\mathcal{SK}_\pi, \mu, L, \text{PP})$, it holds that Pr[Accept $\leftarrow$ SigVer$(\sigma(\mu), \mu, L, \text{PP})] = 1 - neg(\lambda)$.

The scheme also achieves SigLink Correctness. Such property guarantees that two valid signatures $\sigma(\mu)_1$ and $\sigma(\mu)_2$ are signed and linked by an honest signer with overwhelming probability. We show that the LRS-CS scheme satisfies SigLink Correctness property if for any PP $\leftarrow$ Setup(1^λ) with a honest user π runs the protocols $(\mathbf{pk}_\pi, \mathcal{SK}_\pi) \leftarrow$ KeyGen(PP), and $\sigma(\mu)_1 \leftarrow$ SigGen$(\mathcal{SK}_\pi, \mu, L, \text{PP})$, $\sigma(\mu)_2 \leftarrow$ SigGen$(\mathcal{SK}_\pi, \mu, L, \text{PP})$, it holds that Pr[Linked $\leftarrow$ SigLink$(\sigma(\mu)_1, \sigma(\mu)_2)] = 1 - neg(\lambda)$.

The communication model assumes that the parties involve in our computational model are connected by a network of point-to-point and broadcast channels.

6.2 Security Model for LRS-CS

Our security model is motivated by [ATSS+18, BS13] where the adversary corrupts and controls the behaviour of $(N_{CS} - 1)$ cosigners, so forging LRS-CS is as hard as solving the underlying hardness problem. This model also captures anonymity, linkability and non-slanderability as principal properties to secure LRS-CS schemes. We begin by defining the oracles that can be accessed by the adversary.

6.2.1 Oracles for adversaries

The following oracles are available to any adversary who tries to break the security of the L2RS-CS scheme $\forall k \in [N_{in}]$:

- $\mathbf{pk}_i^{(k)} \leftarrow \mathcal{KO}(\perp)$. The *KeyGen Oracle*, on request, adds new user(s) to the system. It runs the KeyGen interactive protocol between the challenger (who controls one cosigner) and the adversary (who controls $(N_{CS}-1)$ cosigners). This oracle returns the aggregate shared public-key $\mathbf{pk}_i^{(k)}$.

  Remark 6.1. The challenger $\mathcal{C}$ generates with the KeyGen algorithm, the aggregate shared public-key $\mathbf{pk}_\pi^{(k)}$ and its pair-keys $(\mathbf{pk}_{\pi,1}^{(k)\dagger}, \mathbf{sk}_{\pi,1}^{(k)\dagger})$, where $L^{sh} = \{\mathbf{pk}_{\pi,1}^{(k)\dagger}, \ldots, \mathbf{pk}_{\pi,N_{CS}}^{(k)}\}$. Without loss of generality, we define the $\mathcal{C}$'s public-key $(\mathbf{pk}_{\pi,1}^{(k)\dagger})$ to occur at least once, and to be in the first position of the L^{sh}. On the other hand, the adversary $\mathcal{A}$ arbitrarily chooses its public-key for $(N_{CS}-1)$ cosigners, so it can control $\{\mathbf{pk}_{\pi,2}^{(k)}, \ldots, \mathbf{pk}_{\pi,N_{CS}}^{(k)}\}$ from L^{sh}. Then, $\mathcal{A}$ can also compute its aggregate shared public-key $\mathbf{pk}_\pi^{(k)}$ by calling the $\mathcal{KO}$ oracle. This means that $\mathcal{A}$ can play the role of all cosigners, except for $\mathbf{pk}_{\pi,1}^{(k)\dagger}$.

- $\sigma(\mu) \leftarrow \mathcal{SO}(L, \mu, \mathbf{pk}_\pi^{(k)})$. The *Signing Oracle*, on input a group size w, a set L as in (6.1), the signer's $\mathbf{pk}_\pi^{(k)}$, and a message μ. This oracle returns a valid signature $\sigma(\mu)$.

6.2.2 One-Time Unforgeability

We point out that forging LRS-CS is infeasible assuming that the adversary is able to corrupt $(N_{CS}-1)$ cosigners. Consequently, the LRS-CS scheme is secure against any existentially unforgeable PPT adversary $\mathcal{A}$ under chosen-message attacks if no $\mathcal{A}$ has a non-negligible advantage. One-time unforgeability property is then defined in the following interactive game between the challenger $\mathcal{C}$ and an existential adversary $\mathcal{A}$ who has access to the oracles in Section 6.2.1:

- $\mathcal{C}$ runs $\mathsf{PP} \leftarrow \mathsf{Setup}(1^\lambda)$ and gives it to $\mathcal{A}$.

- $\mathcal{A}$ queries the $\mathcal{KO}$ oracle Q_k times.
- $\mathcal{A}$ queries the $\mathcal{SO}$ oracle Q_s times on input $(\mu, L, \mathbf{pk}_\pi^{(k)})$ for a message μ, $L = \{\mathbf{pk}_1^{(k)}, \ldots, \mathbf{pk}_\pi^{(k)}, \ldots, \mathbf{pk}_w^{(k)}\}$ (with $w-1$ decoyed users in the ring) as in (6.1), which contains the aggregate shared public-key $(\mathbf{pk}_\pi^{(k)})^{sh}$.
- $\mathcal{A}$ finishes this simulation and outputs a forgery $(L^*, \mu^*, \sigma(\mu^*)^*)$ for a new message μ^*, where $L^* = \left\{\mathbf{pk}_i^{*(k)}\right\}_{i\in[w],k\in[N_{in}]}$.

$\mathcal{A}$ wins the game if:

1. $\mathsf{SigVer}(L^*, \mu^*, \sigma(\mu^*)^*)$ outputs Accept.
2. $\mathcal{SO}$ was queried at most once.
3. $(L^*, \mu^*, \sigma(\mu^*)^*)$ is not an output of $\mathcal{SO}$.
4. For all $i \in [w]$, there exists $k \in [N_{in}]$ such that $\mathbf{pk}_i^{*(k)} \in L^*$ was generated by the $\mathcal{KO}$ oracle.
5. Every $\mathbf{pk}_i^{*(k)}$ was used to query $\mathcal{SO}$ as a signing key rather than a decoy at most once.

The advantage of the adversary $\mathcal{A}$ in breaking the LRS-CS scheme is defined as the probability that $\mathcal{A}$ wins the above game. We say that $\mathcal{A}$ breaks this game with $(\tau, Q_s, Q_k, \epsilon_{uf})$ if $\mathcal{A}$ runs in time at most τ and with negligible probability ϵ_{uf} after having made at most Q_s signing queries, Q_k queries to $\mathcal{KO}$, and $(N_{CS} - 1)$ corrupt cosigners. Thus, we denote this property as $\mathbf{Advantage}_{\mathcal{A}}^{\mathsf{ot\text{-}unf}}(\lambda) = \Pr[\mathcal{A} \text{ wins the game}]$.

Definition 6.2 (One-Time Unforgeability)**.** The LRS-CS scheme is said to be one-time unforgeable if no adversary with $(\tau, Q_s, Q_k, \epsilon_{uf})$ is able to break the scheme.

6.2.3 Unconditional Anonymity

This property requires that any powerful adversaries are incapable of saying which member of the ring created a particular signature. We define that it should be infeasible for an adversary $\mathcal{A}$ to distinguish a signer's $\mathbf{pk}_{\pi}^{(k)}$ with non-negligible advantage, even if the adversary has unlimited computing resources and time. This property for LRS-CS schemes is defined in the following game between a simulator $\mathcal{S}$ and an unbounded adversary $\mathcal{A}$.

- $\mathcal{A}$ may query $\mathcal{KO}$ oracle according to any adaptive strategy.
- $\mathcal{A}$ gives $\mathcal{S}$ the $L = \left\{\mathbf{pk}_{i_0}^{(k)}, \mathbf{pk}_{i_1}^{(k)}\right\}_{k\in[N_{in}]}$, where $i_0, i_1 \in [w]$ which is the output of the $\mathcal{KO}$ oracle, and a message μ.
- $\mathcal{S}$ flips a coin $b = \{0, 1\}$, then $\mathcal{S}$ computes the signature $\sigma(\mu)_b = \mathsf{SigGen}(L, \mathbf{sk}_{i_b}^{(k)}, \mu, \mathsf{PP})$. This signature is given to $\mathcal{A}$.
- $\mathcal{A}$ outputs a bit b'.
- The output of this experiment is defined to be 1 if $b = b'$, otherwise 0.

$\mathcal{A}$ wins the game if:

1. $\mathbf{pk}_{i_0}^{(k)}$, $\mathbf{pk}_{i_1}^{(k)}$, and $\mathbf{sk}^{(k)} \notin \{\mathbf{sk}_{i_0}^{(k)}, \mathbf{sk}_{i_1}^{(k)}\}$ cannot be used by $\mathcal{SO}$.
2. $\mathcal{A}$ outputs b' such $b = b'$.

The unconditional anonymity advantage of the LRS-CS scheme is denoted by $\mathbf{Advantage}_{\mathcal{A}}^{\mathsf{Anon}}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right|$.

Definition 6.3 (Unconditional Anonymity)**.** The LRS-CS scheme is called unconditional anonymous if for any unbounded adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{\mathsf{Anon}}(\lambda)$ is negligible.

6.2.4 Linkability

It should be infeasible for an adversary $\mathcal{A}$ to generate (with same $\mathbf{sk}_{\pi}^{(k)}$) two valid LRS-CS signatures which are Unlinked. To describe this, we use the interaction between a simulator $\mathcal{S}$ and $\mathcal{A}$:

- The $\mathcal{A}$ queries the $\mathcal{KO}$ oracle multiple times.
- The $\mathcal{A}$ outputs two signatures $\sigma(\mu)$ and $\sigma(\mu)'$ and two lists L as in (6.1) and $L' = \left\{\mathbf{pk}_i^{\prime(k)}\right\}_{i\in[w],k\in[N_{in}]}$.

$\mathcal{A}$ wins the game if:

1. By calling SigVer on input $\sigma(\mu)$ and $\sigma(\mu)'$, it outputs Accept on both inputs.
2. The $\mathbf{pk}^{(k)}$'s in L and L' are outputs of $\mathcal{KO}$ oracle.
3. Finally, it gets Unlinked, when calling SigLink on input $\sigma(\mu)$ and $\sigma(\mu)'$.

Thus the advantage of the linkability in the LRS-CS scheme is denoted by $\mathbf{Advantage}_{\mathcal{A}}^{\mathsf{Link}}(\lambda) = \Pr[\mathcal{A} \text{ wins the game}]$.

Definition 6.4 (Linkability)**.** The LRS-CS scheme is linkable if for all PPT adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{\mathsf{Link}}(\lambda)$ is negligible in λ.

6.2.5 Non-Slanderability

It should be infeasible for an adversary $\mathcal{A}$ to output linked for two valid LRS-CS signatures which were correctly generated with different $\mathbf{sk}^{(k)}$'s. This means that an adversary can frame an honest user for signing a valid signature so the adversary can produce another valid signature such that the SigLink algorithm outputs Linked. To describe this, we use the interaction between a simulator $\mathcal{S}$ and an adversary $\mathcal{A}$:

- $\mathcal{A}$ queries the $\mathcal{KO}$ to generate $(\mathbf{pk}_i^{(k)}, \mathbf{sk}_i^{(k)})$ with $i \neq \pi$.

- $\mathcal{S}$ queries the $\mathcal{KO}$ to create $(\mathbf{pk}_{\pi}^{(k)}, \mathbf{sk}_{\pi}^{(k)})$ and gives $\mathbf{pk}_{\pi}^{(k)}$ to $\mathcal{A}$.
- $\mathcal{S}$ calls the $\mathcal{SO}$ with $\mathbf{sk}_{\pi}^{(k)}$ and outputs a valid signature $\sigma(\mu)$, which is then given to $\mathcal{A}$.
- $\mathcal{A}$ produces a second signature $\sigma(\mu)'$ by calling the $\mathcal{SO}$ algorithm.

$\mathcal{A}$ wins the game if:

1. The SigVer, on input $\sigma(\mu)$ and $\sigma(\mu)'$, outputs Accept.
2. When calling the SigLink on input $\sigma(\mu)$ and $\sigma(\mu)'$, it outputs linked.

Thus the advantage of the non-slanderability in the LRS-CS scheme is denoted by $\mathbf{Advantage}_{\mathcal{A}}^{\mathsf{NS}}(\lambda) = \Pr[\mathcal{A} \text{ wins the game}]$.

Definition 6.5 (Non-Slanderability)**.** The LRS-CS scheme is *non-slanderable* if for all PPT adversary $\mathcal{A}$, $\mathbf{Advantage}_{\mathcal{A}}^{\mathsf{NS}}(\lambda)$ is negligible in λ.

6.3 A Lattice-based Construction of the LRS-CS

This section describes technically the Lattice-based Linkable Ring Signature with Co-Signing (MIMO.L2RS-CS) scheme. This construction comprises the following algorithms, Setup, KeyGen, SigGen SigVer, and SigLink.

6.3.1 Setup

By receiving the security parameter λ, this Setup defines $\mathbf{A} = [\mathbf{A}'\|\mathbf{I}] \in \mathcal{R}_q^{2\times(m-1)}$ and $\mathbf{H} = [\mathbf{H}'\|\mathbf{I}] \in \mathcal{R}_q^{2\times(m-1)}$ (as Lemma 6.6), where $\mathbf{A}' \hookleftarrow \mathcal{R}_q^{2\times(m-3)}$, $\mathbf{H}' \hookleftarrow \mathcal{R}_q^{2\times(m-3)}$ are chosen uniformly and randomly, and $\mathbf{I}$ denotes the identity. This algorithm outputs the public parameters (PP): $\mathbf{A}$ and $\mathbf{H}$.

Lemma 6.6. *If $q \geq 4n$, then solving the MSIS-HNF problem with a matrix $\boldsymbol{A} = [\boldsymbol{A}'\|\boldsymbol{I}] \in \mathcal{R}_q^{2\times(m-1)}$, in the Hermite Normal Form (HNF), is as hard as solving the $\boldsymbol{MSIS}_{q,m,k,\beta}^{\mathcal{K}}$ problem with $\boldsymbol{A} = [\boldsymbol{A}'\|\boldsymbol{A}''] \in \mathcal{R}_q^{2\times(m-1)}$ uniformly random.*

Proof. Given the MSIS instance $\mathbf{A} = [\mathbf{A}'\|\mathbf{A}''] \in \mathcal{R}_q^{2\times(m-1)}$, if $\mathbf{A}''^{-1}$ exists, then we can reduce it to MSIS-HNF instance, which is of the form $\bar{\mathbf{A}} = \mathbf{A}''^{-1}_{1,1} \times \mathbf{A}$. Therefore, this reduction works with probability equal to the probability that $\mathbf{A}''^{-1}_{1,1}$ exists; then, it remains to show that this probability is non-negligible. We denote the entries of $\mathbf{A}'' = \begin{bmatrix} \mathbf{A}''_{1,1} & \mathbf{A}''_{1,2} \\ \mathbf{A}''_{2,1} & \mathbf{A}''_{2,2} \end{bmatrix} \in \mathcal{R}_q^{2\times 2}$, so the inverse matrix $\mathbf{A}''^{-1} = \frac{1}{\det(\mathbf{A}'')} \cdot \begin{bmatrix} \mathbf{A}''_{2,2} & -\mathbf{A}''_{1,2} \\ -\mathbf{A}''_{2,1} & \mathbf{A}''_{1,1} \end{bmatrix}$, with $\det(\mathbf{A}'')^{-1} = (\mathbf{A}''_{1,1}\mathbf{A}''_{2,2} - \mathbf{A}''_{1,2}\mathbf{A}''_{2,1})^{-1} \in \mathcal{R}_q$ if the inverse exists. Then, we have that $\mathbf{A}''$ is invertible if and only if $\frac{1}{\det(\mathbf{A}'')}$ exists in $\mathcal{R}_q$. Let's define the events $S_0 = \left\{\mathbf{A}''^{-1} \text{does not exist}\right\}$, and $S_1 = \left\{\det(\mathbf{A}'')^{-1} \text{ does not exist in } \mathcal{R}_q\right\}$. We said that $\Pr_{\mathbf{A}''\hookleftarrow\mathcal{R}_q^{2\times2}}\left[S_0\right] = \Pr\left[S_1\right] = P_1 + P_2$, where $P_1 = \Pr\left[S_1 \mid \mathbf{A}''^{-1}_{2,2} \text{ exists}\right] \times \Pr\left[\mathbf{A}''^{-1}_{2,2} \text{ exists}\right]$, and $P_2 = \Pr\left[S_1 \mid \mathbf{A}''^{-1}_{2,2} \text{ does not exist}\right] \times \Pr\left[\mathbf{A}''^{-1}_{2,2} \text{ does not exist}\right]$. We consider that if $\mathbf{A}''_{1,1} \hookleftarrow \mathcal{R}_q$ and $\mathbf{A}''^{-1}_{2,2}$ exists in $\mathcal{R}_q$, then $\mathbf{A}''_{1,1} \times \mathbf{A}''_{2,2}$ is uniform in $\mathcal{R}_q$, i.e. $\forall\bar{\mathbf{A}} \in \mathcal{R}_q$:

$$\Pr_{\mathbf{A}''_{1,1}\hookleftarrow\mathcal{R}_q}\left[\mathbf{A}''_{1,1} \times \mathbf{A}''_{2,2} = \bar{\mathbf{A}}\right] = \Pr_{\mathbf{A}''_{1,1}\hookleftarrow\mathcal{R}_q}\left[\mathbf{A}''_{1,1} = \bar{\mathbf{A}} \times \mathbf{A}''^{-1}_{2,2}\right] = \frac{1}{|\mathcal{R}_q|} \tag{6.2}$$

Let S_2 be the event where a uniform element in $\mathcal{R}_q$ is not invertible in $\mathcal{R}_q$. We observe that $\Pr\left[S_2\right] \leq \frac{n}{q}$ as in [SSTX09]. Then by using (6.2), we have that $P_1 \leq \Pr\left[S_2\right]$ and $P_2 \leq \Pr\left[\mathbf{A}''^{-1}_{2,2} \text{ does not exist in } \mathcal{R}_q\right]$, which is equivalent to $\Pr\left[S_2\right]$, since $\mathbf{A}''_{2,2}$ is uniformly random element in $\mathcal{R}_q$. Therefore, we argue that $\Pr\left[S_1\right] \leq P_1 + P_2 \leq 2 \times \Pr\left[S_2\right] \leq \frac{2n}{q}$. Subsequently, we want to show that $1 - \Pr\left[S_2\right] \geq$ non-negligible and this is implied by $q \geq 4n$. These conditions lead to the probability that $\det(\mathbf{A}'')^{-1}$ exist in $\mathcal{R}_q$ is: $1 - \Pr\left[S_1\right] = \Pr\left[\det(\mathbf{A}'')^{-1} \text{ exist in } \mathcal{R}_q\right] \geq \frac{2n}{q} \geq \frac{1}{2}$. □

Remark 6.7. Setup incorporates a trapdoor in $\mathbf{A}$ or $\mathbf{H}$, in practice Setup would generate $\mathbf{A}$ and $\mathbf{H}$ based on the cryptographic Hash function H_2 evaluated at two distinct and fixed constants.

Definition 6.8 (Function Lift)**.** This function maps $\mathcal{R}_q^2$ to $\mathcal{R}_{2q}$ with respect to a public parameter $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$. Given $\mathbf{a} \in \mathcal{R}_q^2$, we let $\mathsf{Lift}(\mathbf{A}, \mathbf{a}) \triangleq (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$ with $\mathbf{q} = q \cdot (1, 1)^T$.

6.3.2 Key Generation (KeyGen)

The KeyGen (Algorithm 23) is an interactive protocol where N_{CS} cosigners collaborate to produce a pair of keys. We define the public-key to be $\mathbf{a} \triangleq \mathbf{pk}$, and the secret-key as $\mathbf{S} \triangleq \mathbf{sk}$. Once receiving the public parameters PP, each cosigner creates the corresponding secret-key $\bar{\mathbf{S}}_p^T$ and public-key $\bar{\mathbf{a}}_p$ (steps 2-4). After the cosigners interact to verify their public-keys, the aggregate shared public-key $\mathbf{a}^{sh}$ is jointly computed by each cosigner (step 14). The cosigners also calculate their corresponding secret-key $\mathbf{S}_p^T$ using the list of cosigners (step 16). This solid aggregate shared public-key enables this scheme to be secure against *rogue key attacks* [Alo18, GN18, TCZ+20].

Algorithm 23 Key Generation

Input: PP: $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$.
Output: $\left(\mathbf{a}^{sh}, \mathcal{SK}\right)$, with $\mathcal{SK} = \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}$ being the shared public-key and cosigner's secret-key, respectively.
1: **procedure** KeyGen($\mathbf{A}$)
2: Each cosigner $p \in \{1, \ldots, N_{CS}\}$:
3: Selects $\bar{\mathbf{S}}_p^T = (\bar{\mathbf{s}}_{p,1}, \ldots, \bar{\mathbf{s}}_{p,m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\bar{\mathbf{s}}_{p,i} \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
4: Calculates $\bar{\mathbf{a}}_p = (\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2)^T = \mathbf{A} \cdot \bar{\mathbf{S}}_p \bmod q \in \mathcal{R}_q^2$.
5: $\mathbf{o}_p = H_0(\bar{\mathbf{a}}_p)$
6: Broadcasts $\mathbf{o}_p$ to other cosigners $p' \in [N_{CS}]$
7: Receives $\mathbf{o}_{p'}$ with $p' \neq p$, then "p" sends $\bar{\mathbf{a}}_p$ to the cosigners
8: Receives $\bar{\mathbf{a}}_{p'}$ with $p' \neq p$
9: Each cosigner verifies:
10: **for** $(1 \leq p' \leq N_{CS})$ **do**
11: **if** $\mathbf{o}_{p'} = H_0(\bar{\mathbf{a}}_{p'})$ **then** Accept
12: **else** Abort protocol
13: Each cosigner computes the shared public-key as:
14: $\mathbf{a}^{sh} = \sum_{p'}^{N_{CS}} H_2(\bar{\mathbf{a}}_{p'}, L^{sh}) \cdot \bar{\mathbf{a}}_{p'}$ with $L^{sh} = \left\{\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_{N_{CS}}\right\}$
15: Each cosigner calculates its corresponding secret-key as:
16: $\mathbf{S}_p^T = H_2(\bar{\mathbf{a}}_p, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$
17: **return** $\left(\mathbf{a}^{sh}, \mathcal{SK}\right)$, without loss of generality, each cosigner only outputs and holds its corresponding secret-key $\mathbf{S}_{p'}^T$.

