

Compact Lattice-Based NIZK Arguments for Set Membership and Ring Signatures without RO

Nam Tran^{1,2*}, Khoa Nguyen¹, Dongxi Liu²,
Josef Pieprzyk^{2,3}, and Willy Susilo¹ **

¹ University of Wollongong, Northfields Avenue, Wollongong, NSW 2522, Australia
ndt141@uowmail.edu.au, khoa@uow.edu.au, wsusilo@uow.edu.au

² CSIRO Technology, Marsfield, 26 Pembroke Road, NSW 2122, Australia

³ Institute of Computer Science, Polish Academy of Sciences, Poland
Dongxi.Liu@csiro.au, Josef.Pieprzyk@csiro.au

Abstract. Zero-knowledge proofs of set membership underpin privacy-preserving constructions such as ring signatures and anonymous credentials. Existing succinct constructions rely mainly on the Fiat–Shamir transform in the Random Oracle Model (ROM), while standard-model non-interactive proofs from post-quantum assumptions remain either generic and inefficient or asymptotically compact yet concretely impractical. A key obstacle is that existing lattice-based zero-knowledge systems operate over a single ambient modulus, forcing heterogeneous components to be homogenized, inflating parameters and weakening reductions. We introduce the first *compact lattice-based NIZK arguments for set membership in the standard model* with proof size logarithmic in the set cardinality. Our construction matches the logarithmic proof size of accumulator-based ROM constructions while achieving post-quantum security without random oracles. The main technical ingredient is a new trapdoor Σ -protocol supporting linear relations modulo multiple heterogeneous moduli, allowing such relations to be handled at their native moduli without homogenization. This yields a modular approach to compact proofs compatible with lattice accumulators.

As an application, we construct lattice-based ring signatures of size $O(\log R) \cdot \tilde{O}(\lambda^2)$ bits, improving the dependence on the security parameter λ quadratically over the plain-model construction of Chatterjee et al. (CRYPTO 2021) while retaining optimal logarithmic dependence on the ring size R . Our construction is in the CRS model, which partly enables this improvement. The scheme achieves statistical anonymity and unforgeability under standard Module-LWE and Module-SIS assumptions. We also develop two additional tools of independent interest: (i) a generalized Merkle-tree accumulator over module lattices with base- B decomposition, enabling finer efficiency–assumption trade-offs; and (ii) a message-binding technique that removes the need for costly lattice one-time signatures in standard-model ring signatures.

* Nam Tran is supported by CSIRO Data61 (currently CSIRO Technology) PhD Scholarship and CSIRO Data61 Top-up Scholarship.

** Willy Susilo is supported by the Australian Research Council Australian Laureate Fellowship FL230100033.

1 Introduction

Zero-knowledge proofs of *set membership* capture a fundamental task: a prover convinces a verifier that a secret x belongs to a known finite set $S = \{x_1, \dots, x_n\}$, revealing nothing about x beyond membership. In many applications the prover additionally holds a witness w for a relation R , and the goal is to prove in zero knowledge the compound statement “ $x \in S$ and $(x, w) \in R$.” This primitive lies at the core of modern privacy-preserving cryptography. As Rivest *et al.* observed for ring signatures [53], anonymity can be expressed as a membership proof that the signer holds one of the secret keys in a public ring. In anonymous credential systems [18, 11], users must demonstrate possession of a valid credential without revealing which one. In cryptocurrencies, a spender shows that his/her coin belongs to a public UTXO set without linking it to past activity. Across these settings, membership proofs serve as the abstraction that turns algebraic structure into anonymity, unlinkability, and accountable privacy. More generally, set membership is the canonical “one-out-of-many” statement that sits at the boundary between succinct proof systems and anonymity-preserving cryptographic design, and it repeatedly reappears whenever one wants to hide an index while proving structured correctness. From a theoretical perspective, they provide a canonical interface between combinatorial objects (sets, rings, accumulators) and zero-knowledge reasoning.

The ROM vs. standard model. Most efficient membership proofs in practice are obtained by applying the Fiat–Shamir (FS) transform [24] to interactive arguments in the Random Oracle Model (ROM). This paradigm yields succinct proofs and clean protocols, but only heuristic security guarantees. By contrast, standard-model non-interactive arguments for membership remain rare and, until recently, largely disconnected from post-quantum assumptions. For example, Lipmaa and Parisella [39] constructed pairing-based (non-)membership NIZKs in the standard model, but their techniques are not post-quantum and may require linear-size public parameters. More broadly, there is a persistent gap between what is asymptotically achievable in the ROM and what is currently known in the standard model. ROM-based constructions exploit strong structural properties of specialized protocols, whereas standard-model approaches often rely on generic compilers that obscure this structure and introduce substantial overhead. Bridging this divide—while retaining asymptotic succinctness under post-quantum assumptions—has therefore become a central challenge in modern zero-knowledge, especially as lattice-based components with incompatible parameter regimes are increasingly composed in larger systems.

Generic NIZKs for NP. Recent progress on standard-model NIZKs for general NP relations has established important feasibility milestones. Peikert and Shiehian [50] constructed the first such systems from LWE; Waters [56] revisited the hidden-bit paradigm; and Waters, Wee, and Wu [57] developed a dual-mode NIZK with linear-size CRS. These works demonstrate that succinct post-quantum NIZKs are theoretically attainable and provide powerful general-purpose tools. However, when specialized to structured relations such as set

membership, generic encodings incur inherent asymptotic penalties: expressing $x \in S$ requires reducing to a circuit or NP relation whose size grows with $|S|$, leading to proofs far larger than the $\log |S|$ complexity achieved by accumulator-based protocols. From an asymptotic viewpoint, this suggests that membership proofs should not be treated merely as arbitrary NP statements, but rather as structured relations that demand dedicated cryptographic techniques.

Ring signatures. The asymptotic demands become sharper in the setting of *ring signatures*, where the ideal signature size scales as $O(\log N)$ in the ring size N . Classical constructions achieved this goal without random oracles using one-out-of-many techniques (e.g., [29,1]), illustrating that optimal asymptotics are possible when protocol structure is carefully exploited. In the post-quantum setting, lattice-based schemes have demonstrated feasibility in the standard model via ring trapdoor functions [10], but the signature size grows linearly with the ring size. Subsequent logarithmic-size lattice-based constructions have followed several approaches, including accumulator-based membership proofs [34] and specialized one-out-of-many or set-membership proofs [23,44,42]. Code-based [48], isogeny-based [8], multivariate-based [30] and block cipher/hash-based [31,20,27] approaches achieve similar asymptotics but rely on the (Q)ROM. A notable plain-model advance by Chatterjee *et al.* [17] obtained compact post-quantum ring signatures from LWE, yet their construction requires $\Theta(\lambda)$ repetitions of a subprotocol with $\tilde{O}(\lambda^2)$ cost, resulting in signatures of size $\tilde{O}(\lambda^4)$ bits. These results highlight a broader theme: achieving strong asymptotic scaling in the standard model remains technically demanding, even when feasibility is known.

From accumulators to lattices. The most successful route to logarithmic-size membership proofs has been through *cryptographic accumulators*. Early work on RSA-based accumulators [5,3,12] and bilinear constructions [49] established that sets can be compressed into short digests with efficient zero-knowledge proofs. The lattice-based line initiated by Libert *et al.* [34,35] adapted Merkle-style accumulators and Stern-type arguments [54] to the post-quantum setting, yielding logarithmic-size membership proofs and ring signatures. Subsequent works, including SMILE [44] and BLOOM [42], refined these ideas through ideal-lattice instantiations and bimodal techniques. From an asymptotic viewpoint, these constructions demonstrate that lattice methods can match the succinctness of classical accumulators while remaining compatible with post-quantum hardness assumptions. Nevertheless, all known instantiations ultimately rely on the Fiat–Shamir transform and therefore inherit the conceptual limitations of the ROM.

Related abstractions. Several related frameworks explore alternative formulations of membership. Zero-Knowledge Sets (ZKS) [46,16,38] allow committing to an entire set and later proving membership or non-membership queries, while one-out-of-many proofs [29] provide a structured approach to logarithmic-size arguments. More recent proposals include MPC-in-the-Head membership proofs [27] and zkSNARK-based constructions enforcing membership alongside NP predicates [6]. These directions expand the design space but typically rely on stronger setup assumptions or heavy generic machinery. Importantly, they

do not directly address the fundamental algebraic constraints that arise when designing succinct standard-model proofs from lattice assumptions.

A technical bottleneck. Beyond asymptotics, an underlying technical obstacle remains. Existing lattice zero-knowledge frameworks based on Lyubashevsky-style rejection sampling [41], such as [23,58,43], are formulated over a *single* ambient modulus q . This design choice is intertwined with the analysis of rejection sampling and the algebraic structure of lattice relations, and has implicitly shaped the architecture of subsequent constructions. However, modern cryptographic systems routinely combine components defined over heterogeneous moduli q_i —including hash functions, signatures, commitments, PRFs, and encryption schemes derived from SIS, LWE, or LWR assumptions. This heterogeneity is not an edge case: it is the default in modern lattice cryptography, where components optimize for different moduli, dimensions, and error distributions.

Such a scenario was considered, for example, by Esgin *et al.* [22], where relations over different moduli are handled via homogenization into a common modulus. This already incurs a communication overhead: for n_i equations over modulus q_i , keeping the relations at their native moduli costs proportionally $\sum_i n_i \log q_i$, whereas lifting them to a common modulus Q costs $(\sum_i n_i) \log Q$, an overhead of $\sum_i n_i \log(Q/q_i)$ even before accounting for auxiliary variables introduced by homogenization. In our set-membership proof, each accumulator level combines commitment relations modulo q_C with hash relations modulo q_H , so this overhead occurs throughout the $\log N$ levels of the proof. Our multi-moduli protocol instead keeps these relations at their native moduli and avoids this lifting overhead.

From this standpoint, the single-modulus paradigm therefore constitutes a bottleneck that limits asymptotic efficiency and modularity of lattice-based zero knowledge. Overcoming this barrier requires new technical tools capable of reasoning about relations across multiple moduli simultaneously, without sacrificing the tightness and structure that make lattice protocols efficient.

OUR CONTRIBUTIONS. In this work, we develop new technical tools for constructing succinct lattice-based zero-knowledge proofs in the standard model, removing a long-standing structural barrier and yielding the first compact post-quantum NIZK arguments for set membership without random oracles. Our focus is on advancing the asymptotic foundations of lattice zero knowledge: rather than optimizing concrete parameters, we introduce new algebraic techniques that overcome limitations inherent in existing standard-model constructions. Our main contributions are summarized below.

A multi-moduli trapdoor Σ -protocol. At the heart of our work is a new trapdoor Σ -protocol supporting linear relations modulo multiple heterogeneous moduli. Existing Lyubashevsky-style protocols operate over a single ambient modulus, forcing heterogeneous components to be homogenized into a larger q . We show that these rejection-sampling proofs can be lifted to a multi-moduli setting *without homogenization*, while preserving zero knowledge and soundness. The key observation is that rejection sampling itself does not depend on the moduli of the underlying linear relations, allowing relations sharing the same witness to

be verified directly at their native moduli. This provides a modular foundation for standard-model NIZKs from lattice assumptions. In particular, it enables standard-model lattice ZK for *composed* relations mixing commitments, hashing, and encryption components with distinct moduli—a common pattern in post-quantum protocol design. The protocol achieves inverse-polynomial soundness error and compiles into a multi-theorem NIZK in the CRS model [36].

We distinguish this general observation from the concrete efficiency improvement demonstrated by our construction. While the modulus-agnostic nature of rejection sampling is not specific to our particular trapdoor Σ -protocol, concretely optimized lattice proof systems may impose additional constraints, for example on the structure or size of the challenge space and on the admissible moduli. We therefore do not claim that the same concrete improvement follows immediately for every lattice proof system. Determining how to exploit the multi-moduli abstraction under such additional constraints is an interesting direction for future work.

Succinct standard-model NIZKs for set membership. Building on the above framework, we construct lattice-based NIZK arguments for set membership with logarithmic proof size in the set cardinality. Our construction matches the asymptotic succinctness of ROM-based accumulator protocols while achieving post-quantum security in the standard model. This yields the first succinct lattice-based NIZK for set membership without random oracles or generic NP reductions.

Compact post-quantum ring signatures. As an application of our membership arguments, we obtain a lattice-based ring signature scheme with signature size

$$O(\log R) \cdot \tilde{O}(\lambda^2) \text{ bits,}$$

where R denotes the ring size and λ the security parameter. Compared to the plain-model construction of Chatterjee *et al* [17], our scheme improves the dependence on λ by a quadratic factor while retaining optimal logarithmic scaling in R . As summarized in Table 1, our construction uses a CRS, which partly enables this improvement over the plain-model construction; such setup is inherent to non-trivial NIZKs in the standard model. The scheme achieves *statistical* anonymity against full key exposure [7] and unforgeability under standard Module-SIS and Module-LWE assumptions.

Table 1. Asymptotic efficiency comparison between our ring signature and known lattice-based standard model ring signatures, by Brakerski and Kalai [10] and by Chatterjee *et al* [17], based on security parameter λ and maximum ring size $R \in \text{poly}(\lambda)$. Details of the estimation are provided in Appendix C.5.

Scheme	Model	Signature	Public Key	Secret key
[10]	Plain	$\Omega(R \cdot \lambda \log \lambda)$	$\Omega(\lambda^2 \log \lambda)$	$\mathcal{O}(\lambda^2 \log^2 \lambda)$
[17]	Plain	$\Omega(\log R \cdot \lambda^4)$	$\Omega(\lambda^2)$	$\Omega(\lambda^2)$
Ours	CRS	$\mathcal{O}(\log R \cdot \lambda^2 \log^2 \lambda)$	$\mathcal{O}(\lambda \log \lambda)$	$\mathcal{O}(\lambda \log \lambda)$

Additional technical tools. Complementing the core conceptual advance, we develop two additional tools that support our constructions and improve their efficiency and design.

- A generalized Merkle-tree accumulator that extends the constructions of Libert *et al.* [34,35] from general lattices to module lattices and from binary to base- B decomposition. This auxiliary tool interpolates between general- and ring-lattice regimes and exposes trade-offs between communication complexity, prover cost, and assumption structure.
- A message-binding technique for standard-model ring signatures that integrates the message directly into the accumulation process, removing the need for lattice one-time signatures. Since lattice one-time signatures incur $\Omega(\lambda^3)$ overhead when applied to large NIZK transcripts, eliminating them yields signatures whose size matches that of the underlying membership proof.

Taken together, these contributions address the central obstacles highlighted earlier: they introduce a new algebraic framework for lattice zero-knowledge that supports heterogeneous moduli, yield the first logarithmic-size post-quantum NIZKs for set membership in the standard model, and demonstrate how this framework enables substantially improved asymptotic efficiency for ring signatures without relying on random oracles.

TECHNICAL OVERVIEW. Our starting point is the lattice-based accumulator from [35], built upon a lattice-based hash function defined by $\mathbf{H} = [\mathbf{H}_0 \mid \mathbf{H}_1] \in \mathbb{Z}_{q_H}^{n \times 2n \lceil \log q \rceil}$. To accumulate a set $\mathcal{D} = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\} \subset \{0, 1\}^{2n \lceil \log q_H \rceil}$ of cardinality $N = 2^\ell$, the accumulator constructs a Merkle tree with N leaves in a bottom-up manner, where each parent node $\mathbf{p}$ is obtained as the hash of its two children $\mathbf{c}_0, \mathbf{c}_1$. Using the lattice-based hash function $\mathbf{H}$, the parent-child relation admits a compact linear-algebraic description:

$$\mathbf{H}_0 \mathbf{c}_0 + \mathbf{H}_1 \mathbf{c}_1 - \mathbf{G} \mathbf{p} = \mathbf{0}^n \bmod q_H,$$

where $\mathbf{G} = [1 \ 2 \ \dots \ 2^{\lceil \log q_H \rceil - 1}] \otimes \mathbf{I}_n$ is the base-2 gadget matrix. The root $\mathbf{u}$ of the tree is the accumulated value. For any $\mathbf{d} \in \mathcal{D}$, a witness w that $\mathbf{d}$ is accumulated in $\mathbf{u}$ consists of the path from $\mathbf{u}$ to $\mathbf{d}$ together with the sibling nodes $\mathbf{w}_0, \dots, \mathbf{w}_{\ell-1} \in \{0, 1\}^{n \lceil \log q_H \rceil}$ along this path.

For set membership proofs, the idea of [35] is to prove in zero-knowledge the existence of such an accumulation witness for a hidden element $\mathbf{d} \in \mathcal{D}$. Operationally, this is done by iteratively checking the hash chain at each depth i of the tree: the prover must convince the verifier that the parent node $\mathbf{v}_{i-1}$ is correctly computed from the node $\mathbf{v}_i$ and its sibling $\mathbf{w}_i$. Because the path may go either left or right at each level, the required statement is naturally disjunctive: specifically, one must prove that either

$$\mathbf{H}_0 \mathbf{v}_i + \mathbf{H}_1 \mathbf{w}_i - \mathbf{G} \mathbf{v}_{i-1} = \mathbf{0}^n \bmod q_H$$

or

$$\mathbf{H}_1 \mathbf{v}_i + \mathbf{H}_0 \mathbf{w}_i - \mathbf{G} \mathbf{v}_{i-1} = \mathbf{0}^n \bmod q_H.$$

This leads to our first key idea: we apply the CDS OR-proof technique [19] at each level i , instantiated with an atomic Σ -protocol for proving that a small-norm witness satisfies a linear relation. A crucial observation is that the ℓ level-wise proofs cannot be independent: to ensure global consistency of the Merkle path, the node $\mathbf{v}_{i-1}$ proved at depth i must be exactly the node reused as the child at depth $i-1$. We therefore need to *bind* the protocols across levels. To enforce this binding, we commit to every path node $\mathbf{v}_i$ using a lattice-based commitment scheme—for concreteness, the KTX commitment scheme [32]—and we additionally prove that these commitments are well-formed.

Concretely, let $\mathbf{C}, \mathbf{D}$ be the matrices defining the KTX commitment key under modulus q_C . Then, for $j \in \{0, 1\}$, the atomic relation at depth i consists of (at most) three linear equations:

- $\mathbf{H}_j \mathbf{v}_i + \mathbf{H}_{1-j} \mathbf{w}_i - \mathbf{G} \mathbf{v}_{i-1} = \mathbf{0}^n \bmod q_H$, encoding the hash-chain relation at depth i ;
- $\mathbf{C} \mathbf{v}_i + \mathbf{D} \mathbf{r}_{\mathbf{v}_i} = \mathbf{c}_{\mathbf{v}_i} \bmod q_C$, showing that $\mathbf{c}_{\mathbf{v}_i}$ is a KTX commitment to the depth- i path node $\mathbf{v}_i$ with randomness $\mathbf{r}_{\mathbf{v}_i}$;
- $\mathbf{C} \mathbf{v}_{i-1} + \mathbf{D} \mathbf{r}_{\mathbf{v}_{i-1}} = \mathbf{c}_{\mathbf{v}_{i-1}} \bmod q_C$, showing that $\mathbf{c}_{\mathbf{v}_{i-1}}$ is a KTX commitment to the depth- $(i-1)$ path node $\mathbf{v}_{i-1}$ with randomness $\mathbf{r}_{\mathbf{v}_{i-1}}$.

A useful way to view the above is as an instance of a *generic system of small-norm linear relations over heterogeneous moduli*. Abstractly, we capture it by the following relation:

$$\mathbf{M}_i \cdot \mathbf{m} = \mathbf{t}_i \bmod q_i, \quad \forall i \in \{1, \dots, t\}, \quad (1)$$

where the witness $\mathbf{m}$ has small Euclidean norm, while the public matrices $\mathbf{M}_i$ and statements $\mathbf{t}_i$ are defined with respect to (potentially different) moduli q_i . Each equation modulo q_i in (1) can be proven via the blueprint Σ -protocol of [40,41], in which the prover must produce a small-norm response $\mathbf{z}_{\mathbf{m}}^{(i)}$ satisfying

$$\mathbf{a}_i + c_i \cdot \mathbf{t}_i = \mathbf{M}_i \cdot \mathbf{z}_{\mathbf{m}}^{(i)} \bmod q_i, \quad (2)$$

where c_i is the verifier’s (small) challenge scalar and $\mathbf{a}_i$ is the prover’s first message. The key observation underlying our multi-moduli framework is that rejection sampling, which is used to tune the distribution of $\mathbf{z}_{\mathbf{m}}^{(i)}$ so that it is independent of the witness $\mathbf{m}$, *does not make use of the modulus q_i defining the relation*. Exploiting this, we obtain a Σ -protocol proving all equations in (1) *simultaneously*, by using a *single* challenge c shared across all instances of (2) together with a *single* response $\mathbf{z}_{\mathbf{m}}$.

