Linkable Threshold Ring Signature from Lattices

Navid Abapour[1], Patrick Hough[2], Carlisle Adams[3], and Mahdi Mahdavi[4]

[1] Surrey Centre for Cyber Security, University of Surrey, United Kingdom
n.abapour@surrey.ac.uk
[2] Mathematical Institute, University of Oxford, United Kingdom
patrick.hough@maths.ox.ac.uk
[3] School of Electrical Engineering and Computer Science, University of Ottawa, Canada cadams@uottawa.ca
[4] Electrical Engineering Department, Shahid Beheshti University, Iran
m_mahdavioliaee@sbu.ac.ir

Abstract. The application of ring signatures is increasing, and cryptocurrencies, messaging protocols, voting systems, and whistleblowing are just a few examples. This cryptographic primitive is a type of digital signature that allows a signer to sign a message on behalf of a group of possible signers, ensuring that the actual signer remains anonymous within the group. This work presents the scheme of a lattice-based linkable ring signature based on two new variants of ring learning with errors problem, which is considered promising in terms of efficiency, providing competitive processing time and signature size. We propose using the output signature of the ring signature as the input for a lattice-based secret sharing scheme to add threshold property to increase protection against single points of failure and unauthorized access, reduce ring members' concerns about the consequences of their membership, and provide anonymous signatures. In spite of our decision and the fact that the proposed secret sharing is supposed to be based on RSA, we found a new type of common modulus attack on the Chinese Remainder Theorem model of RSA. In this attack, given one public-private key pair, an attacker can obtain the private key corresponding to a given public key in RSA decryption. In addition, we propose a parallelizable factoring algorithm with an order equivalent to the cyclic attack in the worst-case scenario.

Keywords: Post-quantum Cryptography · Ring Signature · Learning with Errors · Public-key Cryptography · Common Modulus Attack.

1 Introduction

Cryptographic primitives based on problems over algebraic lattices have established themselves as dominant in the landscape of post-quantum cryptography, owing to their success across all categories of the recent standardization process of the National Institute of Standards and Technology (NIST) [1]. Furthermore, lattices have given rise to a number of exotic primitives, such as multi-party computation and fully homomorphic encryption.

While much research has been conducted into the development of lattice-based encryption and digital signatures, far less success has been had in designing privacy-preserving primitives in which issues of trust must be carefully negotiated. In this work, we focus on one such primitive: ring signatures. In brief, a ring signature allows a signer to sign on behalf of a group in an anonymous way. Furthermore, the signer does not need to grant prior permission to allow signatories to create their own group. Despite a number of works in this area [2, 3, 4, 5], more progress is needed to make lattice-based ring signatures practical. Their efficiency only worsens when designs are enhanced with additional properties such as deniability and evidence checking, which often require complex expensive designs.

In addition, considering Linkability allows recognizing if two signatures come from the same signer, real-world implementations of non-linkable ring signatures can be vulnerable to double-spending attacks in applications like cryptocurrency, making 'linkability' and 'threshold security' highly desirable properties of ring signatures. Linkability allows the detection of double-signing by the same group member across different messages, and the second one requires a subset of minimum size to collaborate in order to generate a valid signature; to our knowledge, there is no post-quantum ring signature that can simultaneously support likability and threshold security. Except for very few of them [6, 7, 8, 9, 10], all linkable ring signature designs presented to date utilize number-theoretic assumptions and are vulnerable against quantum attacks [10].

So, how can we design a scheme that can satisfy the core properties of the ring signature, provide lightweight performance to be applicable, utilize post-quantum assumptions, and support linkability and threshold at the same time?

1.1 Related Works in Lattice-based Ring Signature

For several years, cryptographers have been increasing their use of lattice structures for post-quantum era preparation as the risk of quantum computing has emerged. This section will discuss a number of ring signatures that are most similar to our lattice-based scheme.

Xie et al. 2022 [10] present a lattice-based blind ring signature scheme designed to enhance anonymity in blockchain transactions, making them untraceable and reinforcing the privacy of the user's identity within blockchain transactions. The ring signature mechanism described in the paper functions by enabling a user within a group to sign a transaction on behalf of the group without revealing their identity. This is achieved by combining their private key with the public keys of all group members, creating a signature that validates the transaction while keeping the signer anonymous. However, the authors acknowledge the limitations of markedly large storage space for key matrices, which slow operational speeds. Further, they do not address the issue of double-spending attacks or provide proof of non-slanderability for the proposed scheme.

In [6], a non-interactive deniable ring signature scheme, which achieves post-quantum security, is proposed. This scheme is designed anonymously for traceable non-frameability under the hardness of lattice problems' assumptions and

utilizes a variant of the Shortest Vector Problem (SVP). The scheme they proposed allows any member of the ring to deny involvement, yet it permits the genuine signer to prove his identity when necessary—with certain levels of security and accountability. The signing algorithm includes the digital identity of the signer and the rest of the ring members, and it uses public keys to verify the signature.

Mundhe et all [7] outline a lattice-based ring signature scheme that the authors claim is practical and efficient called RAPTOR. The authors give a generic construction based on provable security in a random oracle model, which uses both standard and NTRU lattices for implementation, with the latter being more efficient. The linkable version of the RAPTOR scheme allows the linking of signatures from the same signer while maintaining anonymity and has approximately similar performance characteristics, albeit with slightly larger signature sizes due to additional components required for ensuring likability. This work presents a construction for a new primitive called Chameleon Hash Plus (CH+). Chameleon Hash Plus is a type of hash that incorporates lattice-based security properties. Two instantiations are presented: one based on standard lattice problems such as Short Integer Solution (SIS) and Inhomogeneous Short Integer Solution (ISIS) problems, while another relies on the NTRU lattice framework.

Ye et al. design a technique in the [8] that leverages the algebraic structure of ideal lattices to improve efficiency and security. Ideal lattices, as a generalization of cyclic lattices, offer an algebraic structure that enables smaller key sizes and faster operations compared to Euclidean lattices. The hardness of the scheme relies on the SIS and SVP on ideal lattices. The signing algorithm iterates to ensure the generated signature components fall within predefined polynomial norms. The verification algorithm checks these norms and recomputes hash values to validate the signature.

Cayrel et al. 2010 [11] introduce a scheme that modifies Aguilar's code-based threshold ring signature [12] by incorporating the CLRS identification scheme, which reduces the soundness error to $1/2$ per round. The application of the Fiat-Shamir heuristic further transforms this identification scheme into a signature scheme. The key security advantage lies in the worst-case to average-case reduction typical of lattice-based systems, making breaking even with randomly chosen parameters difficult. The signature generation process in this scheme involves a group of signers, where each signer's public and private keys are associated with a set of SIS problems. The leader among the signers performs operations similar to the CLRS identification protocol, applying permutations and commitments to maintain anonymity and security.

In [6], a scheme is presented in a way that ensures deniability, along with security. The proposed Non-interactive Deniable Ring Signature (NDRS) scheme relies on SVP. The signer's identity remains anonymous initially, but the scheme allows for the generation of evidence that can either confirm or deny a specific individual's involvement in creating the signature. This dual functionality is achieved through a structured approach involving multiple hash functions and polynomial computations within a lattice framework.

1.2 Related Works in Common Modulus Attacks

It is worth noting that despite the ever more likely advent of quantum computers, there exist real-world applications still using traditional public-key cryptosystems like RSA [13]; these include but are not limited to Google Workspace Single Sign-On[14], Telegram [15], OpenVPN[16]. It has been shown that a quantum computer is required to factorize the 2048-bit RSA modulus [17]; nevertheless, some implementations of RSA are vulnerable to attacks even without quantum computation. These attacks have mostly two types: one that aims to factorize the common modulus based on one key pair and the other that aims to decrypt certain ciphertexts generated by two public keys if the keys are co-prime.

In 2019, a survey of the attacks on RSA in the forty years of its life has been published [18]. In this survey, the attacks on RSA are classified into four groups: elementary attacks, weak public exponent attacks, weak private exponent attacks, and large private exponent attacks. Except for elementary attacks on RSA, all of the attacks originate from wrong choices in the public or private key. There are two elementary attacks on RSA: the common modulus attack and the blind signature attack [19]. The conventional common modulus attack assumes two coprime RSA public keys e_a and e_b in the same modulus N, i.e. $\gcd(e_a, e_b) = 1$. Then, the adversary can decrypt any message m encrypted by public keys e_a and e_b [18]. One can classify the attacks based on factorizing N given a public-private key pair [20, 21] as the common modulus attack, where the condition of $\gcd(e_a, e_b) = 1$ is not required anymore.

In the original paper of RSA, [13], it is clearly mentioned that given the public-private key pair (e, d), one can factorize N. This works off [22], which showed that N could be factored, given any multiple of $\phi(N)$ with the complexity of $O(log^3 N)$. Miller shows [22] how knowing the public-private key pair (e, d) is probabilistically equivalent to the factorization of N. Also, [21] presented the first deterministic polynomial time algorithm that factorizes N using the public-private key pair (e, d) with the complexities of $O(log^9 N)$ and $O(log^2 N)$, when $ed < N^2$ and $ed < N^{3/2}$, respectively. It is obvious that if the same modulus is used for Alice and Bob, then Bob can use his own exponents (e_b, d_b) to factorize the modulus N [20] and recover Alice's private key d_a given her public key e_a and the factorization of N.

The public key of RSA is usually set to a small number to provide fast encryption. Hastad presents [23] a small public exponent attack for a broadcast use case based on Coppersmith's method in [24]. In [25], a general version of the small public exponent was proposed. Although choosing a small exponent for the RSA private key provides a fast RSA signature, it makes the scheme vulnerable to small private exponent attacks. Wiener in [26] shows that the RSA system is insecure when $d < N^{0.25}$. However, this attack is useless if $e > N^{1.5}$. Boneh and Durfee in [27] improves Wiener's attack for a higher bound $d < N^{0.292}$. Their attack, which uses the LLL (Lenstra–Lenstra–Lovász) algorithm [28], is effective for $e < N^{\frac{15}{8}} = N^{1.875}$. Although they can not prove that it always succeeded, they have not found evidence of their attack's ineffectiveness [29]. Recently, a special case of Boneh and Durfee's attack is investigated by Mumtaz and Ping

in [30], where the running time of the attack is improved. Mumtaz and Ping propose a large RSA decryption exponent attack in [31]. A cryptanalysis of the RSA cryptosystem with smooth prime sum is proposed in [32]. Moreover, [33] tries to inject backdoors into RSA and other cryptographic primitives based on the integer factoring problem.

1.3 Contribution

Having identified the need for an efficient linkable ring signature that can be used in the post-quantum era and support threshold, we make the following contributions. We begin by defining two new variants of Ring-LWE, which we then use to design a linkable ring signature whose efficiency is competitive with existing schemes. We then present a new secret sharing scheme that yields a threshold ring signature when combined with our ring signature. As an additional contribution, our attempts to design an appropriate secret sharing scheme reveal to us a new common modulus attack on RSA. We list these contributions.

- Proposing Two Lattice-based Problems: Composite Ring Learning with Errors (CRLWE) and Matrix Ring Learning with Errors (MR-LWE), two variants of Ring-LWE, which have the same complexity as Ring-LWE but are slightly more efficient
- Building a Linkable Ring Signature based on CRLWE, which is more efficient than similar schemes [6, 7, 8, 9, 10] in signature size and processing time
- Designing a Secret Sharing Scheme based on MR-LWE and adding it to the signature scheme for building a Threshold Ring Signature
- Presenting an efficient common modulus attack on RSA; also, providing a factoring algorithm regarding a complexity equal to the cyclic attack

1.4 Outline

This paper is structured as follows. Section 2 introduces some concepts in lattice-based cryptography and ring signatures. Section 3 formally defines two variants of lattice-based problems. Section 4 proposes the scheme of ring signature and secret sharing technique. Section 5 illustrates the new attack on the RSA.