6.3.3 Signature Generation (SigGen)

The SigGen (Algorithm 24) is an interactive protocol, among N_{CS} cosigners, which outputs the signature $\sigma(\mu)$. This protocol receives a message μ, the public parameters, the list L that contains the public-keys of w users in the ring, and a set with the consigners' secret keys, $\mathcal{SK} = \left\{\mathbf{S}_{\pi,1}^{(k)}, \ldots, \mathbf{S}_{\pi,p}^{(k)}, \ldots, , \mathbf{S}_{\pi,N_{CS}}^{(k)}\right\}_{k\in[N_{in}]}$ with

N_{in} number of input wallets. The SigGen extends the L2RS [ATSS+18] which follows the Fiat-Shamir transformation and uses the rejection sampling technique (step 40) that hides the secret key from the signature.

Remark 6.9. The number if iterations M of the SigGen until the rejection sampling test accepts is $N = N_{in} \times N_{CS}$ cosigners with inputs $T = M^N$ so that $M = O(1)$ (is constant), the expected time is small with N small, which is the main application in practice.

6.3.4 Signature Verification (SigVer)

The SigVer (Algorithm 25) verifies the generated signature by receiving $(\mu, L, \sigma(\mu), \mathsf{PP})$ and outputing Accept or Reject. Additionally, Theorem 6.10 shows the bound of β_v that is used in this algorithm.

Theorem 6.10. *Let* $\beta_v = \eta\sigma\sqrt{nm}$, $q/4 > \sigma\sqrt{2(\lambda+1)\ln 2 + 2\ln(nm)}$, *and* $\sigma(\mu) = \left(\boldsymbol{c}_1, \left\{\boldsymbol{t}_1^{(k)}, \ldots, \boldsymbol{t}_w^{(k)}\right\}_{k\in[N_{in}]}, \left\{\boldsymbol{h}_\pi^{(k)}\right\}_{k\in[N_{in}]}\right)$ *be generated based on Algorithm 24. Then the output of Algorithm 25 on input* $\sigma(\mu)$ *is accepted with probability* $1 - 2^{-\lambda}$.

Proof. For a desired expected rejection and repetition M, if we take the definition of α where $M = e^{\frac{1}{2\alpha^2}}$, then $\mathbf{t}_\pi^{(k)}$ will be indistinguishable from D_σ if $\sigma \geq \alpha \cdot \|\mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi\|$ [Section 3.2 in [DDLL13]]. We also use [lemma 4.4, parts 1 and 3, in [Lyu12]]. The part 3 of this lemma shows that the bound on Euclidean norm $\beta_v = \eta\sigma\sqrt{nm}$, for a given $\eta > 1$, has a probability $\Pr\left[\|\mathbf{t}_i^{(k)}\|_2 > \eta\sigma\sqrt{nm}\right] \geq 1 - 2^{-\lambda}$. In addition, the bound on infinity norm ($\|\mathbf{t}_i\|_\infty < q/4$) is analysed in part 1 of this lemma where its union bound is also considered. It turns out that η is required such $q/4 > \eta\sigma > \sigma\sqrt{2(\lambda+1)\ln 2 + 2\ln(nm)}$, except with probability $2^{-\lambda}$. □

6.3.5 Correctness of SigGen

We show in the following proof that valid signatures are signed by honest signers, such that $\sigma(\mu) = \left(\mathbf{c}_1, \left\{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\right\}_{k\in[N_{in}]}, \left\{\mathbf{h}_\pi^{(k)}\right\}_{k\in[N_{in}]}\right)$ is the output of the

Algorithm 24 Signature Generation

Input: $\mathcal{SK} = \left\{\mathbf{S}_{\pi,p'}^{(k)}\right\}_{p' \in [N_{CS}], k \in [N_{in}]}, \mu, L = \left\{\mathbf{a}_i^{(k)}\right\}_{i \in [w], k \in [N_{in}]}$ as in (6.1), and PP.

Output: $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$

procedure SigGen($\mathcal{SK}, \mu, L,$ PP)
  for $(1 \le k \le N_{in})$ **do**
    Each cosigner "π, p":
    Computes the linking tag $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H} \cdot \mathbf{S}_{\pi,p}^{(k)} \in \mathcal{R}_q^2$.
    $\bar{\mathbf{o}}_{\pi,p}^{(k)} = H_0(\mathbf{h}_{\pi,p}^{(k)})$
    Broadcasts $\bar{\mathbf{o}}_{\pi,p}^{(k)}$ to other cosigners $p' \in [N_{CS}]$
    Receives $\bar{\mathbf{o}}_{\pi,p'}^{(k)}$ with $p' \neq p$, then "π, p" securely sends $\mathbf{h}_{\pi,p}^{(k)}$ to the cosigners
    Receives $\mathbf{h}_{\pi,p'}^{(k)}$ with $p' \neq p$
    "π, p" verifies:
    for $(1 \le p' \le N_{CS})$ **do**
      if $\bar{\mathbf{o}}_{\pi,p'}^{(k)} = H_0\left(\mathbf{h}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    Computes the shared linking tag $\mathbf{h}_\pi^{(k)} = \sum_{p'}^{N_{CS}} \mathbf{h}_{\pi,p'}^{(k)}$
    Calls $\mathsf{Lift}(\mathbf{H}, \mathbf{h}_\pi^{(k)})$ to obtain $\mathbf{H}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    Calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_\pi^{(k)})$ to obtain $\mathbf{A}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    Chooses $\mathbf{u}_{\pi,p}^{(k)} = (u_{\pi,p,1}, \ldots, u_{\pi,p,m})^T$, where $u_{\pi,p,i} \hookleftarrow D_\sigma^n$, for $1 \le i \le m$.
    Computes $\mathbf{r}_{\pi,p}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)} = \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$
    $\mathbf{o}_{\pi,p}^{(k)} = H_0(\mathbf{r}_{\pi,p}^{(k)}, \mathbf{z}_{\pi,p}^{(k)})$
    Broadcasts $\mathbf{o}_{\pi,p}^{(k)}$ to other cosigners $p' \in [N_{CS}]$
    Receives $\mathbf{o}_{\pi,p'}^{(k)}$ with $p' \neq p$, then "π, p" securely sends $\mathbf{r}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)}$ to the cosigners
    Receives $\mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi,p'}^{(k)}$ with $p' \neq p$
    "π, p" verifies:
    for $(1 \le p' \le N_{CS})$ **do**
      if $\mathbf{o}_{\pi,p'}^{(k)} = H_0\left(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    "π, p" computes $\mathbf{r}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{z}_{\pi,p'}^{(k)}$
  "π, p" performs $\forall k \in [N_{in}]$, $\mathbf{c}_{\pi+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{r}_\pi^{(k)}, \mathbf{z}_\pi^{(k)}\right)$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \le k \le N_{in})$ **do**
      Each cosigner "π, p":
      Selects $\mathbf{t}_{i,p}^{(k)} = (t_{i,p,1}, \ldots, t_{i,p,m})^T$, where $t_{i,p,j} \hookleftarrow D_\sigma^n$, for $1 \le j \le m$.
      Sends $\mathbf{t}_{i,p}^{(k)}$ to other cosigners $p' \in [N_{CS}]$ securely
      Receives $\mathbf{t}_{i,p'}^{(k)}$ with $p' \neq p$ from other cosigners
      Computes $\mathbf{t}_i^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{i,p'}^{(k)}$
      "π, p" calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_i^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    Compute $\forall k \in [N_{in}]$ $\mathbf{c}_{i+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \left\{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}, \left\{\mathbf{H}_{2q,\pi} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}\right)$.
  for $(1 \le k \le N_{in})$ **do**
    Choose $b^{(k)} \hookleftarrow \{0, 1\}$.
    "π, p" computes $\mathbf{t}_{\pi,p}^{(k)} = \mathbf{u}_{\pi,p}^{(k)} + \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \cdot (-1)^{b^{(k)}}$, where $\mathbf{S}_{2q,\pi,p}^{(k)} = [(\mathbf{S}_{\pi,p}^{(k)})^T, 1]^T$.
    Continue with prob. $\left(M \exp\left(-\frac{\|\mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{t}_{\pi,p}^{(k)}, \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
    "π, p" broadcasts $\mathbf{t}_{\pi,p}^{(k)}$ to other cosigners
    "π, p" receives $\mathbf{t}_{\pi,p'}^{(k)}$ with $p' \neq p$ and computes $\mathbf{t}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)}$
  return $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$.

Algorithm 25 Signature Verification

Input: $\sigma(\mu), \mu$,PP, and $L = \left\{\mathbf{a}_i^{(k)}\right\}_{i\in[w],k\in[N_{in}]}$.
Output: Accept or Reject
1: **procedure** SIGVER($\sigma(\mu), \mu, L, \mathsf{PP}$)
2: Computes $\mathbf{H}_{2q}^{(k)} = (2\cdot\mathbf{H}, -2\cdot\mathbf{h}^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$
3: **for** $(i = 1, \ldots, w)$ **do**
4: **for** $(1 \le k \le N_{in})$ **do**
5: "π, p" calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_{i,}^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
6: Compute $\forall k \in [N_{in}],\ \mathbf{c}_{i+1} = H_1\Big(L, \mathbf{H}_{2q}^{(k)}, \mu, \Big\{\mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\Big\}, \Big\{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\Big\}\Big)$
7: Check $\|\mathbf{t}_{i,p'}^{(k)}\|_2 \le \beta_v$ (see Theorem 6.10)
8: Check $\|\mathbf{t}_{i,p'}^{(k)}\|_\infty < q/4$
9: **if** $\mathbf{c}_1 = H_1\Big(L, \mathbf{H}_{2q}^{(k)}, \mu, \Big\{\mathbf{A}_{2q,w}^{(k)}\cdot\mathbf{t}_w^{(k)} + \mathbf{q}\cdot\mathbf{c}_w\Big\}, \Big\{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_w^{(k)} + \mathbf{q}\cdot\mathbf{c}_w\Big\}\Big)$ **then** Accept
10: **else** Reject
11: **return** Accept or Reject

SigGen algorithm on input $(\mu, L, \mathbf{S}_{\pi,p}^{(k)}, \mathsf{PP})$. Then, on input $(\mu, L, \sigma(\mu), \mathsf{PP})$, the SigVer algorithm outputs Accept with overwhelming probability.

We demonstrate that when SigVer (step 9) computes $\forall k, \in [N_{in}]$, $H_1\Big(L, \mathbf{H}_{2q}^{(k)}, \mu, \Big\{\mathbf{A}_{2q,w}^{(k)}\cdot\mathbf{t}_w^{(k)} + \mathbf{q}\cdot\mathbf{c}_w\Big\}, \Big\{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_w^{(k)} + \mathbf{q}\cdot\mathbf{c}_w\Big\}\Big)$, this result should be equal to $\mathbf{c}_1$. The SigVer also verifies $\forall k, w \in [N_{in}], [w]$ that $H_1\Big(L, \mathbf{H}_{2q}^{(k)}, \mu, \Big\{\mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\Big\}, \Big\{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\Big\}\Big) = \mathbf{c}_{i+1}$. This evaluation considers two scenarios:

- When $i \neq \pi$, $\forall k \in [N_{in}]$, SigGen evaluates $\mathbf{c}_{i+1} = H_1\Big(L, \mathbf{H}_{2q}^{(k)}, \mu, \Big\{\mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\Big\}, \Big\{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\Big\}\Big)$, while SigVer computes $\mathbf{c}_{i+1} = H_1\Big(L, \mathbf{H}_{2q}^{(k)}, \mu, \Big\{\mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\Big\}, \Big\{\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\Big\}\Big)$. These are equal since $\mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)}\cdot\mathbf{q}\cdot\mathbf{c}_i$ (in SigGen) $= \mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)}\cdot\mathbf{q}\cdot\mathbf{c}_i$ (in SigVer); and $\mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i$ (in SigGen) $= \mathbf{H}_{2q}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i$ (in SigVer).

- When $i = \pi$, $\forall k \in [N_{in}]$, SigGen checks $\mathbf{c}_{\pi+1} = H_1\Big(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{r}_\pi^{(k)}, \mathbf{z}_\pi^{(k)}\Big)$, whereas SigVer calculates $\mathbf{c}_{\pi+1} = H_1\Big(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{A}_{2q,\pi}^{(k)}\cdot\mathbf{t}_\pi^{(k)}\cdot\mathbf{q}\cdot\mathbf{c}_\pi, \mathbf{H}_{2q,\pi}^{(k)}\cdot\mathbf{t}_\pi^{(k)} + \mathbf{q}\cdot\mathbf{c}_\pi\Big)$. In this case, we need to show that $\mathbf{c}_{\pi+1}$ (in SigGen) $= \mathbf{c}_{\pi+1}$ (in SigVer). In doing so, we evaluate two equalities one related to the public key $\mathbf{r}_\pi^{(k)} = \mathbf{A}_{2q,\pi}^{(k)}\cdot\mathbf{t}_\pi^{(k)} + \mathbf{q}\cdot\mathbf{c}_\pi$, and the other associated to the linking tag $\mathbf{z}_{\pi,p'}^{(k)} = \mathbf{H}_{2q,\pi,p'}^{(k)}\cdot\mathbf{t}_{\pi,p'}^{(k)} + \mathbf{q}\cdot\mathbf{c}_\pi$. These equalities are analysed as follows:

1. The first equality is compared with $\forall(k, p') \in [N_{in}] \times [N_{CS}]$:

$$\mathbf{r}_{\pi}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{t}_{\pi}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$\sum_{p'=1}^{N_{CS}} \mathbf{r}_{\pi,p'}^{(k)} = \left\{ \mathbf{A}_{2q,\pi}^{(k)} \cdot \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)} \right\} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$\sum_{p'=1}^{N_{CS}} \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)} = \left\{ \mathbf{A}_{2q,\pi}^{(k)} \cdot \sum_{p'=1}^{N_{CS}} \left(\mathbf{u}_{\pi,p'}^{(k)} + \mathbf{S}_{2q,\pi,p'}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right) \right\} +$$

$$\mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$\sum_{p'=1}^{N_{CS}} \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)} + \sum_{p'=1}^{N_{CS}} \left\{ \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{S}_{2q,\pi,p'}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right\} +$$

$$\mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \sum_{p'=1}^{N_{CS}} \left\{ \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{S}_{2q,\pi,p'}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right\} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \sum_{p'=1}^{N_{CS}} \left\{ (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_{\pi}^{(k)} + \mathbf{q}) \cdot [\mathbf{S}_{\pi,p'}^{(k)}, 1]^T \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right\} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \left\{ 2 \cdot \mathbf{A}, -2 \cdot \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{a}}_{\pi,p'}^{(k)} + \mathbf{q} \right\} \cdot$$

$$\left\{ \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot [\bar{\mathbf{S}}_{\pi,p'}^{(k)}, 1] \right\}^T \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \left\{ 2 \cdot \mathbf{A} \cdot \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{S}}_{\pi,p'}^{(k)}, -2 \cdot \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{a}}_{\pi,p'}^{(k)} + \mathbf{q} \right\} \cdot$$

$$\mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \left\{ 2 \cdot \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{a}}_{\pi,p'}^{(k)}, -2 \cdot \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{a}}_{\pi,p'}^{(k)} + \mathbf{q} \right\} \cdot$$

$$\mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \mathbf{q} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$-\mathbf{q} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} = \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

We distinguish two cases for b:

- When $b = 0$, we verify that $-\mathbf{q} \cdot \mathbf{c}_{\pi} = \mathbf{q} \cdot \mathbf{c}_{\pi} \bmod 2q$.

– When $b = 1$, we have $\mathbf{q} \cdot \mathbf{c}_\pi = \mathbf{q} \cdot \mathbf{c}_\pi \bmod 2q$.

2. Consequently, the second equality is also examined with $\forall(k, p') \in [N_{in}] \times [N_{CS}]$:

$$\mathbf{z}_{\pi}^{(k)} = \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{t}_{\pi}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$\sum_{p'=1}^{N_{CS}} \mathbf{z}_{\pi,p'}^{(k)} = \left\{ \mathbf{H}_{2q,\pi}^{(k)} \cdot \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)} \right\} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$\sum_{p'=1}^{N_{CS}} \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)} = \left\{ \mathbf{H}_{2q,\pi}^{(k)} \cdot \sum_{p'=1}^{N_{CS}} \left(\mathbf{u}_{\pi,p'}^{(k)} + \mathbf{S}_{2q,\pi,p'}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right) \right\} +$$

$$\mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$\sum_{p'=1}^{N_{CS}} \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)} + \sum_{p'=1}^{N_{CS}} \left\{ \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{S}_{2q,\pi,p'}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right\} +$$

$$\mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \sum_{p'=1}^{N_{CS}} \left\{ \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{S}_{2q,\pi,p'}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right\} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \sum_{p'=1}^{N_{CS}} \left\{ (2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\pi}^{(k)} + \mathbf{q}) \cdot [\mathbf{S}_{\pi,p'}^{(k)}, 1]^T \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right\} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \sum_{p'=1}^{N_{CS}} \left\{ (2 \cdot \mathbf{H}, -2 \cdot \sum_{p'}^{N_{CS}} \mathbf{h}'^{(k)}_{\pi,p'} + \mathbf{q}) \cdot [\mathbf{S}_{\pi,p'}^{(k)}, 1]^T \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} \right\} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \left\{ 2 \cdot \mathbf{H}, -2 \cdot \sum_{p'}^{N_{CS}} \mathbf{H} \cdot \mathbf{S}_{\pi,p'}^{(k)} + \mathbf{q} \right\} \cdot \left\{ \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot [\bar{\mathbf{S}}_{\pi,p'}^{(k)}, 1] \right\}^T \cdot$$

$$\mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \left\{ 2 \cdot \mathbf{H}, -2 \cdot \mathbf{H} \cdot \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{S}}_{\pi,p'}^{(k),T} + \mathbf{q} \right\} \cdot$$

$$\left\{ \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot [\bar{\mathbf{S}}_{\pi,p'}^{(k)}, 1] \right\}^T \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \left\{ 2 \cdot \mathbf{H} \cdot \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{S}}_{\pi,p'}^{(k)}, -2 \cdot \mathbf{H} \cdot \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{S}}_{\pi,p'}^{(k),T} + \mathbf{q} \right\} \cdot$$

$$\mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$0 = \mathbf{q} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} + \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

$$-\mathbf{q} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}} = \mathbf{q} \cdot \mathbf{c}_{\pi} \iff$$

We distinguish between two cases:

- When $b = 0$, it is verified that $-\mathbf{q} \cdot \mathbf{c}_\pi = \mathbf{q} \cdot \mathbf{c}_\pi \bmod 2q$.
- When $b = 1$, we have $\mathbf{q} \cdot \mathbf{c}_\pi = \mathbf{q} \cdot \mathbf{c}_\pi \bmod 2q$.

6.3.6 Signature Linkability (SigLink)

The SigLink (Algorithm 26) checks whether two signatures were correctly produced by the same signer, but it does not reveal the identify of such signer. The correctness proof of this algorithm is described in Appendix 6.3.7.

Algorithm 26 Signature Linkability

Input: $\sigma(\mu)_1$ and $\sigma(\mu)_2$
Output: Linked or Unlinked
1: **procedure** SigLink($\sigma(\mu)_1, \sigma(\mu)_2$)
2: **if** $\Big($SigVer$(\sigma(\mu)_1, *) =$ Accept **and** SigVer$(\sigma(\mu)_2, *)) =$ Accept$\Big)$ **then** Continue [
3: **else if** $\mathbf{h}_{\mu_1}^{(k)} = \mathbf{h}_{\mu_2}^{(k)}$ **then** Linked
4: **else** Unlinked]
5: **return** Linked or Unlinked

6.3.7 Correctness of SigLink

We show that an honest user π who signs two messages μ_1 and μ_2 in the MIMO.L2RS-CS scheme with the list of public-keys L, obtains a Linked output from SigLink algorithm with overwhelming probability. As shown in Algorithm 26, two signatures $\sigma(\mu)_1$ and $\sigma(\mu)_2$ were created, and then successfully verified by SigVer. Therefore, the linkability tags $\mathbf{h}_{\mu_1}^{(k)}$ and $\mathbf{h}_{\mu_2}^{(k)}$ $\forall k \in [N_{in}]$ must be equal. To prove this, we show that:

$$\mathbf{H}_{2q,\mu_1}^{(k)} = \left(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\mu_1}^{(k)} + \mathbf{q}\right) \in \mathcal{R}_{2q}^{2\times m}, \text{where}$$
$$\mathbf{H} = \mathsf{PP} \text{ and } \mathbf{h}_{\mu_1}^{(k)} = \left(\mathbf{H} \cdot \mathbf{S}_\pi^{(k)} + \mathbf{q}\right) \in \mathcal{R}_q^2$$
$$\mathbf{H}_{2q,\mu_2}^{(k)} = \left(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\mu_2}^{(k)} + \mathbf{q}\right) \in \mathcal{R}_{2q}^{2\times m}, \text{where}$$
$$\mathbf{H} = \mathsf{PP} \text{ and } \mathbf{h}_{\mu_2}^{(k)} = \left(\mathbf{H} \cdot \mathbf{S}_\pi^{(k)} + \mathbf{q}\right) \in \mathcal{R}_q^2$$

The first parts of the linkability tag in both MIMO.L2RS-CS signatures have same equality with following probability:

$$\Pr\left[2 \cdot \mathbf{H} = 2 \cdot \mathbf{H}\right] = 1.$$

Ultimately, the second part uses the honest user's secret-key $\mathbf{S}_{\pi}^{(k)}$ is used, so we conclude that:

$$\Pr\left[-2 \cdot \mathbf{h}_{\mu_1}^{(k)} + \mathbf{q} + 2 \cdot \mathbf{h}_{\mu_2}^{(k)} - \mathbf{q} = 0\right] = 1.$$

6.4 Security Analysis

This section presents the results of our security evaluation. It demonstrates that the L2RS-CS is computationally secure in terms of *unforgeability*, *linkability* and *non-slanderability* from the Module-SIS lattice assumption, and it is unconditionally secure for *anonymity* under the Leftover Hash Lemma (LHL).