To obtain an NIZK for (1), we rely on the trapdoor Σ -protocol framework of [13]. Roughly speaking, the trapdoor property requires that the common reference string can be generated together with some trapdoor information τ , and that there is an efficient algorithm which, given τ , any first-move messages $(\mathbf{a}_i)_{i=1}^t$, and any “false” statement $(\mathbf{t}_i)_{i=1}^t$, identifies the *bad challenge* c —namely, the unique challenge to which a prover can produce a response satisfying (2). To realize such trapdooriness, we recall an idea originating in [14] and employed

in [55] to design trapdoor Σ -protocols for lattice relations by coupling the proof with an encryption of the witness.

Concretely, in addition to proving (1), we also prove that a ciphertext $\mathbf{c}$ is an encryption of $\mathbf{m}$ (with randomness $\mathbf{r}$) under a linearly homomorphic encryption scheme $\mathcal{E}$, i.e., $\mathbf{c} = \mathcal{E}(\mathbf{m}; \mathbf{r})$. Since lattice-based encryption schemes [52,25] are already linearly homomorphic in both message and randomness and interact well with Lyubashevsky-type Σ -protocols, proving (1) together with $\mathbf{c} = \mathcal{E}(\mathbf{m}; \mathbf{r})$ can be expressed by enforcing

$$\begin{aligned}\mathbf{a}_i + c \cdot \mathbf{t}_i &= \mathbf{M}_i \cdot \mathbf{z}_{\mathbf{m}} \bmod q_i, \\ \mathbf{a}_{\mathbf{c}} + c \cdot \mathbf{c} &= \mathcal{E}(\mathbf{z}_{\mathbf{m}}; \mathbf{z}_{\mathbf{r}}).\end{aligned}$$

As in [55], the second equation can be interpreted as stating that $(\mathbf{a}_{\mathbf{c}} + c \cdot \mathbf{c})$ is an encryption of $\mathbf{z}_{\mathbf{m}}$. Therefore, the decryption algorithm of $\mathcal{E}$ provides a way to test whether a given challenge c is bad: using the decryption key, we recover the plaintext $\mathbf{z}_{\mathbf{m}}$ underlying the ciphertext $(\mathbf{a}_{\mathbf{c}} + c \cdot \mathbf{c})$ and then check whether this $\mathbf{z}_{\mathbf{m}}$ satisfies the remaining equation(s) modulo q_i . The key difference from [55] is that we decouple witness extraction from relation verification: we extract only $\mathbf{z}_{\mathbf{m}}$ and verify the remaining relations directly at their native moduli. Thus, we need not recover the full response, including the encryption randomness, and the approach is no longer tied to dual-Regev encryption. Viewed differently, this realizes a commit-and-prove protocol with an extractable commitment, where extractability furnishes a trapdoor that enables an efficient search for the bad challenge. This separation allows the methodology to use arbitrary extractable commitments while supporting relations over heterogeneous moduli. Extractable commitments from lattices are closely related to lossy public-key encryption and can be achieved by modifying primal-Regev encryption [52]. Equipped with these ingredients, we obtain a trapdoor Σ -protocol for (1). Finally, to convert this trapdoor Σ -protocol into an NIZK, we employ the compiler of [36], which turns any trapdoor Σ -protocol into a (multi-theorem) NIZK argument.

The resulting NIZK argument directly implies an NIZK for the accumulator-based set membership relation. Its soundness is with respect to the relaxed relation defined by the underlying trapdoor Σ -protocol, as discussed formally in Section 4. In particular, it supports statements of the form “a KTX commitment opens to some element that is accumulated in a value.” While our approach leverages the accumulator of [35], we also introduce several generalizations by: (i) extending the setting to module lattices over power-of-2 cyclotomic number fields, and (ii) employing base- B gadget decomposition in the lattice-based hash function underlying the accumulator. Additionally, our core trapdoor Σ -protocol allows setting up the challenge space as desired. Taken together with the above generalizations, this flexibility provides additional freedom for trading off efficiency and security.

Built upon the above set membership primitive, we propose a logarithmic-size ring signature scheme from standard Module-LWE and Module-SIS assumptions. At a high level, our construction follows the template of [35]: a user’s signing key is a vector $\mathbf{x} \in \mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}}$ consisting of polynomials with binary coefficients, and the

public key is of the form $\mathbf{d} = \mathbf{F} \cdot \mathbf{x} \bmod q_H$, where $\mathbf{F} \in \mathcal{R}_{q_H}^{n \times m_F}$ defines a lattice-based one-way function. Given a ring R and a message M , a signer accumulates the ring into a value $\mathbf{u}$ and produces a signature which is an NIZK proof that “a KTX commitment opens to some $\mathbf{d}$ accumulated in $\mathbf{u}$, and that $\mathbf{x}$ is a secret key corresponding to $\mathbf{d}$.”

A central novelty of our construction lies in how the message M is bound to the proof. In prior standard-model ring signatures (e.g., [15,28,37]), message binding is typically achieved by attaching a one-time signature together with a certified verification key, which signs the NIZK transcript. While simple, this approach is inefficient in one important respect: lattice-based one-time signatures and certification mechanisms contribute an $\Omega(\lambda^3)$ -bit overhead and obscure the algebraic structure of the underlying proof system. To our knowledge, all existing standard-model constructions rely on this paradigm or closely related variants.

Unlike [35], our approach avoids certified one-time signatures altogether. Our key observation is that message binding can be realized *within* the accumulator itself, rather than layered on top of the proof. Concretely, we bind M directly into the accumulation process: we append the message M to each public key in R to form a new ring R' , and then accumulate R' into $\mathbf{u}$. The resulting ring signature is an NIZK proof that “a KTX commitment opens to some $(\mathbf{d} \| M)$ accumulated in $\mathbf{u}$, and that $\mathbf{x}$ is a secret key corresponding to $\mathbf{d}$.” Conceptually, this shifts message binding from an external certification layer to the algebraic core of the membership proof. Besides eliminating the costly one-time signature overhead, this design leads to a cleaner proof architecture: the message becomes part of the accumulated structure itself, so any attempt to modify the message necessarily invalidates the accumulator witness. Intuitively, anonymity follows from the hiding property of the KTX commitment and the zero-knowledge property of the NIZK system, while the security of the accumulator prevents adversaries from copying or modifying a valid signature to another message or ring.

The formal proof of unforgeability is more delicate due to the *relaxed* soundness of the NIZK argument system inherited from Lyubashevsky-type protocols. Nevertheless, we show that any successful forger can be transformed into an algorithm solving appropriate Module-SIS instances, thereby establishing standard-model security under well-studied lattice assumptions.

ORGANIZATION. Section 2 recalls some background on lattices and trapdoor Σ -protocols. Section 3 presents a trapdoor Σ -protocol for linear relations in many different moduli. In Section 4, we explain how to apply the trapdoor Σ -protocol of Section 3 to construct a trapdoor Σ -protocol for set membership relation. Section 5 outlines our ring signature construction. Several technical details are deferred to the Appendix.

2 Preliminaries

By default, logarithms without a base are of base 2. For a positive integer n , we let $[n]$ be the set $\{0, \dots, n-1\}$. If $\mathcal{A}$ is a probabilistic polynomial time (PPT) algorithm, we also write $y \leftarrow \mathcal{A}$ to indicate that y is output from $\mathcal{A}$. We write

$\mathcal{A}(x; r)$ where the input part x and the coin flips r of $\mathcal{A}$ are separated by a semicolon. We write $x \leftarrow \mathcal{D}$ to indicate that x is sampled from a distribution $\mathcal{D}$.

We write $x \xleftarrow{\$} S$ to indicate that x is sampled uniformly from a set S .

For a power of two d and a positive integer q , denote by $\mathcal{R}$ and $\mathcal{R}_q$ the rings $\mathbb{Z}[X]/(X^d + 1)$ and $\mathbb{Z}_q[X]/(X^d + 1)$ respectively. The ring $\mathcal{R}$ is the $2d$ -th cyclotomic ring of integers, defined by the irreducible polynomial $X^d + 1$. We use lower-case letters to denote elements in $\mathcal{R}$ or $\mathcal{R}_q$ and bold lower-case (resp. upper-case) letters represent column vectors (resp. matrices) with coefficients in $\mathcal{R}$ or $\mathcal{R}_q$. The concatenation of two column vectors $\mathbf{v}_1, \mathbf{v}_2$ is a column vector and denoted by $(\mathbf{v}_1, \mathbf{v}_2)$. For a polynomial $w = w_0 + w_1X + \dots + w_{d-1}X^{d-1}$, we write $w[i]$ for the degree- i coefficient. For a matrix $\mathbf{M}$ of dimension $n \times m$, we write $\mathbf{M}[:k]$ for the matrix obtained from selecting the first k rows of $\mathbf{M}$.

For $w \in \mathcal{R}$, the ℓ_∞ -norm and ℓ_2 -norm are defined as $\|w\|_\infty := \max_j \|w[j]\|_\infty$, $\|w\|_2 := (\sum_{i \in [d]} w[i]^2)^{1/2}$. We sometimes omit the subscript in ℓ_2 -norm. For $u, v \in \mathcal{R}$, we have the following bounds

$$\|uv\|_\infty \leq d \cdot \|u\|_\infty \|v\|_\infty, \quad \|uv\| \leq \min\{d \cdot \|u\|_\infty \|v\|, \sqrt{d} \cdot \|u\| \|v\|\}.$$

The ℓ_∞ -norm and ℓ_2 -norm naturally generalize to $\mathbf{w} \in \mathcal{R}^k$: we have that $\|\mathbf{w}\|_\infty := \max_{i \in [k]} \|\mathbf{w}[i]\|_\infty$ and $\|\mathbf{w}\| := \left(\sum_{i \in [k]} \|\mathbf{w}[i]\|^2\right)^{1/2}$. For a positive norm bound β , we define $\mathcal{S}_\beta^n := \{\mathbf{a} \in \mathcal{R}^n : \|\mathbf{a}\| \leq \beta\}$ and $\mathcal{C}_\beta^n := \{\mathbf{a} \in \mathcal{R}^n : \|\mathbf{a}\|_\infty \leq \beta\}$. We let $\mathcal{R}_{\text{bin}} = \{w_0 + w_1X + \dots + w_{d-1}X^{d-1} \in \mathcal{R} : w_i \in \{0, 1\}, \forall i \in [d]\}$. We write $\mathcal{D}_\sigma^l$ to denote the discrete Gaussian on $\mathcal{R}^l$, scaled by $\sigma > 0$.

2.1 Trapdoor Σ -Protocols

We recall the notion of Σ -protocols in the *common reference string* (CRS) model.

Definition 1 (Adapted from [13]). Consider a language $\mathcal{L} = (\mathcal{L}_{\text{zk}}, \mathcal{L}_{\text{snd}})$ associated with two NP relations $\mathcal{R}_{\text{zk}}, \mathcal{R}_{\text{snd}}$. A Σ -protocol in the common reference string model is a tuple of 4 PPT algorithms $\Pi = (\text{Gen}_{\text{par}}, \text{Gen}_{\mathcal{L}}, \text{P}, \text{V})$:

- The PPT algorithm $\text{Gen}_{\text{par}}(1^\lambda)$ on input a security parameter $\lambda \in \mathbb{N}$ outputs public parameters par .
- The PPT algorithm $\text{Gen}_{\mathcal{L}}(\text{par}, \text{info}_{\mathcal{L}})$, on input public parameters par and a language-specific information $\text{info}_{\mathcal{L}}$, outputs the language-reference string $\text{crs}_{\mathcal{L}}$ of the full common reference string $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$.
- **3-Move Form:** P and V both take as inputs $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, with $\text{par} \leftarrow \text{Gen}_{\text{par}}(1^\lambda)$ and $\text{crs}_{\mathcal{L}} \leftarrow \text{Gen}_{\mathcal{L}}(1^\lambda, \mathcal{L})$, and a statement x and proceed as follows: (i) P takes as input $w \in \mathcal{R}_{\text{zk}}(x)$, computes $(\mathbf{a}, st) \leftarrow P(\text{crs}, x, w)$ and sends $\mathbf{a}$ to V ; (ii) V sends back a random challenge c from the challenge space Chall ; (iii) P finally sends a response $\mathbf{z} = P(\text{crs}, x, w, \mathbf{a}, c, st)$ to V ; (iv) On input of a transcript $(\mathbf{a}, c, \mathbf{z})$, V outputs 1 or 0.
- **Completeness:** There is a function $\varepsilon(\lambda)$ such that if $(x, w) \in \mathcal{R}_{\text{zk}}$ and P honestly computes $(\mathbf{a}, \mathbf{z})$ for a challenge c , then $\text{V}(\text{crs}, x, (\mathbf{a}, c, \mathbf{z}))$ outputs 1 with probability $1 - \varepsilon(\lambda)$. The function ε is the completeness error of Π .

- **Special zero-knowledge:** There is a PPT simulator ZKSim that, on input a crs , a statement $x \in \mathcal{L}_{\text{zk}}$ and a challenge $c \in \mathcal{C}$, outputs $(\mathbf{a}, \mathbf{z}) \leftarrow \text{ZKSim}(\text{crs}, x, c)$ such that $(\mathbf{a}, c, \mathbf{z})$ is indistinguishable from a real transcript with challenge Chall (for $w \in \mathcal{R}_{\text{zk}}(x)$).
- **Special soundness:** for any common reference string $\text{crs} \leftarrow \text{Gen}(1^\lambda)$, any $x \notin \mathcal{L}_{\text{snd}}$, and any first message $\mathbf{a}$ sent by the prover, there is at most one challenge $c = f(\text{crs}, x, \mathbf{a})$ for which a valid transcript $(\text{crs}, x, \mathbf{a}, \text{Chall}, \mathbf{z})$ exists for some third message $\mathbf{z}$. The function f is called the “bad challenge function” associated with Π . That is, if $x \notin \mathcal{L}_{\text{snd}}$ and the challenge differs from the bad challenge, the verifier never accepts.

We recall the notion of *trapdoor Σ -protocol* by Canetti *et al.* [13].

Definition 2 (Adapted from [13]). A Σ -protocol $\Pi = (\text{Gen}_{\text{par}}, \text{Gen}_{\mathcal{L}}, \text{P}, \text{V})$ with bad challenge function f for a trapdoor language $\mathcal{L} = (\mathcal{L}_{\text{zk}}, \mathcal{L}_{\text{snd}})$ is a **trapdoor Σ -protocol** if it satisfies the properties of Definition 1. Furthermore, there exist PPT algorithms $(\text{TrapGen}, \text{BadChallenge})$ such that:

- $\text{TrapGen}(\text{par}, \text{info}_{\mathcal{L}})$ is a randomized algorithm, on input public parameters par and an information $\text{info}_{\mathcal{L}}$, outputs the language dependent part $\text{crs}_{\mathcal{L}}$ of the common reference string $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$ and a trapdoor $\tau_{\Sigma} \in \{0, 1\}^{\ell_{\tau}}$ for some $\ell_{\tau}(\lambda)$.
- $\text{BadChallenge}(\tau_{\Sigma}, \text{crs}, x, \mathbf{a})$ on input a trapdoor τ_{Σ} , a CRS $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, an instance x and a first-move message $\mathbf{a}$, outputs a challenge c .

We require the following:

- **CRS Indistinguishability:** The distributions of $\text{crs}_{\mathcal{L}}$, output from Gen_{par} and TrapGen are indistinguishable.
- **Correctness:** For any $x \notin \mathcal{L}_{\text{snd}}$; all pairs $(\text{crs}_{\mathcal{L}}, \tau_{\Sigma}) \leftarrow \text{TrapGen}(\text{par}, \text{info}_{\mathcal{L}})$ and any prover’s first-move message $\mathbf{a}$, we have that

$$\text{BadChallenge}(\tau_{\Sigma}, \text{crs}, x, \mathbf{a}) = f(\text{crs}, x, \mathbf{a}).$$

In a trapdoor Σ -protocol, for any false statement $x \notin \mathcal{L}_{\text{snd}}$ there is *at most one* challenge to which the prover can respond; such challenge can be efficiently computed by BadChallenge when given as inputs a trapdoor information τ (that may depend on the crs), a false statement $x \notin \mathcal{L}_{\text{snd}}$ and a first-move message $\mathbf{a}$.

2.2 Cryptographic Accumulators

A deterministic *accumulator* consists of PPT algorithms TSetup , TAcc , TWitness and TVerify such that:

- $\text{TSetup}(\lambda)$ On input security parameter λ , output the public parameter pp . All other algorithms implicitly take pp as an input.
- $\text{TAcc}(R = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\})$ On input a set $R = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\}$ of $N = \text{poly}(\lambda)$ values, this deterministic algorithm outputs a value $\mathbf{u}$.

- $\text{TWitness}(R, \mathbf{d})$ On input a data set R and a value $\mathbf{d}$, output $\perp$ if $\mathbf{d} \notin R$; otherwise output a witness w for the fact that $\mathbf{d}$ is accumulated in $\text{TAcc}(R)$.
- $\text{TVerify}(\mathbf{u}, \mathbf{d}, w)$ On input accumulator value $\mathbf{u}$ and a value-witness pair $(\mathbf{d}, w)$, output 1 (which indicates that $(\mathbf{d}, w)$ is valid for the accumulator $\mathbf{u}$) or 0.

An accumulator scheme is correct if $\text{TVerify}(\text{TAcc}(R), \mathbf{d}, \text{TWitness}(R, \mathbf{d})) = 1$ for all $pp \leftarrow \text{TSetup}(\lambda)$ and for all $\mathbf{d} \in R$. The security of an accumulator is formalized via the following.

Definition 3 ([3,12]). An accumulator scheme $(\text{TSetup}, \text{TAcc}, \text{TWitness}, \text{TVerify})$ is called secure if for any PPT adversary $\mathcal{A}$ and for all $pp \leftarrow \text{TSetup}(1^\lambda)$

$$\Pr[\mathbf{d}^* \notin R \text{ and } \text{TVerify}(\text{TAcc}(R), \mathbf{d}^*, w^*) = 1 : (R, \mathbf{d}^*, w^*) \leftarrow \mathcal{A}(pp)] \in \text{negl}(\lambda).$$

2.3 Ring Signatures

We recall the definitions and security requirements for ring signatures [7,29]. A ring signature is a tuple of PPT algorithms $(\text{RSetup}, \text{RKeyGen}, \text{RSign}, \text{RVerify})$ that:

- $\text{RSetup}(1^\lambda)$ generates public parameters pp which are made available to all users. All other algorithms implicitly take pp as an input.
- $\text{RKeyGen}(1^\lambda)$ generates a public key pk and the corresponding secret key sk .
- $\text{RSign}(sk, M, R)$ outputs a signature Σ on the message $M \in \{0,1\}^*$ with respect to the ring $R = (pk_0, \dots, pk_{N-1})$. It is required that (pk, sk) is a valid key pair produced by RKeyGen and that $pk \in R$.
- $\text{RVerify}(M, R, \Sigma)$ takes as input a candidate signature Σ , a message M and a ring of public keys R , this deterministic algorithm outputs 1 or 0.

Definition 4 (Correctness). A ring signature $(\text{RSetup}, \text{RKeyGen}, \text{RSign}, \text{RVerify})$ is correct if for any $pp \leftarrow \text{RSetup}(1^\lambda)$, any $(pk, sk) \leftarrow \text{RKeyGen}(1^\lambda)$, any R such that $pk \in R$, any $M \in \{0,1\}^*$, we have $\text{RVerify}(M, R, \text{RSign}(sk, M, R)) = 1$.

Definition 5 (Unforgeability w.r.t. insider corruption). A ring signature scheme $(\text{RSetup}, \text{RKeyGen}, \text{RSign}, \text{RVerify})$ is unforgeable w.r.t. insider corruption if for any $pp \leftarrow \text{RSetup}(1^\lambda)$ and any PPT adversary $\mathcal{A}$,

$$\Pr[\text{RVerify}(M^*, R^*, \Sigma^*) = 1 : (M^*, R^*, \Sigma^*) \leftarrow \mathcal{A}^{\text{PKGen}, \text{Sign}, \text{Corrupt}}(pp)] \in \text{negl}(\lambda),$$

where:

- PKGen on the j -th query runs $(pk_j, sk_j) \leftarrow \text{RKeyGen}(1^\lambda)$ and returns pk_j .
- $\text{Sign}(j, M, R)$ returns the output of $\text{RSign}_{pp}(sk_j, M, R)$ if: (i) (pk_j, sk_j) has been generated by PKGen ; (ii) $pk_j \in R$. Otherwise, it returns $\perp$.
- $\text{Corrupt}(j)$ returns sk_j , if (pk_j, sk_j) has been generated by PKGen .
- $\mathcal{A}$ outputs (M^*, R^*, Σ^*) such that: $\text{Sign}(\cdot, M^*, R^*)$ has not been queried, and R^* is non-empty and only contains public keys pk_j generated by PKGen for which j has not been corrupted.