2 Preliminaries

The notation $\mathbb{Z}_q$ denotes the ring of integers modulo q, where q is typically a prime integer. The term *monic polynomial* refers to a polynomial whose leading coefficient is 1. The notation $\langle f(X) \rangle$ represents the ideal generated by the polynomial $f(X)$. $\mathbf{a}_i$ and $\mathbf{s}$ denoting vectors, and the inner product $\langle \mathbf{a}_i, \mathbf{s} \rangle$ represents the standard dot product. The small error term e_i is typically drawn from a discrete Gaussian or other bounded distribution χ.

2.1 Computational Assumption

Here, we present the existing hardness assumption on which our constructions rely.

The worst-case hardness of standard lattice problems translates to the average-case hardness of LWE, making LWE a foundational problem for post-quantum cryptographic primitives [34]. LWE has been very widely studied and remains conjecturally hard for quantum computers. This hardness serves as the basis for many systems. A variant of the LWE assumption, called $\mathsf{Ring}-\mathsf{LWE}$ [35], adapts the problem to polynomial rings in a way that allows much more efficient computations while imposing similar security constraints around it.

Definition 1 (Ring-LWE). *Consider a ring $\mathbb{Z}_q$ where q is a prime integer, and let $f(X)$ be a fixed, monic polynomial of degree n over $\mathbb{Z}_q$, defining the polynomial ring $R_q = \mathbb{Z}_q[X]/\langle f(X)\rangle$. The Ring Learning with Errors problem is to recover a secret polynomial $s(X) \in R_q$ from a set of pairs $(a_i(X), b_i(X))$ for $i = 1, 2, \ldots, m$, where $a_i(X)$ is chosen randomly from R_q, $b_i(X) = a_i(X)\cdot s(X) + e_i(X) \mod f(X)$, and $e_i(X)$ is a small error polynomial.*

The $\mathsf{Ring}-\mathsf{LWE}$ problem leverages the algebraic structure of polynomial rings to achieve computational efficiency, making it particularly suitable for practical applications. The hardness of the $\mathsf{Ring}-\mathsf{LWE}$ problem is comparable to the hardness of the LWE problem, ensuring robust security guarantees [36].

2.2 Ring Signature Properties

A ring signature scheme is a type of digital signature that allows a member of a group (the ring) to sign a message on behalf of the group without revealing which member actually signed the message. This type of scheme consists of the following algorithms:

- $\mathsf{Setup}(\lambda) \rightarrow \mathtt{params}$: takes a security parameter λ and outputs the system parameters.
- $\mathsf{KeyGen}(\mathtt{params}) \rightarrow (sk, pk)$: A probabilistic algorithm that accepts the system parameters and gives a secret key sk and a public key pk.
- $\mathsf{Sign}(m, sk, \{pk_i\}_{i=1}^n) \rightarrow \sigma$: gets a message m, a secret key sk, and a set of public keys $\{pk_i\}_{i=1}^n$ of the ring members, and gives a signature σ.
- $\mathsf{Verify}(m, \sigma, \{pk_i\}_{i=1}^n) \rightarrow \{0, 1\}$: A deterministic algorithm that takes a message m, a signature σ, and a set of public keys $\{pk_i\}_{i=1}^n$, and outputs 1 if the signature is valid and 0 otherwise.

Generally, there are five properties that capture the core security requirements of a ring signature:

- Correctness assures that the verification algorithm will accept every valid signature.

Definition 2 (Correctness). *A ring signature scheme is correct if, given a message m and a valid signature σ produced by an honest signer, the verification algorithm accepts the signature with high probability. For all messages m and for all valid signatures σ, it holds that:* $\Pr[\mathsf{Verify}(m, \sigma) = 1] \geq 1 - \epsilon$,

– The Anonymity of signers is achieved in such a way that it's impossible to make out who in the group creates which of the provided signatures.

Definition 3 (Anonymity). *The ring signature scheme provides anonymity if, given any two distinct signers, it is computationally infeasible for any adversary to distinguish between signatures produced by these two signers. For any probabilistic polynomial-time adversary $\mathcal{A}$, the advantage in distinguishing the signatures is negligible:*

$$Adv_{\mathcal{A}} = |\Pr[\mathcal{A}(\sigma_i) = i] - \Pr[\mathcal{A}(\sigma_j) = i]| \leq \epsilon,$$

where σ_i and σ_j are signatures produced by the i-th and j-th signer, respectively, and ϵ is a negligible function.

– Unforgeability means that a signature cannot be replicated, and in this case, it includes protection against an adversarial attempt to forge some specific signature, hence allowing only some members of the group to sign messages.

Definition 4 (Unforgeability). *Considering a ring signature, this scheme is unforgeable if it is computationally infeasible for any adversary to produce a valid signature on behalf of any member of the ring without knowing their secret key. For any probabilistic polynomial-time adversary $\mathcal{A}$, the probability of producing a valid signature σ^* for a message m^* without access to the secret key of any legitimate signer is negligible:*

$$\Pr[\mathsf{Verify}(m^*, \sigma^*) = 1 \text{ and } \sigma^* \text{ was not produced by an honest signer}] \leq \epsilon,$$

where ϵ is a negligible function.

– Linkability means that messages can be connected together; hence, if there is a double-signing from a member of the group, then it will easily be noticed.

Definition 5 (Linkability). *A linkable ring signature scheme provides linkability if it is possible to determine whether two valid signatures have been produced by the same signer. For any two valid signatures σ_1 and σ_2, there exists a linkability function* Link *such that:*

$$\mathsf{Link}(\sigma_1, \sigma_2) = \begin{cases} 1 & \text{if the same signer produced both signatures,} \\ 0 & \text{otherwise.} \end{cases}$$

The existence of separated function Link is not necessary, and an inner procedure during signing can handle this too.

– Non-slanderability ensures that combining already sent-out message signatures does not produce any valid new message signature. It is vital for the group to protect message integrity.

Definition 6 (Non-Slanderability). *A linkable ring signature scheme ensures non-slanderability if it is computationally infeasible for any adversary to produce a valid signature that appears to come from another legitimate signer without knowing their secret key. For any probabilistic polynomial-time adversary $\mathcal{A}$, the probability of producing a signature σ^* that links to an honestly generated signature from a signer who never produced σ^* is negligible:*

$$\Pr\left[\begin{array}{c}\mathsf{Verify}(m^*,\sigma^*)=1 \text{ and}\\ \mathsf{Link}(\sigma^*,\sigma)=1 \text{ for an honestly generated } \sigma\\ \text{from a signer who never produced } \sigma^*\end{array}\right] \leq \epsilon,$$

where ϵ is a negligible function.

There have always been ongoing discussions regarding managing these properties. For example, sometimes it has been believed that "*if a linkable ring signature scheme is linkable and non-slanderable, it is also unforgeable*" [37]. This statement is not entirely accurate, as it makes some assumptions about the properties of a "*linkable ring signature scheme*" that may not be universally true, and this implication is not necessarily true. While non-slanderability and linkability are important properties for maintaining the privacy and preventing false accusations in certain applications, unforgeability is a separate property related to preventing unauthorized signature creation. It is possible to have a linkable ring signature scheme that is non-slanderable and linkable but is still vulnerable to forgery if not designed and implemented correctly.

3 Proposed Problems

Derived by Ring-LWE [35], the proposed signature scheme is based on a problem called Composite Ring-LWE (CRLWE).

Definition 7 (CRLWE). *Consider a ring $\mathbb{Z}_q$ where q is a composite integer, and let $f(X)$ be a fixed, monic polynomial of degree n over $\mathbb{Z}_q$, defining the polynomial ring $R_q = \mathbb{Z}_q[X]/\langle f(X)\rangle$. The task is to recover a secret polynomial $s(X) \in R_q$ from a set of pairs $(a_i(X), b_i(X))$ for $i = 1, 2, \ldots, m$, where $a_i(X)$ is chosen randomly from R_q, $b_i(X) = a_i(X) \cdot s(X) + e_i(X) \mod f(X)$, and $e_i(X)$ is a small error polynomial.*

Let $f(X)$ be a monic polynomial of degree n over $\mathbb{Z}_q$. The quotient ring $R_q = \mathbb{Z}_q[X]/\langle f(X)\rangle$ is isomorphic to the ring $\mathbb{Z}_q^n$, where $\mathbb{Z}_q^n$ denotes the set of vectors of length n with components from $\mathbb{Z}_q$. The quotient ring R_q over the monic polynomial $f(X)$ of degree n under a finite field $\mathbb{F}_q$, indeed forms a field. It is a very important distinction since every non-zero element of such a quotient

ring has a multiplicative inverse and, therefore, satisfies the definition of a field. Here we consider the R_q notion since it is more illustrative of the representation in cryptographic applications, particularly those associated with problems on lattices.

Essentially, the Ring-LWE problem is the LWE [38] problem, but defined over the polynomial rings such as R_q, allowing the algorithm to become much more efficient by exploiting the ring structure. The Composite Ring Learning with Errors (CRLWE) problem is a variant of them, so it is similar to LWE like the problem of Ring-LWE, in the sense of making cryptographic protocols on polynomial rings. While, in general, Ring-LWE and LWE are challenges that distinguish only between pure random samples and samples originating from some secret-dependent process, CRLWE generalizes the problem in such a way that it includes composite structures, which accounts for the complexity in the problem set.

CRLWE differs from the usual LWE in that it is hard to distinguish between samples; moreover, working with rings modulo a composite number is considered combinatorially harder than working modulo a prime due to the increased structural complexity and additional factorization properties of composite numbers. This becomes more tricky, introducing new dimensions into the problem, more so regarding arithmetic and algebraic properties between the ring's elements from the point of the lattice-based cryptographic schemes. This assumption comes from the fact that there is, currently, no efficient quantum algorithm that can solve the hard foundational problems on which these cryptographic schemes lie, such as Ring-LWE, or the lattice-based SVP [39].

The following lemma is being brought up to provide a framework and justification for the process being undertaken in the CRLWE problem.

Lemma 1 (Polynomials Interaction). *Let $R_q = \mathbb{Z}_q[X]/\langle f(X)\rangle$, where $f(X)$ is a monic polynomial of degree n over $\mathbb{Z}_q$. For any polynomials $a(X)$, $s(X)$ in R_q, and a small error polynomial $e(X)$, there exist unique polynomials $b(X)$ in R_q such that $b(X) = a(X) \cdot s(X) + e(X) \mod f(X)$, and the degree of $e(X)$ is less than the degree of $f(X)$. The uniqueness of $b(X)$ arises from the unique representation of $a(X) \cdot s(X)$ in R_q and the bounded degree of $e(X)$.*

Proof. Any two polynomials $a(X)$ and $s(X)$, as elements of the R_q, are multiplied in the R_q by using polynomial multiplication under modulo $f(X)$. Since $f(X)$ is a description of the ring R_q, this multiplication modulo $f(X)$ will ensure that the product $a(X) \cdot s(X)$ lies in R_q and therefore maintains the structure of the ring. Adding a small error polynomial $e(X)$ to the term $a(X) \cdot s(X)$ mod $f(X)$, and the result is $b(X) = a(X) \cdot s(X) + e(X) \mod f(X)$. The degree constraint on $e(X)$ makes sure that it does not change the degree properties of $f(X)$ in R_q that are necessary.

The distinctness in $b(X)$ is due to the determinacy of the polynomial multiplication in R_q and the fixed degree constraint in $e(X)$. Having two different representations of $b(X)$ for the same $a(X)$, $s(X)$, and $e(X)$ means having two different outcomes for the same polynomial operation in R_q, gives a contradiction to the rules of well-defined operations in a ring.

Hence, for every pair of polynomials $a(X)$ and $s(X)$ in R_q, and for every small error polynomial $e(X)$, one and only one polynomial $b(X)$ exists such that the relationship $b(X) = a(X) \cdot s(X) + e(X) \mod f(X)$ ensures the integrity of R_q and the bounded degree of $e(X)$ without getting modified. □

To directly address the core challenge posed by the CRLWE Problem, Theorem 1 displays it, which not only confirms the existence of a representative polynomial but also guarantees its uniqueness in CRLWE.