Theorem 6.11 (One-Time Unforgeability)**.** *If there is a PPT algorithm against one-time unforgeability of L2RS-CS that makes Q_{uf} queries to the random oracles $H_0, \mathcal{SO}$ and $\mathcal{KO}$, with non-negligigle probability δ; then, there exist a PPT algorithm that can extract a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem, where $\beta = 2\beta_v$ and with non-negligible probability $\left(\delta - \epsilon_{uf} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \cdot \left(\frac{\delta - \epsilon_{uf} - \frac{1}{|\mathcal{S}_{n,\kappa}|}}{Q_s + Q_1} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right)$. The ϵ_{uf} is neg(n) if the following conditions hold:*

1. $\frac{2 \cdot N_{in} \cdot N_{CS}(2 \cdot Q_{uf}+1)^2}{2^{n+1}} \leq neg(n)$, *with* $Q_{uf} = max(Q_0, Q_s, Q_k)$,
2. $\frac{1}{|\mathcal{S}_{n,\kappa}|} \leq neg(n)$,
3. $\frac{1}{\sqrt{k}} \cdot q^{1/k} \leq neg(n)$.

Proof. The MIMO.L2RS-CS scheme relies on the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem to be secure against any existential forger. This means that a forgery algorithm succeeds with a negligible probability. We conclude that under this probability, the attacker

will also find a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem. We consider the sequence of games in this proof where a PPT $\mathcal{A}$ is the adversary against the MIMO.L2RS-CS scheme.

Game 0 - Real Game: This is defined as the original attack game where the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ interact to produce a forgery. We know that $\mathbf{a} \triangleq \mathbf{pk}$ and $\mathbf{S} \triangleq \mathbf{sk}$; then, the real Game starts with the challenger $\mathcal{C}$ who calls $\mathsf{PP} \leftarrow \mathsf{Setup}(1^{\lambda})$ and gives PP to $\mathcal{A}$. $\mathcal{C}$ runs KeyGen (Algorithm 23), where $\mathcal{C}$ starts computing $\bar{\mathbf{a}}_1^{\dagger}$ and $\bar{\mathbf{S}}_1^{\dagger}$. When the adversary $\mathcal{A}$ sends $\bar{\mathbf{a}}_{p'}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\bar{\mathbf{a}}_1^{\dagger}$ to $\mathcal{A}$. After that, $\mathcal{C}$ performs the aggregate shared public-key as $\mathbf{a}_{\pi}^{sh} = \sum_{p'}^{N_{CS}} H_2(\bar{\mathbf{a}}_{p'}, L^{sh}) \cdot \bar{\mathbf{a}}_{p'}$ with $L^{sh} = \{\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_{N_{CS}}\}$. $\mathcal{C}$ outputs $(\mathbf{a}_{\pi}^{sh}, \mathbf{S}_1^T)$ with its secret-key computed as $\mathbf{S}_1^T = H_2(\bar{\mathbf{a}}_1, L^{sh}) \cdot \bar{\mathbf{S}}_1^T$. $\mathcal{A}$ queries the $\mathcal{KO}$ oracle Q_k times.

The challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ interact to generate a signature $\sigma(\mu)^t$ on (L^t, μ^t) with $L^t = \left\{\mathbf{a}_1^{(k)}, \ldots, (\mathbf{a}_{\pi}^{(k)})^{sh}, \ldots, \mathbf{a}_w^{(k)}\right\}$ $\forall k \in [N_{in}]$ and for any $t \in [1, Q_s]$. We assume that $(\mathbf{a}_{\pi}^{(k)})^{sh}$ was generated following the KeyGen algorithm and from which the challenger $\mathcal{C}$'s public-key $(\bar{\mathbf{a}}_1^{\dagger})$ occurs once. Whenever $\mathcal{A}$ sends interactive queries Q_s with (L^t, μ^t) to $\mathcal{C}$ who behaves as in Algorithm 27 and ultimately returns $\sigma(\mu)^t$ to $\mathcal{A}$.

The adversary $\mathcal{A}$ completes the simulation and outputs a forgery $(L^*, \mu^*, \sigma(\mu)^*)$. $\mathcal{A}$ wins the game if this forgery satisfies the following conditions:

1. $\mathsf{SigVer}(L^*, \mu^*, \sigma(\mu^*)^*)$ outputs Accept.
2. $\mathcal{SO}$ was queried at most once.
3. $(L^*, \mu^*, \sigma(\mu^*)^*)$ is not an output of $\mathcal{SO}$.
4. For all $i \in [w]$, there exists $k \in [N_{in}]$ such that $\mathbf{pk}_i^{*(k)} \in L^*$ was generated by the $\mathcal{KO}$ oracle.
5. Every $\mathbf{pk}_i^{*(k)}$ was used to query $\mathcal{SO}$ as a signing key rather than a decoy at most once.

Algorithm 27 SigGen - Game 0

Input: $\mathcal{SK} = \left\{\mathbf{S}_{\pi,p'}^{(k)}\right\}_{p'\in[N_{CS}],k\in[N_{in}]}, \mu, L = \left\{\mathbf{a}_i^{(k)}\right\}_{i\in[w],k\in[N_{in}]}$ as in (6.1), and PP.

Output: $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k\in[N_{in}]}\right)$

procedure SigGen($\mathcal{SK}, \mu, L,$ PP)
  for $(1 \le k \le N_{in})$ **do**
    The challenger $\mathcal{C}$ computes the linking tag $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H}\cdot\mathbf{S}_{\pi,p}^{(k)} \in \mathcal{R}_q^2$.
    $\mathcal{C}$ sets $\bar{\mathbf{o}}_{\pi,p}^{(k)} = H_0\left(\mathbf{h}_{\pi,p}^{(k)}\right)$
    When $\mathcal{A}$ sends $\bar{\mathbf{o}}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\bar{\mathbf{o}}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{h}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{h}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \le p' \le N_{CS})$ **do**
      if $\bar{\mathbf{o}}_{\pi,p'}^{(k)} = H_0\left(\mathbf{h}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes the shared linking tag $\mathbf{h}_\pi^{(k)} = \sum_{p'}^{N_{CS}} \mathbf{h}_{\pi,p'}^{(k)}$
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{H}, \mathbf{h}_\pi^{(k)})$ to obtain $\mathbf{H}_{2q,\pi}^{(k)} = (2\cdot\mathbf{H}, -2\cdot\mathbf{h}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_\pi^{(k)})$ to obtain $\mathbf{A}_{2q,\pi}^{(k)} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
    $\mathcal{C}$ chooses $\mathbf{u}_{\pi,p}^{(k)} = (u_{\pi,p,1}, \ldots, u_{\pi,p,m})^T$, where $u_{\pi,p,i} \hookleftarrow D_\sigma^n$, for $1 \le i \le m$.
    $\mathcal{C}$ computes $\mathbf{r}_{\pi,p}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)}\cdot\mathbf{u}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)} = \mathbf{H}_{2q,\pi}^{(k)}\cdot\mathbf{u}_{\pi,p}^{(k)}$
    $\mathcal{C}$ sets $\mathbf{o}_{\pi,p}^{(k)} = H_0\left(\mathbf{r}_{\pi,p}^{(k)}, \mathbf{z}_{\pi,p}^{(k)}\right)$
    When $\mathcal{A}$ sends $\mathbf{o}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{o}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{r}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \le p' \le N_{CS})$ **do**
      if $\mathbf{o}_{\pi,p'}^{(k)} = H_0\left(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes $\mathbf{r}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{z}_{\pi,p'}^{(k)}$
  $\mathcal{C}$ performs $\forall k \in [N_{in}]$, $\mathbf{c}_{\pi+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{r}_\pi^{(k)}, \mathbf{z}_\pi^{(k)}\right)$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \le k \le N_{in})$ **do**
      $\mathcal{C}$ selects $\mathbf{t}_{i,p}^{(k)} = (t_{i,p,1}, \ldots, t_{i,p,m})^T$, where $t_{i,p,j} \hookleftarrow D_\sigma^n$, for $1 \le j \le m$.
      When $\mathcal{A}$ sends $\mathbf{t}_{i,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{i,p}^{(k)}$ to $\mathcal{A}$.
      $\mathcal{C}$ computes $\mathbf{t}_i^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{i,p'}^{(k)}$
      $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_i^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
    $\mathcal{C}$ runs $\forall k \in [N_{in}]$ $\mathbf{c}_{i+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \left\{\mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\right\}, \left\{\mathbf{H}_{2q,\pi}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\right\}\right)$.
  for $(1 \le k \le N_{in})$ **do**
    $\mathcal{C}$ chooses $b^{(k)} \hookleftarrow \{0,1\}$.
    $\mathcal{C}$ computes $\mathbf{t}_{\pi,p}^{(k)} = \mathbf{u}_{\pi,p}^{(k)} + \mathbf{S}_{2q,\pi,p}^{(k)}\cdot\mathbf{c}_\pi\cdot(-1)^{b^{(k)}}$, where $\mathbf{S}_{2q,\pi,p}^{(k)} = [(\mathbf{S}_{\pi,p}^{(k)})^T, 1]^T$.
    Continue with prob. $\left(M\exp\left(-\frac{\|\mathbf{S}_{2q,\pi,p}^{(k)}\cdot\mathbf{c}_\pi\|^2}{2\sigma^2}\right)\cosh\left(\frac{\langle\mathbf{t}_{\pi,p}^{(k)}, \mathbf{S}_{2q,\pi,p}^{(k)}\cdot\mathbf{c}_\pi\rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
    When $\mathcal{A}$ sends $\mathbf{t}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{\pi,p}^{(k)}$ to $\mathcal{A}$
    $\mathcal{C}$ computes $\mathbf{t}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)}$
  return $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k\in[N_{in}]}\right)$.

If we define the event S_0 where the adversary $\mathcal{A}$ wins **Game 0**, then we argued that $\mathcal{A}$'s advantage is negligible:

$$\Pr[S_0] \leq \epsilon_0. \tag{6.3}$$

Game 1: This game is similar to **Game 0**, but this time the challenger $\mathcal{C}$ behaves different in the random oracle H_0 as illustrated in Algorithm 28 (step 15). On a y-th query $\mathbf{r}_y^{(k)}$ and $\mathbf{z}_y^{(k)}$ from the adversary $\mathcal{A}$, then $\mathcal{C}$ proceeds as follows:

1. $\mathcal{C}$ returns $H_0\big(\mathbf{r}_y^{(k)}, \mathbf{z}_y^{(k)}\big)$ if this is already defined.
2. $\mathcal{C}$ chooses at random $\mathbf{o}_y^{(k)} \hookleftarrow \mathcal{S}_{n,\kappa}$, otherwise.
3. $\mathcal{C}$ verifies if there exists $p' \in [1, y-1]$ such that $\mathbf{o}_y^{(k)} = H_0\big(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\big)$ for previous queries of $\mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi,p'}^{(k)}$. In the case where p' exists, the game is aborted, otherwise:
4. $\mathcal{C}$ sets $\mathbf{o}_y^{(k)} = H_0\big(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\big)$ and returns $\mathbf{o}_y^{(k)}$ to $\mathcal{A}$.

The difference between the **Game 0** and **Game 1** is that the challenger $\mathcal{C}$ aborts when he tries to set a same hash value H_0 for two different inputs. This game evaluates the probability that $\mathcal{C}$ aborts the game under this situation. The total number of queries Q_0 to H_0 oracle is at most $Q_0 + Q_s$. Then the probability that $\mathcal{C}$ aborts **Game 1** is

$$\sum_{y=1}^{Q_0+Q_s} \Pr\left[\left(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\right) \in \left\{\mathbf{r}_{y'}^{(k)}, \mathbf{z}_{y'}^{(k)}\right\}_{y'<y}\right] \leq$$

$$\sum_{y=1}^{Q_0+Q_s} \sum_{y'=1}^{y-1} \Pr_{\mathbf{u}_{\pi,p'}^{(k)} \hookleftarrow D_\sigma^n}\left[\mathbf{r}_{\pi,p'}^{(k)} = \mathbf{r}_{y'}^{(k)}\right] \leq$$

$$\sum_{y=1}^{Q_0+Q_s} \frac{y-1}{2^n} \leq \frac{(Q_0+Q_s)(Q_0+Q_s+1)}{2^n}$$

Let S_1 be the event where the $\mathcal{A}$ wins this Game with negligible probability $\frac{(Q_0+Q_s)(Q_0+Q_s+1)}{2^n} \leq \epsilon_1$. Then we argue that:

$$|\Pr[S_0] - \Pr[S_1]| \leq \epsilon_1. \tag{6.4}$$

Algorithm 28 SigGen - Game 1

Input: $\mathcal{SK} = \left\{\mathbf{S}_{\pi,p'}^{(k)}\right\}_{p' \in [N_{CS}], k \in [N_{in}]}, \mu$, $L = \left\{\mathbf{a}_i^{(k)}\right\}_{i \in [w], k \in [N_{in}]}$ as in (6.1), and PP.

Output: $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$

procedure SigGen($\mathcal{SK}, \mu, L, \mathsf{PP}$)
  for $(1 \leq k \leq N_{in})$ **do**
    The challenger $\mathcal{C}$ computes the linking tag $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H} \cdot \mathbf{S}_{\pi,p}^{(k)} \in \mathcal{R}_q^2$.
    $\mathcal{C}$ sets $\bar{\mathbf{o}}_{\pi,p}^{(k)} = H_0\left(\mathbf{h}_{\pi,p}^{(k)}\right)$
    When $\mathcal{A}$ sends $\bar{\mathbf{o}}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\bar{\mathbf{o}}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{h}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{h}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \leq p' \leq N_{CS})$ **do**
      if $\bar{\mathbf{o}}_{\pi,p'}^{(k)} = H_0\left(\mathbf{h}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes the shared linking tag $\mathbf{h}_\pi^{(k)} = \sum_{p'}^{N_{CS}} \mathbf{h}_{\pi,p'}^{(k)}$
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{H}, \mathbf{h}_\pi^{(k)})$ to obtain $\mathbf{H}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_\pi^{(k)})$ to obtain $\mathbf{A}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ chooses $\mathbf{u}_{\pi,p}^{(k)} = (u_{\pi,p,1}, \ldots, u_{\pi,p,m})^T$, where $u_{\pi,p,i} \hookleftarrow D_\sigma^n$, for $1 \leq i \leq m$.
    $\mathcal{C}$ computes $\mathbf{r}_{\pi,p}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)} = \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$
    $\boxed{\mathcal{C} \text{ chooses at random } \mathbf{o}_{\pi,p}^{(k)} \hookleftarrow \mathcal{S}_{n,\kappa}}$
    When $\mathcal{A}$ sends $\mathbf{o}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{o}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{r}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \leq p' \leq N_{CS})$ **do**
      if $\mathbf{o}_{\pi,p'}^{(k)} = H_0\left(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes $\mathbf{r}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{z}_{\pi,p'}^{(k)}$
  $\mathcal{C}$ performs $\forall k \in [N_{in}]$, $\mathbf{c}_{\pi+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{r}_\pi^{(k)}, \mathbf{z}_\pi^{(k)}\right)$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \leq k \leq N_{in})$ **do**
      $\mathcal{C}$ selects $\mathbf{t}_{i,p}^{(k)} = (t_{i,p,1}, \ldots, t_{i,p,m})^T$, where $t_{i,p,j} \hookleftarrow D_\sigma^n$, for $1 \leq j \leq m$.
      When $\mathcal{A}$ sends $\mathbf{t}_{i,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{i,p}^{(k)}$ to $\mathcal{A}$.
      $\mathcal{C}$ computes $\mathbf{t}_i^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{i,p'}^{(k)}$
      $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_i^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ runs $\forall k \in [N_{in}]$ $\mathbf{c}_{i+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \left\{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}, \left\{\mathbf{H}_{2q,\pi} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}\right)$.
  for $(1 \leq k \leq N_{in})$ **do**
    $\mathcal{C}$ chooses $b^{(k)} \hookleftarrow \{0, 1\}$.
    $\mathcal{C}$ computes $\mathbf{t}_{\pi,p}^{(k)} = \mathbf{u}_{\pi,p}^{(k)} + \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \cdot (-1)^{b^{(k)}}$, where $\mathbf{S}_{2q,\pi,p}^{(k)} = [(\mathbf{S}_{\pi,p}^{(k)})^T, 1]^T$.
    Continue with prob. $\left(M \exp\left(-\frac{\|\mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{t}_{\pi,p}^{(k)}, \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
    When $\mathcal{A}$ sends $\mathbf{t}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{\pi,p}^{(k)}$ to $\mathcal{A}$
    $\mathcal{C}$ computes $\mathbf{t}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)}$
  return $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$.

Game 2: This game is identical to **Game 1** except that the SigGen algorithm is still modified by the challenger $\mathcal{C}$. When $\mathcal{A}$ sends interactive queries Q_s with (L^t, μ^t) to $\mathcal{C}$ for signing using the SigGen algorithm, then $\mathcal{C}$ behaves as shown in Algorithm 29.

The $\mathcal{C}$ chooses $\mathbf{c}_{\pi+1}$ at random from $\mathcal{S}_{n,\kappa} \subseteq \mathcal{R}_{2q}$, Algorithm 29 (step 22), after that $\mathcal{C}$ programs the answer of the random oracle H_1 $\forall k \in [N_{in}]$ as:

$$H_1\Big(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{r}_{\pi}^{(k)}, \mathbf{z}_{\pi}^{(k)}\Big) = H_1\Big(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{t}_{\pi}^{(k)} \cdot \mathbf{q} \cdot \mathbf{c}_{\pi}, \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{t}_{\pi}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{\pi}\}\Big),$$

without verifying if the values of $\mathbf{r}_{\pi}^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi}^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)}$ $\forall (k, p') \in [N_{in}] \times [N_{CS}]$ were already set. Every time Algorithm 29 is called by $\mathcal{A}$, the probability of generating $\mathbf{u}_{\pi,p'}^{(k)}$, such that $\mathbf{r}_{\pi}^{(k)}$ and $\mathbf{z}_{\pi}^{(k)}$ are equal to one of the previous values that were queried is at most 2^{-n+1}. Therefore, if the SigGen in Game 2 and H_1 are queried Q_s and Q_1 times, respectively, then the probability of getting one collision each time is at most $N_{in} \cdot N_{CS} \cdot (Q_s + Q_1) \cdot 2^{-n+1}$. Additionally, the probability that a collision happens after Q_s queries is at most $N_{in} \cdot N_{CS} \cdot Q_s \cdot (Q_s + Q_1) \cdot 2^{-n+1}$, which is negligible (Based on [DDLL13], Lemma 3.4).

Let S_2 be the event where the $\mathcal{A}$ wins **Game 2** with negligible probability $N_{in} \cdot N_{CS} \cdot Q_s \cdot (Q_s + Q_1) \cdot 2^{-n+1} \leq \epsilon_2$. Then we claim that:

$$|\Pr[S_1] - \Pr[S_2]| \leq \epsilon_2. \tag{6.5}$$

Game 3: In this game the adversary $\mathcal{A}$ queries the random oracle H_0 as in Algorithm 30 (step 5), and stores the answers in Q_{H_0}. The game aborts as in (step 11) if the $\bar{\mathbf{a}}_p$'s are not in the set Q_{H_0}. The successful acceptance of $\bar{\mathbf{a}}_p$ is equal to guess a preimage for the the given $\mathbf{o}_p$ that is committed in (step 5). As a result, the success probability is at most $\frac{1}{|\mathcal{S}_{n,\kappa}|}$.

Algorithm 29 SigGen - Game 2

Input: $\mathcal{SK} = \left\{\mathbf{S}_{\pi,p'}^{(k)}\right\}_{p'\in[N_{CS}],k\in[N_{in}]}, \mu, L = \left\{\mathbf{a}_i^{(k)}\right\}_{i\in[w],k\in[N_{in}]}$ as in (6.1), and PP.