Definition 6. A ring signature scheme $(\text{RSetup}, \text{RKeyGen}, \text{RSign}, \text{RVerify})$ provides anonymity against full key exposure [7] if, for any PPT adversary $\mathcal{A}$ and all $N \in \text{poly}(\lambda)$,

$$\Pr \left[\begin{array}{l} \mathcal{A}(\Sigma^*; \{r_i\}_{i \in [N]}) = b : \\ \begin{array}{l} pp \leftarrow \text{RSetup}(1^\lambda); \\ (pk_i, sk_i) \leftarrow \text{RKeyGen}(1^\lambda; r_i) \text{ for } i \in [N]; \\ (M^*, j_0, j_1, R^*) \leftarrow \mathcal{A}^{\text{Sign}(\cdot)}(\{pk_i\}_{i \in [N]}) \\ b \xleftarrow{\$} \{0, 1\}; \Sigma^* \leftarrow \text{RSign}(sk_{j_b}, M^*, R^*) \end{array} \end{array} \right] = 1/2 + \text{negl}(\lambda),$$

where $pk_{j_0}, pk_{j_1} \in R^*$, and the oracle Sign is defined as in Definition 6.

3 A Trapdoor Σ -Protocol for Multi-Moduli Linear Relations

In this section we construct a trapdoor Σ -protocol proving that a collection of l linear relations of the form

$$\mathbf{t}_i = \mathbf{M}_i \cdot \mathbf{m} \bmod q_i \in \mathcal{R}_{q_i}^{n_i}, i \in [l] \quad (3)$$

is satisfied by a witness $\mathbf{m} \in \mathcal{R}^{l_m}$ such that $\|\mathbf{m}\| \leq B_{\mathbf{m}}$, where $B_{\mathbf{m}} \ll \min\{q_i : i \in [l]\}$. The idea is to commit to $\mathbf{m}$ under the extractable commitment scheme formally defined in Appendix A.4, and prove (3) along with the well-formedness of the commitment via the equations

$$\begin{aligned} \mathbf{a}_i + c \cdot \mathbf{t}_i &= \mathbf{M} \cdot \mathbf{z}_{\mathbf{m}} \bmod q_i, \\ \mathbf{a}_{\mathbf{c}} + c \cdot \mathbf{c} &= \text{Com}(\mathbf{z}_{\mathbf{m}}; \mathbf{z}_{\mathbf{r}}); \end{aligned}$$

in which Com is the commitment algorithm, $\mathbf{c}$ is an extractable commitment to the witness $\mathbf{m}$, c is a verifier's challenge, $(\mathbf{a}_{\mathbf{c}}, (\mathbf{a})_{i \in [l]})$ is prover's first message and $\mathbf{z}_{\mathbf{m}}, \mathbf{z}_{\mathbf{r}}$ are small norm responses. The relations over the witness are linear equations with norm constraints and thus it suffices to apply Lyubashevsky's rejection sampling technique [40,41]. As rejection sampling does not depend on the modulus in the linear relation (see Lemma 4), the Σ -protocols of [40,41] naturally generalize to the setting where there are multiple linear equations with different moduli. Furthermore, extractability of Com allows testing the bad challenge as follows: given a challenge c , we extract the message $\mathbf{z}_{\mathbf{m}}$ under the "commitment" $\mathbf{a}_{\mathbf{c}} + c \cdot \mathbf{c}$, then check if $\mathbf{z}_{\mathbf{m}}$ satisfies the remaining equations. Compared with [55], which couples witness extraction with the encryption relation, our formulation separates extraction of the committed witness from verification of the remaining relations. This removes the dependence on the particular dual-Regev encryption used there and allows the same methodology to accommodate arbitrary extractable commitments and relations over heterogeneous moduli.

Therefore, we obtain an efficient bad challenge searching algorithm when the size of challenge space Chall is polynomial. A simple choice for Chall is

$$\text{Chall} = \{c_0 + c_1 X + \dots + c_{d_{\text{Chall}}-1} X^{d_{\text{Chall}}-1} \in \mathcal{R} : 0 \leq c_i \leq B_{\text{Chall}}\}. \quad (4)$$

We remark that **Chall** can be endowed with a group structure, which is then sufficient for applying the CDS OR technique (formally described in Appendix C.1).

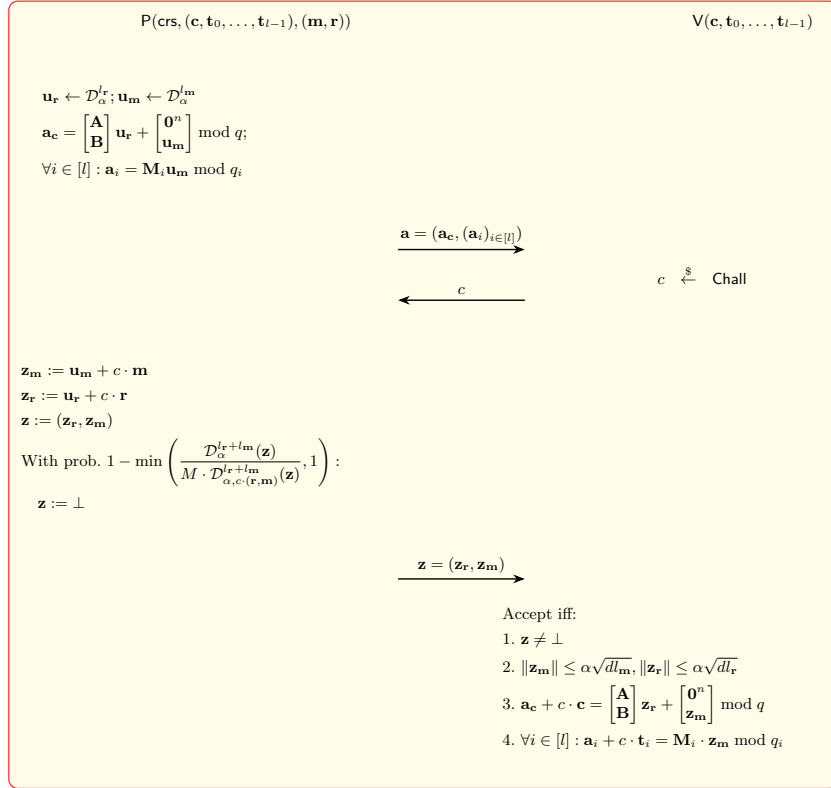


Fig. 1. A trapdoor Σ -protocol for l linear relations across multiple moduli

Protocol Description. We now provide the details of our Σ -protocol:

- $\text{Gen}_{\text{par}}(1^\lambda)$ on input a security parameter $\lambda \in \mathbb{N}$ chooses a power-of-2 d defining the number field $\mathcal{R}$; choose moduli $q_0, \dots, q_{l-1} \in \text{poly}(\lambda)$; choose dimensions $l_m, n_0, \dots, n_{l-1} \in \text{poly}(\lambda)$. The algorithm outputs

$$\text{par} = \{d, \{q_i\}_{i \in [l]}, l_m, \{n_i\}_{i \in [l]}\}.$$

- $\text{Gen}_{\mathcal{L}}(\text{par}, \text{info}_{\mathcal{L}})$ takes as input public parameters par , a language-dependent information $\text{info}_{\mathcal{L}}$ specifying a norm bound $B_m > 0$ and a collection of matrices $(M_i)_{i \in [l]}$ such that $M_i \in \mathcal{R}_{q_i}^{n_i \times l_m}$. The algorithm performs the following steps:
 1. Choose a positive integer B_{Chall} and an integer $0 < d_{\text{Chall}} \leq d$ that together define the challenge space Chall as in (4). Denote by $\overline{\text{Chall}} = \{c - c' : c, c' \in \text{Chall} \text{ and } c \neq c'\}$. We require that $\#\text{Chall} = (B_{\text{Chall}} + 1)^{d_{\text{Chall}}} \in \text{poly}(\lambda)$.

2. Generate public parameters of the extractable commitment scheme of Appendix A.4 in *extraction mode* (by running `ExtSetup`), for committing to messages in the space $\cup_{1 \leq i \leq l_m} \mathcal{C}_{B_m}^i$. Let the public parameters be

$$\text{pp} = (d, q, n, l_m, l_r, B_m, B_r, \mathbf{A}, \mathbf{B}),$$

and $(\mathbf{S}, p)$ be the extraction trapdoor. We note that $\mathbf{S} \in \mathcal{R}_q^{l_m \times n}$ and p is a prime such that $\mathbf{S}\mathbf{A} + p\mathbf{E} = \mathbf{B} \bmod q$ where $\|\mathbf{E}\|_\infty \leq B_r$.

3. Chooses a Gaussian width $\alpha > 0$ and a small parameter $\gamma \geq 1$ such that

$$\alpha = \sqrt{d} \cdot B_{\text{Chall}} \cdot \sqrt{B_m^2 + dl_r B_r^2} \cdot \gamma, \quad (5)$$

and set $M = \exp(14/\gamma + 1/(2\gamma^2))$. The parameter M determines the protocol's rejection rate.

The following constraints on the parameters are required

$$\begin{aligned} \alpha \sqrt{dl_m} &< 1/2 \min\{q, q_0, \dots, q_{l-1}\}, & \alpha \sqrt{dl_r} &< q/2 \\ p \cdot 2dl_r B_r \alpha \sqrt{dl_r l_m} + 2\alpha \sqrt{dl_m} &< q/2, & 2\alpha \sqrt{dl_m} &< p/2 \end{aligned} \quad (6)$$

Let $\text{crs}_{\mathcal{L}} = ((\mathbf{M}_i)_{i \in [l]}, \text{pp}, B_{\text{Chall}}, d_{\text{Chall}}, \alpha)$. The algorithm outputs the reference string $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, where $\text{crs}_{\mathcal{L}}$ defines $\mathcal{L} = (\mathcal{L}_{\text{zk}}, \mathcal{L}_{\text{snd}})$ such that

$$\begin{aligned} \mathcal{L}_{\text{zk}} &= \{(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM} : \exists (\mathbf{m}, \mathbf{r}) \in \mathcal{S}_{B_m}^{l_m} \times \mathcal{C}_{B_r}^{l_r} \\ \text{s.t. } \mathbf{c} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix} \bmod q \text{ and } \mathbf{t}_i = \mathbf{M}_i \cdot \mathbf{m} \bmod q_i, \forall i \in [l]\}, \end{aligned} \quad (7)$$

and

$$\begin{aligned} \mathcal{L}_{\text{snd}} &= \{(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM} : \exists (\bar{\mathbf{c}}, \bar{\mathbf{m}}, \bar{\mathbf{e}}) \in \overline{\text{Chall}} \times \mathcal{S}_{B_m}^{l_m} \times \mathcal{S}_{B_e}^{l_r} \text{ s.t.} \\ (-\mathbf{S} \quad \mathbf{I}_{l_m})(\bar{\mathbf{c}} \cdot \mathbf{c}) &= p \cdot \bar{\mathbf{e}} + \bar{\mathbf{m}} \bmod q \text{ and } \bar{\mathbf{c}} \cdot \mathbf{t}_i = \mathbf{M}_i \cdot \bar{\mathbf{m}} \bmod q_i, \forall i \in [l]\}, \end{aligned} \quad (8)$$

where $\mathcal{STM} = \mathcal{R}_q^{n+l_m} \times \bigotimes_{i \in [l]} \mathcal{R}_{q_i}^{n_i}$, $\bar{B}_m = 2\alpha \sqrt{dl_m}$, $\bar{B}_e = 2dl_r B_r \alpha \sqrt{dl_r l_m}$.

- $\text{P}(\text{crs}, (\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}), (\mathbf{m}, \mathbf{r}))$ and $\text{V}(\text{crs}, (\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}))$ take as common input a CRS $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, a statement $\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1} \in \mathcal{STM}$. The prover P takes as private input a witness $(\mathbf{m}, \mathbf{r}) \in \mathcal{R}^{l_m} \times \mathcal{R}^{l_r}$ and interacts with verifier V as depicted in Figure 1.
- $\text{TrapGen}(\text{par}, \mathcal{L})$ is identical to $\text{Gen}_{\mathcal{L}}$, the difference is that it retains the trapdoor $\tau = (\mathbf{S}, p)$ generated by `ExtSetup`. It sets the trapdoor $\tau_{\Sigma} = (\mathbf{S}, p)$.
- $\text{BadChallenge}(\tau_{\Sigma}, \text{crs}, (\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}), \mathbf{a})$ on input a trapdoor $\tau_{\Sigma} = \mathbf{S} \in \mathcal{R}_q^{l_m \times n}$, a CRS $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, a statement $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM}$ and a first-move message $\mathbf{a} = (\mathbf{a}_{\mathbf{c}}, (\mathbf{a}_i)_{i \in [l]}) \in \mathcal{R}_q^{n+l_m} \times \bigotimes_{i \in [l]} \mathcal{R}_{q_i}^{r_i}$ and proceeds as follows:
 1. For each $c \in \text{Chall}$, write $(-\mathbf{S} \quad \mathbf{I}_{l_m})(\mathbf{a}_{\mathbf{c}} + c \cdot \mathbf{c}) \bmod q = p \cdot \mathbf{e} + \mathbf{z}_m$, where $\|\mathbf{z}_m\|_\infty \leq p/2$; then check if $\|\mathbf{e}\| \leq dl_r B_r \alpha \sqrt{dl_r l_m}$ and $\|\mathbf{z}_m\| \leq \alpha \sqrt{dl_m}$.
 2. For each candidate $(c, \mathbf{z}_m)$ found in the above step, test if it satisfies $\mathbf{a}_i + c \cdot \mathbf{t}_i = \mathbf{M}_i \cdot \mathbf{z}_m \bmod q_i$ for all $i \in [l]$. If there is a single surviving candidate $(c, \mathbf{z}_m)$, return c . Else, return $\perp$.

Proposition 1. *The above protocol is a trapdoor Σ -protocol for the language $\mathcal{L} = (\mathcal{L}_{\text{zk}}, \mathcal{L}_{\text{snd}})$ as specified by (7) and (8).*

Proof. We show that the requirements of Definition 1 are satisfied.

CRS Indistinguishability. This is unconditional as the syntax of $\text{Gen}_{\mathcal{L}}$ and $\text{TrapGen}_{\mathcal{L}}$ are identical except for the treatment of trapdoor τ , which $\text{Gen}_{\mathcal{L}}$ throws away while TrapGen does not.

Completeness. From Lemma 4, the probability that $\mathbf{z} \neq \perp$ is at least $(1 - 2^{-128})/M$. Moreover, as $\mathbf{z}$ follows a distribution statistically close to $\mathcal{D}_{\alpha}^{l_r + l_m}$; we have that $\|\mathbf{z}_r\| \leq \alpha\sqrt{dl_r}$ and $\|\mathbf{z}_m\| \leq \alpha\sqrt{dl_m}$ with overwhelming probability (Lemma 3). Finally, as $\mathbf{z}_r = \mathbf{u}_r + c \cdot \mathbf{r}$ and $\mathbf{z}_m = \mathbf{u}_m + c \cdot \mathbf{m}$, we can verify that

$$\mathbf{a}_c + c \cdot \mathbf{c} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{z}_r + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{z}_m \end{bmatrix} \bmod q,$$

and that $\mathbf{a}_i + c \cdot \mathbf{t}_i = \mathbf{M}_i \cdot \mathbf{z}_m \bmod q_i$ for all $i \in [l]$. Therefore the protocol is complete with error at most $1 - (1 - 2^{-128})/M \approx 1 - 1/M$.

Special soundness and correctness of BadChallenge. It suffices to prove that for a statement $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM}$ and a first-move message $\mathbf{a} = (\mathbf{a}_c, (\mathbf{a}_i)_{i \in [l]}) \in \mathcal{R}_q^{n+l_m} \times \otimes_{i \in [l]} \mathcal{R}_{q_i}^{n_i}$, if there are two distinct challenges $c, c' \in \text{Chall}$ with corresponding valid responses $\mathbf{z} = (\mathbf{z}_r, \mathbf{z}_m)$ and $\mathbf{z}' = (\mathbf{z}'_r, \mathbf{z}'_m)$ then $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{L}_{\text{snd}}$ (8). Note that we have

$$\begin{aligned} \mathbf{a}_c + c \cdot \mathbf{c} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{z}_r + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{z}_m \end{bmatrix} \bmod q \\ \mathbf{a}_i + c \cdot \mathbf{t}_i &= \mathbf{M}_i \mathbf{z}_m \bmod q_i, \end{aligned} \tag{9}$$

and that

$$\begin{aligned} \mathbf{a}_c + c' \cdot \mathbf{c} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{z}'_r + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{z}'_m \end{bmatrix} \bmod q \\ \mathbf{a}_i + c' \cdot \mathbf{t}_i &= \mathbf{M}_i \mathbf{z}'_m \bmod q_i, \end{aligned} \tag{10}$$

Recall that $\mathbf{S}\mathbf{A} + p\mathbf{E} = \mathbf{B} \bmod q$ and $\|\mathbf{E}\|_{\infty} \leq B_{\tau}$. Let $\mathbf{e} = \mathbf{E}\mathbf{z}_r$ and $\mathbf{e}' = \mathbf{E}\mathbf{z}'_r$, then $\|\mathbf{e}\|, \|\mathbf{e}'\| \leq dl_r B_{\tau} \alpha \sqrt{dl_r l_m}$, and

$$\begin{aligned} (-\mathbf{S} \quad \mathbf{I}_{l_m}) (\mathbf{a}_c + c \cdot \mathbf{c}) &= p \cdot \mathbf{e} + \mathbf{z}_m \bmod q \\ \mathbf{a}_i + c \cdot \mathbf{t}_i &= \mathbf{M}_i \mathbf{z}_m \bmod q_i, \end{aligned} \tag{11}$$

and that

$$\begin{aligned} (-\mathbf{S} \quad \mathbf{I}_{l_m}) (\mathbf{a}_c + c' \cdot \mathbf{c}) &= p \cdot \mathbf{e}' + \mathbf{z}'_m \bmod q \\ \mathbf{a}_i + c' \cdot \mathbf{t}_i &= \mathbf{M}_i \mathbf{z}'_m \bmod q_i, \end{aligned} \tag{12}$$

Let $\bar{c} = c - c'$, $\bar{\mathbf{e}} = \mathbf{e} - \mathbf{e}'$ and $\bar{\mathbf{m}} = \mathbf{z}_m - \mathbf{z}'_m$. It follows that $\|\bar{\mathbf{e}}\| \leq 2dB_{\tau}\alpha\sqrt{dl_r l_m}$, $\|\bar{\mathbf{m}}\| \leq 2\alpha\sqrt{dl_m}$ and $\bar{c} \in \overline{\text{Chall}}$. By subtracting (11) and (12), we obtain

$$\begin{aligned} (-\mathbf{S} \quad \mathbf{I}_{l_m}) (\bar{c} \cdot \mathbf{c}) &= p \cdot \bar{\mathbf{e}} + \bar{\mathbf{m}} \bmod q \\ \bar{c} \cdot \mathbf{t}_i &= \mathbf{M}_i \bar{\mathbf{m}} \bmod q_i, \forall i \in [l], \end{aligned}$$

Therefore, if $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \notin \mathcal{L}_{\text{snd}}$, for any first-move message $\mathbf{a}$ there is at most one challenge admitting a valid response.

The above argument also shows that if $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \notin \mathcal{L}_{\text{snd}}$, then the bad challenge c is also the unique challenge so that (11) can be satisfied with some $\mathbf{z}_{\mathbf{m}}$ and $\mathbf{e}$ such that $\|\mathbf{z}_{\mathbf{m}}\| \leq \alpha\sqrt{dl_{\mathbf{m}}}$, $\|\mathbf{e}\| \leq dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\mathbf{m}}}$. By the constraint of parameters as in (6), we have that

$$\|p \cdot \mathbf{e} + \mathbf{z}_{\mathbf{m}}\|_{\infty} \leq p \cdot 2dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\mathbf{m}}} + 2\alpha\sqrt{dl_{\mathbf{m}}} < q/2,$$

and that $\|\mathbf{z}_{\mathbf{m}}\| < p/2$. Therefore, **BadChallenge** correctly recovers c if it exists, and runs in polynomial time since the size of **Chall** is polynomial in λ .

Special zero-knowledge. We construct a simulator that takes as input crs , a statement $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM}$, a challenge $c \in \mathbf{Chall}$ and returns a pair $(\mathbf{a}, \mathbf{z})$ of prover's initial message and response. The simulator proceeds in the following steps:

1. Sample $\mathbf{z}_{\mathbf{r}} \leftarrow \mathcal{D}_{\alpha}^{l_{\mathbf{r}}}$, $\mathbf{z}_{\mathbf{m}} \leftarrow \mathcal{D}_{\alpha}^{l_{\mathbf{m}}}$ and parse $\mathbf{z} = (\mathbf{z}_{\mathbf{r}}, \mathbf{z}_{\mathbf{m}})$.
2. Compute $\mathbf{a}_{\mathbf{c}} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{z}_{\mathbf{r}} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{z}_{\mathbf{m}} \end{bmatrix} - c \cdot \mathbf{c} \bmod q \in \mathcal{R}_q^{n+l_{\mathbf{m}}}$
3. For $i \in [l]$, compute $\mathbf{a}_i = \mathbf{M}_i \mathbf{z}_{\mathbf{m}} - c \cdot \mathbf{t}_i \bmod q_i$ and parse $\mathbf{a} = (\mathbf{a}_{\mathbf{c}}, (\mathbf{a}_i)_{i \in [l]})$;
4. Output $(\mathbf{a}, \mathbf{z})$.