Theorem 1 (Polynomial Recovery). *Given R_q and $f(X)$ as in the context of* CRLWE*, for a set of pairs $(a_i(X), b_i(X))$ for $i = 1, 2, \ldots, m$, where $a_i(X)$ is chosen randomly from R_q and $b_i(X) = a_i(X) \cdot s(X) + e_i(X) \mod f(X)$ with $s(X) \in R_q$ and $e_i(X)$ being small error polynomials, there exists a unique polynomial $s(X) \in R_q$ of degree less than n that satisfies all given pairs under the condition that $m \geq 2n$ and the errors $e_i(X)$ are bounded by $\|e_i(X)\| \leq q/100$.*

Proof. Consider the set of equations defined by the CRLWE problem for $i = 1, 2, \ldots, m$:

$$b_i(X) = a_i(X) \cdot s(X) + e_i(X) \mod f(X),$$

where $a_i(X)$ and $b_i(X)$ are known, $s(X)$ is the secret polynomial to be recovered, and $e_i(X)$ are error polynomials with bounded norm $\|e_i(X)\| \leq \frac{q}{100}$. Define the matrix $\mathbf{A}$ whose rows are coefficients of $a_i(X)$, and vectors $\mathbf{b}$ and $\mathbf{e}$ similarly for $b_i(X)$ and $e_i(X)$. Then, the system can be rewritten in matrix form:

$$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod f(X),$$

where $\mathbf{s}$ is the vector of coefficients of the polynomial $s(X)$. Given that $m \geq 2n$, the system is over-determined. Using the pseudoinverse $\mathbf{A}^+$ of $\mathbf{A}$, the estimate of $\mathbf{s}$ is:

$$\hat{\mathbf{s}} = \mathbf{A}^+\mathbf{b} \mod f(X).$$

The error term $\mathbf{e}$ affects the recovery of $\mathbf{s}$, but given that $\|\mathbf{e}\| \leq \frac{mq}{100}$ and $\mathbf{A}$ has full rank, $\|\mathbf{A}^+\mathbf{e}\|$ will be bounded and small enough not to prevent recovery of $\mathbf{s}$ within the desired error bounds. Since $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq 2n$ and $\text{rank}(\mathbf{A}) = n$, the least squares solution minimizes $\|\mathbf{A}\hat{\mathbf{s}} - \mathbf{b}\|$, ensuring that the impact of $\mathbf{e}$ on $\hat{\mathbf{s}}$ is minimized under the modulus operation. Thus, the unique polynomial $s(X)$ is recoverable from $\hat{\mathbf{s}}$ under the conditions specified, demonstrating that $\mathbf{s}$ and $\hat{\mathbf{s}}$ must coincide modulo $f(X)$, provided that the error $\mathbf{e}$ remains within the specified bounds. □

Theorem 2 (Hardness Equivalence). *Given a ring $R_q = \mathbb{Z}_q[X]/\langle f(X) \rangle$, where q is a composite integer and $f(X)$ is a fixed, monic polynomial of degree n,* CRLWE *is computationally equivalent to the* Ring-LWE *in R_q in terms of hardness, under a pre-defined polynomial compression function ϕ.*

Proof. As setup of Ring-LWE, assume tuples $(a_i(X), b_i(X))$ for $i = 1, \ldots, m$, where $a_i(X)$ is uniformly random in R_q, and $b_i(X) = a_i(X) \cdot s(X) + e_i(X)$

mod $f(X)$, with $s(X), e_i(X)$ randomly drawn from distributions χ_s and χ_e respectively. As setup of CRLWE, let $\phi : R_q \to R'_q$ to be a compression function, leading to compressed samples $(a'_i(X), b'_i(X)) = (\phi(a_i(X)), \phi(b_i(X)))$, where ϕ is a ring homomorphism, and $\phi(e_i(X))$ preserves the subgaussian parameter σ of $e_i(X)$. To construct CRLWE instance from Ring-LWE, due to the homomorphic properties of ϕ, the structure needed for CRLWE is maintained, and we have:

$$\begin{aligned} a'_i(X) &= \phi(a_i(X)), \\ b'_i(X) &= \phi(b_i(X)) = \phi(a_i(X) \cdot s(X) + e_i(X)) \\ &= \phi(a_i(X)) \cdot \phi(s(X)) + \phi(e_i(X)). \end{aligned}$$

$$b'_i(X) = a'_i(X) \cdot s'(X) + e'_i(X), \quad \text{where} \quad s'(X) = \phi(s(X)), \quad e'_i(X) = \phi(e_i(X)).$$

Assume the existence of an adversary $\mathcal{A}$ that solves CRLWE: $\mathcal{A}(a'_i(X), b'_i(X)) \to s'(X)$, construct $\mathcal{B}$ to solve Ring-LWE:

$$\mathcal{B}(a_i(X), b_i(X)) = \phi^{-1}(\mathcal{A}(\phi(a_i(X)), \phi(b_i(X))))$$

By the properties of ϕ, the error term $e'_i(X) = \phi(e_i(X))$ retains subgaussian noise characteristics, ensuring the strength of the CRLWE instance:

$$\rho(e'_i(X)) \approx \rho(e_i(X)) \text{ under } \phi$$

From an error preservation point of view, assuming ϕ is surjective and R'_q is a smaller but sufficient structure, ϕ^{-1} on $s'(X)$ reliably reconstructs $s(X)$, presuming the transformation is invertible or pseudo-invertible on the range of ϕ. If $\mathcal{B}$ solves Ring-LWE by utilizing $\mathcal{A}$ as an oracle, this constructs a polynomial-time reduction from Ring-LWE to CRLWE. □

Moreover, CRLWE offers computational advantages in terms of processing efficiency due to reduced data representation without compromising the hardness.

To relate the gap between the polynomial quotient rings and vector spaces over finite fields, it becomes mandatory to consider the isomorphism of the polynomial rings. This becomes very key in the CRLWE Problem that turns on the identification of representatives in R_q.

The second problem is called Matrix Ring-LWE (MR-LWE), and our secret-sharing scheme will utilize it. The MR-LWE problem can be viewed as an extension or generalization of the standard LWE problem, and it is a collection of LWE samples.

Definition 8 (Error Distribution). *Let θ be a real number representing the standard deviation, and m the dimension of the error vector. The error distribution $\mathcal{D}_\theta^m$ is defined as a multivariate Gaussian distribution with mean zero and covariance matrix $\theta^2\mathbf{I}_m$, where $\mathbf{I}_m$ is the $m \times m$ identity matrix. Each component of the error vector is independently drawn from $\mathcal{N}$ as a distribution*

$$\mathcal{D}_\theta^m \sim \mathcal{N}(0, \theta^2\mathbf{I}_m).$$

Definition 9 (MR-LWE). *Let q and m be positive integers, and let θ be a real number between 0 and 1. Consider an $m \times m$ matrix $\mathbf{A}$ with entries from $\mathbb{Z}_q$ and a secret vector $\mathbf{s}$ uniformly chosen from $\mathbb{Z}_q^m$. Given the matrix $\mathbf{A}$, and the noisy vector*

$$\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q,$$

where $\mathbf{e}$ is a noise vector generated from the error distribution D_θ^m, the task is to recover the secret $\mathbf{s}$.

Unlike standard LWE, which provides multiple samples of the form $\langle \mathbf{a}_i, \mathbf{s} \rangle + e_i$ for $i \in [m]$, MR-LWE encapsulates these samples into a matrix-vector multiplication form, making it more suitable for parallel processing and certain cryptographic constructions where matrix operations are advantageous. In other words, in the LWE problem, one is provided with samples of the form:

$$\langle \mathbf{a}, \mathbf{s} \rangle + e \mod q,$$

where $\mathbf{a}$ is random, $\mathbf{s}$ is a secret vector, and e indicates a small error term. On the other hand, MR-LWE is defined in a matrix-vector setup for samples of the form presented in Definition 9, where LWE samples are derived from inner products, MR-LWE samples arise from matrix-vector multiplications.

But how much noise can an MR-LWE-based system tolerate before the secret becomes irrecoverable? Lemma 2 answers it, which quantifies this noise to represent a threshold θ' where recovery is improbable. It provides insight into the feasibility of recovery of the system as the system tuning parameters are being tuned.

Theorem 3 (MR-LWE **Hardness**). *The* MR-LWE *is computationally equivalent to solving the* Ring-LWE *problem in R, and offers computational improvements by enabling parallel processing and reduced complexity in terms of the number of ring operations required, explicitly quantifiable by matrix multiplication optimizations.*

Proof. For setting up the MR-LWE problem, consider $\mathbf{A} \in R^{m \times m}, \mathbf{s} \in R^m$, and $\mathbf{e} \in R^m$ where each $e_i \sim D_\theta$, compute $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod q$, where D_θ denotes a discrete Gaussian distribution over R with mean 0 and standard deviation θ.

For setting up Ring-LWE, consider $a(x) \in R, s(x) \in R$, and $e(x) \in R$ where $e(x) \sim D_\theta$, find $s(x)$ given $b(x) = a(x) \cdot s(x) + e(x) \mod f(x)$. If each entry $\mathbf{A}_{ij}$ is a polynomial in R, decomposition of the MR-LWE into a set of m individual Ring-LWE problems will be similar to:

$$\text{For each } i, \quad \mathbf{b}_i(x) = \left(\sum_{j=1}^{m} \mathbf{A}_{ij}(x) \cdot \mathbf{s}_j(x) \right) + \mathbf{e}_i(x) \mod f(x).$$

This can also be done by embedding polynomials via converting each element of $\mathbf{A}, \mathbf{s},$ and $\mathbf{e}$ into their equivalents in R.

$$\mathbf{A} = \begin{bmatrix} a_{11}(x) & \cdots & a_{1m}(x) \\ \vdots & \ddots & \vdots \\ a_{m1}(x) & \cdots & a_{mm}(x) \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} s_1(x) \\ \vdots \\ s_m(x) \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e_1(x) \\ \vdots \\ e_m(x) \end{bmatrix}$$

Formulate the MR-LWE as multiple Ring-LWE equations:

$$\mathbf{b}(x) = \mathbf{A}(x)\mathbf{s}(x) + \mathbf{e}(x) = \begin{bmatrix} \sum_{j=1}^{m} a_{1j}(x)s_j(x) + e_1(x) \\ \vdots \\ \sum_{j=1}^{m} a_{mj}(x)s_j(x) + e_m(x) \end{bmatrix} \mod f(x)$$

$$\text{If } \exists\, \mathcal{A}_{\text{Ring-LWE}} : a(x)s(x) + e(x) \to s(x), \text{ then}$$
$$\exists\, \mathcal{B}_{\text{MR-LWE}} : \mathbf{A}(x)\mathbf{s}(x) + \mathbf{e}(x) \to \mathbf{s}(x) \text{ using } \mathcal{A}_{\text{Ring-LWE}}.$$

This transformation means solving MR-LWE is equivalent to solving m simultaneous Ring-LWE problems, and any solver for the Ring-LWE can be adapted to solve MR-LWE by solving each of these equations individually.