Output: $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k\in[N_{in}]}\right)$

procedure SigGen($\mathcal{SK}, \mu, L,$ PP)
  for $(1 \le k \le N_{in})$ **do**
    The challenger $\mathcal{C}$ computes the linking tag $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H}\cdot\mathbf{S}_{\pi,p}^{(k)} \in \mathcal{R}_q^2$.
    $\mathcal{C}$ sets $\bar{\mathbf{o}}_{\pi,p}^{(k)} = H_0\left(\mathbf{h}_{\pi,p}^{(k)}\right)$
    When $\mathcal{A}$ sends $\bar{\mathbf{o}}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\bar{\mathbf{o}}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{h}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{h}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \le p' \le N_{CS})$ **do**
      if $\bar{\mathbf{o}}_{\pi,p'}^{(k)} = H_0\left(\mathbf{h}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes the shared linking tag $\mathbf{h}_\pi^{(k)} = \sum_{p'}^{N_{CS}} \mathbf{h}_{\pi,p'}^{(k)}$
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{H}, \mathbf{h}_\pi^{(k)})$ to obtain $\mathbf{H}_{2q,\pi}^{(k)} = (2\cdot\mathbf{H}, -2\cdot\mathbf{h}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_\pi^{(k)})$ to obtain $\mathbf{A}_{2q,\pi}^{(k)} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
    $\mathcal{C}$ chooses $\mathbf{u}_{\pi,p}^{(k)} = (u_{\pi,p,1}, \ldots, u_{\pi,p,m})^T$, where $u_{\pi,p,i} \hookleftarrow D_\sigma^n$, for $1 \le i \le m$.
    $\mathcal{C}$ computes $\mathbf{r}_{\pi,p}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)}\cdot\mathbf{u}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)} = \mathbf{H}_{2q,\pi}^{(k)}\cdot\mathbf{u}_{\pi,p}^{(k)}$
    $\mathcal{C}$ chooses at random $\mathbf{o}_{\pi,p}^{(k)} \hookleftarrow \mathcal{S}_{n,\kappa}$.
    When $\mathcal{A}$ sends $\mathbf{o}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{o}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{r}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \le p' \le N_{CS})$ **do**
      if $\mathbf{o}_{\pi,p'}^{(k)} = H_0\left(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes $\mathbf{r}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{z}_{\pi,p'}^{(k)}$
    $\boxed{\mathcal{C} \text{ chooses at random } \mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}}$
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \le k \le N_{in})$ **do**
      $\mathcal{C}$ selects $\mathbf{t}_{i,p}^{(k)} = (t_{i,p,1}, \ldots, t_{i,p,m})^T$, where $t_{i,p,j} \hookleftarrow D_\sigma^n$, for $1 \le j \le m$.
      When $\mathcal{A}$ sends $\mathbf{t}_{i,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{i,p}^{(k)}$ to $\mathcal{A}$.
      $\mathcal{C}$ computes $\mathbf{t}_i^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{i,p'}^{(k)}$
      $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_i^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2\cdot\mathbf{A}, -2\cdot\mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2\times m}$.
    $\mathcal{C}$ runs $\forall k \in [N_{in}]$ $\mathbf{c}_{i+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \left\{\mathbf{A}_{2q,i}^{(k)}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\right\}, \left\{\mathbf{H}_{2q,\pi}\cdot\mathbf{t}_i^{(k)} + \mathbf{q}\cdot\mathbf{c}_i\right\}\right)$.
  for $(1 \le k \le N_{in})$ **do**
    $\mathcal{C}$ chooses $b^{(k)} \hookleftarrow \{0,1\}$.
    $\mathcal{C}$ computes $\mathbf{t}_{\pi,p}^{(k)} = \mathbf{u}_{\pi,p}^{(k)} + \mathbf{S}_{2q,\pi,p}^{(k)}\cdot\mathbf{c}_\pi\cdot(-1)^{b^{(k)}}$, where $\mathbf{S}_{2q,\pi,p}^{(k)} = [(\mathbf{S}_{\pi,p}^{(k)})^T, 1]^T$.
    Continue with prob. $\left(M\exp\left(-\frac{\|\mathbf{S}_{2q,\pi,p}^{(k)}\cdot\mathbf{c}_\pi\|^2}{2\sigma^2}\right)\cosh\left(\frac{\langle\mathbf{t}_{\pi,p}^{(k)}, \mathbf{S}_{2q,\pi,p}^{(k)}\cdot\mathbf{c}_\pi\rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
    When $\mathcal{A}$ sends $\mathbf{t}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{\pi,p}^{(k)}$ to $\mathcal{A}$
    $\mathcal{C}$ computes $\mathbf{t}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)}$
  return $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k\in[N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k\in[N_{in}]}\right)$.

Let S_3 be the event where the $\mathcal{A}$ wins **Game 3** with negligible probability over $\mathcal{R}_q^2$ which is at most $\frac{1}{|\mathcal{S}_{n,\kappa}|} \leq \epsilon_3$. Then we claim that:

$$|\Pr[S_2] - \Pr[S_3]| \leq \epsilon_3. \tag{6.6}$$

Algorithm 30 KeyGen - Game 3 and Game 4

Input: PP: $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$.
Output: $\left(\mathbf{a}^{sh}, \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}\right)$, being the shared public-key and cosigner's secret-key, respectively.
1: **procedure** KeyGen($\mathbf{A}$)
2: Each cosigner $p \in \{1, \ldots, N_{CS}\}$:
3: Selects $\bar{\mathbf{S}}_p^T = (\bar{\mathbf{s}}_{p,1}, \ldots, \bar{\mathbf{s}}_{p,m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\bar{\mathbf{s}}_{p,i} \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
4: Calculates $\bar{\mathbf{a}}_p = (\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2)^T = \mathbf{A} \cdot \bar{\mathbf{S}}_p \bmod q \in \mathcal{R}_q^2$.
5: $\mathbf{o}_p = H_0(\bar{\mathbf{a}}_p)$
6: Broadcasts $\mathbf{o}_p$ to other cosigners $p' \in [N_{CS}]$
7: Receives $\mathbf{o}_{p'}$ with $p' \neq p$, then "p" sends $\bar{\mathbf{a}}_p$ to the cosigners
8: Receives $\bar{\mathbf{a}}_{p'}$ with $p' \neq p$
9: Each cosigner verifies:
10: **for** $(1 \leq p' \leq N_{CS})$ **do**
11: **if** $\mathbf{o}_{p'} = H_0(\bar{\mathbf{a}}_{p'})$ **then** Accept
12: **else** Abort protocol
13: Each cosigner computes the shared public-key as:
14: $\mathbf{a}^{sh} = \sum_{p'}^{N_{CS}} H_2(\bar{\mathbf{a}}_{p'}, L^{sh}) \cdot \bar{\mathbf{a}}_{p'}$ with $L^{sh} = \left\{\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_{N_{CS}}\right\}$
15: Each cosigner calculates its corresponding secret-key as:
16: $\mathbf{S}_p^T = H_2(\bar{\mathbf{a}}_p, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$
17: **return** $\left(\mathbf{a}^{sh}, \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}\right)$, without loss of generality, each cosigner only outputs and holds its corresponding secret-key $\mathbf{S}_{p'}^T$.

Game 4: In this game the adversary $\mathcal{A}$ queries the random oracle H_2 as in Algorithm 30 (step 16), and stores the answers in Q_{H_2}. The game aborts if the $\bar{\mathbf{a}}_p, L^{sh}$'s are in the set Q_{H_2}. We upper bound the probability of this abort in this game at most $\frac{Q_{H_2}}{2^P}$ where P is the min-entropy of $\bar{\mathbf{a}}_p$. We use the Leftover Hash Lemma (LHL) argument to show that the distribution of $\bar{\mathbf{a}}_{p'}$ is closed to uniform just by itself. The statistical distance between the distribution $D(\bar{\mathbf{a}}_{p'})$ and the uniform distribution $\mathcal{R}_q^2$ is at most ϵ_{LHL}, where the min-entropy of $\mathcal{R}_q^2 = 2 \cdot n \log q$. Likewise, we argue the min-entropy of $D(\bar{\mathbf{a}}_p) \leq \frac{1}{2^n}$. This proves that if this is not aborting, the output of H_2 and $\bar{\mathbf{a}}_p$ are completely independent of any adversary view.

Let S_4 be the event where the $\mathcal{A}$ wins **Game 4** with negligible probability over $\mathcal{R}_q^2$ which is at most $\frac{Q_{H_2}}{2^n} \leq \epsilon_4$. Then we claim that:

$$|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_4. \tag{6.7}$$

Game 5: This Game now modify the KeyGen in algorithm 31 (step 4), and SigGen, Algorithm 32 (step 3), where the linking tag is computed. We recall that public-key as $\bar{\mathbf{a}}_{\pi,p}^{(k)} = \mathbf{A} \cdot \bar{\mathbf{S}}_{\pi,p}^{(k)} \bmod q \in \mathcal{R}_q^2$. We now choose $\bar{\mathbf{a}}_{\pi,p'}^{(k)}$ $\forall (k, p') \in [N_{CS}] \times [N_{in}]$ uniformly and randomly such $\bar{\mathbf{a}}_{\pi,p'}^{(k)} \hookleftarrow \mathcal{R}_q^2$. Moreover, we now choose $\mathbf{h}_{\pi,p'}^{(k)}$ $\forall (k, p') \in [N_{CS}] \times [N_{in}]$ uniformly and randomly such $\mathbf{h}_{\pi,p'}^{(k)} \hookleftarrow \mathcal{R}_q^2$, rather than computing the linking tag as $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H} \cdot \mathbf{S}_{\pi,p}^{(k)} \in \mathcal{R}_q^2$. We recall that $\mathbf{S}_{\pi,p}^{(k)} = H_2(\bar{\mathbf{a}}_p, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$ (as KeyGen Algorithm 23) where $\bar{\mathbf{S}}_{\pi,p}^{(k)}$ is chosen small and with coefficients in $(-2^\gamma, 2^\gamma)$. We redefine $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H} \cdot H_2(\bar{\mathbf{a}}_p, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$. Then, we now define a new random matrix $\mathbf{H}_{new} = \mathbf{H} \cdot H_2(\bar{\mathbf{a}}_p, L^{sh})$.

We know that the public parameter $\mathbf{A}$ and $\mathbf{H}_{new}$ are uniform and $\bar{\mathbf{S}}_{\pi,p}^{(k)}$ is chosen small and with coefficients in $(-2^\gamma, 2^\gamma)$. Then, multiplying these $\mathbf{A}$ and $\mathbf{H}_{new}$ by the secret key $\bar{\mathbf{S}}_{\pi,p}^{(k)}$, it results in $\mathbf{a}_\pi^{(k)}$ that is close to uniform over $\mathcal{R}_q^2$.

By the Leftover Hash Lemma (LHL) argument (Lemma 3.5), we show that the statistical distance between the distribution of $\mathbf{a}^{(k)} \bmod q$ and the uniform distribution on $\mathcal{R}_q^2$ is at most $N_{in} \cdot N_{CS} \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$, which is negligible in n.

Let S_5 be the event where the $\mathcal{A}$ wins **Game 5** with negligible probability $\mathcal{R}_q^2$ is at most $N_{in} \cdot N_{CS} \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \leq \epsilon_5$. Then we claim that:

$$|\Pr[S_4] - \Pr[S_5]| \leq \epsilon_5. \tag{6.8}$$

Game 6: The challenger $\mathcal{C}$ behaves different in the random oracle H_0 as illustrated in Algorithm 33 (step 11). On a y-th query $\bar{\mathbf{a}}_{p'}$, from the adversary $\mathcal{A}$, then $\mathcal{C}$ proceeds as follows:

1. $\mathcal{C}$ returns $H_0(\bar{\mathbf{a}}_y)$ if this is already defined.
2. $\mathcal{C}$ chooses at random $\mathbf{o}_y \hookleftarrow \mathcal{S}_{n,\kappa}$, otherwise.

Algorithm 31 KeyGen - Game 5

Input: PP: $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$.
Output: $\left(\mathbf{a}^{sh}, \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}\right)$, being the shared public-key and cosigner's secret-key, respectively.

procedure KeyGen($\mathbf{A}$)
  Each cosigner $p \in \{1, \ldots, N_{CS}\}$:
  Selects $\bar{\mathbf{S}}_p^T = (\bar{\mathbf{s}}_{p,1}, \ldots, \bar{\mathbf{s}}_{p,m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\bar{\mathbf{s}}_{p,i} \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
  Choose $\bar{\mathbf{a}}_p \hookleftarrow \mathcal{R}_q^2$
  $\mathbf{o}_p = H_0(\bar{\mathbf{a}}_p)$
  Broadcasts $\mathbf{o}_p$ to other cosigners $p' \in [N_{CS}]$
  Receives $\mathbf{o}_{p'}$ with $p' \neq p$, then "p" sends $\bar{\mathbf{a}}_p$ to the cosigners
  Receives $\bar{\mathbf{a}}_{p'}$ with $p' \neq p$
  Each cosigner verifies:
  for $(1 \leq p' \leq N_{CS})$ **do**
    if $\mathbf{o}_{p'} = H_0(\bar{\mathbf{a}}_{p'})$ **then** Accept
    else Abort protocol
  Each cosigner computes the shared public-key as:
  $\mathbf{a}^{sh} = \sum_{p'}^{N_{CS}} H_2(\bar{\mathbf{a}}_{p'}, L^{sh}) \cdot \bar{\mathbf{a}}_{p'}$ with $L^{sh} = \left\{\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_{N_{CS}}\right\}$
  Each cosigner calculates its corresponding secret-key as:
  $\mathbf{S}_p^T = H_2(\bar{\mathbf{a}}_p, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$
  return $\left(\mathbf{a}^{sh}, \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}\right)$, without loss of generality, each cosigner only outputs and holds its corresponding secret-key $\mathbf{S}_{p'}^T$.

3. $\mathcal{C}$ verifies if there exists $p' \in [1, y-1]$ such that $\mathbf{o}_y^{(k)} = H_0\left(\bar{\mathbf{a}}_{p'}\right)$ for previous queries of $\bar{\mathbf{a}}_{p'}$. In the case where p' exists, the game is aborted, otherwise:

4. $\mathcal{C}$ sets $\mathbf{o}_y^{(k)} = H_0\left(\bar{\mathbf{a}}_{p'}\right)$ and returns $\mathbf{o}_y^{(k)}$ to $\mathcal{A}$.

The difference between the **Game 5** and **Game 6** is that the challenger $\mathcal{C}$ aborts when he tries to set a same hash value H_0 for two different inputs. This game evaluates the probability that $\mathcal{C}$ aborts the game under this situation. The total number of queries Q_0 to H_0 oracle is at most $Q_0 + Q_k$ where Q_k is the number of queries to the $\mathcal{KO}$ oracle. Then the probability that $\mathcal{C}$ aborts **Game 6** is

$$\sum_{y=1}^{Q_0+Q_k} \Pr\left[\left(\bar{\mathbf{a}}_{p'}\right) \in \left\{\bar{\mathbf{a}}_{y'}\right\}_{y'<y}\right] \leq \sum_{y=1}^{Q_0+Q_k} \sum_{y'=1}^{y-1} \Pr_{\mathbf{A}\hookleftarrow\mathcal{R}_q^{2\times(m-1)}}\left[\bar{\mathbf{a}}_{p'} = \bar{\mathbf{a}}_{y'}\right] \leq$$

$$\sum_{y=1}^{Q_0+Q_k} \frac{y-1}{2^n} \leq \frac{(Q_0+Q_k)(Q_0+Q_k+1)}{2^n}$$

Algorithm 32 SigGen - Game 5

Input: $\mathcal{SK} = \left\{\mathbf{S}_{\pi,p'}^{(k)}\right\}_{p' \in [N_{CS}], k \in [N_{in}]}, \mu, L = \left\{\mathbf{a}_i^{(k)}\right\}_{i \in [w], k \in [N_{in}]}$ as in (6.1), and PP.

Output: $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$

procedure SigGen($\mathcal{SK}, \mu, L,$ PP)
  for $(1 \leq k \leq N_{in})$ **do**
    The linking tag is chosen at random $\mathbf{h}_{\pi,p}^{(k)} \hookleftarrow \mathcal{R}_q^2$
    $\mathcal{C}$ sets $\bar{\mathbf{o}}_{\pi,p}^{(k)} = H_0\left(\mathbf{h}_{\pi,p}^{(k)}\right)$
    When $\mathcal{A}$ sends $\bar{\mathbf{o}}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\bar{\mathbf{o}}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{h}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{h}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \leq p' \leq N_{CS})$ **do**
      if $\bar{\mathbf{o}}_{\pi,p'}^{(k)} = H_0\left(\mathbf{h}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes the shared linking tag $\mathbf{h}_\pi^{(k)} = \sum_{p'}^{N_{CS}} \mathbf{h}_{\pi,p'}^{(k)}$
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{H}, \mathbf{h}_\pi^{(k)})$ to obtain $\mathbf{H}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_\pi^{(k)})$ to obtain $\mathbf{A}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ chooses $\mathbf{u}_{\pi,p}^{(k)} = (u_{\pi,p,1}, \ldots, u_{\pi,p,m})^T$, where $u_{\pi,p,i} \hookleftarrow D_\sigma^n$, for $1 \leq i \leq m$.
    $\mathcal{C}$ computes $\mathbf{r}_{\pi,p}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)} = \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$
    $\mathcal{C}$ chooses at random $\mathbf{o}_{\pi,p}^{(k)} \hookleftarrow \mathcal{S}_{n,\kappa}$.
    When $\mathcal{A}$ sends $\mathbf{o}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{o}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{r}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \leq p' \leq N_{CS})$ **do**
      if $\mathbf{o}_{\pi,p'}^{(k)} = H_0\left(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes $\mathbf{r}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{z}_{\pi,p'}^{(k)}$
  $\mathcal{C}$ chooses at random $\mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \leq k \leq N_{in})$ **do**
      $\mathcal{C}$ selects $\mathbf{t}_{i,p}^{(k)} = (t_{i,p,1}, \ldots, t_{i,p,m})^T$, where $t_{i,p,j} \hookleftarrow D_\sigma^n$, for $1 \leq j \leq m$.
      When $\mathcal{A}$ sends $\mathbf{t}_{i,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{i,p}^{(k)}$ to $\mathcal{A}$.
      $\mathcal{C}$ computes $\mathbf{t}_i^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{i,p'}^{(k)}$
      $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_i^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ runs $\forall k \in [N_{in}]$ $\mathbf{c}_{i+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \left\{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}, \left\{\mathbf{H}_{2q,\pi} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}\right)$.
  for $(1 \leq k \leq N_{in})$ **do**
    $\mathcal{C}$ chooses $b^{(k)} \hookleftarrow \{0, 1\}$.
    $\mathcal{C}$ computes $\mathbf{t}_{\pi,p}^{(k)} = \mathbf{u}_{\pi,p}^{(k)} + \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \cdot (-1)^{b^{(k)}}$, where $\mathbf{S}_{2q,\pi,p}^{(k)} = [(\mathbf{S}_{\pi,p}^{(k)})^T, 1]^T$.
    Continue with prob. $\left(M \exp\left(-\frac{\|\mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{t}_{\pi,p}^{(k)}, \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
    When $\mathcal{A}$ sends $\mathbf{t}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{\pi,p}^{(k)}$ to $\mathcal{A}$
    $\mathcal{C}$ computes $\mathbf{t}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)}$
  return $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$.

Let S_6 be the event where the $\mathcal{A}$ wins this Game with negligible probability $\frac{(Q_0+Q_k)(Q_0+Q_k+1)}{2^n} \leq \epsilon_6$. Then we argue that:

$$|\Pr[S_5] - \Pr[S_6]| \leq \epsilon_6. \tag{6.9}$$

Algorithm 33 KeyGen - Game 6

Input: PP: $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$.
Output: $\left(\mathbf{a}^{sh}, \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}\right)$, being the shared public-key and cosigner's secret-key, respectively.
1: **procedure** KeyGen($\mathbf{A}$)
2: Each cosigner $p \in \{1, \ldots, N_{CS}\}$:
3: Selects $\bar{\mathbf{S}}_p^T = (\bar{\mathbf{s}}_{p,1}, \ldots, \bar{\mathbf{s}}_{p,m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\bar{\mathbf{s}}_{p,i} \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
4: Choose $\bar{\mathbf{a}}_p \hookleftarrow \mathcal{R}_q^2$
5: $\mathbf{o}_p = H_0(\bar{\mathbf{a}}_p)$
6: Broadcasts $\mathbf{o}_p$ to other cosigners $p' \in [N_{CS}]$
7: Receives $\mathbf{o}_{p'}$ with $p' \neq p$, then "p" sends $\bar{\mathbf{a}}_p$ to the cosigners
8: Receives $\bar{\mathbf{a}}_{p'}$ with $p' \neq p$
9: Each cosigner verifies:
10: **for** $(1 \leq p' \leq N_{CS})$ **do**
11: **if** $\boxed{\text{Choose } \mathbf{o}_{p'} \hookleftarrow \mathcal{R}_q^2}$ **then** Accept
else Abort protocol
12: Each cosigner computes the shared public-key as:
13: $\mathbf{a}^{sh} = \sum_{p'}^{N_{CS}} H_2(\bar{\mathbf{a}}_{p'}, L^{sh}) \cdot \bar{\mathbf{a}}_{p'}$ with $L^{sh} = \left\{\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_{N_{CS}}\right\}$
14: Each cosigner calculates its corresponding secret-key as:
15: $\mathbf{S}_p^T = H_2(\bar{\mathbf{a}}_p, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$
16: **return** $\left(\mathbf{a}^{sh}, \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}\right)$, without loss of generality, each cosigner only outputs and holds its corresponding secret-key $\mathbf{S}_{p'}^T$.

Game 7: This Game performs similar to **Game 6** but we now modify (for the signer π) the KeyGen Algorithm 34 (step 14). The aggregate public-key as $\mathbf{a}_\pi^{sh(k)} = \sum_{p'}^{N_{CS}} H_2(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{a}}_{\pi,p'}^{(k)}$. We now choose $\mathbf{a}_\pi^{sh(k)}$ $\forall(k) \in [N_{in}]$ uniformly and randomly such $\mathbf{a}_\pi^{sh(k)} \hookleftarrow \mathcal{R}_q^2$. As in **Game 1**, it shows that $\bar{\mathbf{a}}_{\pi,p'}^{(k)}$ is uniformly random. We assume that $\forall\ h \hookleftarrow \mathcal{S}_{n,\kappa}$ where h is the output of the hash function H_2. We said that h needs to be invertible in $\mathcal{R}_q^2$, then to achieve this condition, we choose $\mathcal{S}_{n,\kappa}$ such that $\|h\|_\infty < \frac{1}{\sqrt{k}} \cdot q^{1/k}$ as shown in ([LS18], Corollary 1.2), with probability 1.

Let S_7 be the event where the $\mathcal{A}$ wins **Game 7** with negligible probability, that is $1 \leq \epsilon_7$. Then we claim that:

$$|\Pr[S_6] - \Pr[S_7]| \leq \epsilon_7. \tag{6.10}$$

Algorithm 34 KeyGen - Game 7

Input: PP: $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$.
Output: $\left(\mathbf{a}^{sh}, \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}\right)$, being the shared public-key and cosigner's secret-key, respectively.

procedure KeyGen($\mathbf{A}$)
  Each cosigner $p \in \{1, \ldots, N_{CS}\}$:
  Selects $\bar{\mathbf{S}}_p^T = (\bar{\mathbf{s}}_{p,1}, \ldots, \bar{\mathbf{s}}_{p,m-1}) \in \mathcal{R}_q^{1\times(m-1)}$, where $\bar{\mathbf{s}}_{p,i} \hookleftarrow (-2^\gamma, 2^\gamma)^n$, for $1 \leq i \leq m-1$
  Choose $\bar{\mathbf{a}}_p \hookleftarrow \mathcal{R}_q^2$
  $\mathbf{o}_p = H_0(\bar{\mathbf{a}}_p)$
  Broadcasts $\mathbf{o}_p$ to other cosigners $p' \in [N_{CS}]$
  Receives $\mathbf{o}_{p'}$ with $p' \neq p$, then "p" sends $\bar{\mathbf{a}}_p$ to the cosigners
  Receives $\bar{\mathbf{a}}_{p'}$ with $p' \neq p$
  Each cosigner verifies:
  for $(1 \leq p' \leq N_{CS})$ **do**
    if $\mathbf{o}_{p'} = H_0(\bar{\mathbf{a}}_{p'})$ **then** Accept
    else Abort protocol
  Each cosigner computes the shared public-key as:
  Choose $\mathbf{a}^{sh} \hookleftarrow \mathcal{R}_q^2$
  Each cosigner calculates its corresponding secret-key as:
  $\mathbf{S}_p^T = H_2(\bar{\mathbf{a}}_p, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$
  return $\left(\mathbf{a}^{sh}, \left\{\mathbf{S}_1^T, \ldots, \mathbf{S}_{N_{CS}}^T\right\}\right)$, without loss of generality, each cosigner only outputs and holds its corresponding secret-key $\mathbf{S}_{p'}^T$.