From Lemma 4, in a real execution of the protocol the distribution of $\mathbf{z}$ is within statistical distance $2^{-128}/M$ from $\mathcal{D}_{\alpha}^{l_{\mathbf{r}}+l_{\mathbf{m}}}$. Since $\mathbf{a}$ is uniquely determined from the challenge c and response $\mathbf{z}$, the distribution of $(\mathbf{a}, \mathbf{z})$ output by the simulator is statistically close to a non-aborting transcript from a real execution with c . $\square$

Boosting soundness error. The Σ -protocol in Figure 1 has an inverse polynomial soundness error and thus requires $\kappa = \Theta(\lambda/\log \lambda)$ parallel executions. Namely, the prover $\mathbf{P}$ computes κ first messages $(\mathbf{a}^{(i)})_{i \in [\kappa]}$, then computes the responses $(\mathbf{z}_i)_{i \in [\kappa]}$ after receiving the challenges $(c_i)_{i \in [\kappa]}$. $\mathbf{P}$ sets the response vector $\mathbf{z} = (\mathbf{z}_0, \dots, \mathbf{z}_{\kappa-1}) \in \mathcal{R}^{\kappa(l_{\mathbf{m}}+l_{\mathbf{r}})}$ and $c \otimes (\mathbf{m}, \mathbf{r}) = (c_0 \cdot (\mathbf{m}, \mathbf{r}), \dots, c_{\kappa-1} \cdot (\mathbf{m}, \mathbf{r})) \in \mathcal{R}^{\kappa(l_{\mathbf{m}}+l_{\mathbf{r}})}$, then outputs $\mathbf{z}$ with probability

$$\min \left(\frac{\mathcal{D}_{\alpha}^{\kappa(l_{\mathbf{m}}+l_{\mathbf{r}})}(\mathbf{z})}{\mathcal{D}_{\alpha, c \otimes (\mathbf{m}, \mathbf{r})}^{\kappa(l_{\mathbf{m}}+l_{\mathbf{r}})}(\mathbf{z})}, 1 \right).$$

Lemma 4 requires the Gaussian width

$$\alpha = B_{\text{Chall}} \sqrt{d\kappa} \cdot \sqrt{B_{\mathbf{m}}^2 + dl_{\mathbf{r}}B_{\mathbf{r}}^2} \cdot \gamma$$

to ensure that the joint rejection probability is close to $1 - 1/M$, where

$$M = \exp(14/\gamma + 1/(2\gamma^2)).$$

The rejection sampling is performed jointly over all κ parallel executions. If the joint rejection test rejects, the prover discards the entire proving attempt and

restarts with fresh randomness, generating fresh first messages for all κ executions and, when applying the NIZK compiler of Appendix B, recomputing the corresponding hash-derived challenges and responses. The prover repeats this procedure until the joint rejection test accepts. Since each attempt is accepted with probability approximately $1/M$, the expected number of attempts is M . Under the parameter selection of Appendix C.3, where $\gamma = \Theta(1)$, we have $M = \mathcal{O}(1)$.

Communication cost. The total bit size of prover’s first message and response in a single execution of the protocol in Figure 1 is

$$(n + l_{\mathbf{m}})d \log q + d \sum_{i \in [\ell]} n_i \log q_i + (l_{\mathbf{m}} + l_{\mathbf{r}})d \log(\alpha \sqrt{d(l_{\mathbf{m}} + l_{\mathbf{r}})}). \quad (13)$$

Using the Σ -protocol as the atomic protocol, we employ standard CDS OR technique [19] to construct a trapdoor Σ -protocol for proving the existence of an index $j^* \in [t]$ and $\mathbf{m} \in \mathcal{R}^{l_{\mathbf{m}}}$ such that $\|\mathbf{m}\| \leq B_{\mathbf{m}} \ll \min\{q_i : i \in [\ell]\}$ and

$$\mathbf{t}_i = \mathbf{M}_i^{(j^*)} \cdot \mathbf{m} \bmod q_i \in \mathcal{R}_{q_i}^{n_i}, \forall i \in [\ell];$$

where $\{\mathbf{M}_i^{(j)} \in \mathcal{R}_{q_i}^{n_i \times l_{\mathbf{m}}}\}_{i \in [\ell], j \in [t]}$. The details of the proposed trapdoor OR Σ -protocol are given in Appendix C.1.

4 Compact Lattice-Based Set Membership Proof

In this section, we present a lattice-based Σ -protocol for proving a set membership relation. Our approach is to employ an accumulator to “compress” a set S to a single value u , and construct a trapdoor Σ -protocol proving the existence of a witness w that $x \in S$ is accumulated in u . Thus, we rely on the lattice-based accumulator of [35] but make some generalizations. Recall that Libert *et al.* built the accumulator based on general lattices and at every level of the tree, they decomposed each hash value into binary. We generalize their construction in two directions: (i) from general lattices to module lattices, which interpolate between the general-lattice and ring-lattice extremes, thereby enabling a spectrum of trade-offs between security guarantees and parameter sizes; and (ii) from binary to base- B decomposition with arbitrary $B \geq 2$, which, in combination with Lyubashevsky-style rejection sampling, provides a knob for calibrating prover and verifier cost against communication cost. We use this generalization as an auxiliary tool for exploring efficiency–assumption trade-offs. In particular, increasing B reduces the decomposition length, but also increases the norm bound of the underlying Module-SIS problem, thereby requiring hardness for a larger solution bound and hence weakening the corresponding Module-SIS assumption.

Throughout this section, we work with parameters $n, q_{\mathbf{H}}, q_{\mathbf{C}}, B, \beta, k_{\mathbf{H}}$ and m that are polynomial in the security parameter λ . Specifically: n is a module dimension such that the full lattice dimension $dn = \Theta(\lambda)$; $q_{\mathbf{H}}, q_{\mathbf{C}} \in \text{poly}(\lambda)$ are moduli; $B > 1$ is an integer and $\beta = B - 1$; $k_{\mathbf{H}} = \lceil \log_B q_{\mathbf{H}} \rceil$ and $m = 2nk_{\mathbf{H}}$.

4.1 A Generalized Lattice-Based Accumulator

We consider a decomposition function $\text{dec}_B(\cdot)$, that on a ring element $a \in \mathcal{R}_{q_H}$, returns a vector $\text{dec}_B(a) \in \mathcal{C}_\beta^{k_H}$ by performing the steps below:

1. Write $a = a_0 + a_1X + \dots + a_{d-1}X^{d-1}$, where $a_0, \dots, a_{d-1} \in \mathbb{Z}_{q_H}$;
2. For $i \in [d]$, let $(a_{i,0}, \dots, a_{i,k_H-1})^\top \in [B]^{k_H} = \{0, \dots, \beta\}^{\lceil \log_B q_H \rceil}$ such that $a_i = \sum_{j=0}^{k_H-1} a_{i,j} \cdot B^j \bmod q_H$. Note that, $(a_{i,0}, \dots, a_{i,k_H-1})^\top$ is the base- B representation of the unique non-negative integer $A_i \in [q_H]$ such that $a_i = A_i \bmod q_H$;
3. For $i \in [k_H]$, let $\tilde{a}_i = \sum_{j=0}^{d-1} a_{j,i} X^j \in \mathcal{R}$. Output $(\tilde{a}_0, \dots, \tilde{a}_{k_H-1}) \in \mathcal{C}_\beta^{k_H}$.

The function $\text{dec}_B(\cdot)$ naturally extends to $\mathcal{R}_{q_H}^n$. Namely, for $\mathbf{a} \in \mathcal{R}_{q_H}^n$, $\text{dec}_B(\mathbf{a}) = (\text{dec}_B(\mathbf{a}[0]), \dots, \text{dec}_B(\mathbf{a}[n-1])) \in \mathcal{C}_\beta^{nk_H}$.

Observe that $\text{dec}_B(\cdot)$ generalizes the base-2 gadget decomposition of [47] to the cyclotomic ring $\mathcal{R}$. Therefore, if we consider the *base- B gadget vector*

$$\mathbf{g}_B := (1 \ B \ \dots \ B^{k_H-1}) \in \mathbb{Z}^{k_H},$$

and the *base- B gadget matrix*

$$\mathbf{G}_B := \begin{pmatrix} \mathbf{g}_B & & \\ & \mathbf{g}_B & \\ & & \ddots \\ & & & \mathbf{g}_B \end{pmatrix} \in \mathbb{Z}^{n \times nk_H},$$

then it follows that $\mathbf{a} = \mathbf{G}_B \cdot \text{dec}_B(\mathbf{a}) \bmod q$, for any $\mathbf{a} \in \mathcal{R}_q^n$.

We now recall the family of 2-input lattice-based collision-resistant hash functions of [35], using the base- B decomposition $\text{dec}_B(\cdot)$ as a component.

Definition 7. *The function family $\mathcal{H}$ mapping $\mathcal{C}_\beta^{nk_H} \times \mathcal{C}_\beta^{nk_H}$ to $\mathcal{C}_\beta^{nk_H}$ is defined as $\mathcal{H} = \{f_{\mathbf{H}} \mid \mathbf{H} \in \mathcal{R}_{q_H}^{n \times m}\}$, where for $\mathbf{H} = [\mathbf{H}_0 \mid \mathbf{H}_1]$ with $\mathbf{H}_0, \mathbf{H}_1 \in \mathcal{R}_{q_H}^{n \times nk_H}$, and for any $(\mathbf{u}_0, \mathbf{u}_1) \in (\mathcal{C}_\beta^{nk_H})^2$, we have:*

$$f_{\mathbf{H}}(\mathbf{u}_0, \mathbf{u}_1) = \text{dec}_B(\mathbf{H}_0 \cdot \mathbf{u}_0 + \mathbf{H}_1 \cdot \mathbf{u}_1 \bmod q_H) \in \mathcal{C}_\beta^{nk_H}.$$

Similar to [35], we have the following.

Lemma 1. *Under $\text{MSIS}_{n,m,\beta,q_H}^{(\infty)}$ assumption, the function family $\mathcal{H}$ of Definition 7 is collision-resistant.*

We now recall the lattice-based accumulator of [35]. The accumulator is built from a binary Merkle tree with $N = 2^\ell$ leaves where ℓ is a positive integer, and from the family of lattice-based hash function $\mathcal{H}$ in Definition 7.

- **TSetup**(1^λ) Choose a modulus $q_H \in \text{poly}(\lambda)$, a dimension n , an integer base $B > 1$ specifying an norm bound $\beta = B - 1$, a parameter $k_H = \lceil \log_B q_H \rceil$ and a dimension $m = 2nk_H$. The algorithm then samples $\mathbf{H} \xleftarrow{\$} \mathcal{R}_{q_H}^{n \times m}$, and outputs $pp = (q_H, n, B, \mathbf{H} = [\mathbf{H}_0 \mid \mathbf{H}_1])$, where $\mathbf{H}_0, \mathbf{H}_1 \in \mathcal{R}_{q_H}^{nk_H}$.

- TAcc($R = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\} \subset \mathcal{C}_\beta^{nk_H}$) For $j \in [n]$, let $(j_0, \dots, j_{\ell-1}) \in \{0, 1\}^\ell$ be the binary representation of j , and let $\mathbf{d}_j = \mathbf{u}_{j_0, \dots, j_{\ell-1}}$. Form the tree of depth $\ell = \log N$ with N leaves $\mathbf{u}_{0,0,\dots,0}, \dots, \mathbf{u}_{1,1,\dots,1}$, where the nodes are computed from the bottom as follows:
 - at depth $i \in [\ell] \setminus \{0\}$, for all $(b_0, \dots, b_{i-1}) \in \{0, 1\}^i$ the node $\mathbf{u}_{b_0, \dots, b_{i-1}} \in \mathcal{C}_\beta^{nk_H}$ is defined as $f_{\mathbf{H}}(\mathbf{u}_{b_0, \dots, b_{i-1}, 0}, \mathbf{u}_{b_0, \dots, b_{i-1}, 1})$;
 - at depth 0, the root $\mathbf{u} \in \mathcal{C}_\beta^{nk_H}$ is defined as $f_{\mathbf{H}}(\mathbf{u}_0, \mathbf{u}_1)$.
- TWitness($R, \mathbf{d}$) If $\mathbf{d} \notin R$, return $\perp$. Otherwise, $\mathbf{d} = \mathbf{d}_j = \mathbf{u}_{j_0, \dots, j_{\ell-1}}$ for some $j \in [N]$ with binary representation $(j_0, \dots, j_{\ell-1}) \in \{0, 1\}^\ell$. Output the witness w defined as:

$$w = ((j_0, \dots, j_{\ell-1}), (\mathbf{w}_0, \dots, \mathbf{w}_{\ell-1})) \in \{0, 1\}^\ell \times (\mathcal{C}_\beta^{nk_H})^\ell,$$

where $\mathbf{w}_i = \mathbf{u}_{j_0, \dots, 1-j_i}$ is the sibling of the node $\mathbf{u}_{j_0, \dots, j_i}$ on the path from $\mathbf{d}$ to the root $\mathbf{u}$.

- TVerify($\mathbf{u}, \mathbf{d}, w$) Parse the witness w as:

$$w = ((j_0, \dots, j_{\ell-1}), (\mathbf{w}_0, \dots, \mathbf{w}_{\ell-1})) \in \{0, 1\}^\ell \times (\mathcal{C}_\beta^{nk_H})^\ell,$$

the algorithm sets $\mathbf{v}_{\ell-1} = \mathbf{d}$ and computes the nodes $\mathbf{v}_{\ell-2}, \dots, \mathbf{v}_0, \mathbf{v}_{-1}$ on the path from $\mathbf{d}$ to $\mathbf{u}$, using the following recursive formula

$$\mathbf{v}_{k-1} = \text{dec}_B(\mathbf{H}_{j_k} \cdot \mathbf{v}_k + \mathbf{H}_{1-j_k} \cdot \mathbf{w}_k \bmod q_H), \forall k \in [\ell].$$

Then it returns 1 if $\mathbf{v}_{-1} = \mathbf{u}$. Otherwise, it returns 0.

Similar to [35], we have the following.

Theorem 1. *The above accumulator scheme is correct and secure in the sense of Definition 3, under $\text{MSIS}_{n,m,\beta,q_H}^{(\infty)}$ assumption.*

4.2 A Trapdoor Σ -Protocol for Set Membership Relation

Overview We construct a Σ -protocol that allows a prover P to prove that a KTX commitment [32] (recalled in Appendix A.3) opens to some element in a public (multi)set. Formally, let $\mathcal{D} = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\} \subset \mathcal{C}_\beta^{nk_H}$, $\mathbf{c}_\mathbf{d} \in \mathcal{R}_{q_C}^{nc}$, and $\mathbf{C} \in \mathcal{R}_{q_C}^{nc \times nk_H}$, $\mathbf{D} \in \mathcal{R}_{q_C}^{nc \times n_R}$ be public parameters of the KTX commitment scheme, P proves that there exists $\mathbf{d} \in \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\}$ and $\mathbf{r}_\mathbf{d} \in \mathcal{C}_{B_R}^{n_R}$ such that $\mathbf{c}_\mathbf{d} = \mathbf{C}\mathbf{d} + \mathbf{D}\mathbf{r}_\mathbf{d} \bmod q_C$.

The prover P first accumulates $\mathcal{D}$ into a value $\mathbf{u}$ via the accumulator of Section 4.1, then proves that: $\mathbf{c}_\mathbf{d}$ is a commitment to $\mathbf{d}$; and $\mathbf{d}$ is properly accumulated into $\mathbf{u}$. The latter essentially requires P to prove the existence of

$$w = ((j_0, \dots, j_{\ell-1}), (\mathbf{w}_0, \dots, \mathbf{w}_{\ell-1})) \in \{0, 1\}^\ell \times (\mathcal{C}_\beta^{nk_H})^\ell,$$

such that $\text{TVerify}(\mathbf{u}, \mathbf{d}, w) = 1$. By considering the nodes $\mathbf{v}_{\ell-1} = \mathbf{d}, \mathbf{v}_{\ell-1}, \dots, \mathbf{v}_0$, and $\mathbf{v}_{-1} = \mathbf{u}$ on the path computed by $\text{TVerify}(\mathbf{u}, \mathbf{d}, w)$, we can rewrite the statement $\text{TVerify}(\mathbf{u}, \mathbf{d}, w) = 1$ as a set of linear equations:

$$\forall i \in [\ell] : \mathbf{v}_{i-1} = \begin{cases} \text{dec}_B(\mathbf{H}_0 \cdot \mathbf{v}_i + \mathbf{H}_1 \cdot \mathbf{w}_i \bmod q_H), & \text{if } j_i = 0; \\ \text{dec}_B(\mathbf{H}_1 \cdot \mathbf{v}_i + \mathbf{H}_0 \cdot \mathbf{w}_i \bmod q_H), & \text{if } j_i = 1. \end{cases}$$

The above equations capture the hash chain relation at every depth in the Merkle tree underlying the accumulator, and they are equivalent to the following

$$\forall i \in [\ell] : \mathbf{G}_B \mathbf{v}_{i-1} = \begin{cases} \mathbf{H}_0 \cdot \mathbf{v}_i + \mathbf{H}_1 \cdot \mathbf{w}_i \bmod q_H, & \text{if } j_i = 0; \\ \mathbf{H}_1 \cdot \mathbf{v}_i + \mathbf{H}_0 \cdot \mathbf{w}_i \bmod q_H, & \text{if } j_i = 1; \end{cases} \quad (14)$$

The equations of (14) are special cases of the OR linear relations specified by (27). Therefore, to prove (14), prover P can simply compose ℓ instances of the trapdoor Σ -protocol of Appendix C.1 to simultaneously prove all the hash chain relations. A caveat is that the intermediate nodes $\mathbf{v}_{\ell-2}, \dots, \mathbf{v}_1$ are reused in these ℓ protocols as witnesses, thus P needs to commit to all of these nodes and additionally proves that the commitments are well-formed. More concretely, for $\ell - 1 > i \geq 0$, the prover samples commitment randomness $\mathbf{r}_{\mathbf{v}_i}$ and computes

$$\mathbf{c}_{\mathbf{v}_i} = \mathbf{C} \cdot \mathbf{v}_i + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_i} \bmod q_C.$$

The full linear relations at depth $\ell - 1 \geq i > 0$ now become

$$\begin{aligned} \mathbf{c}_{\mathbf{v}_i} &= \mathbf{C} \cdot \mathbf{v}_i + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_i} \bmod q_C, \\ \mathbf{c}_{\mathbf{v}_{i-1}} &= \mathbf{C} \cdot \mathbf{v}_{i-1} + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_{i-1}} \bmod q_C, \\ \exists j \in \{0, 1\} : \mathbf{0} &= [\mathbf{H}_j | \mathbf{H}_{1-j} | -\mathbf{G}_B] \cdot (\mathbf{v}_{\ell-1}, \mathbf{w}_{\ell-1}, \mathbf{v}_{\ell-2}) \bmod q_H; \end{aligned}$$

where $\mathbf{c}_{\mathbf{v}_{\ell-1}} = \mathbf{c}_{\mathbf{d}}$ and $\mathbf{r}_{\mathbf{v}_{\ell-1}} = \mathbf{r}_{\mathbf{d}}$. The prover now forms the witness $\mathbf{m}_i = (\mathbf{v}_i, \mathbf{r}_{\mathbf{v}_i}, \mathbf{v}_{i-1}, \mathbf{r}_{\mathbf{v}_{i-1}}, \mathbf{w}_i)$ and proves that

$$\begin{aligned} (\mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) &= \mathbf{M}_0 \cdot \mathbf{m}_i \bmod q_C, \\ \exists j \in \{0, 1\} : \mathbf{0}^n &= \mathbf{M}_0^{(j)} \mathbf{m}_i \bmod q_H; \end{aligned} \quad (15)$$

where

$$\begin{aligned} \mathbf{M}_0 &= \begin{bmatrix} \mathbf{C} & \mathbf{D} & \mathbf{0}^{n_C \times nk_H} \\ & \mathbf{C} & \mathbf{D} & \mathbf{0}^{n_C \times nk_H} \end{bmatrix} \in \mathcal{R}_{q_C}^{2n_C \times (3nk_H + 2n_R)}, \\ \mathbf{M}_0^{(j)} &= [\mathbf{H}_j | \mathbf{0}^{n \times n_R} | -\mathbf{G}_B | \mathbf{0}^{n \times n_R} | \mathbf{H}_{1-j}] \in \mathcal{R}_{q_H}^{n \times (3nk_H + 2n_R)}; \end{aligned} \quad (16)$$

and $\|\mathbf{m}_i\| \leq \sqrt{d[3\beta^2 nk_H + 2B_R^2 n_R]}$. Similarly, at depth 0 we have

$$\begin{aligned} \mathbf{c}_{\mathbf{v}_0} &= \mathbf{C} \cdot \mathbf{v}_0 + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_0} \bmod q_C, \\ \exists j \in \{0, 1\} : \mathbf{G}_B \cdot \mathbf{u} &= [\mathbf{H}_j | \mathbf{H}_{1-j}] \cdot (\mathbf{v}_0, \mathbf{w}_0) \bmod q_H. \end{aligned}$$

The above can be rewritten as

$$\begin{aligned} \mathbf{c}_{\mathbf{v}_0} &= [\mathbf{C}|\mathbf{D}|\mathbf{0}^{n_C \times nk_H}] \cdot \mathbf{m}_0 \bmod q_C, \\ \exists j \in \{0, 1\} : \mathbf{G}_B \mathbf{u} &= [\mathbf{H}_j|\mathbf{0}^{n \times nk_H}|\mathbf{H}_{1-j}] \cdot \mathbf{m}_0 \bmod q_H, \end{aligned} \quad (17)$$

where the witness $\mathbf{m}_0 = (\mathbf{v}_0, \mathbf{r}_{\mathbf{v}_0}, \mathbf{w}_0)$ satisfies $\|\mathbf{m}_0\| \leq \sqrt{d(2\beta^2 nk_H + B_R^2 n_R)}$. The prover then proves (15) and (17) by composing ℓ instances of the OR trapdoor Σ -protocol of Appendix C.1, tied altogether via a single verifier's challenge.