The multiplication $\mathbf{As}$ involves m^2 multiplications in R, which are typically $O(n^2)$ operations given that polynomial multiplication in R is $O(n^2)$ under standard algorithms. However, utilizing algorithms like the Karatsuba or FFT-based methods, the complexity reduces to $O(n \log n)$. The matrix-vector multiplication $\mathbf{As}$ can be parallelized, distributing m vector products across multiple processors, effectively reducing the wall-clock time by a factor of m in an ideal parallel computation environment. □

Lemma 2 (Noise Tolerance). *Assuming that the matrix* $\mathbf{A}$ *is of full rank and the noise vector* $\mathbf{e}$ *is generated from the error distribution* $\mathcal{D}_\theta^m$*, there exists a threshold* θ' *such that if* $\theta < \theta'$*, the secret vector* $\mathbf{s}$ *can be recovered with high probability.*

Proof. Let the error vector $\mathbf{e} = (e_1, e_2, \ldots, e_m)$ consist of components e_i independently distributed according to a Gaussian distribution $\mathcal{N}(0, \theta^2)$, where θ represents the standard deviation, adjusted for the modulus q. To consider the probability bounds for e_i, we employ Gaussian tail bounds. The probability that the magnitude of any component e_i exceeds θq is given by:

$$\Pr(|e_i| \geq \theta q) = 2\left(1 - \Phi\left(\frac{\theta q}{\theta}\right)\right),$$

where Φ is the cumulative distribution function of the standard Gaussian distribution. This probability can be approximated using the tail bound:

$$\Pr(|e_i| \geq \theta q) \approx 2\left(1 - \Phi\left(q\right)\right) \leq 2\exp\left(-\frac{q^2}{2}\right),$$

assuming $\theta = 1$ for simplification, and quantifies the robustness against the deviation of e_i from zero, ensuring e_i remains sufficiently small with high probability. □

Lemma 2 captures intuitions about the distribution of noise and its relation to the recoverability of the secret. This sets a threshold (θ') so that there exists a clear separation of the noise, and if the noise does not cross this threshold,

the secret vector $\mathbf{s}$ will be recoverable with high confidence. This is majorly significant since, by its demarcation, it points the way towards the necessary parameters that are used in cryptographic implementations based on MR-LWE.

This full-rank requirement for the matrix $\mathbf{A}$ is relatively crucial, showing the importance of the matrix structure in the MR-LWE problem. A full-rank matrix ensures that the system of equations is non-degenerate, hence making the secret recoverable. The other natural question of interest is this: under what conditions can we hope to recover $\mathbf{s}$ with high confidence? Theorem 4, although the answer it gives could be considered trivial, is highly nontrivial in as much as it underscores that while noise introduces ambiguity, this ambiguity isn't impenetrable. Such a delineation through the threshold θ' would make sure the secret $\mathbf{s}$ could be recovered reliably unless the noise goes high up to some problematic extent, in other words, the cryptographic system based on MR-LWE would not deviate from being secure. MR-LWE, on the other hand, has its theoretical ground in the presence of such noise, strategically devised computational methods that can break through this layer of uncertainty to recover the original secret with great confidence. A fine line—this noise balance secret between the noise causing this recovery and the non-recoverable noise is key to practical application and security in MR-LWE-based systems.

Theorem 4. *Let* $\mathbf{A}$ *be a full rank matrix and* $\mathbf{e}$ *be a noise vector generated from the error distribution* $\mathcal{D}_\theta^m$*. If* $\mathbf{s}$ *is a secret vector uniformly chosen from* $\mathbb{Z}_q^m$ *and* $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e}\,(mod q)$*, then there exists an algorithm that recovers* $\mathbf{s}$ *with high probability, given appropriate assumptions on* $\mathcal{D}_\theta^m$*.*

Proof. The goal is to show the existence of an algorithm that can recover $\mathbf{s}$ with high probability. To this end, the concept of maximum likelihood estimation is being employed. Define the likelihood function $\mathcal{L}(\mathbf{s})$ as:

$$\mathcal{L}(\mathbf{s}) = \Pr(\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} | \mathbf{s}, \mathbf{A}).$$

Given that $\mathbf{e}$ is independent and identically distributed according to $\mathcal{D}_\theta^m$, the likelihood can be decomposed as:

$$\mathcal{L}(\mathbf{s}) = \prod_{i=1}^{m} \Pr(e_i | \mathbf{s}, \mathbf{A}).$$

The goal of MLE is to find $\mathbf{s}$ that maximizes $\mathcal{L}(\mathbf{s})$. Equivalently, the log-likelihood can get maximized:

$$L(\mathbf{s}) = \sum_{i=1}^{m} \log(\Pr(e_i | \mathbf{s}, \mathbf{A})).$$

Optimizing $L(\mathbf{s})$ yields an estimate for $\mathbf{s}$. Since $\mathbf{A}$ is full rank, it's invertible. Utilizing this property, along with optimization techniques (e.g., gradient descent or expectation-maximization), we can maximize $L(\mathbf{s})$ to get an estimate of $\mathbf{s}$. The accuracy of this estimate and the recovery of $\mathbf{s}$ depend on assumptions about $\mathcal{D}_\theta^m$, such as boundedness and small variance. Thus, considering Algorithm 1, under appropriate conditions on $\mathcal{D}_\theta^m$, the theorem is proven. □

Algorithm 1 MLE-based Secret Vector Recovery

procedure RecoverSecretVector($\mathbf{A}$, $\mathbf{b}$, $\mathcal{D}_\theta^m$)
  Input: Measurement matrix $\mathbf{A}$, measurement vector $\mathbf{b}$, noise distribution $\mathcal{D}_\theta^m$
  Output: Estimated secret vector $\hat{\mathbf{s}}$
  Initialize a starting estimate $\mathbf{s}_0$ (e.g., a zero vector)
  while not converged **do**
    for $i = 1$ to m **do**
      Compute gradient $\nabla_i L(\mathbf{s})$ using $\frac{\partial}{\partial s_i} \log(\Pr(e_i \mid \mathbf{s}, \mathbf{A}))$
      Update s_i in the direction of $\nabla_i L(\mathbf{s})$ (e.g., using a gradient ascent)
    end for
  end while
  Output $\hat{\mathbf{s}}$ as the estimate that maximizes $L(\mathbf{s})$
end procedure

To see better performance of MR-LWE in practice, it is crucial to ensure that the vectors in the reduced lattice basis $\mathbf{B}'$ are not only shorter but also more orthogonal than those in the original basis $\mathbf{B}$. This modification serves dual purposes:

I. **Efficiency:** Functions such as sampling from a lattice or solving for the shortest vectors become much more efficient when a well-reduced base is used. This is at the core of making protocols more practical through faster key generation and quicker encryption/decryption [40].
II. **Security:** It should be that reducing the lattice basis does not simplify the lattice to a point where the reduced lattice is attackable. The reduction should be hidden enough to reach the security level intended for MR-LWE-based cryptographic systems. It should avoid configurations that make the solution to hard lattice problems, like SVP or CVP at large dimensions, easy because these are basically the foundations of the security assumption under lattice-based cryptography [41].

Lemma 3 (Lattice Reduction). *Given a lattice basis* $\mathbf{B}$ *in* $\mathbb{Z}_q^{m\times m}$*, there exists an algorithm that computes a reduced lattice basis* $\mathbf{B}'$ *such that the vectors in* $\mathbf{B}'$ *are both shorter and more orthogonal than those in* $\mathbf{B}$*.*

Proof. The Lenstra–Lenstra–Lovász (LLL) algorithm provides a method for reducing a lattice basis. Given a lattice basis $\mathbf{B}$ in $\mathbb{Z}_q^{m\times m}$, the LLL algorithm, computes a reduced lattice basis $\mathbf{B}'$ with the desired properties. The LLL procedure can be outlined as Algorithm 2.

The Lenstra–Lenstra–Lovász algorithm ensures that:

- Vectors in $\mathbf{B}'$ are shorter: $\|\mathbf{b}'_i\| \leq \|\mathbf{b}_i\|$ for all i.
- Vectors in $\mathbf{B}'$ are more orthogonal: $|\mathbf{b}'_i \cdot \mathbf{b}'_j| \leq \frac{1}{2}$ for all $i \neq j$.

□

Algorithm 2 Lenstra–Lenstra–Lovász Lattice Reduction

procedure LLLReduction($\mathbf{B}$)
  Input: Lattice basis $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_m]$
  Output: Reduced lattice basis $\mathbf{B}'$
  Initialize Gram-Schmidt orthogonalization matrix $\mathbf{G}$ as the identity matrix.
  repeat
    for each vector pair $(\mathbf{b}_i, \mathbf{b}_j)$ where $i > j$ **do**
      Compute $\mu = \frac{\mathbf{b}_i \cdot \mathbf{g}_j}{\mathbf{g}_j \cdot \mathbf{g}_j}$
      Update $\mathbf{b}_i = \mathbf{b}_i - \lfloor \mu \rfloor \mathbf{b}_j$
      Adjust $\mathbf{G}$ with $\mathbf{g}_i = \mathbf{b}_i - \mathbf{b}_j$
    end for
    for each $\mathbf{b}_i$ **do**
      if squared length of $\mathbf{b}_i$ exceeds $\frac{3}{4}\|\mathbf{b}_i - \mathbf{b}_{i-1}\|^2$ **then**
        Update $\mathbf{b}_i = \mathbf{b}_i - \left\lfloor \frac{\mathbf{b}_i \cdot \mathbf{b}_{i-1}}{\mathbf{b}_{i-1} \cdot \mathbf{b}_{i-1}} \right\rfloor \mathbf{b}_{i-1}$
        Adjust $\mathbf{G}$ accordingly
      end if
    end for
  until no further progress
  return $\mathbf{B}'$
end procedure

4 Linkable Threshold Ring Signature

Here, we present the linkable ring signature scheme that is based on the CRLWE problem. In general, our linkable ring signature scheme starts with the choice of a random polynomial in the ring and proceeds with a number of polynomial operations that are applied to compute related polynomials. The signature contains an identifier to ensure that the linkability is unique. Verification includes the verification of a hash and polynomial computations to ensure that the scheme is secure. The signature is built in such a way that the sharing of secrets takes place among the participants. There is a threshold provision ascertaining that only a coalition can form a valid signature. Such an approach to a threshold divides trust among the participants and increases the resistance of the signature against forgery.

4.1 Ring Signature Scheme

The following, along with Figure 1, is the way that the ring signature processes.

Stern's Zero-Knowledge Protocol The Zero-Knowledge Proof protocol developed by Stern [42] allows a prover to convince the verifier that he knows something without revealing the actual value and does so without ever revealing the secret. The prover does so through both the generation of random values and the creation of different commitments. Essentially, these commitments are blinded computations with the secret. The prover then computes a challenge by

hashing these commitments and responds to the challenge by generating specific responses that inter-product the secret and random values that are verifiable but not reverse-engineerable to reveal the secret. The verifier checks if the responses are valid according to the commitments and the challenge. Assuming everything goes well and the prover convinces the verifier of their identity, then the verifier can be sure that the prover possesses the secret without knowing anything about it directly, whereas the secret remains secure. This protocol fits well within the proposed linkable ring structure and seems efficient for our lattice-based problems.

The Prove procedure works by first selecting random polynomials y, z, and w from the ring R_q. These random values are used to compute three commitments: $c_1 = y \cdot a_i + z \cdot H(m) \mod q$, $c_2 = y \cdot g_i + z \cdot u_i \mod q$, and $c_3 = y \cdot r_i + w \cdot e_i \mod q$. The prover then computes a hash challenge c by hashing the concatenation of these commitments, $c = H(c_1 \parallel c_2 \parallel c_3)$. Next, the prover calculates the responses r_y, r_z, and r_w using the secret polynomial s_i, the random values, and the challenge: $r_y = y + c \cdot s_i \mod q$, $r_z = z + c \cdot t \mod q$, and $r_w = w + c \cdot \sigma_i \mod q$. The proof π consists of the tuple (c, r_y, r_z, r_w).

In the Verify procedure, the verifier recomputes the commitments c'_1, c'_2, and c'_3 using the responses and checks if these recomputed commitments, when concatenated and hashed, yield the original challenge c. Specifically, the recomputed commitments are $c'_1 = r_y \cdot a_i + r_z \cdot H(m) - c \cdot s_i \cdot a_i \mod q$, $c'_2 = r_y \cdot g_i + r_z \cdot u_i - c \cdot s_i \cdot g_i \mod q$, and $c'_3 = r_y \cdot r_i + r_w \cdot e_i - c \cdot s_i \cdot r_i \mod q$. The verifier computes $c' = H(c'_1 \parallel c'_2 \parallel c'_3)$ and checks if $c = c'$. If they match, the proof is considered valid, and the verifier returns $\top$; otherwise, it returns $\bot$.