Game 8: This Game now changes are made on the $\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}$ from the SigGen algorithm. When $\mathcal{A}$ sends interactive queries Q_s with (L^t, μ^t) to $\mathcal{C}$ for signing using the SigGen algorithm, then $\mathcal{C}$ behaves as in Algorithm 35. This time, the $\mathcal{C}$ chooses $\mathbf{t}_{\pi,p}^{(k)}$ at random from $D_\sigma^{n\times m}$ as in Algorithm 35 (step 33) instead of computing it as $\mathbf{t}_{\pi,p}^{(k)} = \mathbf{u}_{\pi,p}^{(k)} + \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \cdot (-1)^{b^{(k)}}$ (Based on [DDLL13], Lemma 3.5). We claim that this Game is forgeable when $\mathcal{A}$ finds a PPT algorithm $\mathcal{F}$ to solve the $\mathbf{MSIS}_{q,m,k,\beta}^{\mathcal{K}}$ problem. This attack performs as follows:

1. Random coins are selected for the forger ϕ and signer ψ.
2. The random oracle H_1 is called to generate the responses of the users in the SigGen scheme, $(\mathbf{c}_1, \ldots, \mathbf{c}_w) \hookleftarrow \mathcal{S}_{n,\kappa}$.
3. These create a SubRoutine that takes as input $(\mathbf{A}_{2q,\pi}^{(k)}, \phi, \psi, \mathbf{c}_1, \ldots, \mathbf{c}_w)$.
4. $\mathcal{F}$ is initialized and run by providing the $\mathbf{A}_{2q,\pi}^{(k)}$ and forger's random coins ϕ.
5. The SubRoutine signs the message μ using the signer's coins ψ in the Algorithm 35, this produces a signature $\sigma_L(\mu)$.
6. During the signing process, $\mathcal{F}$ calls the oracle H_1 and answers are placed in the list $(\mathbf{c}_1, \ldots, \mathbf{c}_w)$, the queries are kept in a table in the event that same queries are used in this oracle.

Algorithm 35 SigGen - Game 8

Input: $\mathcal{SK} = \left\{\mathbf{S}_{\pi,p'}^{(k)}\right\}_{p' \in [N_{CS}], k \in [N_{in}]}, \mu, L = \left\{\mathbf{a}_i^{(k)}\right\}_{i \in [w], k \in [N_{in}]}$ as in (6.1), and PP.

Output: $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$

procedure SigGen($\mathcal{SK}, \mu, L, \mathsf{PP}$)
  for $(1 \leq k \leq N_{in})$ **do**
    The linking tag is chosen at random $\mathbf{h}_{\pi,p}^{(k)} \hookleftarrow \mathcal{R}_q^2$.
    $\mathcal{C}$ sets $\bar{\mathbf{o}}_{\pi,p}^{(k)} = H_0\left(\mathbf{h}_{\pi,p}^{(k)}\right)$
    When $\mathcal{A}$ sends $\bar{\mathbf{o}}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\bar{\mathbf{o}}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{h}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{h}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \leq p' \leq N_{CS})$ **do**
      if $\bar{\mathbf{o}}_{\pi,p'}^{(k)} = H_0\left(\mathbf{h}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes the shared linking tag $\mathbf{h}_\pi^{(k)} = \sum_{p'}^{N_{CS}} \mathbf{h}_{\pi,p'}^{(k)}$
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{H}, \mathbf{h}_\pi^{(k)})$ to obtain $\mathbf{H}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_\pi^{(k)})$ to obtain $\mathbf{A}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ chooses $\mathbf{u}_{\pi,p}^{(k)} = (u_{\pi,p,1}, \ldots, u_{\pi,p,m})^T$, where $u_{\pi,p,i} \hookleftarrow D_\sigma^n$, for $1 \leq i \leq m$.
    $\mathcal{C}$ computes $\mathbf{r}_{\pi,p}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)} = \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$
    $\mathcal{C}$ chooses at random $\mathbf{o}_{\pi,p}^{(k)} \hookleftarrow \mathcal{S}_{n,\kappa}$.
    When $\mathcal{A}$ sends $\mathbf{o}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{o}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{r}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \leq p' \leq N_{CS})$ **do**
      if $\mathbf{o}_{\pi,p'}^{(k)} = H_0\left(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes $\mathbf{r}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{z}_{\pi,p'}^{(k)}$
  $\mathcal{C}$ chooses at random $\mathbf{c}_{\pi+1} \hookleftarrow \mathcal{S}_{n,\kappa}$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \leq k \leq N_{in})$ **do**
      $\mathcal{C}$ selects $\mathbf{t}_{i,p}^{(k)} = (t_{i,p,1}, \ldots, t_{i,p,m})^T$, where $t_{i,p,j} \hookleftarrow D_\sigma^n$, for $1 \leq j \leq m$.
      When $\mathcal{A}$ sends $\mathbf{t}_{i,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{i,p}^{(k)}$ to $\mathcal{A}$.
      $\mathcal{C}$ computes $\mathbf{t}_i^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{i,p'}^{(k)}$
      $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_i^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ runs $\forall k \in [N_{in}]$ $\mathbf{c}_{i+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \left\{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}, \left\{\mathbf{H}_{2q,\pi} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}\right)$.
  for $(1 \leq k \leq N_{in})$ **do**
    $\mathcal{C}$ chooses $b^{(k)} \hookleftarrow \{0, 1\}$.

    $\boxed{\mathcal{C} \text{ chooses } \mathbf{t}_{\pi,p}^{(k)} \hookleftarrow D_\sigma^{n \times m}}$

    Continue with prob. $\left(M \exp\left(-\frac{\|\mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{t}_{\pi,p}^{(k)}, \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
    When $\mathcal{A}$ sends $\mathbf{t}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{\pi,p}^{(k)}$ to $\mathcal{A}$
    $\mathcal{C}$ computes $\mathbf{t}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)}$
  return $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$.

7. $\mathcal{F}$ is stops this simulation and outputs a forgery $\sigma(\mu)^* = \left(\mathbf{c}_1^*, \left\{\mathbf{t}_1^{*(k)}, \ldots, \mathbf{t}_w^{*(k)}\right\}, \mathbf{h}_\pi^{*(k)}\right)$, with negligible probability. This output has to be successfully accepted by the SigVer algorithm.

If the random oracle was not called using some input $\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{*(k)} \cdot \mathbf{q} \cdot \mathbf{c}_i^*, \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{*(k)} + \mathbf{q} \cdot \mathbf{c}_i^*$ $(\forall i, k \in [w] \times [N_{in}])$, then $\mathcal{F}$ has $1/|\mathcal{S}_{n,\kappa}|$ chances of producing a $\mathbf{c}_{i+1}^*$ such that $\mathbf{c}_{i+1}^* = H_1\left(L^*, \mathbf{H}_{2q}^{(k)}, \mu^*, \mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{*(k)} \cdot \mathbf{q} \cdot \mathbf{c}_i^*, \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_i^{*(k)} + \mathbf{q} \cdot \mathbf{c}_i^*\right)$. We claim that $\epsilon_7 - 1/|\mathcal{S}_{n,\kappa}|$ is the probability that $\mathbf{c}_{i+1}^* = \mathbf{c}_{j+1}$ for some j. In this analysis, we now consider two types of forgeries:

Forgery 1. We consider that $\mathbf{c}_{j+1}$ is the result after using $\mathcal{F}$ which is $\mathbf{c}_{j+1} = H_1\left(L', \mathbf{H}_{2q}^{(k)}, \mu', \mathbf{A}_{2q,j}^{(k)} \cdot \mathbf{t}_j^{\prime(k)} \cdot \mathbf{q} \cdot \mathbf{c}_j, \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_j^{\prime(k)} + \mathbf{q} \cdot \mathbf{c}_j\right)$. Then we have $\forall k \in [N_{in}]$:

$$H_1\left(L^*, \mathbf{H}_{2q}^{(k)}, \mu^*, \mathbf{A}_{2q,j}^{(k)} \cdot \mathbf{t}_j^{*(k)} \cdot \mathbf{q} \cdot \mathbf{c}_j, \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_j^{*(k)} + \mathbf{q} \cdot \mathbf{c}_j\right) =$$
$$H_1\left(L', \mathbf{H}_{2q}^{(k)}, \mu', \mathbf{A}_{2q,j}^{(k)} \cdot \mathbf{t}_j^{\prime(k)} \cdot \mathbf{q} \cdot \mathbf{c}_j, \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_j^{\prime(k)} + \mathbf{q} \cdot \mathbf{c}_j\right),$$

$\mathcal{F}$ finds a preimage of $\mathbf{c}_{j+1}$ if $\mu^* \neq \mu'$ or $\mathbf{A}_{2q,j}^{(k)} \cdot \mathbf{t}_j^{*(k)} \cdot \mathbf{q} \cdot \mathbf{c}_j \neq \mathbf{A}_{2q,j}^{(k)} \cdot \mathbf{t}_j^{\prime(k)} \cdot \mathbf{q} \cdot \mathbf{c}_j$ or $\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_j^{*(k)} + \mathbf{q} \cdot \mathbf{c}_j \neq \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_j^{\prime(k)} + \mathbf{q} \cdot \mathbf{c}_j$. Then, we have with overwhelming probability that $\mu^* = \mu'$ or $\mathbf{A}_{2q,j}^{(k)} \cdot \mathbf{t}_j^{*(k)} \cdot \mathbf{q} \cdot \mathbf{c}_j = \mathbf{A}_{2q,j}^{(k)} \cdot \mathbf{t}_j^{\prime(k)} \cdot \mathbf{q} \cdot \mathbf{c}_j$ or $\mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_j^{*(k)} + \mathbf{q} \cdot \mathbf{c}_j = \mathbf{H}_{2q}^{(k)} \cdot \mathbf{t}_j^{\prime(k)} + \mathbf{q} \cdot \mathbf{c}_j$. These equalities will result in: $\mathbf{A}_{2q,j}^{(k)}(\mathbf{t}_j^{*(k)} - \mathbf{t}_j^{\prime(k)}) = 0 \bmod q$ and $\mathbf{H}_{2q}^{(k)}(\mathbf{t}_j^{*(k)} - \mathbf{t}_j^{\prime(k)}) = 0 \bmod q$. We assume that both $\mathbf{t}_j^{*(k)}$ and $\mathbf{t}_j^{\prime(k)}$ are different and they met the SigVer Algorithm conditions, so it results in $\mathbf{t}_j^{*(k)} - \mathbf{t}_j^{\prime(k)} \neq 0 \bmod q$, and $\|\mathbf{t}_j^{*(k)} - \mathbf{t}_j^{\prime(k)}\| \leq 2\beta_v$.

Forgery 2. We assume that $\mathbf{c}_{j+1}$ was a response to a random oracle H_1 query made by $\mathcal{A}$ and it records the $\mathbf{c}_{j+1}$ and the signature $\sigma(\mu)$ on message μ. Then, fresh random elements are generated as $(\mathbf{c}'_j, \ldots, \mathbf{c}'_w) \hookleftarrow \mathcal{S}_{n,\kappa}$. We use the forking lemma [BN06] to show the probability of $\mathbf{c}_{j+1} \neq \mathbf{c}'_{j+1}$ and the forger uses an oracle

response $\mathbf{c}'_{j+1}$ is at least:

$$\left(\Pr[S_7] - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \cdot \left(\frac{\Pr[S_7] - \frac{1}{|\mathcal{S}_{n,\kappa}|}}{Q_s + Q_1} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right),$$

which is negligible. Therefore, with this probability, $\mathcal{A}$ creates a signature $\sigma(\mu)' = \left(\mathbf{c}'_1, \{\mathbf{t}'^{(k)}_1, \ldots, \mathbf{t}'^{(k)}_w\}, \mathbf{h}'^{(k)}_\pi\right)$ where $\mathbf{A}^{(k)}_{2q,j} \cdot \mathbf{t}^{*(k)}_j \cdot \mathbf{q} \cdot \mathbf{c}_j = \mathbf{A}^{(k)}_{2q,j} \cdot \mathbf{t}'^{(k)}_j \cdot \mathbf{q} \cdot \mathbf{c}'_j$ and $\mathbf{H}^{(k)}_{2q} \cdot \mathbf{t}^{*(k)}_j + \mathbf{q} \cdot \mathbf{c}_j = \mathbf{H}^{(k)}_{2q} \cdot \mathbf{t}'^{(k)}_j + \mathbf{q} \cdot \mathbf{c}'_j$. We now obtain $\mathbf{A}^{(k)}_{2q,j} \cdot (\mathbf{t}^{*(k)}_j - \mathbf{t}'^{(k)}_j) = \mathbf{q}(\mathbf{c}_j - \mathbf{c}'_j) \bmod 2q$ and $\mathbf{H}^{(k)}_{2q}(\mathbf{t}^{*(k)}_j - \mathbf{t}'^{(k)}_j) = \mathbf{q}(\mathbf{c}_j - \mathbf{c}'_j) \bmod 2q$. Since $\mathbf{c}_j - \mathbf{c}'_j \neq 0 \bmod 2$, so in both equations, we have $\mathbf{t}^{*(k)}_j - \mathbf{t}'^{(k)}_j \neq 0 \bmod 2q$ where $\|\mathbf{t}^{*(k)}_j - \mathbf{t}'^{(k)}_j\|_\infty < q/2$. By applying mod$q$ reduction, we find a small non-zero vector $\mathbf{v}^{(k)} = \mathbf{t}^{*(k)}_j - \mathbf{t}'^{(k)}_j \neq 0 \bmod q$. This $\mathbf{v}^{(k)}$ will compute $\mathbf{A}^{(k)}_{2q,j} \cdot \mathbf{v}^{(k)} = \mathbf{0} \bmod q$ and $\mathbf{H}^{(k)}_{2q} \cdot \mathbf{v}^{(k)} = \mathbf{0} \bmod q$ with $\|\mathbf{v}^{(k)}\| \leq 2\beta_v$. Since $\mathbf{v}^{(k)}$ is same for both $\mathbf{A}^{(k)}_{2q,j}$ and $\mathbf{H}^{(k)}_{2q}$, we only use the former to continue this analysis. We say that $\mathbf{A}^{(k)}_{2q,j} \bmod q = 2(\mathbf{A}, -\mathbf{a}^{(k)}) \bmod q$, then $2(\mathbf{A}, -\mathbf{a}^{(k)})\mathbf{v}^{(k)} = \mathbf{0} \bmod q$, this implies that $(\mathbf{A}, -\mathbf{a}^{(k)})\mathbf{v}^{(k)} = \mathbf{0} \bmod q$, since q is odd. Therefore, this vector $\mathbf{v}$ will be a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem, where $\beta = 2\beta_v$, with non-negligible probability and with respect to $(\mathbf{A}, -\mathbf{a}^{(k)})$ over $\mathcal{R}^2_q$.

Let S_8 be the event where the $\mathcal{A}$ wins **Game 8** with negligible probability $\left(\Pr[S_7] - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \cdot \left(\frac{\Pr[S_7] - \frac{1}{|\mathcal{S}_{n,\kappa}|}}{Q_s + Q_1} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right)$ to solve the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem.

Combining the results of the above Games (6.3), (6.4), (6.5), (6.6), (6.7), (6.8), (6.9), and (6.10) we obtain:

$$\left|\left(\Pr[S_7] - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right) \cdot \left(\frac{\Pr[S_7] - \frac{1}{|\mathcal{S}_{n,\kappa}|}}{Q_s + Q_1} - \frac{1}{|\mathcal{S}_{n,\kappa}|}\right)\right| \leq \Pr[\text{Solve } \mathbf{MSIS}],$$

Since $\Pr[S_7] \geq \Pr[S_0] - \epsilon_{uf}$ with $\epsilon_{uf} = \sum_{i=1}^{7} \epsilon_i$, and we let $\delta = \Pr[S_0]$ then

$$\left|\left(\delta-\epsilon_{uf}-\frac{1}{|\mathcal{S}_{n,\kappa}|}\right)\cdot\left(\frac{\delta-\epsilon_{uf}-\frac{1}{|\mathcal{S}_{n,\kappa}|}}{Q_s+Q_1}-\frac{1}{|\mathcal{S}_{n,\kappa}|}\right)\right|\leq \Pr[\text{Solve } \mathbf{MSIS}]$$

□

Theorem 6.12 (Anonymity)**.** *Suppose that the quantities:* $\frac{N_{in}\cdot N_{CS}}{2}\cdot\sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$ *and* $\frac{2\cdot Q_{anon}\cdot(2\cdot Q_{anon}+2\cdot Q_{anon}\cdot N_{CS}+1)}{2^n}$ *are negligible in* n *with* $Q_{anon}=max(Q_0,Q_1,Q_s)$*. Then, the L2RS-CS scheme provides unconditionally anonymity against any adversary who makes* Q_{anon} *queries to the random oracles* H_0, H_1*, and* $\mathcal{SO}$*.*

Proof. We prove the anonymity property of the MIMO.L2RS-CS scheme by using the sequence-of-games approach [Sho04]:

Game 0 - Real Game: This Game follows the definition of unconditional anonymity in Section 5.1.3. We assume that an adversary $\mathcal{A}$, by using the $\mathcal{KO}$, creates a list of $\mathbf{pk}^{(k)}$'s $L=\left(\mathbf{pk}_{i_0}^{(k)},\mathbf{pk}_{i_1}^{(k)}\right)$ $\forall k\in[N_{in}]$ and $\forall i_0,i_1\in[w]$. $\mathcal{A}$ gives the L and a message μ to the challenger. The challenger then flips a coin $b\hookleftarrow\{0,1\}$, then creates a signature $\sigma(\mu)_b=\mathsf{SigGen}(\mathbf{S}_{i_b}^{(k)},\mu,L,\mathsf{PP})$ and gives $\sigma(\mu)_b$ to $\mathcal{A}$. The adversary $\mathcal{A}$ outputs a guess b'. $\mathcal{A}$ wins this game if the following conditions are achieved:

1. $\mathbf{pk}_{i_0}^{(k)}$ and $\mathbf{pk}_{i_1}^{(k)}$ must not be used by $\mathcal{CO}$ and $\mathcal{SO}$.
2. $\mathcal{A}$ outputs b' such $b=b'$, with $\Pr=1/2$.

If we define the event S_0 where the adversary $\mathcal{A}$ wins **Game 0**, then we claim that $\mathcal{A}_2$'s advantage is $\frac{1}{2}+\epsilon_0$.

$$|\Pr[S_0]-\tfrac{1}{2}|\leq\epsilon_0. \tag{6.11}$$

Game 1: In this game, we analyse the KeyGen Algorithm 23 in order to show that $\bar{\mathbf{a}}_{\pi,p}^{(k)}$ is independent to $\bar{\mathbf{a}}_{\pi,p'}^{(k)}$. In the step 11 of this protocol, the challenger verifies that $\mathbf{o}_{\pi,p'}=H_0(\bar{\mathbf{a}}_{\pi,p'})$. Then, there are two cases to be considered:

- Case 1: $\bar{\mathbf{a}}_{\pi,p'}$ was queried by $\mathcal{A}$ to the random oracle H_0 before $\bar{\mathbf{a}}_{\pi,p}$ was sent. We define the event E_1 where the adversary $\mathcal{A}$ queries the H_0 up to revealing $\bar{\mathbf{a}}_{\pi,p}$. E_2 to be the event when $\mathcal{A}$ guesses $\bar{\mathbf{a}}_{\pi,p}$, with no information of $\bar{\mathbf{a}}_{\pi,p'}$. Then, we state that with the following probability $\bar{\mathbf{a}}_{\pi,p'}$ is independent of $\bar{\mathbf{a}}_{\pi,p}$:

$$\Pr[E_1] = \Pr[E_2] \leq \frac{1}{2^n}$$

, where 2^n is the min-entropy of $\bar{\mathbf{a}}_{\pi,p}$.

- Case 2: $\bar{\mathbf{a}}_{\pi,p'}$ was not queried, which means that the chance to satisfy the following condition is negligible:

$$\Pr[\mathbf{o}_{\pi,p'} = H_0(\bar{\mathbf{a}}_{\pi,p'})] \leq \frac{1}{2^{|H_0|}}$$

Let S_1 be the event where the $\mathcal{A}$ wins **Game 1** with negligible probability $\mathcal{R}_q^2$ is at most $\frac{1}{2^n} \leq \epsilon_1$. Then we claim that:

$$|\Pr[S_0] - \Pr[S_1]| \leq \epsilon_1. \tag{6.12}$$

Game 2: This Game now modifies the KeyGen Algorithm 23 (step 4), and SigGen Algorithm 24 (step 3), where the linking tag is computed. We know that the public-key is computed as $\bar{\mathbf{a}}_{\pi,p}^{(k)} = \mathbf{A} \cdot \bar{\mathbf{S}}_{\pi,p}^{(k)} \bmod q \in \mathcal{R}_q^2$. Then, we choose $\bar{\mathbf{a}}_{\pi,p'}^{(k)}$ $\forall (k,p') \in [N_{CS}] \times [N_{in}]$ uniformly and randomly such $\bar{\mathbf{a}}_{\pi,p'}^{(k)} \hookleftarrow \mathcal{R}_q^2$. Moreover, we select $\mathbf{h}_{\pi,p'}^{(k)}$ $\forall (k,p') \in [N_{CS}] \times [N_{in}]$ uniformly and randomly such $\mathbf{h}_{\pi,p'}^{(k)} \hookleftarrow \mathcal{R}_q^2$, rather than computing the linking tag as $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H} \cdot \mathbf{S}_{\pi,p}^{(k)} \in \mathcal{R}_q^2$. We recall that $\mathbf{S}_{\pi,p}^{(k)} = H_2(\bar{\mathbf{a}}_p, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$ (as KeyGen Algorithm 23) where $\bar{\mathbf{S}}_{\pi,p}^{(k)}$ is chosen small and with coefficients in $(-2^\gamma, 2^\gamma)$. We redefine $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H} \cdot H_2(\bar{\mathbf{a}}_{\pi,p}^{(k)}, L^{sh}) \cdot \bar{\mathbf{S}}_p^T$, where a new random matrix $\mathbf{H}_{new} = \mathbf{H} \cdot H_2(\bar{\mathbf{a}}_{\pi,p}^{(k)}, L^{sh})$.