Protocol Description We now give a trapdoor Σ -protocol proving a KTX commitment opens to an element in a set of cardinality $N = 2^\ell$. The details of the algorithms underlying the proposed trapdoor Σ -protocol are given as follows:

- $\text{Gen}_{\text{par}}(1^\lambda)$: On input security parameter $\lambda \in \mathbb{N}$, choose a power-of-2 d defining number field $\mathcal{R}$; choose two prime moduli:
 - $q_C \in \text{poly}(\lambda)$ (for the KTX commitment);
 - $q_H \in \text{poly}(\lambda)$ (for the 2-input hash function of Section 4.1).

The algorithm then chooses module dimensions $n, n_C, n_R \in \text{poly}(\lambda)$ such that the corresponding lattice dimensions $dn, dn_C \in \Theta(\lambda)$, then chooses an integer base $B > 1$ and a norm bound B_R . These parameters additionally define $k_H = \lceil \log_B q_H \rceil$, a norm bound $\beta = B - 1$, a dimension $m = 2nk_H$ and a dimension $l_{\max} = 3nk_H + 2n_R$ that specifies the maximum message dimension in the extractable commitment scheme formally defined in Appendix A.4. The algorithm outputs

$$\text{par} = \{d, q_C, q_H, n, n_C, n_R, B, B_R\}. \quad (18)$$

- $\text{Gen}_{\mathcal{L}}(\text{par}, \text{info}_{\mathcal{L}})$: Takes as input par and language information $\text{info}_{\mathcal{L}} = N = 2^\ell \in \text{poly}(\lambda)$, the algorithm then proceeds as follows:
 1. Choose a positive integer B_{Chall} and an integer $0 < d_{\text{Chall}} \leq d$ that together define the challenge space Chall as in (4). Denote by $\overline{\text{Chall}} = \{\mathbf{c} - \mathbf{c}' : \mathbf{c}, \mathbf{c}' \in \text{Chall} \text{ and } \mathbf{c} \neq \mathbf{c}'\}$ and define a binary operation $\oplus$ over Chall such that for all $\mathbf{c}_0, \mathbf{c}_1 \in \text{Chall}$

$$\mathbf{c}_0 \oplus \mathbf{c}_1 = \mathbf{c} \in \text{Chall} \text{ iff } \forall i \in [l] : \mathbf{c}[i] = \mathbf{c}_0[i] + \mathbf{c}_1[i] \bmod (B_{\text{Chall}} + 1).$$

We require that $\#\text{Chall} = (B_{\text{Chall}} + 1)^{d_{\text{Chall}}}$ is a polynomial in λ .

2. Let $B_{\mathbf{m}} = \sqrt{d(3\beta^2 nk_H + 2B_R^2 n_R)}$, then generate public parameters of the extractable commitment scheme of Appendix A.4 in *extraction mode* (by running ExtSetup), for committing to messages in $\cup_{1 \leq i \leq l_{\max}} \mathcal{C}_{B_{\mathbf{m}}}^i$. Let the public parameters be

$$\text{pp} = (d, q, n, l_{\max}, l_r, B_{\mathbf{m}}, B_r, \mathbf{A}, \mathbf{B}),$$

and $(\mathbf{S}, p)$ be the extraction trapdoor. We note that $\mathbf{S} \in \mathcal{R}_q^{l_{\max} \times n}$ and p is a prime such that $\mathbf{S}\mathbf{A} + p\mathbf{E} = \mathbf{B} \bmod q$ where $\|\mathbf{E}\|_\infty \leq B_\tau$.

3. Sample a matrix $\mathbf{H} = [\mathbf{H}_0 | \mathbf{H}_1] \xleftarrow{\$} \mathcal{R}_{q_H}^{n \times m}$, where $\mathbf{H}_0, \mathbf{H}_1 \in \mathcal{R}_{q_H}^{n \times nk_H}$. The matrix $\mathbf{H}$ defines a 2-input lattice-based hash function of Section 4.1;
4. Sample matrices $\mathbf{C} \xleftarrow{\$} \mathcal{R}_{q_C}^{n_C \times nk_H}$, $\mathbf{D} \xleftarrow{\$} \mathcal{R}_{q_C}^{n_C \times n_R}$ that define the public parameters of the KTX commitment scheme [32];
5. Choose a Gaussian width $\alpha > 0$ and a parameter $\gamma \geq 1$ such that

$$\alpha = \sqrt{d} \cdot B_{\text{Chall}} \cdot \sqrt{B_{\mathbf{m}}^2 + dl_{\mathbf{r}} B_{\mathbf{r}}^2} \cdot \gamma$$

Set $M = \exp(14/\gamma + 1/(2\gamma^2))$.

From (29), we require that

$$\begin{aligned} \alpha \sqrt{dl_{\max}} &< 1/2 \min\{q, q_H, q_C\}, & \alpha \sqrt{dl_{\mathbf{r}}} &< q/2, \\ p \cdot 2dl_{\mathbf{r}} B_{\tau} \alpha \sqrt{dl_{\mathbf{r}} l_{\max}} + 2\alpha \sqrt{dl_{\max}} &< q/2, & 2\alpha \sqrt{dl_{\max}} &< p/2. \end{aligned}$$

The algorithm sets $\text{crs}_{\mathcal{L}} = (q, l_{\mathbf{r}}, \mathbf{B}_{\mathbf{m}}, B_{\mathbf{r}}, \mathbf{A}, \mathbf{B}, \mathbf{H}, \mathbf{C}, \mathbf{D}, B_{\text{Chall}}, d_{\text{Chall}}, \alpha)$ and outputs $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$. For $i \in [\ell]$ and $j \in \{0, 1\}$, $\text{crs}_{\mathcal{L}}$ defines the languages $\mathcal{L}_{\text{set}} = (\mathcal{L}_{\text{set}, \text{zk}}, \mathcal{L}_{\text{set}, \text{snd}})$ such that

$$\begin{aligned} \mathcal{L}_{\text{set}, \text{zk}} &= \{(\mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}) \in \mathcal{STM} \text{ s.t.} \\ \forall i \in [\ell] \text{ and } i > 0 : (\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) &\in \mathcal{L}_{\text{set}, \text{zk}}^{(i)} \text{ and } (\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{L}_{\text{set}, \text{zk}}^{(0)}\}, \end{aligned} \quad (19)$$

and

$$\begin{aligned} \mathcal{L}_{\text{set}, \text{snd}} &= \{(\mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}) \in \mathcal{STM} \text{ s.t.} \\ \forall i \in [\ell] \text{ and } i > 0 : (\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) &\in \mathcal{L}_{\text{set}, \text{snd}}^{(i)} \text{ and } (\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{L}_{\text{set}, \text{snd}}^{(0)}\}, \end{aligned} \quad (20)$$

where $\mathcal{STM} = \mathcal{C}_{\beta}^{nk_H} \times \mathcal{R}_q^{n+m+n_R} \times (\mathcal{R}_q^{n+l_{\max}})^{\ell-1} \times (\mathcal{R}_{q_C}^{n_C})^{\ell}$. The languages

$$\mathcal{L}_{\text{set}, \text{zk}}^{(i)} := \cup_{j \in \{0, 1\}} \mathcal{L}_{\text{set}, \text{zk}}^{(i, j)}, \quad \mathcal{L}_{\text{set}, \text{snd}}^{(i)} := \cup_{j \in \{0, 1\}} \mathcal{L}_{\text{set}, \text{snd}}^{(i, j)}$$

capture the linear relations at depth i of the accumulator, namely:

- For $\ell - 1 \geq i > 0$, we have that

$$\begin{aligned} \mathcal{L}_{\text{set}, \text{zk}}^{(i, j)} &= \{(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) \in \mathcal{R}_q^{n+l_{\max}} \times (\mathcal{R}_{q_C}^{n_C})^2 : \\ \exists (\mathbf{m}_i, \mathbf{r}_i) &\in (\mathcal{C}_B^{nk_H} \times \mathcal{C}_{B_R}^{n_R})^2 \times \mathcal{C}_B^{nk_H} \times \mathcal{C}_{B_{\mathbf{r}}}^{l_{\mathbf{r}}} \text{ s.t.} \\ \mathbf{c}_i &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r}_i + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_i \end{bmatrix} \bmod q, \quad (\mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) = \mathbf{M}_0 \cdot \mathbf{m}_i \bmod q_C, \\ \text{and } \mathbf{0}^n &= \mathbf{M}_0^{(j)} \mathbf{m}_i \bmod q_H\}, \end{aligned} \quad (21)$$

and

$$\begin{aligned} \mathcal{L}_{\text{set}, \text{snd}}^{(i, j)} &= \{(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) \in \mathcal{R}_q^{n+l_{\max}} \times (\mathcal{R}_{q_C}^{n_C})^2 : \\ \exists (\bar{\mathbf{c}}_i, \bar{\mathbf{m}}_i, \bar{\mathbf{e}}_i) &\in \overline{\text{Chall}} \times \mathcal{S}_{B_{\mathbf{m}}}^{l_{\max}} \times \mathcal{S}_{B_{\mathbf{e}}}^{l_{\max}} \text{ s.t.} \\ [-\mathbf{S} \quad \mathbf{I}_{l_{\max}}] &(\bar{\mathbf{c}}_i \cdot \mathbf{c}_i) = p \cdot \bar{\mathbf{e}}_i + \bar{\mathbf{m}}_i \bmod q, \\ \bar{\mathbf{c}}_i \cdot (\mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) &= \mathbf{M}_0 \cdot \bar{\mathbf{m}}_i \bmod q_C \text{ and } \mathbf{0}^n = \mathbf{M}_0^{(j)} \bar{\mathbf{m}}_i \bmod q_H\}, \end{aligned} \quad (22)$$

where $\bar{\mathbf{B}}_{\mathbf{m}} = 2\alpha \sqrt{dl_{\mathbf{m}}}$, $\bar{B}_{\mathbf{e}} = 2dl_{\mathbf{r}} B_{\tau} \alpha \sqrt{dl_{\mathbf{r}} l_{\mathbf{m}}}$; $\mathbf{M}_0, \mathbf{M}_0^{(j)}$ are defined by (16).

- At depth $i = 0$, we have that

$$\begin{aligned}
\mathcal{L}_{\text{set,zk}}^{(0,j)} &= \{(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{C}_\beta^{nk_H} \times \mathcal{R}_q^{n+m+n_R} \times \mathcal{R}_{q_C}^{n_C} : \\
&\exists(\mathbf{m}_0, \mathbf{r}_0) \in (\mathcal{C}_B^{nk_H} \times \mathcal{C}_{B_R}^{n_R})^2 \times \mathcal{C}_B^{nk_H} \times \mathcal{C}_{B_r}^{l_r} \text{ s.t.} \\
\mathbf{c}_0 &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B}[: m + n_R] \end{bmatrix} \mathbf{r}_0 + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_0 \end{bmatrix} \bmod q, \\
\mathbf{c}_{\mathbf{v}_0} &= [\mathbf{C}|\mathbf{D}|\mathbf{0}^{n_C \times nk_H}] \cdot \mathbf{m}_0 \bmod q_C, \\
&\text{and } \mathbf{G}_B \mathbf{u} = [\mathbf{H}_j|\mathbf{0}^{n \times nk_H}|\mathbf{H}_{1-j}] \cdot \mathbf{m}_0 \bmod q_H\}.
\end{aligned} \tag{23}$$

$$\begin{aligned}
\mathcal{L}_{\text{set,snd}}^{(0,j)} &= \{(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{C}_\beta^{nk_H} \times \mathcal{R}_q^{n+m+n_R} \times \mathcal{R}_{q_C}^{n_C} : \\
&\exists(\bar{\mathbf{c}}_0, \bar{\mathbf{m}}_0, \bar{\mathbf{e}}_0) \in \overline{\text{Chall}} \times \mathcal{S}_{B_m}^{m+n_R} \times \mathcal{S}_{B_e}^{m+n_R} \text{ s.t.} \\
&[-\mathbf{S}[: m + n_R] \quad \mathbf{I}_{m+n_R}] (\bar{\mathbf{c}}_0 \cdot \mathbf{c}_0) = p \cdot \bar{\mathbf{e}}_0 + \bar{\mathbf{m}}_0 \bmod q, \\
\bar{\mathbf{c}}_0 \cdot \mathbf{c}_{\mathbf{v}_0} &= [\mathbf{C}|\mathbf{D}|\mathbf{0}^{n_C \times nk_H}] \cdot \bar{\mathbf{m}}_0 \bmod q_C, \\
&\text{and } \bar{\mathbf{c}}_0 \cdot \mathbf{G}_B \mathbf{u} = [\mathbf{H}_j|\mathbf{0}^{n \times n_R}|\mathbf{H}_{1-j}] \cdot \bar{\mathbf{m}}_0 \bmod q_H\},
\end{aligned} \tag{24}$$

$$\text{where } \bar{B}_m = 2\alpha\sqrt{dl_m}, \bar{B}_e = 2dl_r B_\tau \alpha\sqrt{dl_r l_m}.$$

- The prover $\mathbf{P}$ and verifier $\mathbf{V}$ take as common input a CRS crs , a statement $(\mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, \dots, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}) \in \mathcal{STM}$. $\mathbf{P}$ takes as private input a witness $((\mathbf{m}_i, \mathbf{r}_i))_{i \in [\ell]}$ that $(\mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}) \in \mathcal{L}_{\text{set,zk}}$. Then for $i \in [\ell]$, $\mathbf{P}$ runs the prover in the trapdoor Σ -protocol Π_i of Appendix C.1 for $\mathcal{L}_{\text{set,zk}}^{(i)}$ as follows:
 - For $i = 0$, $\mathbf{P}$ runs the prover of the Σ -protocol Π_0 for $\mathcal{L}_{\text{set}}^{(0)}$ (see (23), (24)), with input CRS crs , statement $(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0})$ and witness $(\mathbf{m}_0, \mathbf{r}_0)$;
 - For $\ell - 1 \geq i > 0$, $\mathbf{P}$ runs the prover of the Σ -protocol Π_i for $\mathcal{L}_{\text{set}}^{(i)}$ (see (21), (22)), with input CRS crs , statement $(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}})$ and witness $(\mathbf{m}_i, \mathbf{r}_i)$;
Let the prover's first message in the protocol Π_i be MSG_i . $\mathbf{P}$ then sends $(\text{MSG}_i)_{i \in [\ell]}$ to the verifier $\mathbf{V}$.

Upon receiving $\mathbf{V}$'s challenge $c \xleftarrow{\$} \text{Chall}$, $\mathbf{P}$ executes the provers of Π_i to obtain the corresponding response RSP_i . Let the final response be $(\text{RSP}_i)_{i \in [\ell]}$. Receiving $\mathbf{P}$'s response $(\text{RSP}_i)_{i \in [\ell]}$, $\mathbf{V}$ runs the verifier of each protocol Π_i to check that $(\text{MSG}_i, c, \text{RSP}_i)$ is a valid transcript of Π_i and outputs 1 if and only if all checks pass.

- $\text{TrapGen}(\text{par}, \mathcal{L})$ is identical to $\text{Gen}_{\mathcal{L}}$, the difference is that it retains the trapdoor $\tau = (\mathbf{S}, p)$. It sets the trapdoor $\tau_\Sigma = (\mathbf{S}, p)$.
- $\text{BadChallenge}_{\text{Set}}(\tau_\Sigma, \text{crs}, \text{STM}, (\text{MSG}_i)_{i \in [\ell]})$ takes input a trapdoor $\tau_\Sigma = \mathbf{S} \in \mathcal{R}_q^{l_m \times n}$, a CRS $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, a statement $\text{STM} = (\mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}) \in \mathcal{STM}$ and a first message $(\text{MSG}_i)_{i \in [\ell]}$. It then proceeds as follows.
 1. For $i \in [\ell]$, execute the $\text{BadChallenge}_{\text{OR}}$ algorithm underlying the trapdoor protocol Π^i for the language $(\mathcal{L}_{\text{set,zk}}^{(i)}, \mathcal{L}_{\text{set,snd}}^{(i)})$, namely:
 - When $i = 0$, execute the $\text{BadChallenge}_{\text{OR}}$ algorithm underlying the trapdoor protocol Π_0 on input the trapdoor τ_Σ , the string crs , the statement $(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0})$ and first message MSG_0 ;

- When $\ell - 1 \geq i > 0$, execute the $\text{BadChallenge}_{\text{OR}}$ algorithm underlying the trapdoor protocol Π_i on input the trapdoor τ_Σ , the string crs , the statement $(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}})$ and first message MSG_i .
- 2. If $c_0 = \dots = c_{\ell-1} := c \in \text{Chall}$, output c . Else, output $\perp$.

Proposition 2. *The above protocol is a trapdoor Σ -protocol for the language $\mathcal{L}_{\text{set}} = (\mathcal{L}_{\text{set,zk}}, \mathcal{L}_{\text{set,snd}})$ specified by (19), (20), (21), (22), (23) and (24).*

Proof. The above Σ -protocol is unconditionally CRS indistinguishable and special zero-knowledge since the trapdoor Σ -protocol of Appendix C.1 is special zero-knowledge. For special soundness, assume that for a statement

$$\text{STM} = (\mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}) \in \mathcal{C}_\beta^{n_{\text{KH}}} \times (\mathcal{R}_q^{n+l_{\text{max}}})^{\ell-1} \times \mathcal{R}_q^{n+m+n_R} \times (\mathcal{R}_{qc}^{nc})^\ell$$

and a first message $(\text{MSG}_i)_{i \in [\ell]}$, there are two valid pairs of challenge-response

$$c \in \text{Chall}, (\text{RSP}_i)_{i \in [\ell]},$$

and

$$c' \in \text{Chall}, (\text{RSP}'_i)_{i \in [\ell]}$$

such that $c \neq c'$. This implies that for $i > 0$, (c, RSP_i) and (c', RSP'_i) are two valid pairs of challenge-response with respect to the first message MSG_i and statement $(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}})$, in the trapdoor Σ -protocol for the language $(\mathcal{L}_{\text{set,zk}}^{(i)}, \mathcal{L}_{\text{set,snd}}^{(i)})$; and that (c, RSP_0) and (c', RSP'_0) are two valid pairs of challenge-response with respect to the first message MSG_0 and statement $(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0})$, in the trapdoor Σ -protocol for the language $(\mathcal{L}_{\text{set,zk}}^{(0)}, \mathcal{L}_{\text{set,snd}}^{(0)})$. Therefore, by Proposition 4, we have that $(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) \in \mathcal{L}_{\text{set,snd}}^{(i)}$ for all $i > 0$, and that $(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{L}_{\text{set,snd}}^{(0)}$. Therefore $\text{STM} \in \mathcal{L}_{\text{set,snd}}$.

For correctness of $\text{BadChallenge}_{\text{Set}}$, note that if $\text{STM} \notin \mathcal{L}_{\text{set,snd}}$, then there exists some $i \in [\ell]$ such that $(\mathbf{c}_j, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) \notin \mathcal{L}_{\text{set,snd}}^{(i)}$ if $i > 0$, or that $(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \notin \mathcal{L}_{\text{set,snd}}^{(0)}$. In either case, the input statement to the trapdoor Σ -protocol Π_i for $(\mathcal{L}_{\text{set,zk}}^{(i)}, \mathcal{L}_{\text{set,snd}}^{(i)})$ is a false statement and therefore there can be at most one challenge c so that the prover can respond properly. Due to the correctness of $\text{BadChallenge}_{\text{OR}}$, the correctness of $\text{BadChallenge}_{\text{Set}}$ then follows. Furthermore, the runtime of $\text{BadChallenge}_{\text{Set}}$ scales linearly in $\ell = \log \lambda$ and therefore is polynomial. $\square$

The NIZK obtained by applying the compiler of Appendix B to the above protocol inherits this relaxed soundness guarantee. In particular, honest proofs are generated for statements in $\mathcal{L}_{\text{set,zk}}$, while soundness guarantees only that an accepting statement belongs to $\mathcal{L}_{\text{set,snd}}$. Thus, as a standalone primitive, the resulting NIZK certifies set membership only with respect to the relaxed relation $\mathcal{L}_{\text{set,snd}}$, rather than exact membership in $\mathcal{L}_{\text{set,zk}}$. We do not provide a generic transformation from a relaxed witness to an exact membership witness; applications requiring exact standalone membership would need an additional mechanism to rule out relaxed-only witnesses. In our ring-signature application, this is unnecessary, as the unforgeability proof works directly with the relaxed relation and reduces a successful forgery to the underlying Module-SIS problems.