Signing Procedure The Setup process for signature consists of selecting a cyclotomic polynomial ring and setting the parameters for a discrete Gaussian distribution. Every party generates its public and secret keys in the KeyGen algorithm. Assuming a ring $R_q = \mathbb{Z}q[X]/(\Phi_n(X))$, defined by the n-th cyclotomic polynomial $\Phi_n(X)$ and modulo q, a large prime, it picks a discrete Gaussian distribution $\mathcal{D}_{\mathbb{Z},\sigma}$ with standard deviation σ to sample error terms and secret keys. Next, in KeyGen, every party in the ring independently generates its keys: the secret key s_i is sampled from $\mathcal{D}_{\mathbb{Z}^n,\sigma}$ and the public key g_i is calculated as:

$$g_i = a_i \cdot s_i + e_i \mod q$$

where a_i is selected randomly from R_q^n and e_i is an error term from $\mathcal{D}_{\mathbb{Z}^n,\sigma}$.

The signing procedure, Sign, is initiated by the signer to produce a signature (σ_i, u_i, π) for a message m. Masking polynomial r_i is getting randomly selected to preserve the secrecy of s_i during the signing process, and commitment u_i is calculated as $u_i = r_i \cdot H(m) + e'_i \mod q$, incorporating a hash of the message and an error term e'_i, binding the message to the signature. The challenge t is derived from $H(m||u_i)$ and used to compute $\sigma_i = r_i + t \cdot s_i \mod q$, linking the secret key with the message and the commitment. Proof π demonstrates knowledge of s_i and the validity of σ_i without revealing s_i.

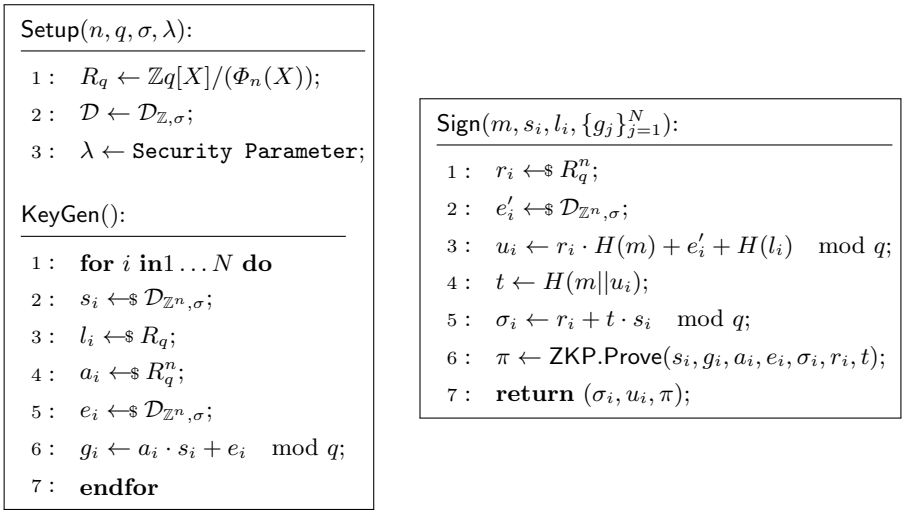

Fig. 1. Signing Process in Linkable Threshold Lattice-based Ring Signature

Verification Procedure Verification by Verify consists of checking the validity of signature (σ_i, u_i, π) by recomputing challenge t', being $H(m||u_i)$, and checking it is equal to t which was used in σ_i, and then checking the validity of π takes place so, at this point, integrity is checked and is made with knowledge of valid s_i. Finally, it is checked whether $\sigma_i \cdot H(m) - t' \cdot \sum_{j=1}^N g_j \equiv u_i \mod q$, where all the components of signature are indeed matching, and ensuring the integrity of the signature by bounding the norm via γ which represents a threshold value for the norm of the signature.

4.2 Properties of Proposed Ring Signature Scheme

The assurance of correctness, anonymity, unforgeability, linkability, and non-slanderability in a linkable ring signature scheme is paramount for its security and functionality [43]. Without these properties, the scheme's full trustworthiness and practical usability would be significantly compromised.

Theorem 5 (Correctness and Linkability). *The linkable* CRLWE*-based ring signature scheme defined in Figure 1 ensures verification correctness and linking correctness.*

Proof. Given a message m, the signer computes a masking polynomial r_i from R_q^n and a commitment u_i defined by:

$$u_i = r_i \cdot H(m) + e_i' + H(l_i) \mod q$$

where e_i' is a Gaussian error and H is a hash function. The challenge t is computed as: $t = H(m\|u_i)$, and the signature is then: $\sigma_i = r_i + t \cdot s_i \mod q$. As a

Verify$(m, (\sigma_i, u_i, \pi), g_j{}_{j=1}^{N})$:

1 : $t' \leftarrow H(m||u_i)$;
2 : ZKP.Verify(π);
3 : **if** $\sigma_i \cdot H(m) - t' \cdot \sum j = 1^N g_j \stackrel{?}{=} u_i \mod q$ **then**
4 : **if** $||\sigma_i|| \leq \gamma$ **then**
5 : $l_i \leftarrow u_i \cdot g_1^{-1} \mod q$;
6 : **if** $\exists\ k,\ l_i = g_k$ **then**
7 : **return** $\perp$;
8 : **else**
9 : **return** $\top$;
10 : **endif**
11 : **else**
12 : **return** $\perp$;
13 : **endif**
14 : **else**
15 : **return** $\perp$;
16 : **endif**

Fig. 2. Verification in Proposed Scheme

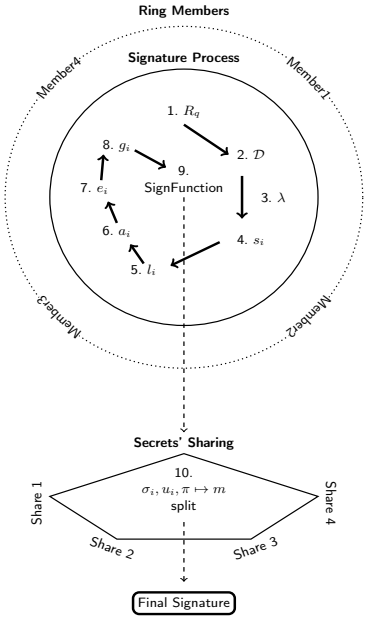

Fig. 3. Overview of Lattice-based Linkable Threshold Ring Signature Scheme (e.g., Four Members)

ZKP.Prove$(s_i, g_i, a_i, e_i, \sigma_i, r_i, t)$:

1 : $y \leftarrow_\$ R_q$; $z \leftarrow_\$ R_q$; $w \leftarrow_\$ R_q$;
2 : $c_1 \leftarrow y \cdot a_i + z \cdot H(m) \mod q$;
3 : $c_2 \leftarrow y \cdot g_i + z \cdot u_i \mod q$;
4 : $c_3 \leftarrow y \cdot r_i + w \cdot e_i \mod q$;
5 : $c \leftarrow H(c_1||c_2||c_3)$;
6 : $r_y \leftarrow y + c \cdot s_i \mod q$;
7 : $r_z \leftarrow z + c \cdot t \mod q$;
8 : $r_w \leftarrow w + c \cdot \sigma_i \mod q$;
9 : **return** (c, r_y, r_z, r_w);

ZKP.Verify$(c, r_y, r_z, r_w, s_i, g_i, a_i, e_i, \sigma_i, r_i, u_i)$:

1 : $c'_1 \leftarrow r_y \cdot a_i + r_z \cdot H(m) - c \cdot s_i \cdot a_i \mod q$;
2 : $c'_2 \leftarrow r_y \cdot g_i + r_z \cdot u_i - c \cdot s_i \cdot g_i \mod q$;
3 : $c'_3 \leftarrow r_y \cdot r_i + r_w \cdot e_i - c \cdot s_i \cdot r_i \mod q$;
4 : $c' \leftarrow H(c'_1||c'_2||c'_3)$;
5 : **if** $c \stackrel{?}{=} c'$ **then**
6 : **return** $\top$;
7 : **else**
8 : **return** $\perp$;
9 : **endif**

Fig. 4. Stern's Zero-knowledge Protocol in Ring Signature

result:

$$\begin{aligned}
\sigma_i \cdot H(m) - t' \cdot \sum_{j=1}^{N} g_j &= (r_i + t \cdot s_i) \cdot H(m) - t \cdot \sum_{j=1}^{N} g_j \\
&= r_i \cdot H(m) + t \cdot s_i \cdot H(m) - t \cdot \sum_{j=1}^{N} g_j \\
&= r_i \cdot H(m) + t \cdot (s_i \cdot H(m) - \sum_{j=1}^{N} g_j) \\
&= r_i \cdot H(m) \text{ (since } g_i = a_i \cdot s_i + e_i \text{ and sums cancel)} \\
&= u_i \mod q \text{ (by definition of } u_i)
\end{aligned}$$

The commitment u_i includes $H(l_i)$, the hash of the secret linking tag. Given that l_i is unique to each signer and securely generated, any two signatures that yield the same l_i must originate from the same signer. Therefore, this scheme ensures that signatures can be effectively linked to the same signer when appropriate, achieving both verification and linking correctness. □

Theorem 6 (Anonymity). *Under the hardness of* CRLWE*, the linkable ring signature based on this problem maintains anonymity, and no polynomial-time adversary can distinguish between signatures produced by any two distinct members of the ring with non-negligible probability.*

Proof. We define a formal security game involving an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ to model the anonymity property of the ring signature scheme. $\mathcal{C}$ runs the $\mathsf{Setup}(n, q, \sigma, \lambda)$ to initialize the scheme. $\mathcal{C}$ generates key pairs for N participants using $\mathsf{KeyGen}()$ and provides the public keys $\{g_k\}_{k=1}^{N}$ to $\mathcal{A}$, and $\mathcal{A}$ chooses two distinct indices i and j from $\{1, \ldots, N\}$ and a message m. $\mathcal{C}$ selects a random bit $b \in \{0, 1\}$. Depending on b, $\mathcal{C}$ generates a signature using either s_i or s_j by executing $\mathsf{Sign}(m, s_{b+1}, l_{b+1}, \{g_k\}_{k=1}^{N})$ to produce (σ_b, u_b, π_b). The signature is given to $\mathcal{A}$. After analyzing the signature, $\mathcal{A}$ outputs a guess b'. The advantage of $\mathcal{A}$ in this game is defined as:

$$\mathrm{Adv}_{\mathcal{A}} = \left| \Pr[b' = b] - \frac{1}{2} \right|$$

where $\Pr[b' = b]$ is the probability that $\mathcal{A}$ guesses b correctly. The CRLWE assumption states that for any polynomial-time adversary, given polynomially many samples of the form $(a(X), a(X) \cdot s(X) + e(X))$, it is computationally hard to distinguish $s(X)$ from a random element in R_q, where $e(X)$ is a small error polynomial.