Since the public parameter $\mathbf{A}$ and $\mathbf{H}_{new}$ are uniform and $\bar{\mathbf{S}}_{\pi,p}^{(k)}$ is chosen small and with coefficients in $(-2^\gamma, 2^\gamma)$, then multiplying these $\mathbf{A}$ and $\mathbf{H}_{new}$ by the secret key $\bar{\mathbf{S}}_{\pi,p}^{(k)}$, it results in $\bar{\mathbf{a}}_{\pi,p}^{(k)}$ and $\mathbf{h}_{\pi,p}^{(k)}$ that are close to uniform over $\mathcal{R}_q^2$. By the Leftover

Hash Lemma (LHL) argument (Lemma 3.5), we show that the statistical distance between the distribution of $\mathbf{a}^{(k)} \bmod q$ and the uniform distribution on $\mathcal{R}_q^2$ is at most $N_{in} \cdot N_{CS} \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}}$, which is negligible in n.

Let S_2 be the event where the $\mathcal{A}$ wins **Game 2** with negligible probability $\mathcal{R}_q^2$ is at most $N_{in} \cdot N_{CS} \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \leq \epsilon_2$. Then we claim that:

$$|\Pr[S_1] - \Pr[S_2]| \leq \epsilon_2. \tag{6.13}$$

Game 3: Rather that computing $\bar{\mathbf{o}}_{\pi,p}^{(k)} = H_0\big(\mathbf{h}_{\pi,p}^{(k)}\big)$, $\bar{\mathbf{o}}_{\pi,p}^{(k)}$ is now chosen at random as seen in Algorithm 36, in step 5. On a y-th query $\mathbf{h}_y^{(k)}$ from the adversary $\mathcal{A}$, then $\mathcal{C}$ proceeds as follows:

1. $\mathcal{C}$ returns $H_0\big(\mathbf{h}_y^{(k)}\big)$ if this is already defined.
2. $\mathcal{C}$ chooses at random $\bar{\mathbf{o}}_y^{(k)} \hookleftarrow \mathcal{S}_{n,\kappa}$, otherwise.
3. $\mathcal{C}$ verifies if there exists $p' \in [1, y-1]$ such that $\bar{\mathbf{o}}_y^{(k)} = H_0\big(\mathbf{h}_{\pi,p'}^{(k)}\big)$ for previous queries of $\mathbf{h}_{\pi,p'}^{(k)}$. In the case where p' exists, the game is aborted, otherwise:
4. $\mathcal{C}$ sets $\bar{\mathbf{o}}_y^{(k)} = H_0\big(\mathbf{h}_{\pi,p'}^{(k)}\big)$ and returns $\bar{\mathbf{o}}_y^{(k)}$ to $\mathcal{A}$.

The difference between the **Game 2** and **Game 3** is that the challenger $\mathcal{C}$ aborts when he tries to set a same hash value H_0 for two different inputs. This game evaluates the probability that $\mathcal{C}$ aborts the game under this situation. The total number of queries Q_0 to H_0 oracle is at most $Q_0 + Q_s$. Then the probability that $\mathcal{C}$ aborts **Game 3** is

$$\sum_{y=1}^{Q_0+Q_s} \Pr\left[\left(\mathbf{h}_{\pi,p'}^{(k)}\right) \in \left\{\mathbf{h}_{y'}^{(k)}\right\}_{y'<y}\right] \leq \sum_{y=1}^{Q_0+Q_s} \sum_{y'=1}^{y-1} \Pr_{\bar{\mathbf{o}}_y^{(k)} \hookleftarrow \mathcal{S}_{n,\kappa}}\left[\mathbf{h}_{\pi,p'}^{(k)} = \mathbf{h}_{y'}^{(k)}\right] \leq$$

$$\sum_{y=1}^{Q_0+Q_s} \frac{y-1}{2^n} \leq \frac{(Q_0+Q_s)(Q_0+Q_s+1)}{2^n}$$

Let S_3 be the event where the $\mathcal{A}$ wins this Game with negligible probability $\frac{(Q_0+Q_s)(Q_0+Q_s+1)}{2^n} \leq \epsilon_3$. Then we argue that:

$$|\Pr[S_2] - \Pr[S_3]| \leq \epsilon_3. \tag{6.14}$$

Algorithm 36 SigGen - Game 3

Input: $\mathcal{SK} = \left\{\mathbf{S}_{\pi,p'}^{(k)}\right\}_{p' \in [N_{CS}], k \in [N_{in}]}, \mu$, $L = \left\{\mathbf{a}_i^{(k)}\right\}_{i \in [w], k \in [N_{in}]}$ as in (6.1), and PP.

Output: $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$

procedure SigGen($\mathcal{SK}, \mu, L, \mathsf{PP}$)
  for $(1 \leq k \leq N_{in})$ **do**
    The challenger $\mathcal{C}$ computes the linking tag $\mathbf{h}_{\pi,p}^{(k)} = \mathbf{H} \cdot \mathbf{S}_{\pi,p}^{(k)} \in \mathcal{R}_q^2$.
    $\mathcal{C}$ sets $\bar{\mathbf{o}}_{\pi,p}^{(k)} = H_0\left(\mathbf{h}_{\pi,p}^{(k)}\right)$
    $\mathcal{C}$ chooses at random $\bar{\mathbf{o}}_{\pi,p}^{(k)} \hookleftarrow \mathcal{S}_{n,\kappa}$
    When $\mathcal{A}$ sends $\bar{\mathbf{o}}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\bar{\mathbf{o}}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{h}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{h}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \leq p' \leq N_{CS})$ **do**
      if $\bar{\mathbf{o}}_{\pi,p'}^{(k)} = H_0\left(\mathbf{h}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes the shared linking tag $\mathbf{h}_\pi^{(k)} = \sum_{p'}^{N_{CS}} \mathbf{h}_{\pi,p'}^{(k)}$
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{H}, \mathbf{h}_\pi^{(k)})$ to obtain $\mathbf{H}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_\pi^{(k)})$ to obtain $\mathbf{A}_{2q,\pi}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_\pi^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ chooses $\mathbf{u}_{\pi,p}^{(k)} = (u_{\pi,p,1}, \ldots, u_{\pi,p,m})^T$, where $u_{\pi,p,i} \hookleftarrow D_\sigma^n$, for $1 \leq i \leq m$.
    $\mathcal{C}$ computes $\mathbf{r}_{\pi,p}^{(k)} = \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)} = \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p}^{(k)}$
    $\mathcal{C}$ sets $\mathbf{o}_{\pi,p}^{(k)} = H_0\left(\mathbf{r}_{\pi,p}^{(k)}, \mathbf{z}_{\pi,p}^{(k)}\right)$
    When $\mathcal{A}$ sends $\mathbf{o}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{o}_{\pi,p}^{(k)}$ to $\mathcal{A}$.
    When $\mathcal{A}$ sends $\mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ sends $\mathbf{r}_{\pi,p}^{(k)}$ and $\mathbf{z}_{\pi,p}^{(k)}$ to $\mathcal{A}$. Then, $\mathcal{C}$ computes as follows:
    for $(2 \leq p' \leq N_{CS})$ **do**
      if $\mathbf{o}_{\pi,p'}^{(k)} = H_0\left(\mathbf{r}_{\pi,p'}^{(k)}, \mathbf{z}_{\pi,p'}^{(k)}\right)$ **then** Accept
      else Abort protocol
    $\mathcal{C}$ computes $\mathbf{r}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{r}_{\pi,p'}^{(k)}$ and $\mathbf{z}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{z}_{\pi,p'}^{(k)}$
  $\mathcal{C}$ performs $\forall k \in [N_{in}]$, $\mathbf{c}_{\pi+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{r}_\pi^{(k)}, \mathbf{z}_\pi^{(k)}\right)$.
  for $(i = \pi+1, \pi+2, \ldots, w, 1, 2, \ldots, \pi-1)$ **do**
    for $(1 \leq k \leq N_{in})$ **do**
      $\mathcal{C}$ selects $\mathbf{t}_{i,p}^{(k)} = (t_{i,p,1}, \ldots, t_{i,p,m})^T$, where $t_{i,p,j} \hookleftarrow D_\sigma^n$, for $1 \leq j \leq m$.
      When $\mathcal{A}$ sends $\mathbf{t}_{i,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{i,p}^{(k)}$ to $\mathcal{A}$.
      $\mathcal{C}$ computes $\mathbf{t}_i^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{i,p'}^{(k)}$
      $\mathcal{C}$ calls $\mathsf{Lift}(\mathbf{A}, \mathbf{a}_i^{(k)})$ to obtain $\mathbf{A}_{2q,i}^{(k)} = (2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_i^{(k)} + \mathbf{q}) \in \mathcal{R}_{2q}^{2 \times m}$.
    $\mathcal{C}$ runs $\forall k \in [N_{in}]$ $\mathbf{c}_{i+1} = H_1\left(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \left\{\mathbf{A}_{2q,i}^{(k)} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}, \left\{\mathbf{H}_{2q,\pi} \cdot \mathbf{t}_i^{(k)} + \mathbf{q} \cdot \mathbf{c}_i\right\}\right)$.
  for $(1 \leq k \leq N_{in})$ **do**
    $\mathcal{C}$ chooses $b^{(k)} \hookleftarrow \{0, 1\}$.
    $\mathcal{C}$ computes $\mathbf{t}_{\pi,p}^{(k)} = \mathbf{u}_{\pi,p}^{(k)} + \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \cdot (-1)^{b^{(k)}}$, where $\mathbf{S}_{2q,\pi,p}^{(k)} = [(\mathbf{S}_{\pi,p}^{(k)})^T, 1]^T$.
    Continue with prob. $\left(M \exp\left(-\frac{\|\mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi\|^2}{2\sigma^2}\right) \cosh\left(\frac{\langle \mathbf{t}_{\pi,p}^{(k)}, \mathbf{S}_{2q,\pi,p}^{(k)} \cdot \mathbf{c}_\pi \rangle}{\sigma^2}\right)\right)^{-1}$ otherwise **Restart**.
    When $\mathcal{A}$ sends $\mathbf{t}_{\pi,p'}^{(k)}$ with $p' \in [2, N_{CS}]$, $\mathcal{C}$ returns $\mathbf{t}_{\pi,p}^{(k)}$ to $\mathcal{A}$
    $\mathcal{C}$ computes $\mathbf{t}_\pi^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)}$
  return $\sigma(\mu) = \left(\mathbf{c}_1, \{\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}\}_{k \in [N_{in}]}, \{\mathbf{h}_\pi^{(k)}\}_{k \in [N_{in}]}\right)$.

Game 4: This Game performs similar to **Game 3** but we now modify (for the signer π) the KeyGen, Algorithm 23 (step 14). The aggregate public-key as $\mathbf{a}_{\pi}^{(k)} = \sum_{p'}^{N_{CS}} H_2(\bar{\mathbf{a}}_{\pi,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{a}}_{\pi,p'}^{(k)}$. We now choose $\bar{\mathbf{a}}_{\pi}^{(k)}$ $\forall (k) \in [N_{in}]$ uniformly and randomly such $\mathbf{a}_{\pi}^{(k)} \hookleftarrow \mathcal{R}_q^2$. As in **Game 1**, it shows that $\bar{\mathbf{a}}_{\pi,p'}^{(k)}$ is uniformly random. We assume that $\forall\ h \hookleftarrow \mathcal{S}_{n,\kappa}$ where h is the output of the hash function H_2. we said that h needs to be invertible in $\mathcal{R}_q^2$, then to achieve this condition, we choose $\mathcal{S}_{n,\kappa}$ such that $\|h\|_{\infty} < \frac{1}{\sqrt{k}} \cdot q^{1/k}$ as shown in ([LS18], Corollary 1.2), with probability 1.

Let S_4 be the event where the $\mathcal{A}$ wins **Game 4** with negligible probability, that is $1 \leq \epsilon_4$. Then we claim that:

$$|\Pr[S_3] - \Pr[S_4]| \leq \epsilon_4. \tag{6.15}$$

Game 5: Changes on this game are made in the remaining public-keys $\mathbf{a}_i^{(k)}$ $(1 \leq i \leq w,\ i \neq \pi)$, $\forall k \in [N_{in}]$ which are in the list of the ring L. We know that $\mathbf{a}_i^{(k)} = \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{i,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{a}}_{i,p'}^{(k)}$ and secret-key $\mathbf{S}_i^{(k)} = \sum_{p'}^{N_{CS}} H(\bar{\mathbf{a}}_{i,p'}^{(k)}, L^{sh}) \cdot \bar{\mathbf{S}}_{i,p'}^{(k)}$, where $\bar{\mathbf{a}}_{i,p'}^{(k)} = \mathbf{A} \cdot \bar{\mathbf{S}}_{i,p'}^{(k)}$ $\forall (k, p') \in [N_{CS}] \times [N_{in}]$. We now choose uniformly random $\bar{\mathbf{a}}_{i,p'}^{(k)}$, and all $\bar{\mathbf{S}}_{i,p'}^{(k)}$'s are chosen small with coefficients in $(-2^{\gamma}, 2^{\gamma})$. When the $\bar{\mathbf{S}}_{i,p'}^{(k)}$'s are multiplied by the public parameter $\mathbf{A}$, it gives $(\bar{\mathbf{a}}_{i,p'}^{(k)})$'s that are close to uniform over $\mathcal{R}_q^2$.

By the Leftover Hash Lemma (LHL) argument (Lemma 3.5), we show that the statistical distance between the distribution of the $(\mathbf{A} \cdot \mathbf{S}_i^{(k)} \bmod q)$'s and the uniform distribution on $\mathcal{R}_q^2 \times \mathcal{R}_q^2$ is at most $N_{in} \cdot N_{CS} \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \cdot (w-1)$.

We define the event S_5 where $\mathcal{A}$ wins **Game 5** with negligible probability $N_{in} \cdot N_{CS} \cdot \frac{1}{2} \cdot \sqrt{\frac{q^{4n}}{2^{(\gamma+1)\cdot(m-1)\cdot n}}} \cdot (w-1) \leq \epsilon_5$.

$$|\Pr[S_4] - \Pr[S_5]| \leq \epsilon_5. \tag{6.16}$$

Game 6: It changes the behaviour of the random oracle H_1 in the SigGen, Algorithm 24 (step 21). The challenger chooses $\mathbf{c}_{\pi+1}$ at random from $\mathcal{S}_{n,\kappa} \subseteq \mathcal{R}_{2q}$, after

that, the answer of the random oracle is programmed H_1 $\forall k \in [N_{in}]$ as:

$$H_1\Big(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{r}_{\pi}^{(k)}, \mathbf{z}_{\pi}^{(k)}\Big) = H_1\Big(L, \mathbf{H}_{2q,\pi}^{(k)}, \mu, \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{t}_{\pi}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{\pi}, \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{t}_{\pi}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{\pi}\}\Big)$$

without verifying if the values of $\mathbf{r}_{\pi}^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{A}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)}$ and $\mathbf{z}_{\pi}^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{H}_{2q,\pi}^{(k)} \cdot \mathbf{u}_{\pi,p'}^{(k)}$ were already set $\forall p' \in [N_{CS}]$. We argue that the probability of $\mathcal{A}$ generating $\mathbf{u}_{\pi,p'}^{(k)}$, such that $\mathbf{r}_{\pi}^{(k)}$ and $\mathbf{z}_{\pi}^{(k)}$ are equal to one of previous queries is at most 2^{-n+1}. Therefore, if the SigGen (in this Game 3) and H_1 are queried Q_s and Q_1 times, respectively, then the probability of getting one collision each time is at most $N_{CS} \cdot (Q_s + Q_1)2^{-n+1}$. Additionally, the probability that a collision happens after Q_s queries is at most $N_{CS} \cdot Q_s \cdot (Q_s + Q_1)2^{-n+1}$, which is negligible (Based on [DDLL13], Lemma 3.4).

Let S_6 be the event where the $\mathcal{A}$ wins **Game 6** with negligible probability $N_{CS} \cdot Q_s \cdot (Q_s + Q_1)2^{-n+1} \leq \epsilon_6$. Then we claim that:

$$|\Pr[S_5] - \Pr[S_6]| \leq \epsilon_6. \tag{6.17}$$

Game 7: Changes in this game are made on the $\mathbf{t}_1^{(k)}, \ldots, \mathbf{t}_w^{(k)}$ from the SigGen, Algorithm 24 (step 39). This time, the challenger chooses $\forall p'$, $\mathbf{t}_{\pi,p'}^{(k)}$ now directly from the Gaussian distribution $D_{\sigma}^{n \times m}$, instead of computing it as $\mathbf{t}_{\pi}^{(k)} = \sum_{p'=1}^{N_{CS}} \mathbf{t}_{\pi,p'}^{(k)}$ with $\mathbf{t}_{\pi,p'}^{(k)} = \mathbf{u}_{\pi,p'}^{(k)} + \mathbf{S}_{2q,\pi,p'}^{(k)} \cdot \mathbf{c}_{\pi} \cdot (-1)^{b^{(k)}}$ (Based on [DDLL13], Lemma 3.5). Since $\mathbf{t}_{\pi,p'}^{(k)}$ is computed using rejection sampling (as Lemma 3.8), thus it is always sample from the Gaussian distribution $D_{\sigma}^{n}(\mu)$. This means that any adversary will have no advantage in breaking the anonymity property in this Game due to both cases have same distribution.

Let S_7 be the event where the $\mathcal{A}$ wins **Game 7** with zero probability $0 = \epsilon_7$. Then we claim that:

In this game, the view of the adversary $\mathcal{A}$ is independent of b; therefore,

$$\Pr[S_7] = \Pr[b' = b] = \tfrac{1}{2}. \tag{6.18}$$

The results of the Games are combined from (6.11), (6.12), (6.13), (6.14), (6.15), (6.16), (6.17), and (6.18) we obtain

$$\Pr[S_0] = \Pr[S_7] + \sum_{i=1}^{6} \epsilon_i,$$

by replacing (6.18) in the $\Pr[S_0]$, we have

$$\Pr[S_0] = \frac{1}{2} + \sum_{i=1}^{6} \epsilon_i, \text{ then we conclude that } \Pr\left[S_0 - \frac{1}{2}\right] = \sum_{i=1}^{6} \epsilon_i \text{ is negligible.}$$

□

Theorem 6.13 (Linkability)**.** *The L2RS-CS scheme is linkable in the random oracle model if the* $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ *problem is hard with* $\beta \leq 2\beta_v(1 + \sqrt{n}N_{in}2^{\gamma})$.

Proof. We construct a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$ to solve the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem. They run the linkability attack game (Def. 6.4) $\forall k \in [N_{in}]$, namely:

1. $\mathcal{C}$ generates using the KeyGen (Algorithm 23) all secret-keys $\mathbf{S}_i^{(k)}$'s with the corresponding public-keys $\mathbf{a}_i^{(k)}$'s, then $\mathcal{C}$ gives $\mathbf{S}_{\pi}^{(k)} = \sum_{p'}^{N_{CS}} \mathbf{S}_{\pi,p'}^{(k)}$ to the adversary $\mathcal{A}$.

2. $\mathcal{A}$ outputs two signatures $\sigma(\mu_1)$ and $\sigma'(\mu_1')$ along with their corresponding lists L and L', respectively. These signatures are successfully verified by SigVer (Algorithm 25) with their linkability tags different such that $\mathbf{h}_{\mu_1}^{(k)} \neq \mathbf{h}_{\mu_1'}^{(k)}$.

3. $\mathcal{C}$ computes the linking tags as $\mathbf{h}_{\pi}^{(k)} = \mathbf{H} \cdot \mathbf{S}_{\pi,p}^{(k)} \bmod q$, where "$\pi$" is the legitimate signer. This $\mathbf{h}_{\pi}^{(k)}$ can then be compared with the linkability tags $\mathbf{h}_{\mu_1}^{(k)}$ and $\mathbf{h}_{\mu'}^{(k)}$ that were outputted by $\mathcal{A}$ (in step 2) and one of them would be different.

4. Without loss of generality, suppose $\mathbf{h}_{\mu_1}^{(k)} \neq \mathbf{h}_{\pi}^{(k)} \bmod q$. Using the forking lemma [BN06], $\mathcal{C}$ rewinds the attacker $\mathcal{A}$ to the random oracle "H_1" query that corresponds to the SigVer of the signature $\sigma_L(\mu_1)$. $\mathcal{C}$ reruns $\mathcal{A}$ with a different response of H_1 and obtains two signatures: $\sigma(\mu_2)$ and $\sigma'(\mu_2')$. Then, we use this signature $\sigma(\mu_1)$ and $\sigma(\mu_2)$ to extract a solution to the $\mathbf{MSIS}^{\mathcal{K}}_{q,m,k,\beta}$ problem if the adversary $\mathcal{A}$ finds an efficient algorithm to unlink these signatures (as further shown in step 7).