Communication cost Recall that the prover executes ℓ Σ -protocols Π_i of Section C.1 for proving linear relations with two moduli q_C, q_H , witness's dimension $l_{\max} = 3nk_H + 2n_R$ and $t = 2$. By (33) it follows that for $i > 0$, the prover's communication cost in Π_i is

$$2(n + 3nk_H + 2n_R)d \log q + 2n_R d \log q_C + 2nd \log q_H + \\ 2(3nk_H + 2n_R + l_r)d \log(\alpha \sqrt{d((3nk_H + 2n_R + l_r))}) + 2d_{\text{Chall}} \log(B_{\text{Chall}} + 1),$$

and the prover's communication cost in Π_0 is

$$2(n + m + n_R)d \log q + 2n_R d \log q_C + 2nd \log q_H + \\ 2(m + n_R + l_r)d \log(\alpha \sqrt{d(m + n_R + l_r)}) + 2d_{\text{Chall}} \log(B_{\text{Chall}} + 1).$$

Combining the above, the total prover's communication cost in a single execution of the trapdoor Σ -protocol for $\mathcal{L}_{\text{set}}$ is of order $\mathcal{O}(\ell \cdot (nk_H d \log \lambda + n_R d \log \lambda)) = \mathcal{O}(\ell \cdot \lambda \log^2 \lambda)$, which is logarithmic in set size.

5 A Compact Ring Signature in the Standard Model

5.1 Overview

In this section, we present a standard-model ring signature scheme with logarithmic signature size, making use of the trapdoor Σ -protocol for set membership of Section 4.2 as a key component. At a high level, our scheme adapts the set membership proof to the ring setting by binding the signer's secret key through an additional linear relation, while preserving logarithmic size. We also introduce a new method for binding the message to the ring, by binding the message M *directly* to the generation of accumulated value. Namely, if $R = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\}$ is a ring of public keys, then given a message M , we transform R and M to a new set $R' = \{(\mathbf{d}_0, M), \dots, (\mathbf{d}_{N-1}, M)\}$. Then, similar to [35], a signer with public key $\mathbf{d} \in R$ runs the accumulation algorithm on R' to obtain a value $\mathbf{u}$. It then produces an NIZK proof that:

- There exists a public key $\mathbf{d}$ such that $(\mathbf{d}, M)$ is accumulated in $\mathbf{u}$, and;
- There exists a secret key $\mathbf{x}$ corresponding to the public key $\mathbf{d}$.

Consequently, the first item (existence of a Merkle-path) certifies the binding of the message M and the signer's membership in the ring, while the second item certifies that the signer is valid. This ensures that the final ring signature size remains the same asymptotic order as that of the underlying set membership proof, yielding an efficiency gain of at least a factor $\Omega(\lambda)$ over the prior approach of appending a one-time signature.

In more detail, a user's signing key is a vector $\mathbf{x}$ of $\mathcal{R}_{\text{bin}}^{m_F}$ consisting of polynomials with binary coefficients. The public key is the base- B decomposition of $\mathbf{F} \cdot \mathbf{x} \bmod q_H$ where $\mathbf{F} \in \mathcal{R}_{q_H}^{n \times m_F}$ defines a lattice-based one-way function. To sign a message $M \in \mathcal{R}_{\text{bin}}^{l_M}$ in a ring of public keys $R = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\}$, the

signer with public key $\mathbf{d} \in R$ creates a set $R' = \{(\mathbf{d}_0, M), \dots, (\mathbf{d}_{N-1}, M)\}$, runs the accumulation algorithm $\text{TAcc}(R') = \mathbf{u}$, then generates an NIZK proof that $(\mathbf{d}, M)$ is accumulated in $\mathbf{u}$, and that there is a secret $\mathbf{x} \in \mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}}$ satisfying $\mathbf{G}_B \mathbf{d} = \mathbf{F} \mathbf{x} \bmod q_H$. Note that, in the set membership proof we reveal a part of the secret element of R' (the message M), and introduce an additional linear relation involving the signer's secret and public keys. Therefore, the trapdoor Σ -protocol for set membership, for which we apply the NIZK compiler of [36], is slightly modified. The details are given in Appendix C.2.

For security, we note that our construction provides *statistical anonymity* as the extractable commitment scheme, formally defined in Appendix A.4, is statistically hiding in normal mode. Regarding unforgeability, we rely on the extraction mode of the same commitment scheme to reduce an adversary to one that solves some instances of MSIS. Although the zero-knowledge guarantee in extraction mode is only computational, the public parameters generated in normal and extraction modes are computationally indistinguishable. This allows the scheme to achieve statistical anonymity in normal mode while using extraction mode to establish computational unforgeability. The reduction, in comparison with [35], is more complicated due to the relaxed soundness of the proof system.

5.2 Scheme Description and Analyses

We construct a ring signature scheme with signature size logarithmic in the ring size, based on MSIS and MLWE assumptions. Let λ denote the security parameter. Our construction works over the power-of-2 cyclotomic ring $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ where d is a power of 2, and with the following parameters:

- Moduli $q_C, q_H, q \in \text{poly}(\lambda)$ for KTX commitment, accumulator hash, and extractable commitment respectively;
- Dimension $l_M \in \text{poly}(\lambda)$ specifying the message space $\mathcal{R}_{\text{bin}}^{l_M}$;
- An integer $B > 1$, a norm bound $\beta = B - 1$, a parameter $k_H = \lceil \log_B q_H \rceil$;
- Dimensions $n, n_{\mathbf{F}}, m_{\mathbf{F}}, n_C, n_R \in \text{poly}(\lambda)$. We require that $m_{\mathbf{F}} > n_{\mathbf{F}} \log q_H$, and that $m_{\mathbf{H}} = 2nk_H = 2(n_{\mathbf{F}}k_H + l_M)$;
- A maximal ring size $N = 2^\ell \in \text{poly}(\lambda)$ for some $\ell \in \mathbb{N}$;
- Norm bounds $B_R, B_{\mathbf{r}} \in \text{poly}(\lambda)$ such that $\sqrt{(1 + q_C^{n_C}/(2B_R + 1)^{n_R})^d} - 1 \in \text{negl}(\lambda)$, which is satisfied by choosing $n_R = \mathcal{O}(n_C \log q_C)$;
- A Gaussian width $\alpha > 0$ defining the distribution for rejection sampling;
- A repetition parameter $\kappa = \Omega(\lambda/\log \lambda)$.

The algorithms of our ring signature scheme are described as follows

- $\text{RSetup}(1^\lambda)$ takes as input the security parameter λ and does the following:
 1. Choose a power-of-2 d defining number field $\mathcal{R}$; choose two prime moduli $q_C, q_H \in \text{poly}(\lambda)$ and choose:
 - Dimensions $n, n_{\mathbf{F}}, m_{\mathbf{F}}$ such that $dn, dn_{\mathbf{F}} \in \Theta(\lambda)$ and $m_{\mathbf{F}} \gtrsim n_{\mathbf{F}} \log q_H$;
 - An integer base $B > 1$, defining a norm bound $\beta = B - 1$ and a decomposition parameter $k_H = \lceil \log_B q_H \rceil$;

- Dimensions $l_M, m_H \in \text{poly}(\lambda)$ such that $m_H = 2nk_H = 2(n_F k_H + l_M)$;
 - A norm bound B_R , dimensions $n_C, n_R \in \text{poly}(\lambda)$ such that $dn_C \in \Theta(\lambda)$.
- The above parameters additionally define $l_m = 3nk_H + 2n_R$, $l_m^{(0)} = 2nk_H + n_R$ and $l_m^{(\ell-1)} = n_F k_H + 2nk_H + 2n_R + m_F$.
2. Let $l_{\max} = \max\{l_m, l_m^{(0)}, l_m^{(\ell-1)}\}$ be the maximum message length in the extractable commitment scheme of Appendix A.4. Using public parameters

$$\text{par} = \{d, q_C, q_H, n, n_F, m_F, l_M, m_H, n_C, n_R, l_m, l_m^{(0)}, l_m^{(\ell-1)}, B, B_R\},$$

and language-dependent information $\text{info}_L = N = 2^\ell$, execute algorithm Gen_L of the trapdoor Σ -protocol of Appendix C.2, with the exception that in Step 2, the public parameters of the commitment scheme of Appendix A.4 is generated with Setup algorithm (i.e., in *normal mode*). Let the output be

$$\text{crs}_L = (q, l_r, B_m, B_r, \mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{H}, \mathbf{C}, \mathbf{D}, B_{\text{Chall}}, d_{\text{Chall}}, \alpha)$$

as in (35), where

$$B_m = \max \left\{ \sqrt{d[3\beta^2 nk_H + 2B_R^2 n_R]}, \sqrt{d[\beta^2(2n + n_F)k_H + 2B_R^2 n_R + m_F]} \right\}.$$

We require that

$$\begin{aligned} \alpha\sqrt{dl_{\max}} &< 1/2 \min\{q, q_H, q_C\}, \\ \alpha\sqrt{dl_r} &< q/2, \quad \sqrt{(1 + q^{n+l_{\max}}/(2B_r + 1)^{l_r})^d - 1} \in \text{negl}(\lambda), \end{aligned} \quad (25)$$

and that there exist a prime modulus $p < q$ and an ℓ_∞ -norm bound B_τ such that $p \cdot 2dl_r B_\tau \alpha\sqrt{dl_r l_{\max}} + 2\alpha\sqrt{dl_{\max}} < q/2$ and $2\alpha\sqrt{dl_{\max}} < p/2$.

3. Generate a common reference string $\text{crs} = (\text{par}, \text{crs}_L, k, \kappa, pk)$ of the NIZK system of Appendix B by doing the following:
 - Generate a key $k \leftarrow \text{Gen}(1^\lambda)$ for the somewhere CI hash family $\mathcal{H}$ (Definition 13).
 - Generate public parameters $\Gamma \leftarrow \text{LPKE.Param}(1^\lambda)$ and lossy encryption keys $(pk, sk) \leftarrow \text{LPKE.LKeygen}(\Gamma)$ for encrypting the first-move messages of the trapdoor Σ -protocol of Appendix C.2.
 - Choose a repetition parameter $\kappa = \Theta(\lambda/\log \lambda)$.

Finally, RSetup outputs system parameters $\text{pp} = (\text{par}, \text{crs}_L, k, \kappa, pk)$.

- RKeyGen(1^λ) takes as input security parameter λ . It samples $\mathbf{x} \xleftarrow{\$} \mathcal{R}_{\text{bin}}^{m_F}$, computes $\mathbf{d} = \text{dec}_B(\mathbf{F} \cdot \mathbf{x} \bmod q_H) \in \mathcal{C}_\beta^{n_F k_H}$ then returns $(pk, sk) = (\mathbf{d}, \mathbf{x})$.
- RSign(sk, M, R) takes as input a secret key $sk = \mathbf{x} \in \mathcal{R}_{\text{bin}}^{m_F}$, message $M \in \mathcal{R}_{\text{bin}}^{l_M}$, and a ring $R = (\mathbf{d}_0, \mathbf{d}_1, \dots, \mathbf{d}_{N-1}) \subset \mathcal{C}_\beta^{n_F k_H}$ such that $\mathbf{d} = \text{dec}_B(\mathbf{F} \cdot \mathbf{x} \bmod q_H) \in R$. It then proceeds as follows:
 1. Let $R' = \{\mathbf{d}'_i = (\mathbf{d}_i, M) : i \in [N]\} \subset \mathcal{C}_\beta^{n_{k_H}}$, then using the accumulator of Section 4.1 with $\mathbf{H}$ defining the 2-input hash and B defining the decomposition base, compute the accumulated value $\text{TAcc}(R') = \mathbf{u} \in \mathcal{C}_\beta^{n_{k_H}}$.

2. Let $\mathbf{d}' = (\mathbf{d}, M)$, execute $\text{TWitness}(R', \mathbf{d}')$ to obtain

$$((j_0, \dots, j_{\ell-1}), (\mathbf{w}_0, \dots, \mathbf{w}_{\ell-1})) \in \{0, 1\}^\ell \times (\mathcal{C}_\beta^{n_{k_H}})^\ell.$$

Let $\mathbf{v}_{\ell-1} = \mathbf{d}', \dots, \mathbf{v}_0, \mathbf{v}_{-1} = \mathbf{u}$ be the nodes on the path from $\mathbf{d}'$ to $\mathbf{u}$;

3. For $i \in [\ell]$, sample $\mathbf{r}_{\mathbf{v}_i} \xleftarrow{\$} \mathcal{C}_{B_R}^{n_R}$, compute the KTX commitments to $\mathbf{v}_i$ as

$$\mathbf{c}_{\mathbf{v}_i} = \mathbf{C} \cdot \mathbf{v}_i + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_i} \bmod q_C.$$

4. Let $\mathbf{m}_{\ell-1} = (\mathbf{d}, \mathbf{r}_{\mathbf{v}_{\ell-1}}, \mathbf{v}_{\ell-2}, \mathbf{r}_{\mathbf{v}_{\ell-2}}, \mathbf{w}_{\ell-1}, \mathbf{x}) \in \mathcal{R}^{l_{\mathbf{m}}^{(\ell-1)}}$; sample $\mathbf{r}_{\ell-1} \xleftarrow{\$} \mathcal{C}_{B_r}^{l_r}$ and compute

$$\mathbf{c}_{\ell-1} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}[:, l_{\mathbf{m}}^{(\ell-1)}] \end{bmatrix} \mathbf{r}_{\ell-1} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_{\ell-1} \end{bmatrix} \bmod q.$$

Note that we have $\|\mathbf{m}_{\ell-1}\| \leq \sqrt{d[2\beta^2(2n + n_{\mathbf{F}})k_H + 2B_R^2 n_R + m_{\mathbf{F}}]} \leq B_{\mathbf{m}}$.

5. Let $\mathbf{m}_0 = (\mathbf{v}_0, \mathbf{r}_{\mathbf{v}_0}, \mathbf{w}_0) \in \mathcal{R}^{l_{\mathbf{m}}^{(0)}}$; sample $\mathbf{r}_0 \xleftarrow{\$} \mathcal{C}_{B_r}^{l_r}$ and compute

$$\mathbf{c}_0 = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}[:, l_{\mathbf{m}}^{(0)}] \end{bmatrix} \mathbf{r}_0 + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_0 \end{bmatrix} \bmod q.$$

Note that we have $\|\mathbf{m}_0\| \leq \sqrt{d[2\beta^2 n k_H + B_R^2 n_R]} \leq B_{\mathbf{m}}$.

6. For $\ell - 1 > i > 0$, let $\mathbf{m}_i = (\mathbf{v}_i, \mathbf{r}_{\mathbf{v}_i}, \mathbf{v}_{i-1}, \mathbf{r}_{\mathbf{v}_{i-1}}, \mathbf{w}_{i-1}) \in \mathcal{R}^{l_{\mathbf{m}}}$, sample $\mathbf{r}_i \xleftarrow{\$} \mathcal{C}_{B_r}^{l_r}$ then compute

$$\mathbf{c}_i = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}[:, l_{\mathbf{m}}] \end{bmatrix} \mathbf{r}_i + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_i \end{bmatrix} \bmod q;$$

Note that we have $\|\mathbf{m}_i\| \leq \sqrt{d[3\beta^2 n k_H + 2B_R^2 n_R]} \leq B_{\mathbf{m}}$.

7. Generate an NIZK proof that

$$(M, \mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]})$$

is a statement of the language $\mathcal{L}_{\text{ring}} = (\mathcal{L}_{\text{ring, zk}}, \mathcal{L}_{\text{ring, snd}})$ defined by (36) and (37), using witness $((\mathbf{m}_i, \mathbf{r}_i))_{i \in [\ell]}$. Let the final proof be π .

Output the final signature

$$\Sigma = (\mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}, \pi). \quad (26)$$

- $\text{RVerify}(M, R, \Sigma)$ takes as input a message $M \in \mathcal{R}_{\text{bin}}^{l_M}$, a ring $R \subset \mathcal{C}_\beta^{n_{\mathbf{F}} k_H}$ such that $R = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\} \subset \mathcal{C}_\beta^{n_{\mathbf{F}} k_H}$, and a signature Σ . The algorithm parses Σ as in (26) and proceeds as follows:

1. Let $R' = \{\mathbf{d}' = (\mathbf{d}_i, M) : \mathbf{d} \in R\}$, then execute algorithm $\text{TAcc}(R')$ to obtain accumulated value $\mathbf{u}'$ and check whether $\mathbf{u}' = \mathbf{u}$. Return 0 if it is not the case.

2. Verify that π is a valid NIZK proof that $(M, \mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]})$ is a statement of $\mathcal{L}_{\text{ring}}$. Return 1 if the proof is valid and 0 otherwise.

CORRECTNESS AND SECURITY ANALYSIS. We formalize the correctness of our construction via the following theorem.

Theorem 2. *The ring signature scheme of Section 5.2 is correct provided that the NIZK for $\mathcal{L}_{\text{ring}} = (\mathcal{L}_{\text{ring}, \text{zk}}, \mathcal{L}_{\text{ring}, \text{snd}})$, defined by (36) and (37), is correct.*

Proof. If an honestly-generated signature $\Sigma = (\mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}, \pi)$ as in (26) that verifies with message M and a ring R , then it must be the case that $\mathbf{u} = \text{TAcc}(\{(\mathbf{d}, M) : \mathbf{d} \in R\})$, and that the signer possesses a witness $((\mathbf{m}_i, \mathbf{r}_i))_{i \in [\ell]}$ that $(M, \mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]})$ is a statement of $\mathcal{L}_{\text{ring}, \text{zk}}$. Combining with the correctness of the NIZK system for $\mathcal{L}_{\text{ring}}$, Theorem 2 follows. $\square$

For security of our construction, we have the following:

Theorem 3. *Assuming that the parameters satisfy*

$$\begin{aligned} \sqrt{(1 + q^{n+l_{\max}}/(2B_{\mathbf{r}} + 1)^{l_{\mathbf{r}}})^d - 1} &\in \text{negl}(\lambda), \\ \sqrt{(1 + q_C^{n_C}/(2B_{\mathbf{R}} + 1)^{n_{\mathbf{R}}})^d - 1} &\in \text{negl}(\lambda). \end{aligned}$$

and the NIZK system is statistically zero-knowledge. Then the ring signature scheme of Section 5.2 provides statistical anonymity under full key exposure.

Furthermore, assume that: (i) $q_{\mathbf{H}} = 5 \bmod 8$, $B_{\text{Chall}} < \sqrt{q_{\mathbf{H}}}$; (ii) there exists a prime modulus $p < q$ and an infinity norm bound B_{τ} such that $p \cdot 2dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\max}} + 2\alpha\sqrt{dl_{\max}} < q/2$ and $2\alpha\sqrt{dl_{\max}} < p/2$; (iii) the NIZK system is sound; (iv) MSIS and MLWE assumptions hold. Then the ring signature scheme of Section 5.2 is unforgeable against insider corruption.

Due to space restriction, we defer the proof of Theorem 3 to Appendix C.4.

References

1. M. Backes, N. Döttling, L. Hanzlik, K. Klucznik, and J. Schneider. Ring signatures: Logarithmic-size, no setup – from standard assumptions. In *EUROCRYPT 2019*, volume 11478 of *LNCS*, pages 281–311. Springer, 2019.
2. W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, 1993.
3. N. Baric and B. Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes Without Trees. In *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 480–494. Springer, 1997.
4. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, 2009.
5. J. Benaloh and M. de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Sinatures. In *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 274–285. Springer, 1993.