Suppose there exists an adversary $\mathcal{A}$ that can win the above anonymity game with a non-negligible advantage. We will construct an algorithm $\mathcal{B}$ that solves the CRLWE problem using $\mathcal{A}$'s ability to distinguish between signatures. $\mathcal{B}$ receives a CRLWE challenge $(a(X), b(X) = a(X) \cdot s(X) + e(X))$ where $s(X)$ is either a

sample from the secret distribution or a random element in R_q. $\mathcal{B}$ embeds this challenge into the ring signature setup by setting $g_i = b(X)$ and other g_k's as random valid public keys. $\mathcal{B}$ then simulates the anonymity game with $\mathcal{A}$, using i and any other $j \neq i$. If $\mathcal{A}$ can distinguish signatures, this implies it can tell whether g_i was formed using a valid secret key or not, thus solving the CRLWE challenge. Given that $\mathcal{A}$'s ability to distinguish between two signatures would directly lead to a solution for the CRLWE problem, and under the assumption that CRLWE is hard, we conclude that no such $\mathcal{A}$ can exist. □

Theorem 7 (Unforgeability and Non-Slanderability). *The* CRLWE*-based Ring Signature scheme ensures that it is computationally infeasible for any polynomial-time adversary $\mathcal{A}$ to forge a valid signature on behalf of any group member without knowing their secret key and to forge a valid signature by combining elements of signatures from different messages not directly signed by the signer.*

Proof. Defining a formal security game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ to prove the unforgeability of the ring signature scheme.

Setup: $\mathcal{C}$ runs $\mathsf{Setup}(n, q, \sigma, \lambda)$ to initialize system parameters. Then, $\mathcal{C}$ executes $\mathsf{KeyGen}()$ to generate keys for N participants, providing $\mathcal{A}$ with all public keys $\{g_k\}_{k=1}^{N}$.

Adaptive Queries: $\mathcal{A}$ may request signatures for any messages from any participants, except for a secretly chosen target j by $\mathcal{C}$, and $\mathcal{A}$ requests signatures on any set of messages under any subset of the public keys.

Forgery Attempt: $\mathcal{A}$ outputs a message m^*, a signature (σ^*, u^*, π^*), claiming it to be from participant j, and/or claiming it to be a new valid signature. $\mathcal{C}$ checks if (σ^*, u^*, π^*) is valid under any public key g_k for the new message m^*.

Winning Condition: $\mathcal{A}$ wins if (σ^*, u^*, π^*) verifies correctly and as valid for m^* under j's public key and m^* was never signed by j during the query phase.

Any adversary $\mathcal{A}$ winning the above game with non-negligible probability implies an algorithm $\mathcal{B}$ that can solve the CRLWE problem, leading to a contradiction. Now, to check non-slanderability, assume $\mathcal{A}$ can forge signatures with non-negligible probability. Construct a reduction algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to solve an instance of the CRLWE problem.

- $\mathcal{B}$ receives a CRLWE challenge $(a(X), b(X) = a(X) \cdot s(X) + e(X))$.
- $\mathcal{B}$ embeds this challenge into the public key of j (the challenge participant), setting $g_j = b(X)$.
- $\mathcal{B}$ simulates all other aspects of the ring signature environment for $\mathcal{A}$, using valid secret keys for all but j.

If $\mathcal{A}$ produces a valid forged signature (σ^*, u^*, π^*) for j, then $\mathcal{B}$ uses components of this signature to extract information about $s(X)$, contradicting the hardness of CRLWE. The ability of $\mathcal{A}$ to forge a signature implies a breach in the CRLWE assumption. Since CRLWE is assumed hard, $\mathrm{Adv}_{\mathcal{A}}$ must be negligible, establishing that forging a signature is computationally infeasible. □

4.3 Comparison of Lattice-based Ring Signatures

Evaluation of schemes has been done on the Apple MacBook Air (2022) M2 chip, A15 Bionic (64-bit ARM-based), 8-core CPU with a base clock speed of 3.49 GHz., and 16GB memory; the value of security parameter λ is considered 512, and experiments run on two ring members. As Table 1 shows, the proposed scheme has a promising potential even by using the zero-knowledge protocol, which takes more time to compute.

Table 1. Quantitative Comparison of Related Works

Measure / Work	Signature Time (ms)	Verification Time (ms)	ZKP Utilization	Signature Size (kb)
Gao et al., 2019 [6]	1.2	1.3	✗	0.7
Mundhe et al., 2020 [7]	0.011	0.005	✗	8.313
Ye et al., 2022 [8]	0.066	0.062	✗	8.617
Cayrel et al., 2010 [9]	0.465	0.0018	✓	9.01
Xie et al., 2022 [10]	0.019	0.025	✗	0.1
Ours	0.01	0.008	✓	2.8

To ensure resistance against quantum attacks for our scheme, we use the following preferred inputs (minimum) with exact values as an example.

- Degree n: Choose $n = 2^{11}$, which is a common parameter in Ring-LWE based schemes for efficiency.
- Prime q: Select a prime q slightly above 2^{32}, such as $q = 4294967311$, to allow efficient polynomial multiplication.
- Matrix A: For an $m \times m$ matrix A, with $m \approx 512$, a security level analogous to AES-128 against quantum adversaries can get achieved.
- Error Distribution D_θ^m: Set the standard deviation θ of the discrete Gaussian noise distribution to around $\frac{q}{\sqrt{2\pi}} \cdot \texttt{noise rate}$, with a noise rate of approximately 1/8 or 1/16.

These parameters should be chosen with caution, and these values are based on recommendations for post-quantum cryptography; note that as the field size and degree increase, the computational complexity of the scheme also increases.

4.4 Secret Sharing Scheme

Consider a threshold value t, defining the minimum number of participants that can generate valid signatures. Using a secret sharing scheme, our scheme shares the polynomial $r(x)$ in $t+1$ shares. In our case, shares are given out to $t+1$ participants, and only t participants can reconstruct $r(x)$. Each participant then calculates his shares of $s'(x)$ and $s''(x)$ based on his share of $r(x)$. At the time

of signature generation, t or more participants combine their shares of $s'(x)$ and $s''(x)$ to produce the final signature. Our secret sharing scheme, along with Algorithm 3, is formulated from MR-LWE and described below: Both parties generate a randomly shared polynomial $f(x)$ of degree $p-1$, with coefficients from $\mathbb{Z}_q$, and compute their own share of the secret vector s by substituting their own participant number into the polynomial.

Then the participant creates his noisy share s'_i by multiplying his share of the secret vector $\mathbf{s}$ with the matrix $\mathbf{A}$ and adding his share of the noise vector $e(\mathrm{mod}q)$. More precisely, the i th participant computes $s'_i = A_i s + e_i(\mathrm{mod}q)$, where A_i denotes the i th row of matrix $\mathbf{A}$ and e_i the ith element of vector e. The participant then sends s'_i to all other participants.

The information amalgamation process through the noisy shares can be passed on without directly revealing the share that belongs to each participant. Even if the noisy shares do not directly represent the share of the original shares, the reconstruction process will remain free from the intimate revelation of the information. In any case, the participant computes the difference between their noisy share s'_i and those of others, that is, s'_j. Participant i computes $\delta_{ij} = s'_i - s'_j \mod q$, where δ_{ij} is the share difference between the participant i and the participant j whose shares are noisy. To mask his own noisy share, each participant adds a random vector m_i drawn from the error distribution $\mathcal{D}_\theta^m$. We thus get $s_i = s'_i$.

Afterward, every participant sums the values of masked noisy shares $\mathbf{s}_i$ from others element-wise. Participant i computes: $s = \sum_i s_i \mod q$, where $\sum_i s_i$.

Each party applies error correction matched to the distribution of errors $\mathcal{D}_\theta^m$ on the amalgamated masked noisy shares s to heal the errors and decode its shares. The particular error correction techniques are dependent on the distribution of errors $\mathcal{D}_\theta^m$ and may include statistical techniques.

Finally, the participants reconstruct the secret vector $\mathbf{s}$ from the shares using Lagrange interpolation, in which the corrected shares are used as interpolation points. The corrected shares are used such that, through the property of Lagrange interpolation, the reconstructed secret vector s is returned.

The result of this algorithm is the secret vector $\mathbf{s}$ following secret reconstruction: secure aggregation, error correction, and secret reconstruction steps.

The last part of Algorithm 3 illustrating the verification protocol for our secret sharing scheme. Incorporating these verification measures guarantees the integrity and security of each phase of the scheme.

5 An Efficient Common Modulus Attack on RSA

During this research, we intended to base the secret sharing scheme on RSA to have a multiplicative homomorphic structure [44] but found a vulnerability, and the way of exploiting it and attacking will be detailed here as an additional contribution. Suppose that two RSA public-private key pairs are generated in a common modulus for two users. The following lemma and theorem demonstrate that each user can derive a value identical to the private key of the other user

Algorithm 3 MR-LWE-based Secret Sharing

procedure ShareSecret($\texttt{msg}, n, p, q$)
  Input: $\texttt{msg} \in \mathbb{Z}_q$, number of participants n, polynomial degree p, modulus q
  Output: Shares $\mathbf{s}_i$ for $i = 1, \ldots, n$
  for each participant $i = 1$ to n **do**
    Initialize $f_i(x)$ to a random polynomial of degree $\leq p-1$ with constant $\texttt{msg}$
    Assign x_i a unique value in $\mathbb{Z}_q$
    Compute share $\mathbf{s}_i = f_i(x_i)$
  end for
end procedure
procedure ReconstructSecret($\mathbf{A}, \mathbf{e}, q, \texttt{msg}, n$)
  Input: $\mathbf{A} \in \mathbb{Z}_q^{m \times m}$, $\mathbf{e} \in \mathbb{Z}_q^m$, $q \in \mathbb{N}$, $\texttt{msg} \in \mathbb{Z}_q$, number of participants n
  Output: Secret $\mathbf{s} \in \mathbb{Z}_q^m$
  for each participant $i = 1$ to n **do**
    Compute noisy share $\mathbf{s}'_i = \mathbf{A}\mathbf{s}_i + \mathbf{e}_i \mod q$
  end for
  for each participant $i = 1$ to n **do**
    for each participant $j = 1$ to n **do**
      if $i \neq j$ **then**
        Broadcast $\mathbf{s}'_i$ to participant j
      end if
    end for
  end for
  for each participant $i = 1$ to n **do**
    for each participant $j = 1$ to n **do**
      if $i \neq j$ **then**
        Compute $\delta_{ij} = \mathbf{s}'_i - \mathbf{s}'_j \mod q$
      end if
    end for
    Sample $\mathbf{m}_i$ from D_θ^m
    Adjust share $\mathbf{s}'_i = \mathbf{s}'_i + \mathbf{m}_i \mod q$
  end for
  Initialize $\mathbf{s}_{\text{agg}} = \mathbf{0}$
  for each participant $i = 1$ to n **do**
    $\mathbf{s}_{\text{agg}} = \mathbf{s}_{\text{agg}} + \mathbf{s}'_i \mod q$
  end for
  Apply error correction to $\mathbf{s}_{\text{agg}}$ to obtain $\mathbf{s}$
  return $\mathbf{s}$
end procedure
procedure VerifySecret($\mathbf{A}, \mathbf{e}, q, p, \mathcal{D}_\theta^m, \texttt{msg}$)
  Input: Matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times m}$, vector $\mathbf{e} \in \mathbb{Z}_q^m$, modulus $q \in \mathbb{N}$,
  polynomial degree $p \in \mathbb{N}$, distribution $\mathcal{D}_\theta^m$, $\texttt{msg} \in \mathbb{Z}_q$
  Output: Status $\in \{\text{Valid}, \text{Invalid}\}$
  Check $f_i(x)$ degree $\leq p-1$ and constant equals $\texttt{msg}$ for all i, coefficients $\in \mathbb{Z}_q$
  Verify $\mathbf{A}$ structure and elements in $\mathbb{Z}_q$
  Validate $\mathbf{s}'_i \in \mathbb{Z}_q^m$ for all noisy shares $\mathbf{s}'_i$
  Verify received $\mathbf{s}'_j \in \mathbb{Z}_q^m$ and count for all j
  Check masks $\mathbf{m}_i$ sampled from $\mathcal{D}_\theta^m$ for all i
  Validate aggregated $\mathbf{s}_{\text{agg}} \in \mathbb{Z}_q^m$
  Verify error correction against $\mathcal{D}_\theta^m$
  Check reconstruction algorithm correctness
  if all checks pass **then**
    return Valid
  else
    return Invalid
  end if
end procedure

during the decryption process. Consequently, they can decrypt any ciphertext initially encrypted for the other user.