5. The adversary $\mathcal{A}$ matches the challenge message of both signatures where $\mathbf{H}_{2q,\mu_1}^{(k)}, \mathbf{A}_{2q,w,\mu_1}^{(k)}$ and $\mathbf{q}$ are fixed. Subsequently, we obtain the following relations:

$$\begin{aligned} \mathbf{A}_{2q,w,\mu_1}^{(k)} \cdot \mathbf{t}_{w,\mu_1}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{w,\mu_1} &= \mathbf{A}_{2q,w,\mu_1}^{(k)} \cdot \mathbf{t}_{w,\mu_2}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{w,\mu_2} \\ \mathbf{H}_{2q,\mu_1}^{(k)} \cdot \mathbf{t}_{w,\mu_1}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{w,\mu_1} &= \mathbf{H}_{2q,\mu_1}^{(k)} \cdot \mathbf{t}_{w,\mu_2}^{(k)} + \mathbf{q} \cdot \mathbf{c}_{w,\mu_2} \end{aligned} \tag{6.19}$$

These expressions can be represented as:

$$\begin{aligned} \mathbf{A}_{2q,w,\mu_1}^{(k)} \cdot (\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) &= \mathbf{q} \cdot (\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1}) \\ \mathbf{H}_{2q,\mu_1}^{(k)} \cdot (\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) &= \mathbf{q} \cdot (\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1}) \end{aligned} \tag{6.20}$$

Reducing (6.20) mod$q$ with $(\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1}) \neq \mathbf{0} \bmod 2$, it results in:

$$\begin{aligned} \mathbf{A}_{2q,w,\mu_1}^{(k)} \cdot (\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) &= \mathbf{0} \bmod q \\ \mathbf{H}_{2q,\mu_1}^{(k)} \cdot (\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) &= \mathbf{0} \bmod q \end{aligned} \tag{6.21}$$

We recall the definition of $\mathbf{H}_{2q,\mu_1}^{(k)}$ and $\mathbf{A}_{2q,w,\mu_1}^{(k)}$ in SigGen, Algorithm 24 (steps 8 and 9), respectively, then we have:

$$\begin{aligned} \left(2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_{\mu_1}^{(k)} + \mathbf{q}\right) \cdot (\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) &= \mathbf{0} \bmod q \\ \left(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\mu_1}^{(k)} + \mathbf{q}\right) \cdot (\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) &= \mathbf{0} \bmod q \end{aligned} \tag{6.22}$$

Afterwards, if we define $(\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)})$ as:

$$\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)} = \begin{pmatrix} \mathbf{t}_{w,\mu_1}^{\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime(k)} \\ \mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)} \end{pmatrix} \in \mathcal{R}_q^m \tag{6.23}$$

Then, by replacing (6.22) in (6.23), it results in:

$$\begin{aligned} \left(2 \cdot \mathbf{A}, -2 \cdot \mathbf{a}_{\mu_1}^{(k)} + \mathbf{q}\right) \cdot \begin{pmatrix} \mathbf{t}_{w,\mu_1}^{\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime(k)} \\ \mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)} \end{pmatrix} &= \mathbf{0} \bmod q \\ \left(2 \cdot \mathbf{H}, -2 \cdot \mathbf{h}_{\mu_1}^{(k)} + \mathbf{q}\right) \cdot \begin{pmatrix} \mathbf{t}_{w,\mu_1}^{\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime(k)} \\ \mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)} \end{pmatrix} &= \mathbf{0} \bmod q. \end{aligned} \tag{6.24}$$

Since we reduce (6.24) to mod$q$, q is odd, and $\mathbf{H} \cdot (\mathbf{t}_{w,\mu_1}^{\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime(k)}) = \mathbf{h}_{\mu_1}^{(k)} \cdot (\mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)}) \bmod q$. We claim that $(\mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)}) \neq \mathbf{0}$ is invertible in $\mathcal{R}_q$. To show this, we have $(\mathbf{c}_{w,\mu_2} - \mathbf{c}_{w,\mu_1}) \neq \mathbf{0} \bmod 2$. Therefore, using (6.20), we conclude $(\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) \neq \mathbf{0} \bmod 2$, and $(\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}) \neq \mathbf{0} \bmod 2q$. Additionally, we know that $\|\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}\|_\infty < q/2$ and $\|\mathbf{t}_{w,\mu_1}^{(k)} - \mathbf{t}_{w,\mu_2}^{(k)}\|_2 < 2\beta_v$ as SigVer, Algorithm 25, which implies that $(\mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)}) \neq \mathbf{0} \bmod q$. Furthermore, since $2\beta_v < \frac{1}{\sqrt{k}} \cdot q^{1/k}$ as in ([LS18], Corollary 1.2), then $(\mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)})$ is invertible in $\mathcal{R}_q$. After that, we establish $\mathbf{h}_{\mu_1}^{(k)}$ as:

$$\mathbf{h}_{\mu_1}^{(k)} = \mathbf{H} \cdot \frac{(\mathbf{t}_{w,\mu_1}^{\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime(k)})}{(\mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)})} \bmod q \tag{6.25}$$

6. Then, $\bar{\mathbf{S}}_{\mu_1}^{(k)}$ is well-defined since $(\mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)})$ is invertible in $\mathcal{R}_q$, then we said that:

$$\bar{\mathbf{S}}_{\mu_1}^{(k)} \triangleq \frac{(\mathbf{t}_{w,\mu_1}^{\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime(k)})}{(\mathbf{t}_{w,\mu_1}^{\prime\prime(k)} - \mathbf{t}_{w,\mu_2}^{\prime\prime(k)})} \bmod q \tag{6.26}$$

7. By using $\mathbf{S}_\pi^{(k)}$ from (step 3), we consider two cases, when $\bar{\mathbf{S}}_{\mu_1}^{(k)} = \mathbf{S}_\pi^{(k)} \bmod q$ and $\bar{\mathbf{S}}_{\mu_1}^{(k)} \neq \mathbf{S}_\pi^{(k)} \bmod q$. These cases are analysed as follows:

   (a) **Case 1:** If $\bar{\mathbf{S}}_{\mu_1}^{(k)} = \mathbf{S}_\pi^{(k)} \bmod q$, we show that $\mathbf{h}_{\mu_1}^{(k)} = -2 \cdot \mathbf{H} \cdot \bar{\mathbf{S}}_{\mu_1}^{(k)} = -2 \cdot \mathbf{H} \cdot \mathbf{S}_\pi^{(k)} = \mathbf{h}_\pi^{(k)} \bmod q$, which is a contradiction with respect to the above assumption (step 4), where $\mathbf{h}_{\mu_1}^{(k)} \neq \mathbf{h}_\pi^{(k)} \bmod q$.

(b) **Case 2:** When $\bar{\mathbf{S}}_{\mu_1}^{(k)} \neq \mathbf{S}_{\pi}^{(k)} \bmod q$, we have $\mathbf{a}_{\mu_1}^{(k)} = \mathbf{A} \cdot \bar{\mathbf{S}}_{\mu_1}^{(k)} = \mathbf{A} \cdot \mathbf{S}_{\pi}^{(k)} = \mathbf{a}_{\pi}^{(k)} \bmod q$, then using (6.26) we have:

$$\mathbf{A} \cdot \frac{(\mathbf{t}_{w,\mu_1}'^{(k)} - \mathbf{t}_{w,\mu_2}'^{(k)})}{(\mathbf{t}_{w,\mu_1}''^{(k)} - \mathbf{t}_{w,\mu_2}''^{(k)})} = \mathbf{A} \cdot \mathbf{S}_{\pi}^{(k)} \bmod q \iff$$

$$\mathbf{A} \cdot (\mathbf{t}_{w,\mu_1}'^{(k)} - \mathbf{t}_{w,\mu_2}'^{(k)}) = \mathbf{A} \cdot \mathbf{S}_{\pi}^{(k)} \cdot (\mathbf{t}_{w,\mu_1}''^{(k)} - \mathbf{t}_{w,\mu_2}''^{(k)}) \bmod q \iff$$

$$\mathbf{A} \cdot \Big((\mathbf{t}_{w,\mu_1}'^{(k)} - \mathbf{t}_{w,\mu_2}'^{(k)}) - \mathbf{S}_{\pi}^{(k)} \cdot (\mathbf{t}_{w,\mu_1}''^{(k)} - \mathbf{t}_{w,\mu_2}''^{(k)}) \Big) = 0 \bmod q$$

then we let this small non-zero vector $\mathbf{v} \triangleq \Big((\mathbf{t}_{w,\mu_1}'^{(k)} - \mathbf{t}_{w,\mu_2}'^{(k)}) - \mathbf{S}_{\pi}^{(k)} \cdot (\mathbf{t}_{w,\mu_1}''^{(k)} - \mathbf{t}_{w,\mu_2}''^{(k)}) \Big)$ be the output of the adversary $\mathcal{A}$, and this vector is a solution to the $\mathbf{MSIS}_{q,m,k,\beta}^{\mathcal{K}}$ problem with respect to the public parameter $\mathbf{A} \in \mathcal{R}_q^{2\times(m-1)}$, where $\beta = \|\mathbf{v}\|$ and $\|\mathbf{v}\| \leq 2\beta_v(1 + \sqrt{n}N_{in}2^{\gamma})$.

□

Theorem 6.14 (Non-Slanderability)**.** *For any linkable ring signature, if it satisfies unforgeability and linkability, then it satisfies non-slanderability.*

Proof. Let's suppose there is a non-slanderability adversary $\mathcal{A}_{Sland}$ who is given $\mathbf{pk}_i, \mathbf{sk}_i, i \neq \pi,$ and $i \in \{1, \dots w\}$, and he produces a valid signature $\sigma'(\mu)$ with linkability tag $\mathbf{h}_{\sigma'(\mu)}$ which is equal to $\mathbf{h}_{\sigma(\mu)}$, $\sigma(\mu)$ being the legitimate signature generated with respect to $\mathbf{sk}_{\pi}$. This means that $\mathcal{A}_{Sland}$ can create a signature with the linkability tag $\mathbf{h}_{\sigma(\mu)}$ without knowing $\mathbf{sk}_{\pi}$. The adversary can also compute a valid $\sigma''(\mu)$ with $\mathbf{sk}_i$, $i \neq \pi,$ and $i \in \{1, \dots w\}$ for which $\mathbf{h}_{\sigma''(\mu)} \neq \mathbf{h}_{\sigma'(\mu)}$. We give $(\sigma''(\mu), \sigma'(\mu))$ to the forger, which can turn it to an $\mathbf{MSIS}_{q,m,k,\beta}^{\mathcal{K}}$ solution. In particular, it will be computationally secure when two valid signatures created by different users are unlinked using the L2RS-CS algorithms. An adversary $\mathcal{A}$ will break these properties with negligible probability as demonstrated in Theorems (6.11 and 6.13), and with these probabilities the $\mathcal{A}$ will find a $\mathbf{MSIS}_{q,m,k,\beta}^{\mathcal{K}}$ solution. Therefore, non-slanderability is implied by the definitions of the unforgeability (Def. 6.2) and linkability (Def. 6.4), and their security analyses, respectively. □

Corollary 6.15 (Non-Slanderability)**.** *The L2RS-CS scheme is non-slanderable under the assumptions of Theorem 6.11 and Theorem 6.13.*

6.5 Performance Analysis

Remark 6.16. This research project did not consider the implementation of the scheme L2RS-CS, as a result there is not run time analysis. The project only evaluates the signature and key sizes of the proposed construction.

After consolidating the conditions (shown in Table 6.2) from the correctness and security analyses, which were discussed in earlier sections, we chose the optimal parameters of our L2RS-CS with Hermite factor $\delta = 1.0045$ and security parameter $\lambda = 128$ bits. This evaluation follows the analysis of the attack on SIS from [MR09] that we use to estimate secure values for the parameters. In our experiment, we then set the polynomial ring degree $n = 2^8$ instead of $n = 2^7$ since it yields a shorter signature size and a optimal value for $\log_2(q) = 58$, as illustrated in Figures 6.1.**a** and 6.1.**b**, respectively. Consequently, we selected the number of ring elements of the matrices of the PP to be $m = 23$. This also allowed us to determine the Hamming weight of each challenge vector ($\kappa = 23$), the Gaussian standard deviation ($\sigma = 188416$), and the $\log \beta = 38.9$ (which also solves the lattice assumption). With these results, we attained a signature size of 1.26 MB with the cosigner's pair of keys ($|\mathbf{sk}|$=10 KB, $|\mathbf{pk}|$=3.6 KB). This evaluation was restricted to ring size $w = 100$, $N_{in} = 1$ and $N_{out} = 1$, which was compared with existing lattice-based TRS schemes [BS13, CLRS10], as shown in Table 6.1. In a different experiment, we analysed how the signature size grows with the ring size and N_{CS} cosigners while comparing our L2RS-CS with [BS13, CLRS10]. Despite all approaches growing linearly with the ring size w, our L2RS-CS scheme generated shorter signature sizes than previous constructions (Figure 6.2.**a**). In terms of the N_{CS} cosigners (Figure 6.2.**b**), our proposed scheme achieved constant time and provided better signature sizes than other lattice-based TRS, in particular when $N_{CS} > 2^2$.

TABLE 6.2: List conditions for MIMO.L2RS-CS's performance analysis

Order	Condition	Description
1	$\mathcal{S}_{n,\kappa} = \binom{n}{\kappa} \cdot 2^{\kappa} > 2^{\lambda}$	The challenge space
2	$\gamma \geq \log(nk)$	n/a
3	$\|Sc\| \leq \sqrt{mnk}2^{\gamma}$	Rejection sampling
4	$\sigma \geq \alpha\|Sc\|$	Rejection sampling from BLISS, $\alpha = \{0.5, 0.55, 0.7, 1\}$
5	$\beta_v = \eta\sigma\sqrt{nm}$	SigVer and Correctness, with $(\eta = 1.1)$
6	$N_{in}N_{CS}\frac{1}{2}\sqrt{\frac{q^{4n}}{2^{(\gamma+1)(m-1)n}}} \leq 2^{-\lambda}$	Left Over Hash Lemma, with security parameter $(\lambda = 128)$
7	$\beta \leq 2\beta_v + 2\beta_v\sqrt{n}2^{\gamma}$	β from the linkability analysis
8	$\min\left(q, 2^{2\sqrt{2n\log(q)\log(\delta)}}\right) > \beta$	Shortest vector length ([MR09]-P156), with Hermite factor $(\delta = 1.0045)$

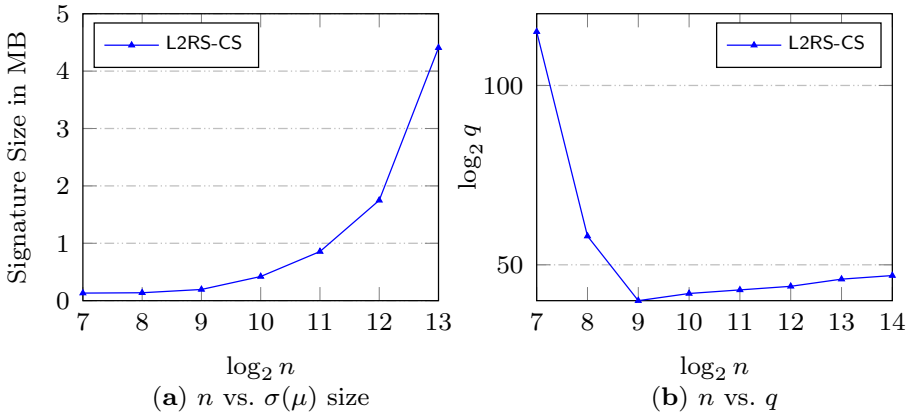
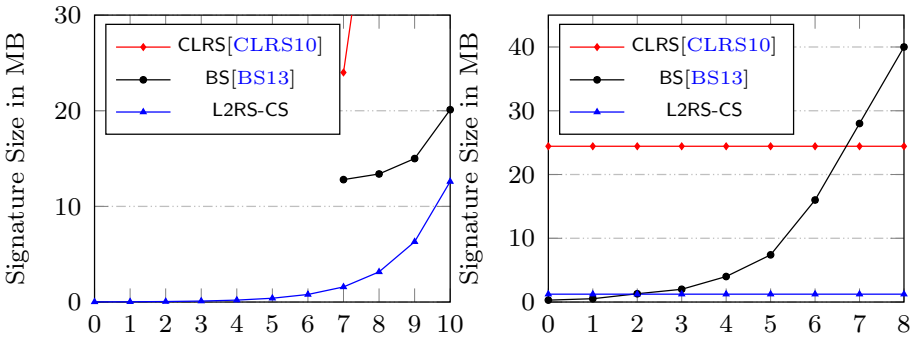


FIGURE 6.1: Analysis of signature size and q versus n with fixed $w = 11$.

Finally, we also explored how the signature size grows when selecting regular values for N_{in} and N_{out}. We also set $w = 11$ since it currently offers secure anonymity, according to Monero's blockchain[1]. The outcome of this evaluation is presented in Table 6.3. This reveals that the signature size grows linearly with the N_{in} for any $N_{CS} > 2$.

[1]https://moneroblocks.info/



(**a**) $\sigma(\mu)$ size vs. w ($N_{CS} = 50$) (**b**) $\sigma(\mu)$ size vs. N_{CS} ($w = 100$)

FIGURE 6.2: Analysis of signature size versus w and N_{CS}.

TABLE 6.3: Size estimation for L2RS-CS for any $N_{CS} \geq 2$

L2RS-CS	$(N_{in}, N_{out}) = (1, 2)$	$(N_{in}, N_{out}) = (2, 2)$	$(N_{in}, N_{out}) = (3, 2)$
Signature size ($w = 11$)	$\approx$ 138.8 KB	$\approx$ 289.8 KB	$\approx$ 452.9 KB
Private-key size	$\approx$ 10.6 KB	$\approx$ 11.1 KB	$\approx$ 11.6 KB
Public-key size	$\approx$ 3.6 KB	$\approx$ 3.8 KB	$\approx$ 4 KB

6.6 Summary

In this chapter, we presented the new post-quantum cryptographic mechanism, called Lattice-based Linkable Ring Signature with Co-Signing (L2RS-CS), which offers a distributed authorisation feature to protect electronic wallets. A novel security model for L2RS-CS was also formalised that captures the security and privacy requirements to protect transactions in applications to blockchain cryptocurrency protocols, such as the RingCT. To address key-generation security concerns, and to support compression of keys and signatures, the L2RS-CS incorporates a distributed key generation along with a solid public-key aggregation. Finally, we proved the security of our constructed L2RS-CS in the random oracle model and the standard lattice-based Module-SIS hardness assumption.

Chapter 7

Conclusions and Future Research

This thesis points out the importance of constructing post-quantum cryptographic primitives to deal with the predicted quantum attacks that pose a threat to traditional cryptography. All the proposed schemes in this thesis used the post-quantum lattice-based cryptography, which is considered one of the most feasible alternatives to overcome the threat of quantum attacks. The *Linkable Ring Signature* and *Threshold Signature* are the core schemes that were instantiated by the use of lattice-based cryptography. These schemes currently have a wide application in today's digital world, cryptocurrencies in particular. Following this trend, this research utilised the above proposed schemes to extend, devise and construct a post-quantum cryptocurrency protocol. The study focused on the Ring Confidential Transaction (or RingCT) that is widely used by Monero's cryptocurrency application. Additionally, with the *Threshold Signature*, an authorisation setting is incorporated to increase the level of security of digital wallets. It is expected

that the result of this thesis motivates future studies within the area of applied cryptography.

The first contribution of this thesis, the design and construction of the post-quantum Lattice-based Linkable Ring Signature (L2RS), was described in Chapter 4. This scheme achieved unconditional anonymity, meaning that even a powerful adversary with unlimited computational resources and time, would be incapable of breaking into this property. Other properties such as the unforgeability, linkability and non-slanderability are computationally secure under the standard Ring Short Integer Solution (Ring-SIS) lattice hardness assumption. Moreover, to extend the L2RS scheme, a novel post-quantum cryptocurrency protocol (the LRCT) was devised and constructed, inheriting the post-quantum security guarantees from the L2RS. The performance results illustrated that signature size grows linearly with the number of users in the ring. However, these proposals had some limitations. For example, they only enable transfers from a Single Output wallet to a Single Output wallet (SISO). In the RingCT model, signatures are *one-time*. If one then needs to receive change after making a transfer, a new output wallet is required, so this reveals the importance of supporting multiple input and output in digital wallets. Furthermore, having more than one output wallet also introduces a new security problem like the negative output amount (or out-of-range) attack [BBB+18], where an adversary can create extra coins (of free money). Although this attack was addressed in the previous RingCT versions [Noe15] by employing a range proof technique, it is post-quantum insecure. These limitations were in part the motivation for the subsequent contributions of this thesis.

Another post-quantum primitive was later presented in Chapter 5, addressing the limitations of the prior proposals. A second version of the Lattice-based Ring Confidential Transactions (LRCT) was designed and constructed, supporting transactions of Multiple-Input and Multiple-Output wallets (MIMO). This scheme extended the SISO.LRCT cryptocurrency protocol, as well as its underlying building block, the L2RS signature. The MIMO.LRCT also inherited the post-quantum security guarantees from the SISO.LRCT and L2RS; in other words, the hardness of the lattice mathematical assumption (the Ring-SIS) was added to secure the

balance security property. The MIMO.LRCT captured the amount privacy attacks and user anonymity which were then proved to be unconditionally secure. The performance evaluation showed preliminary parameters and signature sizes that can be referred for future research. The result of this evaluation demonstrated that the signature size also grows linearly with the number of users in the ring. In the end, this study has served as a motivation for further studies. For instance, in a recent work, [EZS+19] produced a significant improvement of the lattice-based RingCT which resulted in practical signatures sizes.

A further development was thereafter proposed to improve the security of the LRCT cryptocurrency protocol. It included an authorisation model for the expenditure of digital wallets by segregating its corresponding secret-keys. This improvement was introduced in Chapter 6, where the new scheme was denominated as Lattice-based Linkable Ring Signature with Co-Signing (L2RS-CS), which offers a distributed authorisation feature to protect such wallets. A novel model associated with this construction (the LRS-CS) that can be utilised in further research (i.e. it can be instantiated by different security assumptions) was also proposed. The building blocks employed in this scheme included, Threshold Ring Signatures (TRS), Distributed Key Generation (DKG) and key aggregation. These methods led not only to support the security of the scheme but also to achieve a certain level of compression of the cryptographic keys that are linked to the wallets. The security of the L2RS-CS was proved in the random oracle model with the lattice-based Module-SIS hardness assumption, and the anonymity property was inherited from the previous constructions of the (L2RS). The outcomes of the performance evaluation demonstrated that the signature size of L2RS-CS also grows linearly with the number of users in the ring, and the number of input wallets N_{in}. However, the analysis showed that the signature size is constant and independent of the number of cosigners N_{CS} while achieving better signature sizes in comparison with prior and similar constructions.

7.1 Future Research

There are still possible research directions after having constructed several post-quantum cryptographic primitives within this thesis. They might cover certain types of improvements, new features and perhaps new applications.