6. D. Benarroch, M. Campanelli, D. Fiore, K. Gurkan, and D. Kolonelos. Zero-knowledge proofs for set membership: efficient, succinct, modular. *Des. Codes Cryptogr.*, 91(11):3457–3525, 2023.
7. A. Bender, J. Katz, and R. Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. *J. Cryptology*, 22(1):114–138, 2009.
8. W. Beullens, S. Katsumata, and F. Pintore. Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In *ASIACRYPT 2020*, volume 12492 of *LNCS*, pages 464–492. Springer, 2020.
9. K. Boudgoust, C. Jeudy, A. Roux-Langlois, and W. Wen. On the hardness of module learning with errors with short distributions. *J. Cryptol.*, 36(1):1, 2023.
10. Z. Brakerski and Y. T. Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. IACR Cryptology ePrint Archive, 2010.
11. J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, 2001.
12. J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer, 2002.
13. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. N. Rothblum, R. D. Rothblum, and D. Wichs. Fiat-shamir: from practice to theory. In *STOC 2019*, pages 1082–1090. ACM, 2019.
14. P. Chaidos and J. Groth. Making sigma-protocols non-interactive without random oracles. In *PKC 2015*, volume 9020 of *LNCS*, pages 650–670. Springer, 2015.
15. N. Chandran, J. Groth, and A. Sahai. Ring signatures of sub-linear size without random oracles. In *ICALP 2007*, LNCS, pages 423–434. Springer, 2007.
16. M. Chase, A. Healy, A. Lysyanskaya, T. Malkin, and L. Reyzin. Mercurial commitments with applications to zero-knowledge sets. In *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 422–439. Springer, 2005.
17. R. Chatterjee, S. Garg, M. Hajiabadi, D. Khurana, X. Liang, G. Malavolta, O. Pandey, and S. Shiehian. Compact ring signatures from learning with errors. In *CRYPTO 2021*, volume 12825 of *LNCS*, pages 355–385. Springer, 2021.
18. D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.
19. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *CRYPTO 1994*, volume 839 of *LNCS*, pages 174–187. Springer, 1994.
20. D. Derler, S. Ramacher, and D. Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In *PQCrypto 2018*, LNCS, pages 419–440. Springer, 2018.
21. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.
22. M. F. Esgin, R. Steinfeld, D. Liu, and S. Ruj. Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrf. In *CRYPTO 2023*, LNCS, pages 484–517. Springer, 2023.
23. M. F. Esgin, R. Steinfeld, A. Sakzad, J. K. Liu, and D. Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *ACNS 2019*, volume 11464 of *LNCS*, pages 67–88. Springer, 2019.

24. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194. Springer, 1986.
25. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. In *STOC 2008*, pages 197–206. ACM, 2008.
26. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO 2013*, volume 8042 of *LNCS*, pages 75–92. Springer, 2013.
27. A. Goel, M. Green, M. Hall-Andersen, and G. Kaptchuk. Efficient set membership proofs using mpc-in-the-head. *Proc. Priv. Enhancing Technol.*, 2022(2):304–324, 2022.
28. A. González. Shorter ring signatures from standard assumptions. In *PKC 2019*, *LNCS*, pages 99–126. Springer, 2019.
29. J. Groth and M. Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT 2015*, volume 9057 of *LNCS*, pages 253–280. Springer, 2015.
30. S. Katsumata, G. Niot, I. Tucker, and T. Wiggers. Comprehensive deniability analysis of signal handshake protocols: X3dh, PQXDH to fully post-quantum with deniable ring signatures. In *USENIX 2025*, pages 6797–6816. USENIX Association, 2025.
31. J. Katz, V. Kolesnikov, and X. Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *ACM CCS 2018*, pages 525–537. ACM, 2018.
32. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems. In *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 372–389. Springer, 2008.
33. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.*, 75(3):565–599, 2015.
34. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT 2016*, volume 9666 of *LNCS*, pages 1–31. Springer, 2016.
35. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. *J. Cryptol.*, 36(3):23, 2023.
36. B. Libert, K. Nguyen, A. Passelègue, and R. Titui. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In *ASIACRYPT 2020*, volume 12491 of *LNCS*, pages 128–158. Springer, 2020.
37. B. Libert, K. Nguyen, T. Peters, and M. Yung. One-shot fiat-shamir-based NIZK arguments of composite residuosity and logarithmic-size ring signatures in the standard model. In *EUROCRYPT 2022*, *LNCS*, pages 488–519. Springer, 2022.
38. B. Libert, K. Nguyen, B. H. M. Tan, and H. Wang. Zero-knowledge elementary databases with more expressive queries. In *PKC 2019*, volume 11442 of *LNCS*, pages 255–285. Springer, 2019.
39. H. Lipmaa and R. Parisella. Set (non-)membership nizks from determinantal accumulators. In *LATINCRYPT 2023*, volume 14168 of *LNCS*, pages 352–374. Springer, 2023.
40. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *PKC 2008*, volume 4939 of *LNCS*, pages 162–179. Springer, 2008.
41. V. Lyubashevsky. Lattice Signatures without Trapdoors. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755. Springer, 2012.

42. V. Lyubashevsky and N. K. Nguyen. BLOOM: Bimodal lattice one-out-of-many proofs and applications. In *ASIACRYPT 2022*, volume 13794 of *LNCS*, pages 95–125. Springer, 2022.
43. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In *CRYPTO 2022*, volume 13508 of *LNCS*, pages 71–101. Springer, 2022.
44. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Smile: Set membership from ideal lattices with applications to ring signatures and confidential transactions. In *CRYPTO 2021*, volume 12825 of *LNCS*, pages 629–658. Springer, 2021.
45. V. Lyubashevsky and G. Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *EUROCRYPT 2018*, volume 10820 of *LNCS*, pages 204–224. Springer, 2018.
46. S. Micali, M. O. Rabin, and J. Kilian. Zero-knowledge sets. In *FOCS 2003*, pages 80–91. IEEE, 2003.
47. D. Micciancio and C. Peikert. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. In *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 700–718. Springer, 2012.
48. K. Nguyen, H. Tang, H. Wang, and N. Zeng. New code-based privacy-preserving cryptographic constructions. In *ASIACRYPT 2019*, volume 11922 of *LNCS*, pages 25–55. Springer, 2019.
49. L. Nguyen. Accumulators from bilinear pairings and applications. In *CT-RSA 2005*, volume 3376 of *LNCS*, pages 275–292. Springer, 2005.
50. C. Peikert and S. Shiehian. Noninteractive zero knowledge for NP from (plain) LWE. In *CRYPTO 2019*, volume 11694 of *LNCS*, pages 89–114. Springer, 2019.
51. C. Peikert, V. Vaikuntanathan, and B. Waters. A Framework for Efficient and Composable Oblivious Transfer. In *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, 2008.
52. O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *Journal of the ACM*, 56(6), 2009.
53. R. L. Rivest, A. Shamir, and Y. Tauman. How to Leak a Secret. In *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 552–565. Springer, 2001.
54. J. Stern. A New Paradigm for Public Key Identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.
55. N. Tran, K. Nguyen, D. Liu, J. Pieprzyk, and W. Susilo. Lattice-based group signatures in the standard model, revisited. In *ASIACRYPT 2025*, volume 16248 of *LNCS*, pages 70–102. Springer, 2025.
56. B. Waters. A new approach for non-interactive zero-knowledge from learning with errors. In *STOC 2024*, pages 399–410. ACM, 2024.
57. B. Waters, H. Wee, and D. Wu. New techniques for preimage sampling: Improved nizks and more from LWE. In *EUROCRYPT 2025*, LNCS. Springer, 2025.
58. R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO 2019*, volume 11692 of *LNCS*, pages 147–175. Springer, 2019.

Supplementary Materials

A Cryptographic Primitives

A.1 Lattice-Based Cryptography

Hardness Assumptions We recall the Module-SIS (MSIS) and the Module-LWE (MLWE) problems [33,21], which are defined over the ring $\mathcal{R}$.

Definition 8 (MSIS). Let $n, m, \beta, q > 0$ be functions in security parameter λ such that $\beta < q$. For $i \in \{2, \infty\}$, the Module-SIS problem w.r.t. to ℓ_i -norm, denoted as $\text{MSIS}_{n,m,\beta,q}^{(i)}$, asks to find $\mathbf{z} \in \mathcal{R}^m$ such that $\mathbf{A}\mathbf{z} = \mathbf{0} \bmod q$ and $0 < \|\mathbf{z}\|_i \leq \beta$, where $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$. The $\text{MSIS}_{n,m,\beta,q}^{(i)}$ assumption states that for $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}$ and all PPT adversaries $\mathcal{A}$, $\Pr[\mathbf{A}\mathbf{z} = \mathbf{0} \bmod q, 0 < \|\mathbf{z}\|_i \leq \beta : \mathbf{z} \leftarrow \mathcal{A}(\mathbf{A})]$ is negligible in λ .

Definition 9 (MLWE). Let n, m, q be functions in security parameter λ and χ is a distribution over $\mathcal{R}$. Let $\mathbf{s} \xleftarrow{\$} \mathcal{R}_q^n$, the Module-LWE distribution $\mathcal{D}_{\mathbf{s},\chi}$ over $\mathcal{R}_q^{m \times n} \times \mathcal{R}_q^m$ is defined as $\{(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) : \mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{m \times n}, \mathbf{e} \xleftarrow{\$} \chi^n\}$.

The Module-LWE problem, denoted as $\text{MLWE}_{n,m,\chi,q}$, asks an adversary $\mathcal{A}$ to distinguish between $\mathcal{D}_{\mathbf{s},\chi}$ and the uniform distribution over $\mathcal{R}_q^{m \times n} \times \mathcal{R}_q^m$. The $\text{MLWE}_{n,m,\chi,q}$ assumption states that for all PPT adversaries $\mathcal{A}$, the following is a negligible function of λ :

$$|\Pr[1 \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{b}) : (\mathbf{A}, \mathbf{b}) \leftarrow \mathcal{D}_{\mathbf{s},\chi}] - \Pr[1 \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{b}) : (\mathbf{A}, \mathbf{b}) \xleftarrow{\$} \mathcal{R}_q^{m \times n} \times \mathcal{R}_q^m]|.$$

Leftover Hash Lemma (LHL) We use the following version of the LHL.

Lemma 2 (Adapted from [9, Lemma 2.8]). Let d be a power of 2, n, m be positive integers and q be a prime modulus. Then the statistical distance between the distribution $\{(\mathbf{A}, \mathbf{A}\mathbf{x} \bmod q) : \mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times m}, \mathbf{x} \xleftarrow{\$} \mathcal{C}_\beta^m\}$ and the uniform distribution over $\mathcal{R}_q^{n \times m} \times \mathcal{R}_q^n$ is at most $1/2 \cdot \sqrt{(1 + q^n/(2\beta + 1)^m)^d - 1}$.

Discrete Gaussian and Rejection Sampling The discrete Gaussian on a lattice $\mathcal{R}^l$ centered at $\mathbf{c} \in \mathcal{R}^l$ with width $\sigma > 0$ is given by the mass function

$$\mathcal{D}_{\mathbf{c},\sigma}^l(\mathbf{z}) = \frac{\exp(-\pi\|\mathbf{z} - \mathbf{c}\|^2/2\sigma^2)}{\sum_{\mathbf{v} \in \mathcal{R}^l} \exp(-\pi\|\mathbf{v} - \mathbf{c}\|^2/2\sigma^2)}.$$

When $\mathbf{c} = \mathbf{0}^l$, we simply write $\mathcal{D}_\sigma^l$.

Lemma 3 ([2]). For $\sigma > 0$, we have $\Pr[\|\mathbf{v}\| > \sigma\sqrt{ld}] \leq 2^{-2ld}$ where $\mathbf{v} \leftarrow \mathcal{D}_\sigma^l$.

We recall the rejection sampling technique of [41].

Lemma 4 ([41,43]). *Let $V \subseteq \mathcal{R}^l$ of which all elements have ℓ_2 -norms at most T , let σ be a real number such that $\sigma = \omega(T\sqrt{\log m})$, and $h : V \rightarrow [0, 1]$ be a probability distribution. Then, there exists a real number M such that the distribution of the following algorithm $\mathcal{A}$:*

1. $\mathbf{v} \leftarrow h; \mathbf{z} \leftarrow \mathcal{D}_{\sigma, \mathbf{v}}^l$
2. output $(\mathbf{z}, \mathbf{v})$ with probability $\min\left(\frac{\mathcal{D}_{\sigma}^l(\mathbf{z})}{M \cdot \mathcal{D}_{\sigma, \mathbf{v}}^l(\mathbf{z})}, 1\right)$

is within statistical distance $\frac{2^{-\omega(\log m)}}{M}$ from the output distribution produced by the following algorithm $\mathcal{F}$:

1. $\mathbf{v} \leftarrow h; \mathbf{z} \leftarrow \mathcal{D}_{\sigma}^l$
2. output $(\mathbf{z}, \mathbf{v})$ with probability $1/M$.

Moreover, the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-\omega(\log m)}}{M}$. More concretely, if $\sigma = \alpha T$ for any positive α , then $M = e^{14/\alpha + 1/(2\alpha^2)}$, the output of $\mathcal{A}$ is within statistical distance $2^{-128}/M$ of the output of $\mathcal{F}$, and the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-128}}{M}$.

Short Invertible Elements of $\mathcal{R}_q$ We recall the result of [45] regarding the invertibility of short ring elements modulo a prime in the ring $\mathcal{R}$.

Lemma 5 ([45]). *Let $d \geq k > 1$ be powers of 2, $q = 2k + 1 \bmod 4k$ be a prime, $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ and $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d + 1)$, then $s \in \mathcal{R}$ is invertible in $\mathcal{R}_q$ provided that $0 < \|s\|_{\infty} < 1/\sqrt{k} \cdot q^{1/k}$ or $0 < \|s\| < q^{1/k}$.*

A.2 Lossy Public-key Encryption

Lossy encryption was first formulated by Peikert *et al.* [51] under the notion of dual-mode encryption, and later generalized by Böhl *et al.* [4]. In essence, a lossy public-key encryption scheme has two modes of key generation: a normal mode where encryption is injective, and a “lossy” (or “messy”) mode where encryption under lossy keys loses information about plaintexts.

Definition 10 (Adapted from [51,4]). *A lossy encryption scheme is a tuple of PPT algorithms $\text{LPKE} = (\text{Param}, \text{Keygen}, \text{LKeygen}, \text{Enc}, \text{Dec}, \text{Opener})$ that*

- $\text{Param}(1^{\lambda}, 1^L)$, on input security parameter λ and a message length L , outputs public parameters par_{LE} defining plaintext space M , ciphertext space C , randomness space R , a distribution D_R for sampling encryption randomness.
- $\text{Keygen}(1^{\lambda}, \text{par}_{\text{LE}})$, on input security parameter λ and public parameters par_{LE} , outputs injective keys $(pk_{\text{inj}}, sk_{\text{inj}})$.
- $\text{LKeygen}(1^{\lambda}, \text{par}_{\text{LE}})$, on input security parameter λ and public parameters par_{LE} , outputs lossy keys $(pk_{\text{lossy}}, sk_{\text{lossy}})$.

- **Opener**($pk_{\text{lossy}}, sk_{\text{lossy}}, c, m$), on input lossy keys $(pk_{\text{lossy}}, sk_{\text{lossy}})$, a ciphertext c , and a message m , outputs randomness r .

In addition, we require the following:

1. **Decryption correctness w.r.t. injective keys.** There is a negligible function $\epsilon(\lambda)$ such that for $m \in \mathcal{M}$,

$$\Pr[(pk_{\text{inj}}, sk_{\text{inj}}) \leftarrow \text{Keygen}(1^\lambda, \text{par}_{\text{LE}}); \\ r \leftarrow D_{\mathbb{Z}} : \text{Dec}(sk_{\text{inj}}, \text{Enc}(pk_{\text{inj}}, m, r)) = m] = 1 - \epsilon(\lambda).$$

2. **Indistinguishability of lossy keys.** The two distributions

$$\{pk : (pk, sk) \leftarrow \text{Keygen}(1^\lambda, \text{par}_{\text{LE}})\}, \\ \{pk : (pk, sk) \leftarrow \text{LKeygen}(1^\lambda, \text{par}_{\text{LE}})\}$$

are computationally indistinguishable.

3. **Lossiness of lossy keys.** For all lossy key pk_{lossy} and for all $m_0, m_1 \in \mathcal{M}$, the statistical distance between the distributions $\{\text{Enc}(pk_{\text{lossy}}, m_0, r) : r \leftarrow D_{\mathbb{Z}}\}$ and $\{\text{Enc}(pk_{\text{lossy}}, m_1, r) : r \leftarrow D_{\mathbb{Z}}\}$ is negligible in λ .
4. **Efficient opening under lossy keys.** Let $(pk_{\text{lossy}}, sk_{\text{lossy}})$ be a lossy key pair generated by **LKeygen**. For any $m_0, m_1 \in \mathcal{M}$ and $r_0 \leftarrow D_{\mathbb{Z}}$, set $c = \text{Enc}(pk_{\text{lossy}}, m_0, r_0)$. Let r_1 be the randomness output by **Opener** for c and m_1 . Then, except with negligible probability,

$$\text{Enc}(pk_{\text{lossy}}, m_1, r_1) = c.$$

There exists a simple construction of lattice-based lossy encryption by modifying the primal-Regev public key encryption [52]. Namely, encryption randomness is sampled from a discrete Gaussian, lossy keys are generated as a (statistically) uniform matrix with a trapdoor for sampling from arbitrary shifted discrete Gaussian (e.g. GPV trapdoor [25] or MP trapdoor [47]).

A.3 Commitment Schemes

A commitment scheme **COM** is a tuple of algorithms (**Setup**, **Com**) that:

- **Setup**(1^λ): On input the security parameter λ , it returns system parameters pp_{C} which defines the message space $\mathcal{M}$ and the randomness space $\mathcal{R}$, along with a distribution of randomnesses $\mathcal{D}_{\mathcal{R}}$.
- **Com**($\text{pp}_{\text{C}}, m; r$): On input pp_{C} , a message $m \in \mathcal{M}$ and randomness $r \in \mathcal{R}$, this algorithm computes a commitment C .

Hiding. A commitment scheme **COM** is computationally/statistically hiding if for $\text{pp}_{\text{C}} \leftarrow \text{Setup}(1^\lambda)$ and any distinct messages $m_0, m_1 \in \mathcal{M}$, the distributions $\{C : r \leftarrow \mathcal{D}_{\mathcal{R}}, C = \text{Com}(\text{pp}_{\text{C}}, m_0; r)\}$ and $\{C : r \leftarrow \mathcal{D}_{\mathcal{R}}, C = \text{Com}(\text{pp}_{\text{C}}, m_1; r)\}$ are computationally/statistically indistinguishable.

Binding. A commitment scheme COM is binding if, for any PPT adversary $\mathcal{A}$, the following probability

$$\Pr \left[\text{Com}(\text{pp}_C, m_0; r_0) = \text{Com}(\text{pp}_C, m_1; r_1) : \begin{array}{l} \text{pp}_C \leftarrow \text{Setup}(1^\lambda), \\ (\text{com}, m_0, r_0, m_1, r_1) \leftarrow \mathcal{A}(\text{pp}_C) \end{array} \right] \wedge (m_0 \neq m_1)$$

is negligible in λ .

KTX COMMITMENT. We present a variant of the KTX commitment scheme [32] that works with module lattices. The commitment scheme works with a prime modulus $q = \text{poly}(\lambda)$; parameters $n_C, n_R, l \in \text{poly}(\lambda)$ specifying dimensions of commitment, randomness and messages, an ℓ_∞ -norm bound B on the committed message and an ℓ_∞ -norm bound B_R on the commitment randomness. The algorithm of the commitment scheme are as follows:

- **Setup**(1^λ): on input security parameter λ , choose a power-of-2 d defining $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, choose a prime modulus $q = \text{poly}(\lambda)$, choose dimensions $n_C, n_R, l \in \text{poly}(\lambda)$, and choose norm bounds $B, B_R = \text{poly}(\lambda)$. We require that $\sqrt{(1 + q^{n_C}/(2B_R + 1)^{n_R})^d} - 1 \in \text{negl}(\lambda)$.
The algorithm then samples $(\mathbf{C}, \mathbf{D}) \xleftarrow{\$} \mathcal{R}_q^{n_C \times l} \times \mathcal{R}_q^{n_C \times n_R}$ and output public parameters $\text{pp} = (q, n_C, n_R, l, B, B_R, \mathbf{C}, \mathbf{D})$. The public parameters define message space $\mathcal{C}_B^l$ and randomness space $\mathcal{C}_{B_R}^{n_R}$, equipped with uniform distribution.
- **Com**($\text{pp}, \mathbf{x}; \mathbf{r}$): to commit to a message $\mathbf{x} \in \mathcal{C}_B^l$ with a randomness $\mathbf{r} \xleftarrow{\$} \mathcal{C}_{B_R}^{n_R}$, compute the commitment $\mathbf{c} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{r} \bmod q$.

By Leftover Hash Lemma (Lemma 2), the KTX commitment is statistically hiding. In turn, the binding property holds under $\text{MSIS}_{n_C, n_R + l, 2 \max\{B, B_R\}}^{(\infty)}$ assumption.

A.4 A Lattice-Based Commitment with Extractability

We present a lattice-based commitment scheme with extractability. Simply put, in addition to the normal **Setup** algorithm, there exists an alternative setup algorithm **ExtSetup** outputting public parameters indistinguishable to that of **Setup**, along with an extraction trapdoor. Furthermore, there exists an extraction algorithm **Ext** taking the trapdoor as input, that can recover the message under any commitment. The algorithms underlying the extractable commitment scheme are given as below:

- **Setup**(1^λ): choose a power-of-2 d , prime moduli q, p such that $q > p$; choose dimensions $n, l_{\max}, l_r$; norm bounds $B_{\mathbf{m}}, B_{\mathbf{r}}$ and sample $(\mathbf{A}, \mathbf{B}) \xleftarrow{\$} \mathcal{R}_q^{n \times l_r} \times \mathcal{R}_q^{l_{\max} \times l_r}$. Set $\text{pp} = (d, q, n, l_{\max}, l_r, B_{\mathbf{m}}, B_{\mathbf{r}}, \mathbf{A}, \mathbf{B})$, which defines the message space $\mathcal{M} = \cup_{i=1}^{l_{\max}} \mathcal{C}_{B_{\mathbf{m}}}^i$, the randomness space $\mathcal{C}_{B_{\mathbf{r}}}^{l_r}$ with uniform distribution.