Theorem 8. *Let (e_1, d_1) and (e_2, d_2) be two public-private key pairs in a common modulus $N = pq$, and s_2 is the second Bézout coefficient for gcd($\frac{e_1 d_1 - 1}{g}, e_2$), i.e. $\frac{e_1 d_1 - 1}{g} \cdot s_1 + e_2 \cdot s_2 = 1$, where $g = \gcd(e_1 d_1 - 1, e_2)$. Then, $s_2 = d_2 \bmod \phi(N)$, i.e. s_2 is equivalent to the private key d_2 in the decryption of RSA.*

Proof. Since $e_1 d_1 = 1 \bmod \phi(N)$, so

$$e_1 d_1 - 1 = k \cdot \phi(N); \quad k \neq 0 \tag{1}$$

Based on (1), set $g = \gcd(e_1 d_1 - 1, e_2) = \gcd(k \cdot \phi(N), e_2)$. Since $\gcd(e_2, \phi(N)) = 1$, we have $g = \gcd(k, e_2)$, which means that $g|k$, and hence $k' = \frac{k}{g}$ is an integer. On the other hand, since $g = \gcd(e_1 d_1 - 1, e_2)$, it can be concluded that $\gcd(\frac{e_1 d_1 - 1}{g}, e_2) = 1$. Therefore, we have:

$$\gcd(\frac{e_1 d_1 - 1}{g}, e_2) = \gcd(\frac{k \cdot \phi(N)}{g}, e_2) = \gcd(\frac{k}{g} \cdot \phi(N), e_2) = \gcd(k' \cdot \phi(N), e_2) = 1 \tag{2}$$

Using the extended Euclidean algorithm, we can find Bézout coefficients s_1 and $s_2 \in \mathbb{Z}$ such that $k' \phi(N) \cdot s_1 + e_2 \cdot s_2 = 1$. This equation implies that,

$$e_2 \cdot s_2 = k'' \cdot \phi(N) + 1 \tag{3}$$

where $k'' = -k' \cdot s_1$ is an integer. Equation (3) shows that s_2 is the modular inverse of e_2 and is equivalent to d_2 in the decryption of ciphertexts encrypted by e_2.

To verify the claim, assume we have the ciphertext C of a message m encrypted by e_2, written as $C = m^{e_2} \pmod N$. To decrypt C, all we need to do the following calculation:

$$C^{s_2} = (m^{e_2})^{s_2} = m^{k'' \phi(N) + 1} = m \mod N \tag{4}$$

So, we can use s_2 for decryption instead of d_2. □

Figure 5 presents a summary of our attack. According to Theorem 8, this algorithm is guaranteed to succeed. It is evident that the algorithm in Figure 5 is fast, as it only involves one multiplication, one gcd computation, one division, and finally, one Bézout algorithm.

Remark 1. The complexity of the attack based on Theorem (8) is $O(logN)$ since it requires two (extended) Euclidean Algorithm runs. It has no extra condition on the key pairs, the common modulus, or the ciphertext to be decrypted.

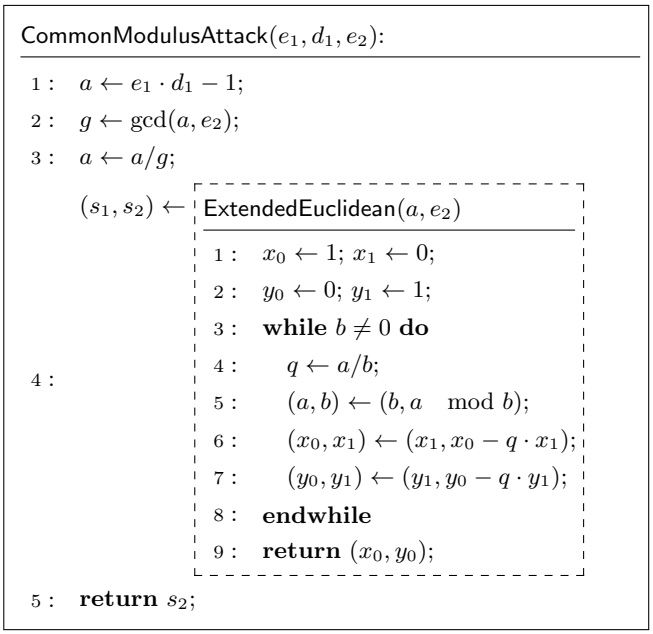

Fig. 5. The Proposed Common Modulus Attack

Remark 2. The proposed attack is valid also when (e_1, d_1) are in Euler RSA system and (e_2, d_2) are Carmichael or Euler. If both have Euler secret keys, this is already proved in Theorems 3 and 8. To prove the second type, suppose that $\phi(N) = t.\lambda(N)$. Set $g = \gcd(e_1 d_1 - 1, e_2) = \gcd(k.\phi(N), e_2) = \gcd(k.t.\lambda(N), e_2)$. Note that $\gcd(e_2, \lambda(N)) = 1$. The rest of the proof is similar to Theorem 8 by replacing k with $k.t$. To find s_2 in these types, Figure 5 is run, too.

Remark 3. We prove that the proposed attack is valid when (e_1, d_1) are in Carmichael's RSA system and (e_2, d_2) are either Carmichael or Euler. The following theorems prove this:

Theorem 9. *The attack is valid if (e_1, d_1) are in Carmichael RSA system and (e_2, d_2) are either in Carmichael or Euler.*

Proof. If both parties have Carmichael's secret keys, it can be demonstrated easily by substituting $\lambda(N)$ for $\phi(N)$ in equations (1), (2), and (3). To prove the second type, it should be noted that $\gcd(e_2, \lambda(N)) = 1$ because $\gcd(e_2, \phi(N)) = 1$ and $\lambda(N)$ divides $\phi(N)$. The remainder of the proof is similar to the previous one. To determine s_2 in these instances, Algorithm 5 is applied. □

Corollary 1. *The attack proposed in Theorem 9 can be applied to the RSA system without considering Euler or Carmichael using Theorems of 9 and 10.*

Theorem 10. *Let (e_1, d_{1p}, d_{1q}) and (e_2, d_{2p}, d_{2q}) be two public-private key pairs for CRT decryption in a common modulus $N = pq$. Then, Algorithm 5 is valid to compute $s_{2p} = d_{2p} \bmod (p-1)$ and $s_{2q} = d_{2q} \bmod (q-1)$, i.e. s_{2p} and s_{2q} are equivalent to the private keys d_{2p} and d_{2q} respectively in the CRT decryption of RSA.*

Proof. In this case, the algorithm in Figure 5 is run twice with inputs of the first (e_1, d_{1p}, d_{1q}) and then (e_2, d_{2p}, d_{2q}). In the case of (e_1, d_{1p}, e_2), we have $e_1 d_{1p} - 1 = k \cdot (p-1)$. Set $g = \gcd(e_1.d_{1p} - 1, e_2) = \gcd(k \cdot (p-1), e_2)$. Since $\gcd(e_2, \phi(N)) = 1$ and $(p-1)|\phi(N)$ then $\gcd(e_2, p-1) = 1$. Similar to the discussion in the proof of Theorem 8, we can write

$$\gcd(\frac{e_1 d_{1p} - 1}{g}, e_2) = \gcd(\frac{k \cdot (p-1)}{g}, e_2) = \gcd(\frac{k}{g} \cdot (p-1), e_2) = \gcd(k' \cdot (p-1), e_2) = 1 \tag{5}$$

So, there are s_{1p} and s_{2p} such that

$$k'(p-1) \cdot s_{1p} + e_2 \cdot s_{2p} = 1 \rightarrow e_2 \cdot s_{2p} = k'' \cdot (p-1) + 1 \rightarrow s_{2p} = d_{2p} \mod (p-1) \tag{6}$$

The proof for d_{2q} is the same as for the previous case. □

Remark 4. If an attacker has the secret keys of RSA in the CRT model (without knowing the factors), then they can attack other users that use the same module. This becomes from the fact that if $\gcd(\phi(N), e_2) = 1$ or $\gcd(\lambda(N), e_2) = 1$ then we can conclude that $\gcd(p-1, e_2) = \gcd(q-1, e_2) = 1$. On the other side, we know that $\gcd(d_{p1}, p-1) = \gcd(d_{q1}, q-1) = 1$ is held. Therefore, we can write $\gcd(e_2.d_{p1}, p-1) = \gcd(e_2.d_{q1}, q-1) = 1$, $e_2.d_{p1} = 1 \mod (p-1)$ and $e_2.d_{q1} = 1 \mod (q-1)$. So, the attacker can use Algorithm **??** to recover message m.

Corollary 2. *Based on Corollary 1 and Remark 4, if an attacker has a public-private key pair in any of the Euler, Carmichael, and black-boxed CRT RSA models, they can attack the ciphertexts of other users in a common modulus in any of the Euler, Carmichael, and black-boxed RSA models.*

Generalization and Factoring Algorithm

In this section, we present a generalized version of the presented attack. Unlike before, the attacker no longer needs to be the owner of the (e_1, d_1) key pair in modulus N. Instead, the attacker only needs to have leaked information of the form (α, β), where $f(\alpha) = \beta \mod (\phi(N))$ holds under the condition $k \neq 0$ in $f(\alpha) = \beta + k.\phi(N)$. Here, f can be any function composed of multiplication, division, summation, extraction, or exponentiation.

In the our attack case, assume that $\alpha = d_1$, $\beta = 1$, and $f(x) = e_1 \cdot x \bmod (\phi(N))$. Then we have $f(\alpha) = \beta = 1$. For the cyclic attack, we have $\alpha = \beta = e$ and $f(x) = x^k$ such that $f(\alpha) = \beta$. By computing $f(\alpha) - \beta = k \cdot \phi(N)$, the

attacker can apply the proposed attack in Theorem 8. This yields $\frac{f(\alpha)-\beta}{k} = \phi(N)$ and $g = \gcd(f(\alpha)-\beta, e_2)$. The remaining steps of the attack are identical to those outlined in Theorem 8. To apply this generalized attack, follow the algorithm proposed in Figure 5. In the generalized version of Theorem 3, suppose that the attacker knows $f(\alpha) = \beta \mod (p-1)$ and $g(\alpha') = \beta' \mod (q-1)$. In this case, Figure 5 is run twice to return s_{2_p} and s_{2_q}.

Now, we propose an algorithm for breaking RSA encryption, which is inspired by Figure 5 and can be executed without prior knowledge of the RSA secret key. To do this, we can run the algorithm of Figure 5 with only the knowledge of the public key N to find the function $f(x)$. Once we have found the function f, we can apply Figure 5's algorithm to break RSA encryption successfully. Line 11 in Algorithm 5 guarantees $k \neq 0$ in relation $\beta = a \cdot \alpha^k + k \cdot \lambda(N)$.

In the worst case, the algorithm's running time is equivalent to the cyclic attack if we set $\alpha = e$ and $f(x) = x^k$. However, unlike a cyclic attack, Figure 5 can be used for factoring in cases beyond RSA. This means that Figure 5 does not need the value of e, which can be an added advantage. Moreover, the algorithm of Figure 5 can run be parallelized for multiple α and β, a noteworthy feature.