- **Performance**: all the schemes introduced in this thesis (L2RS, LRCT and L2RS-CS) are subject to improving signature size. Since the size of these schemes grows linearly with the number of users in the ring, it would be interesting to study how this size can somehow be improved (in constant or logarithmic size, for instance). The LRCT cryptocurrency protocol could be further improved, particularly to split the transferred amount or using amortisation techniques to reduce the signature size.

- **Applications**: whilst this thesis did evaluate the application in cryptocurrencies; however, the L2RS and L2RS-CS have the potential for other privacy preserving protocol applications such as supply chain [MQ18], e-voting [KY19], or direct anonymous attestation that has a direct application in trusted computing [TW05].

- **Security and Functionalities**: besides the security properties examined in the proposed constructions, there are other possible aspects that can be analysed. For example, forward security would protect past signatures or transactions if the secret keys had been compromised. The L2RS-CS has the possibility for being extended to t-out-of-N_{CS} (with $t < N_{CS}$) since the current threshold signature evaluates N_{CS}-out-of-N_{CS} [Bra19]. This extension would bring a robustness property where $t-1$ malicious parties cannot prevent the protocol from producing a valid signature, so this enables the schemes to be more resilient and tolerant to failures.

Bibliography

[AABN02] Michel Abdalla, Jee An, Mihir Bellare, and Chanathip Namprempre. From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security. In *EUROCRYPT*, pages 418–433. Springer, 2002.

[ABB10] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient Lattice (H) IBE in the Standard Model. In *EUROCRYPT*, volume 6110, pages 553–572. Springer, 2010.

[ACST06] Man Ho Au, Sherman S. M. Chow, Willy Susilo, and Patrick P. Tsang. Short Linkable Ring Signatures Revisited. In *EuroPKI*, pages 101–115. Springer, 2006.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing - STOC '97*, pages 284–293, New York, 1997. ACM Press.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems. In *STOC*, pages 99–108. ACM, 1996.

[Alo18] Kurt Alonso. Zero to Monero: Multisig Chapter. *https://github.com/SarangNoether/zero-to-monero/blob/master/multisig_chapter-1-0.pdf*, 2018.

[ALSY06] Man Ho Au, Joseph K Liu, Willy Susilo, and Tsz Hon Yuen. Secure ID-based linkable and revocable-iff-linked ring signature with

constant-size construction. In *INDOCRYPT*, volume 4329, pages 364–378. Springer, 2006.

[ALSY07] Man Ho Au, Joseph K. Liu, Willy Susilo, and Tsz Hon Yuen. Certificate Based (Linkable) Ring Signature. In *ISPEC*, pages 79–92. Springer, 2007.

[AMBB+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. Springer, 2013.

[ATBS15] Wilson Abel Alberto Torres, Nandita Bhattacharjee, and Bala Srinivasan. Privacy-preserving biometrics authentication systems using fully homomorphic encryption. *International Journal of Pervasive Computing and Communications*, 11(2):151–168, 6 2015.

[ATKS+19] Wilson Alberto Torres, Veronika Kuchta, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Jacob Cheng. Lattice RingCT v2.0 with Multiple Input and Output Wallets. In *ACISP*, pages 156–175. Springer, 2019.

[ATSS+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.

[ATSSK20] Wilson Alberto Torres, Ron Steinfeld, Amin Sakzad, and Veronika Kuchta. Post-Quantum Linkable Ring Signature Enabling Distributed Authorised Ring Confidential Transactions in Blockchain. In *Cryptology ePrint Archive: Report 2020/1121*, 2020.

[BBB+18] Benedikt Bunz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short Proofs for

Confidential Transactions and More. In *IEEE Symposium on Security and Privacy*. IEEE, 2018.

[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.

[BdM93] Josh Benaloh and Michael de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Signatures. In *EUROCRYPT*, pages 274–285. Springer, 1993.

[BDN18] Dan Boneh, Manu Drijvers, and Gregory Neven. Compact Multi-signatures for Smaller Blockchains. In *ASIACRYPT*, pages 435–464. Springer, 12 2018.

[BHS18] Carsten Baum, Lin Huang, and Oechsner Sabine. Towards Practical Lattice-Based One-Time Linkable Ring Signatures. *https://eprint.iacr.org/2018/107*, 2018.

[BK10] Zvika Brakerski and Yael Tauman Kalai. A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model. *https://eprint.iacr.org/2010/086*, 2010.

[BL17a] Daniel J. Bernstein and Tanja Lange. Post-quantum cryptography. *Nature*, 549(7671):188–194, 9 2017.

[BL17b] Daniel J Bernstein and Tanja Lange. Post-quantum cryptography-dealing with the fallout of physics success. *IACR Cryptology ePrint Archive*, 2017:314, 2017.

[BLM17] Johannes Buchmann, Kristin Lauter, and Michele Mosca. Postquantum Cryptography State-of-the-Art. *IEEE Symposium on Security and Privacy*, 15(4):12–13, 2017.

[BLM18] Johannes Buchmann, Kristin Lauter, and Michele Mosca. Postquantum Cryptography, Part 2. *IEEE Symposium on Security and Privacy*, 16(5):12–13, 9 2018.

[BLP+13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 575–584. ACM, 2013.

[BN06] Mihir Bellare and Gregory Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *CCS*, page 390. ACM, 2006.

[Bol03] Alexandra Boldyreva. Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme. In *PKC*, pages 31–46. Springer, 2003.

[Boy13] Xavier Boyen. Attribute-Based Functional Encryption on Lattices. In *TCC*, volume 7785, pages 122–142. Springer, 2013.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73, 1993.

[BR96] Mihir Bellare and Phillip Rogaway. The Exact Security of Digital Signatures-How to Sign with RSA and Rabin. pages 399–416. Springer, Berlin, Heidelberg, 1996.

[Bra19] Luís T. A. N. Brandão. Towards Standardization of Threshold Schemes at NIST. In *Proceedings of ACM Workshop on Theory of*

Implementation Security Workshop, pages 29–29, New York, 2019. ACM Press.

[BS13] Slim Bettaieb and Julien Schrek. Improved Lattice-Based Threshold Ring Signature Scheme. In *PQCRYPTO*, pages 34–51. Springer, 2013.

[BSS02] Emmanuel Bresson, Jacques Stern, and Michael Szydlo. Threshold Ring Signatures and Applications to Ad-hoc Groups. In *CRYPTO*, pages 465–480. Springer, 2002.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. *Theory of Cryptography*, pages 253–273, 2011.

[Byt15] Bytecoin Team. Aggregate Addresses in CryptoNote: Towards Efficient Privacy. *https://bytecoin.org/static/files/docs/aggregate-addresses.pdf*, 2015.

[CCJ+16] Lily Chen, Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. *Report on post-quantum cryptography*. NIST, 2016.

[CELR18] Mauro Conti, Sandeep Kumar E, Chhagan Lal, and Sushmita Ruj. A Survey on Security and Privacy Issues of Bitcoin. *IEEE Communications Surveys and Tutorials*, 20(4):3416 – 3452, 2018.

[CHGL19] Jiangshan Chen, Yupu Hu, Wen Gao, and Hongmei Liang. Lattice-based Threshold Ring Signature with Message Block Sharing. *TIIS*, 13(2):1003–1019, 2019.

[CL06] Melissa Chase and Anna Lysyanskaya. On Signatures of Knowledge. In *CRYPTO*. Springer, 2006.

[CLRS10] Pierre-Louis Cayrel, Richard Lindner, Markus Rückert, and Rosemberg Silva. A Lattice-Based Threshold Ring Signature Scheme. In *LATINCRYPT*, pages 255–272. Springer, 2010.

[CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. In *ASIACRYPT*, pages 1–20. Springer, 2011.

[CVH91] David Chaum and Eugène Van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265. Springer, 1991.

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *CRYPTO*, pages 40–56. Springer, 2013.

[DF89] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In *CRYPTO*, pages 307–315. Springer, 1989.

[DH76] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 11 1976.

[DHS03] V. Daza, J. Herranz, and G. Saez. Some protocols useful on the Internet from threshold signature schemes. In *14th International Workshop on Database and Expert Systems Applications*, pages 359–363. IEEE, 2003.

[DLL+18] Léo Ducas, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS – Dilithium: Digital Signatures from Module Lattices. In *IACR Transactions on Symmetric Cryptology*, pages 238–268, 2018.

[dPLNS17] Rafaël del Pino, Vadim Lyubashevsky, Gregory Neven, and Gregor Seiler. Practical Quantum-Safe Voting from Lattices. In *CCS*, pages 1565–1581. ACM Press, 2017.

[EBS16] Rachid El Bansarkhani and Jan Sturm. An Efficient Lattice-Based Multisignature Scheme with Applications to Bitcoins. In *CANS*, pages 140–155. Springer, 2016.

[ElG84] Taher ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *CRYPTO*, pages 10–18. Springer, 1984.

[Elg85] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[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 *CCS*, pages 567–584. ACM Press, 2019.

[FS86] Amos Fiat and Adi Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO*, pages 186–194. Springer, 1986.

[FS07] Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In *PKC*, volume 4450, pages 181–200. Springer, 2007.

[Fuj11] Eiichiro Fujisaki. Sub-linear Size Traceable Ring Signatures without Random Oracles. In *CT-RSA*, volume 11, pages 393–415. Springer, 2011.

[GBGN17] Steven Goldfeder, Joseph Bonneau, Rosario Gennaro, and Arvind Narayanan. Escrow Protocols for Cryptocurrencies: How to Buy Physical Goods Using Bitcoin. In *Financial Cryptography*, pages 321–339. Springer, 2017.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *CRYPTO*, page 112. Springer, 1997.

[GGH13] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate Multilinear Maps from Ideal Lattices. pages 1–17. Springer, Berlin, Heidelberg, 2013.

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate Indistinguishability Obfuscation and Functional Encryption for All Circuits. *SIAM Journal on Computing*, 45(3):882–929, 1 2016.

[GGN16] Rosario Gennaro, Steven Goldfeder, and Arvind Narayanan. Threshold-Optimal DSA/ECDSA Signatures and an Application to Bitcoin Wallet Security. In *ACNS*, pages 156–174. Springer, 2016.

[GJKR96] Rosario Gennaro, Stanisław Jarecki, Hugo Krawczyk, and Tal Rabin. Robust Threshold DSS Signatures. In *EUROCRYPT*, pages 354–371. Springer, 1996.

[GJKR03] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure Applications of Pedersen's Distributed Key Generation Protocol. In *CT-RSA*, pages 373–390. Springer, 2003.

[GJKR07] Rosario Gennaro, Stanislaw Jarecki, Hugo Krawczyk, and Tal Rabin. Secure Distributed Key Generation for Discrete-Log Based Cryptosystems. *Journal of Cryptology*, 20(1):51–83, 1 2007.

[GK96] Lov K. Grover and Lov K. A fast quantum mechanical algorithm for database search. In *STOC*, pages 212–219. ACM, 1996.

[GMR88] Shafi Goldwasser, Silvio Micali, and Ronald L Rivest. A digital signature scheme secure against adaptive chosen-message attacks. 17(2), 1988.

[GN18] Brandon Goodell and Sarang Noether. Thring Signatures and their Applications to Spender-Ambiguous Digital Currencies. In *Cryptology ePrint Archive: Report 2018/774*, 2018.

[Go09] Craig Gentry and others. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, page 197. ACM, 2008.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288. Springer, 1998.

[HPS01] Jeffrey Hoffstein, Jill Pipher, and Joseph Silverman. NSS: An NTRU lattice-based signature scheme. In *EUROCRYPT*, pages 211–228. Springer, 2001.

[HPSS08] Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[HZW06] Xin-Yi Huang, Fu-Tai Zhang, and Wei Wu. Identity-based ring signcryption scheme. *Dianzi Xuebao(Acta Electronica Sinica)*, 34(2):263–266, 2006.

[Kat10] Jonathan Katz. *Digital signatures*. Springer, 2010.

[KG13] Cameron F Kerry and Patrick D Gallagher. Digital signature standard (DSS). *FIPS PUB*, pages 184–186, 2013.

[KKM14] Philip Koshy, Diana Koshy, and Patrick McDaniel. An Analysis of Anonymity in Bitcoin Using P2P Network Traffic. In *Financial Cryptography*, pages 469–485. Springer, 2014.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC, 2nd edition, 2014.

[KY19] Alexandra Kugusheva and Yury Yanovich. Ring Signature-Based Voting on Blockchain. In *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*, pages 70–75, New York, 12 2019. ACM.

[LAH+14] Joseph K. Liu, Man Ho Au, Xinyi Huang, Willy Susilo, Jianying Zhou, and Yong Yu. New Insight to Preserve Online Survey Accuracy and Privacy in Big Data Era. In *ESORICS*, pages 182–199. Springer, 2014.

[LASZ14] Joseph K Liu, Man Ho Au, Willy Susilo, and Jianying Zhou. Linkable ring signature with unconditional anonymity. *IEEE Transactions on Knowledge and Data Engineering*, 26(1):157–165, 2014.

[Lau17] Kristin Lauter. Postquantum Opportunities: Lattices, Homomorphic Encryption, and Supersingular Isogeny Graphs. *IEEE Symposium on Security and Privacy*, 15(4):22–27, 2017.

[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 12 1982.

[LLM+07] Dennis Y. W. Liu, Joseph K. Liu, Yi Mu, Willy Susilo, and Duncan S. Wong. Revocable Ring Signature. *Journal of Computer Science and Technology*, 22(6):785–794, 11 2007.

[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.

[LLNW18] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-Based Zero-Knowledge Arguments for Integer Relations. In *CRYPTO*. Springer, 2018.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *International Colloquium on Automata, Languages, and Programming*, pages 144–155. Springer, 2006.

[LM08] Vadim Lyubashevsky and Daniele Micciancio. Asymptotically efficient lattice-based digital signatures. *Theory of Cryptography*, pages 37–54, 2008.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23. Springer, 2010.

[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.

[LSW06] Joseph K. Liu, Willy Susilo, and Duncan S. Wong. Ring Signature with Designated Linkability. In *IWSEC*, pages 104–119. Springer, 2006.

[LW05a] Joseph K. Liu and Duncan S. Wong. Linkable Ring Signatures: Security Models and New Schemes. In *ICCSA*, pages 614–623. Springer, 2005.

[LW05b] Joseph K. Liu and Duncan S. Wong. On the Security Models of (Threshold) Ring Signature Schemes. In *ICISC*, pages 204–217. Springer, 2005.

[LWLW06] Joseph K Liu, Duncan S Wong, J K Liu, and D S Wong. Enhanced security models and a generic construction approach for linkable ring signature. *Int. J. Found. of Comput. Sci.*, 17(6):1403–1422, 2006.

[LWW04a] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. A Separable Threshold Ring Signature Scheme. In *ICISC*, pages 12–26. Springer, 2004.

[LWW04b] Joseph K Liu, Victor K Wei, and Duncan S Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In *ACISP*, 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*. Springer, 2012.

[Mac15] Surae; Monero Core Team Mackenzie, Adam; Noether. Improving Obfuscation in the CryptoNote Protocol. *https://lab.getmonero.org/pubs/MRL-0004.pdf*, 2015.

[Max15] Greg Maxwell. Confidential Transactions. *https://xiph.org/confidential_values.txt*, 2015.

[Mic07] Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

[MMB17] John Mulholland, Michele Mosca, and Johannes Braun. The Day the Cryptography Dies. *IEEE Symposium on Security and Privacy*, 15(4):14–21, 2017.

[Mon14] Monero. About Monero — Monero - secure, private, untraceable. *https://getmonero.org/resources/about/*, 2014.

[MPSW19] Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple Schnorr multi-signatures with applications to Bitcoin. *Designs, Codes and Cryptography*, 87(9):2139–2164, 9 2019.

[MQ18] Mark H. Meng and Yaou Qian. The Blockchain Application in Supply Chain Management: Opportunities, Challenges and Outlook. Technical report, 10 2018.

[MR09] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[Nak09] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. *https://bitcoin.org/bitcoin.pdf*, 2009.

[NNM14] Surae Noether, Sarang Noether, and Adam Mackenzie. A Note on Chain Reactions in Traceability in CryptoNote 2.0. *https://lab.getmonero.org/pubs/MRL-0001.pdf*, 2014.

[Noe14] Shen Noether. CryptoNote whitepaper review by Monero - CryptoNote News. *https://cryptonote.org/news/2014/7/15/cryptonote-whitepaper-review-by-monero*, 2014.

[Noe15] Shen Noether. Ring Signature Confidential Transactions for Monero. In *Cryptology ePrint Archive: Report 2015/1098*, 2015.

[NR09] Phong Q. Nguyen and Oded Regev. Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures. *Journal of Cryptology*, 22(2):139–160, 4 2009.

[OTYO18] Takeshi Okamoto, Raylin Tso, Michitomo Yamaguchi, and Eiji Okamoto. A k-out-of-n Ring Signature with Flexible Participation for Signers. In *Cryptology ePrint Archive: Report 2018/728*, 2018.

[Ped91] Torben Pryds Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In *CRYPTO*. Springer, 1991.

[Pei08] Chris Peikert. Limits on the Hardness of Lattice Problems in p Norms. *computational complexity*, 17(2):300–351, 5 2008.

[Po16] Chris Peikert and others. *Decade of Lattice Cryptography*. World Scientific, 2016.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Theory of Cryptography Conference*, pages 145–166. Springer, 2006.

[Reg04] Oded Regev. Quantum Computation and Lattice Problems. *SIAM Journal on Computing*, 33(3):738–760, 1 2004.

[Reg10] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 2010.

[RS13] Dorit Ron and Adi Shamir. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography*, pages 6–24. Springer, 2013.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[RST01] Ronald Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In *ASIACRYPT*, pages 552–565. Springer, 2001.

[RST06] Ronald L Rivest, Adi Shamir, and Yael Tauman. How to leak a secret: Theory and applications of ring signatures. *Essays in memory of Shimon Even*, 3895:164–186, 2006.

[SALY17] Shi-Feng 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*, pages 456–474. Springer, 2017.

[Sch89] C. P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Advances in Cryptology — CRYPTO' 89 Proceedings*, pages 239–252. Springer New York, New York, NY, 1989.

[Sch91] C.P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[Sho00] Victor Shoup. Practical Threshold Signatures. In *EUROCRYPT*, pages 207–220. Springer, 2000.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. In *Cryptology ePrint Archive: Report 2004/332*, 2004.

[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.

[TAL+10] Patrick P Tsang, Man Ho Au, Joseph K Liu, Willy Susilo, and Duncan S Wong. A suite of non-pairing id-based threshold ring signature schemes with different levels of anonymity. In *ProvSec*, volume 6402, pages 166–183. Springer, 2010.

[TBS14] Wilson Abel Alberto Torres, Nandita Bhattacharjee, and Bala Srinivasan. Effectiveness of Fully Homomorphic Encryption to Preserve the Privacy of Biometric Data. In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services - iiWAS '14*, pages 152–158, New York, New York, USA, 2014. ACM Press.

[TCZ+20] Alin Tomescu, Robert Chen, Yiming Zheng, Ittai Abraham, Benny Pinkas, Guy Golan, and Srinivas Devadas. Towards Scalable Threshold Cryptosystems. In *IEEE Symposium on Security and Privacy*, 2020.

[TW05] Patrick P Tsang and Victor K Wei. Short linkable ring signatures for e-voting, e-cash and attestation. In *ISPEC*, volume 3439, pages 48–60. Springer, 2005.

[TWC+04] Patrick P. Tsang, Victor K. Wei, Tony K. Chan, Man Ho Au, Joseph K. Liu, and Duncan S. Wong. Separable Linkable Threshold Ring Signatures. In *INDOCRYPT*, pages 384–398. Springer, 2004.

[Vai11] Vinod Vaikuntanathan. Lecture notes of lattices in computer science, taught at MIT University, 2011.

[VN51] John Von Neumann. Various Techniques Used in Connection with Random Digits. *National Bureau of Standards Applied Mathematics Series*, 12(13):36–38, 1951.

[VS13] Nicolas Van Saberhagen. CryptoNote v 2.0. *https://cryptonote.org/whitepaper.pdf*, 2013.

[WDZ+14] Baodian Wei, Yusong Du, Huang Zhang, Fangguo Zhang, Haibo Tian, and Chongzhi Gao. Identity Based Threshold Ring Signature from Lattices. In *NSS*, pages 233–245. Springer, 2014.

[WFLW03] Duncan S. Wong, Karyin Fung, Joseph K. Liu, and Victor K. Wei. On the RS-Code Construction of Ring Signature Schemes and a Threshold Setting of RST. In *ICICS*, pages 34–46. Springer, 2003.

[WS11] Jin Wang and Bo Sun. Ring Signature Schemes from Lattice Basis Delegation. In *ICICS*, pages 15–28. Springer, 2011.

[WW12] Chen Wang and Huaixi Wang. A New Ring Signature Scheme from NTRU Lattice. In *ICCIS*, pages 353–356. IEEE, 2012.

[YHAL+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. *https://eprint.iacr.org/2017/781*, 2017.

[YLA+11] Tsz Hon Yuen, Joseph K. Liu, Man Ho Au, Willy Susilo, and Jianying Zhou. Threshold ring signature without random oracles. In *ASIACCS*, page 261. ACM Press, 2011.

[YLA+13] T. H. Yuen, J. K. Liu, M. H. Au, W. Susilo, and J. Zhou. Efficient Linkable and/or Threshold Ring Signature Without Random Oracles. *The Computer Journal*, 56(4):407–421, 4 2013.

[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. In *Cryptology ePrint Archive*. Report 2019/508, 2019.

[ZLCL07] Dong Zheng, Xiangxue Li, Kefei Chen, and Jianhua Li. Linkable Ring Signatures from Linear Feedback Shift Register. In *EUC*, pages 716–727. Springer, Berlin, Heidelberg, 2007.

[ZSNL04] Fangguo Zhang, Reihaneh Safavi-Naini, and Chin-Yin Lin. Some new proxy signature schemes from pairings. *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE.*, pages 59–66, 2004.

[ZZTA17] Huang Zhang, Fangguo Zhang, Haibo Tian, and Man Ho Au. Anonymous Post-Quantum Cryptocash (Full Version). *https://eprint.iacr.org/2017/716*, 2017.