GeneralizedAttack(α, β, f, e_2):

1 : $b \leftarrow f(\beta)$; $a \leftarrow \alpha \cdot b - 1$;
2 : $g \leftarrow \gcd(a, e_2)$; $a \leftarrow a/g$;
3 : $x_0 \leftarrow 1$; $x_1 \leftarrow 0$;
4 : $y_0 \leftarrow 0$; $y_1 \leftarrow 1$;
5 : **while** $e_2 \neq 0$ **do**
6 : $q \leftarrow a/e_2$;
7 : $(a, e_2) \leftarrow (e_2, a \mod e_2)$;
8 : $(x_0, x_1) \leftarrow (x_1, x_0 - q \cdot x_1)$;
9 : $(y_0, y_1) \leftarrow (y_1, y_0 - q \cdot y_1)$;
10 : **endwhile**
11 : **return** y_0;

Fig. 6. Generalization of Attack

FindFunction(N):

1 : $k \leftarrow 1$; $m \leftarrow 2$;
2 : $\alpha, \beta, a \leftarrow\$ \mathbb{Z}_N$;
3 : $c_1 \leftarrow m^{a \cdot \alpha} \mod N$;
4 : $c_2 \leftarrow m^{\beta} \mod N$;
5 : **while** $c_1 \neq c_2$ **do**
6 : $k \leftarrow k + 1$;
7 : $c_1 \leftarrow c_1^{\alpha} \mod N$;
8 : **endwhile**
9 : **if** $\beta = a \cdot \alpha^k$ **then**
10 : $k \leftarrow k + 1$;
11 : $c_1 \leftarrow c_1^{\alpha} \mod N$;
12 : go to line 8;
13 : **endif**
14 : **return** $\alpha, \beta, f(x) = a \cdot x^k$;

Fig. 7. Finding Function f

To factorize N, we can use the algorithm's function. By computing $\gcd(m^{f(\alpha)} - m^{\beta}, N)$ using this function, we can find the factors of N. Another way to factorize N using the algorithm of Figure 5 is to compute $f(\alpha) - \beta = k\lambda(N)$, where

$k \neq 0$, and then run the Miller algorithm to recover factors of N. The complexity order of Figure 5's algorithm is equivalent to the order of cyclic attack in the worst case, as the cyclic attack is a special case of ours. While the complexity order of the cyclic attack on the RSA cryptosystem is not explicitly stated in the provided search results, we can say that our factoring attack can achieve a complexity order of less than the cyclic attack's complexity order.

Table 2. Comparison of the Common Modulus Attacks on RSA

Attack Aim	Data required	Success Rate	Conditions	Complexity	Reference
Factoring N	$((e_1, d_1), N)$	Probabilistic (1/2)	-	$O(log^3 N)$	[22]
Factoring N	$((e_1, d_1), N)$	Deterministic	$e_1 d_1 < N^2$	$O(log^9 N)$	[21]
Factoring N	$((e_1, d_1), N)$	Deterministic	$e_1 d_1 < N^{3/2}$	$O(log^2 N)$	[21]
Obtaining Plaintext[5]	$(e_1, e_2, m^{e_1}, m^{e_2}, N)$	Deterministic	$\gcd(e_1, e_2) = 1$	$O(logN)$	[20]
Obtaining Plaintext	$((e_1, d_1), e_2, N)$	Deterministic	-	$O(logN)$	Ours
Obtaining Plaintext	$((e_1, d_{p1}, d_{q1}), e_2, N)$	Deterministic	-	$O(logN)$	Ours
Obtaining Plaintext	$(e, N), f(\alpha)$	Deterministic	-	$O(logN)$	Ours

References

[1] *Announcing PQC Candidates to be Standardized, Plus Fourth Round Candidates | CSRC — csrc.nist.gov*. https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4. [Accessed 21-05-2024].

[2] Xingye Lu, Man Ho Au, and Zhenfei Zhang. "Raptor: A Practical Lattice-Based (Linkable) Ring Signature". In: *Applied Cryptography and Network Security*. Ed. by Robert H. Deng et al. Springer. 2019, pp. 110–130.

[3] Yongli Tang et al. "Identity-Based Linkable Ring Signature on NTRU Lattice". In: *Security and Communication Networks* 2021 (2021). Article ID 9992414.

[4] Shiyuan Xu et al. "A Lattice-Based Ring Signature Scheme to Secure Automated Valet Parking". In: *Wireless Algorithms, Systems, and Applications*. Ed. by Zhe Liu, Fan Wu, and Sajal K. Das. Vol. 12938. Lecture Notes in Computer Science. Springer. 2021, pp. 70–83.

[5] Chunhong Jiao and Xinyin Xiang. "Anti-Quantum Lattice-Based Ring Signature Scheme and Applications in VANETs". In: *Entropy* 23.10 (2021), p. 1364.

[6] Wen Gao et al. "Lattice-based deniable ring signatures". In: *International Journal of Information Security* 18.3 (Aug. 2018), pp. 355–370. ISSN: 1615-5270.

[7] Pravin Mundhe et al. "Efficient Lattice-Based Ring Signature for Message Authentication in VANETs". In: *IEEE Systems Journal* 14.4 (Dec. 2020), pp. 5463–5474. ISSN: 2373-7816.

[5] This attack just decrypts the ciphertexts generated by encryption of the same message under the two public keys.

[8] Qing Ye et al. "Efficient Lattice-Based Ring Signature Scheme without Trapdoors for Machine Learning". In: *Computational Intelligence and Neuroscience* 2022 (Sept. 2022). Ed. by Le Sun, pp. 1–13. ISSN: 1687-5265.

[9] Shangping Wang, Ru Zhao, and Yaling Zhang. "Lattice-based ring signature scheme under the random oracle model". In: *International Journal of High Performance Computing and Networking* 11.4 (2018), p. 332. ISSN: 1740-0570.

[10] Yi-Yang Xie, Xiu-Bo Chen, and Yi-Xian Yang. "A New Lattice-Based Blind Ring Signature for Completely Anonymous Blockchain Transaction Systems". In: *Security and Communication Networks* 2022 (2022). Article ID 4052029, pp. 1–12.

[11] Pierre-Louis Cayrel et al. "A Lattice-Based Threshold Ring Signature Scheme". In: *Progress in Cryptology – LATINCRYPT 2010*. Springer Berlin Heidelberg, 2010, pp. 255–272. ISBN: 9783642147128.

[12] Carlos Aguilar Melchor, Pierre-Louis Cayrel, and Philippe Gaborit. "A New Efficient Threshold Ring Signature Scheme Based on Coding Theory". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 1–16. ISBN: 9783540884033.

[13] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[14] *Generate Keys and Certificates for SSO - Google Workspace Admin Help — support.google.com*. https://support.google.com/a/answer/6342198?hl=en. [Accessed 21-05-2024].

[15] *FAQ for the Technically Inclined — core.telegram.org*. https://core.telegram.org/techfaq. [Accessed 21-05-2024].

[16] *Hardening OpenVPN Security | OpenVPN — openvpn.net*. https://openvpn.net/community-resources/hardening-openvpn-security. [Accessed 21-05-2024].

[17] Élie Gouzien and Nicolas Sangouard. "Factoring 2048-bit RSA integers in 177 days with 13 436 qubits and a multimode memory". In: *Physical review letters* 127.14 (2021), p. 140503.

[18] Majid Mumtaz and Luo Ping. "Forty years of attacks on the RSA cryptosystem: A brief survey". In: *Journal of Discrete Mathematical Sciences and Cryptography* 22.1 (2019), pp. 9–29.

[19] David Chaum. "Blind signature system". In: *Advances in cryptology*. Springer. 1984, pp. 153–153.

[20] Dan Boneh et al. "Twenty years of attacks on the RSA cryptosystem". In: *Notices of the AMS* 46.2 (1999), pp. 203–213.

[21] Jean-Sébastien Coron and Alexander May. "Deterministic polynomial-time equivalence of computing the RSA secret key and factoring". In: *Journal of Cryptology* 20.1 (2007), pp. 39–50.

[22] Gary L Miller. "Riemann's hypothesis and tests for primality". In: *Journal of computer and system sciences* 13.3 (1976), pp. 300–317.

[23] Johan Hastad. "Solving simultaneous modular equations of low degree". In: *siam Journal on Computing* 17.2 (1988), pp. 336–341.
[24] Don Coppersmith. "Small solutions to polynomial equations, and low exponent RSA vulnerabilities". In: *Journal of cryptology* 10.4 (1997), pp. 233–260.
[25] Mengce Zheng, Honggang Hu, and Zilong Wang. "Generalized cryptanalysis of RSA with small public exponent". In: *Science China Information Sciences* 59.3 (2016), pp. 1–10.
[26] Michael J Wiener. "Cryptanalysis of short RSA secret exponents". In: *IEEE Transactions on Information theory* 36.3 (1990), pp. 553–558.
[27] Dan Boneh and Glenn Durfee. "Cryptanalysis of RSA with private key d less than N/sup 0.292". In: *IEEE transactions on Information Theory* 46.4 (2000), pp. 1339–1349.
[28] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. "Factoring polynomials with rational coefficients". In: *Mathematische annalen* 261.ARTICLE (1982), pp. 515–534.
[29] Dan Boneh and Glenn Durfee. "New results on the cryptanalysis of low exponent RSA". In: *IEEE Transactions on Information Theory* 46.4 (2000), pp. 1339–1349.
[30] Majid Mumtaz and Luo Ping. "Cryptanalysis of a Special Case of RSA Large Decryption Exponent Using Lattice Basis Reduction Method". In: *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*. IEEE. 2021, pp. 714–720.
[31] Majid Mumtaz and Luo Ping. "An improved cryptanalysis of large RSA decryption exponent with constrained secret key". In: *International Journal of Information and Computer Security* 14.2 (2021), pp. 102–117.
[32] Meryem Cherkaoui Semmouni, Abderrahmane Nitaj, and Mostafa Belkasmi. "Cryptanalysis of RSA with smooth prime sum". In: *Journal of Discrete Mathematical Sciences and Cryptography* (2022), pp. 1–21.
[33] Marco Cesati. "A new idea for RSA backdoors". In: *arXiv preprint arXiv:2201.13153* (2022).
[34] Adeline Langlois and Damien Stehle. *Worst-Case to Average-Case Reductions for Module Lattices*. Cryptology ePrint Archive, Paper 2012/090. 2012.
[35] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by Henri Gilbert. Springer Berlin Heidelberg, 2010, pp. 1–23. ISBN: 978-3-642-13190-5.
[36] David Balbás. *The Hardness of LWE and Ring-LWE: A Survey*. Cryptology ePrint Archive, Paper 2021/1358. 2021.
[37] Xueli Wang, Yu Chen, and Xuecheng Ma. *Adding Linkability to Ring Signatures with One-Time Signatures*. Cryptology ePrint Archive, Paper 2019/371. 2019.
[38] Oded Regev. "On Lattices, Learning with Errors, Random Linear Codes, and Cryptography". In: *Proceedings of the Thirty-Seventh Annual ACM*

Symposium on Theory of Computing. STOC '05. Baltimore, MD, USA: Association for Computing Machinery, 2005, pp. 84–93. ISBN: 1581139608.

[39] Chris Peikert. *How (Not) to Instantiate Ring-LWE*. Cryptology ePrint Archive, Paper 2016/351. 2016.

[40] Shahriar Ebrahimi and Siavash Bayat-Sarmadi. "Lightweight and DPA-Resistant Post-Quantum Cryptoprocessor based on Binary Ring-LWE". In: *2020 20th International Symposium on Computer Architecture and Digital Systems (CADS)*. 2020, pp. 1–6.

[41] Phong Q. Nguyen. "Lattice Reduction". In: *Encyclopedia of Cryptography and Security*. Ed. by Henk C. A. van Tilborg. Boston, MA: Springer US, 2005, pp. 346–347. ISBN: 978-0-387-23483-0.

[42] Jacques Stern. "A new identification scheme based on syndrome decoding". In: *Advances in Cryptology — CRYPTO' 93*. Lecture notes in computer science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 13–21.

[43] Maxime Buser et al. "Post-quantum ID-Based Ring Signatures from Symmetric-Key Primitives". In: *Applied Cryptography and Network Security*. Ed. by Giuseppe Ateniese and Daniele Venturi. Springer. 2022, pp. 892–912.

[44] Sathi Sarveswara Reddy, Sharad Sinha, and Wei Zhang. "Design and Analysis of RSA and Paillier Homomorphic Cryptosystems Using PSO-Based Evolutionary Computation". In: *IEEE Transactions on Computers* (2023), pp. 1–14.