- $\text{Com}(\text{pp}, \mathbf{m}; \mathbf{r})$: for a message $\mathbf{m} \in \mathcal{C}_{B_{\mathbf{m}}}^{l_{\mathbf{m}}}$ such that $1 \leq l_{\mathbf{m}} \leq l_{\max}$, and randomness $\mathbf{r} \in \mathcal{C}_{B_{\mathbf{r}}}^{l_{\mathbf{r}}}$, compute the commitment $\mathbf{c} \in \mathcal{R}_q$ as

$$\mathbf{c} = \begin{bmatrix} \mathbf{A} \\ \mathbf{B}[:, l_{\mathbf{m}}] \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix} \bmod q \in \mathcal{R}_q^{n+l_{\mathbf{m}}}.$$

- $\text{ExtSetup}(1^\lambda)$: similar to Setup , except that the algorithm also chooses a prime modulus p and a norm bound B_τ , then samples $\mathbf{A} \xleftarrow{\$} \mathcal{R}_q^{n \times l_{\mathbf{r}}}$, $\mathbf{S} \xleftarrow{\$} \mathcal{R}_q^{l_{\max} \times n}$ and $\mathbf{E} \xleftarrow{\$} \mathcal{C}_{B_\tau}^{l_{\max} \times l_{\mathbf{r}}}$. It sets $\mathbf{B} = \mathbf{S}\mathbf{A} + p\mathbf{E} \bmod q$, outputs extraction trapdoor $\tau = (\mathbf{S}, p)$ and public parameters $\text{pp} = (d, q, n, l_{\max}, l_{\mathbf{r}}, B_{\mathbf{m}}, B_{\mathbf{r}}, \mathbf{A}, \mathbf{B})$.
- $\text{Ext}(\text{pp}, \mathbf{c}, \tau = (\mathbf{S}, p))$: the extraction algorithm takes as input pp , a commitment $\mathbf{c} \in \mathcal{R}_q^{n+l_{\mathbf{m}}}$ where $1 \leq l_{\mathbf{m}} \leq l_{\max}$, and an extraction trapdoor $\tau = \mathbf{S} \in \mathcal{R}_q^{l_{\max} \times n}$. It parses $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{R}_q^n \times \mathcal{R}_q^{l_{\mathbf{m}}}$, computes $\mathbf{m} = (\mathbf{c}_1 - \mathbf{S}[:, l_{\mathbf{m}}]\mathbf{c}_0 \bmod q) \bmod p$ and output $\mathbf{m} \in \mathcal{R}^{l_{\mathbf{m}}}$.

Proposition 3. *Suppose that*

$$p \cdot dl_{\mathbf{r}}B_\tau B_{\mathbf{r}} + B_{\mathbf{m}} < \frac{q}{2}, \quad B_{\mathbf{m}} < p/2; \quad \sqrt{(1 + q^{n+l_{\max}}/(2B_{\mathbf{r}} + 1)^{l_{\mathbf{r}}})^d - 1} \in \text{negl}(\lambda).$$

Then the above commitment scheme satisfies the following:

- The distributions $\{\text{pp} \leftarrow \text{Setup}(1^\lambda)\}$ and $\{\text{pp} \leftarrow \text{ExtSetup}(1^\lambda)\}$ are computationally indistinguishable under $\text{MLWE}_{l_{\max}, l_{\mathbf{r}}, \chi}$ assumption, where χ is uniform over $\mathcal{C}_{B_{\mathbf{r}}}$;
- For $\text{pp} \leftarrow \text{Setup}(1^\lambda)$, the scheme is statistically hiding;
- For $(\text{pp}, \tau) \leftarrow \text{ExtSetup}(1^\lambda)$, we have that $\text{Ext}(\text{pp}, \text{Com}(\text{pp}, \mathbf{m}; \mathbf{r}), \tau) = \mathbf{m}$ for any $\mathbf{m} \in \mathcal{C}_{B_{\mathbf{m}}}^{l_{\mathbf{m}}}$ and any $\mathbf{r} \in \mathcal{C}_{B_{\mathbf{r}}}^{l_{\mathbf{r}}}$, where $1 \leq l_{\mathbf{m}} \leq l_{\max}$.

Proof. The first two items are direct consequences of $\text{MLWE}_{l_{\max}, l_{\mathbf{r}}, \chi}$ assumption and the Leftover Hash Lemma (Lemma 2). For the third item, note that if $\mathbf{c} = \text{Com}(\text{pp}, \mathbf{m}; \mathbf{r}) \in \mathcal{R}_q^{n+l_{\mathbf{m}}}$, then

$$\begin{bmatrix} -\mathbf{S}[:, l_{\mathbf{m}}] & \mathbf{I}_{l_{\mathbf{m}}} \end{bmatrix} \cdot \mathbf{c} = p\mathbf{E}[:, l_{\mathbf{m}}] \cdot \mathbf{r} + \mathbf{m} \bmod q.$$

The ℓ_∞ -norm of the right hand side can be bounded by

$$\|p\mathbf{E}[:, l_{\mathbf{m}}] \cdot \mathbf{r} + \mathbf{m}\|_\infty \leq p \cdot \|\mathbf{E}[:, l_{\mathbf{m}}] \cdot \mathbf{r}\|_\infty + \|\mathbf{m}\|_\infty \leq pdl_{\mathbf{r}}B_\tau B_{\mathbf{r}} + B_{\mathbf{m}} < q/2,$$

thus we have that $\begin{bmatrix} -\mathbf{S}[:, l_{\mathbf{m}}] & \mathbf{I}_{l_{\mathbf{m}}} \end{bmatrix} \cdot \mathbf{c} \bmod q = p\mathbf{E}[:, l_{\mathbf{m}}] \cdot \mathbf{r} + \mathbf{m}$. As $\|\mathbf{m}\|_\infty < p/2$, reducing the right hand side modulo p gives $\mathbf{m}$. $\square$

We remark that extractability of a commitment also implies binding. Furthermore, for hiding property one can set $l_{\mathbf{r}} = \Theta((n + l_{\max}) \cdot \log q)$ so that Lemma 2 is satisfied.

A.5 Non-interactive Zero Knowledge Argument Systems

We recall the formalization of NIZK systems following [36].

Definition 11. Let $\mathcal{R} = (\mathcal{R}_{\text{zk}}, \mathcal{R}_{\text{snd}})$ be a pair of NP-relation such that $\mathcal{R}_{\text{zk}} \subseteq \mathcal{R}_{\text{snd}}$, the associated language be $\mathcal{L} = (\mathcal{L}_{\text{zk}}, \mathcal{L}_{\text{snd}})$. An NIZK argument system for $\mathcal{R}$ is a tuple $\text{NIZK} = (\text{Setup}, \text{Prove}, \text{Verify}, \text{Sim})$ of PPT algorithms, defined as follows.

- $\text{Setup}(1^\lambda, \text{info}_{\mathcal{L}})$ taking as input λ , a language-specific information $\text{info}_{\mathcal{L}}$, outputs a common reference string $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$ and a simulation trapdoor τ_{sim} . Here, $\text{crs}_{\mathcal{L}}$ is a language reference string giving the full description of $\mathcal{L}$.
- $\text{Prove}(\text{crs}, x, w) \rightarrow \pi$ taking as inputs crs , a statement x and a witness w , outputs a proof/argument π .
- $\text{Verify}(\text{crs}, x, \pi) \rightarrow \{0, 1\}$ is deterministic, taking as inputs crs , a statement x and a proof π , outputs either 1 or 0.
- $\text{Sim}(\text{crs}, \tau_{\text{sim}}, x) \rightarrow \{\pi^*, \perp\}$ taking as inputs crs , a simulation trapdoor τ_{sim} and statement x , outputs a simulated argument π^* or symbol $\perp$ indicating failure.

We require the following:

COMPLETENESS. There is a negligible function $\epsilon(\lambda)$ such that for $(x, w) \in \mathcal{R}_{\text{zk}}$

$$\Pr \left[\text{Verify}(\text{crs}, x, \pi) = 1 \mid \begin{array}{l} \text{crs} \leftarrow \text{Setup}(\lambda, \text{info}_{\mathcal{L}}), \\ \pi \leftarrow \text{Prove}(\text{crs}, x, w) \end{array} \right] = 1 - \epsilon(\lambda),$$

SOUNDNESS. For every PPT adversary $\mathcal{A}$ and for all $x \notin \mathcal{L}_{\text{snd}}$, it holds that

$$\Pr \left[\text{Verify}(\text{crs}, x, \pi) = 1 \mid \begin{array}{l} \text{crs} \leftarrow \text{Setup}(\lambda, \text{info}_{\mathcal{L}}), \\ \pi \leftarrow \mathcal{A}(\text{crs}, x) \end{array} \right] \in \text{negl}(\lambda).$$

COMPUTATIONAL/STATISTICAL ZERO-KNOWLEDGE. For $(\text{crs}, \tau_{\text{sim}}) \leftarrow \text{Setup}(1^\lambda, \text{info}_{\mathcal{L}})$, and for $(x, w) \in \mathcal{R}_{\text{zk}}$, the distributions

$$\{\pi \leftarrow \text{Prove}(\text{crs}, x, w)\},$$

and

$$\{\pi \leftarrow \text{Sim}(\text{crs}, \tau_{\text{sim}}, x)\},$$

are computationally/statistically indistinguishable.

A.6 Correlation Intractable Hash Functions

We provide the definition of correlation intractable hash functions following Canetti *et al.* [13]. They considered a class of a binary relation $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ that is unique-output, i.e. for any $x \in \mathcal{X}$, there is at most one $y \in \mathcal{Y}$ such that $(x, y) \in \mathcal{R}$. Such a relation $\mathcal{R}$ is *efficiently searchable in time T* if there

is a function f efficiently computable in time T such that if $(x, y) \in \mathcal{R}$ then $f(x) = y$.

For a unique-output, efficiently searchable relation $\mathcal{R}$, a hash family $\mathcal{H}$ that is *correlation intractable* w.r.t $\mathcal{R}$ satisfies that, for a hash function h with key k generated from $\mathcal{H}$, it is computationally hard for a PPT adversary to find $(x, y) \in \mathcal{R}$, such that $h(k, x) = y$. More formally, let $\lambda \in \mathbb{N}$ be a security parameter. A hash family with input length $n(\lambda)$ and output length $m(\lambda)$ is a collection $\mathcal{H} = \{h_\lambda : \{0, 1\}^{s(\lambda)} \times \{0, 1\}^{n(\lambda)} \rightarrow \{0, 1\}^{m(\lambda)}\}$ of keyed functions realized by efficient algorithms $(\text{Gen}, \text{Hash})$ where $\text{Gen}(1^\lambda)$ outputs a key $k \in \{0, 1\}^{s(\lambda)}$ and $\text{Hash}(k, x)$ computes a hash value $h_\lambda(k, x)$.

Definition 12 ([13]). For a relation ensemble $\{\mathcal{R}_\lambda \subseteq \{0, 1\}^{n(\lambda)} \times \{0, 1\}^{m(\lambda)}\}$, a hash function family $\mathcal{H} = \{h_\lambda : \{0, 1\}^{s(\lambda)} \times \{0, 1\}^{n(\lambda)} \rightarrow \{0, 1\}^{m(\lambda)}\}$ is $\mathcal{R}$ -correlation intractable if, for any PPT adversary $\mathcal{A}$, we have

$$\Pr[k \leftarrow \text{Gen}(1^\lambda), x \leftarrow \mathcal{A}(k) : (x, h_\lambda(k, x)) \in \mathcal{R}] \in \text{negl}(\lambda).$$

Canetti et al. [13] also considered a stronger definition of *somewhere statistically correlation intractable* for a hash family $\mathcal{H}$. Namely, the family $\mathcal{H}$ has two indistinguishable key generation algorithms KeyGen and StatGen ; and with high probability over the hash key k output by StatGen , there is no $x \in \mathcal{X}$ such that $(x, h(k, x)) \in \mathcal{R}$.

Definition 13 ([13]). Given a collection of relation ensemble $\mathfrak{R}$, a hash family $\mathcal{H}$ is *somewhere statistically correlation intractable* w.r.t. $\mathfrak{R}$ if there is an efficient fake key generation algorithm StatGen with the following properties:

- $\text{StatGen}(1^\lambda, \text{aux})$ takes as inputs a security parameter λ and an auxiliary input aux . It outputs a hashing key k ;
- For any $\mathcal{R} \in \mathfrak{R}$, there is an auxiliary input $\text{aux}_\mathcal{R}$ with these properties:
 - **Key indistinguishability:** The distributions $\{k : k \leftarrow \text{Gen}(1^\lambda)\}$ and $\{k : k \leftarrow \text{StatGen}(1^\lambda, \text{aux}_\mathcal{R})\}$ are computationally indistinguishable.
 - **Statistical Correlation Intractability:** With high probability over the choice of $k \leftarrow \text{StatGen}(1^\lambda, \text{aux}_\mathcal{R})$, no pair $(k, h(k, x))$ satisfies $\mathcal{R}$, i.e. we have

$$\Pr_{k \leftarrow \text{StatGen}(1^\lambda, \text{aux}_\mathcal{R})} [\exists x \in \{0, 1\}^{n(\lambda)} : (x, h(k, x)) \in \mathcal{R}] \leq 2^{-\Omega(\lambda)}.$$

- **Universality:** for any $\lambda \in \mathbb{N}$, input $x \in \{0, 1\}^n$, and output $y \in \{0, 1\}^m$, we have $\Pr[h(k, x) = y \mid k \leftarrow \text{Gen}(1^\lambda)] = 2^{-m}$.
- **Programmability:** there exists an efficient algorithm $\text{Sample}(1^\lambda, x, y)$ that samples from the conditional distribution $\{k \leftarrow \text{Gen}(1^\lambda) \mid h(k, x) = y\}$.

Peikert and Shiehian [50] provided a construction of somewhere statistically correlation intractable hash functions for efficiently searchable relations. Their construction is based on the GSW fully homomorphic encryption scheme [26] and relies on LWE with a polynomial approximation factor.

B NIZK from Trapdoor Σ -Protocols

We recall the compiler of [36], which turns any trapdoor Σ -protocol into a multi-theorem zero-knowledge and sound NIZK argument system. The following building blocks are needed:

- A trapdoor Σ -protocol $\Pi = (\text{Gen}_{\text{par}}, \text{Gen}_{\mathcal{L}}, \text{TrapGen}_{\mathcal{L}}, \text{P}, \text{V})$ for a language $\mathcal{L} = (\mathcal{L}_{\text{zk}}, \mathcal{L}_{\text{snd}})$ (e.g. the trapdoor Σ -protocols in Section 3 and Section C.1).
- A CI hash family $\mathcal{H} = (\text{Gen}, \text{Hash})$ with output length $\text{poly}(\lambda)$ for the class $\mathcal{R}_{\text{CI}}$ of efficiently searchable relations within time T (see Section A.6). We assume that the **BadChallenge** function of Π is computable in time T .
- A lossy PKE scheme $\text{LPKE} = (\text{Param}, \text{Keygen}, \text{LKeygen}, \text{Enc}, \text{Dec}, \text{Opener})$ (see Section A.2). For compatibility, we assume that the first-move messages in the trapdoor Σ -protocol Π can be embedded into the message space of LPKE. We also assume that algorithm **Dec** is computable in time T . Let R_{LPKE} be the randomness space of encryption, and $D_{R_{\text{LPKE}}}$ be the distribution from which the randomness of **Enc** is sampled.

We give the description of the NIZK argument system as follows:

- **Setup**($1^\lambda, \text{info}_{\mathcal{L}}$): taking as inputs a security parameter λ and language-specific information $\mathcal{L}$ specifying a norm bound $B > 0$, the algorithm does the following:
 1. Run $\text{Gen}_{\text{par}}(1^\lambda)$ to obtain $\text{par} \leftarrow \text{Gen}_{\text{par}}(1^\lambda)$;
 2. Generate a common reference string $(\text{par}, \text{crs}_{\mathcal{L}})$ for the Σ -protocol Π , by computing its language-dependent part as $\text{crs}_{\mathcal{L}} \leftarrow \text{Gen}_{\mathcal{L}}(\text{par}, \text{info}_{\mathcal{L}})$;
 3. Generate a key $k \leftarrow \text{Gen}(1^\lambda)$ for the somewhere CI hash family $\mathcal{H}$;
 4. Generate public parameters $\Gamma \leftarrow \text{LPKE.Param}(1^\lambda)$ for LPKE, then generate lossy encryption keys $(pk, sk) \leftarrow \text{LPKE.LKeygen}$;
 5. Choose a number of parallel repetitions κ . We remark that with the trapdoor Σ -protocols in Section 3, we can set $\kappa = \Theta(\lambda / \log \lambda)$.

In the real setup, output $\text{crs} = ((\text{par}, \text{crs}_{\mathcal{L}}), k, \kappa, pk)$ and discard sk . For simulation, the same setup procedure is used, but sk is retained as the simulation trapdoor $\tau_{\text{sim}} = sk$. Thus, the real and simulated setups produce identically distributed common reference strings.
- **Prove**(crs, x, w): To prove a statement $x \in \mathcal{L}_{\text{zk}}$ using a witness $w \in R_{\text{zk}}(x)$,
 1. Compute κ first messages $(\mathbf{u} = (u_1, \dots, u_\kappa), \text{st})$ via κ invocations of the prover of Π ;
 2. For each $i \in [\kappa]$, encrypt $\mathbf{a}_i \leftarrow \text{LPKE.Enc}(pk, u_i; r_i)$ using random coins $r_i \leftarrow D_{R_{\text{LPKE}}}$. Let $\mathbf{a} = (a_1, \dots, a_\kappa)$ and $\mathbf{r} = (r_1, \dots, r_\kappa)$;
 3. Compute $\text{Chall} = \text{Hash}(k, (x, \mathbf{a})) \in \mathcal{C}^\kappa$;
 4. For each $i \in [\kappa]$, compute the response

$$\mathbf{z}_i = \text{P}(\text{crs}_{\mathcal{L}}, (x, w), u_i, \text{Chall}[i], \text{st})$$

and parse $\mathbf{z} = (z_1, \dots, z_\kappa)$. Output $\pi = (\mathbf{u}, \mathbf{z}, \mathbf{r})$.

- **Verify**(crs, x, π): Given a statement x and a candidate proof $\pi = (\mathbf{u}, \mathbf{z}, \mathbf{r})$, parse $\mathbf{u} = (u_1, \dots, u_\kappa)$, $\mathbf{z} = (z_1, \dots, z_\kappa)$ and $\mathbf{r} = (r_1, \dots, r_\kappa)$. Then do the following:
 - For each $i \in [\kappa]$, compute $a_i \leftarrow \text{LPKE.Enc}(pk, u_i; r_i)$;
 - Parse $\mathbf{a} = (a_1, \dots, a_\kappa)$ and compute $\text{Chall} = \text{Hash}(k, (x, \mathbf{a})) \in \mathcal{C}^\kappa$;
 - For each $i \in [\kappa]$, execute the verifier of Π to check if $(u_i, \text{Chall}[i], z_i)$ is a valid proof for the statement x . Return 1 if and only if all checks pass.

Theorem 4 ([36]). *The above NIZK argument system is multi-theorem computational/statistical zero-knowledge and sound, assuming: (i) the trapdoor Σ -protocol Π is special computational/statistical zero-knowledge; (ii) LPKE is a lossy encryption scheme; (ii) the family $\mathcal{H}$ is somewhere correlation-intractable for efficiently searchable relations within time T , where T is the maximal running time of algorithm **BadChallenge** underlying Π .*

C Deferred Analysis of Our Ring Signature Scheme

C.1 Composing Lyubashevsky’s Σ -Protocol via OR Proof Technique

We present a trapdoor Σ -protocol built upon the protocol of Section 3 and the CDS OR technique [19]. Namely, given a collection of tl matrices $\{\mathbf{M}_i^{(j)}\}_{i \in [l], j \in [t]}$, the proposed Σ -protocol proves the existence of an index $j^* \in [t]$ such that

$$\mathbf{t}_i = \mathbf{M}_i^{(j^*)} \cdot \mathbf{m} \bmod q_i \in \mathcal{R}_{q_i}^{n_i}, \forall i \in [l]; \quad (27)$$

are satisfied by a witness $\mathbf{m} \in \mathcal{R}^{l_m}$ and that $\|\mathbf{m}\| \leq B_{\mathbf{m}}$, where $B_{\mathbf{m}} \ll \min\{q_i : i \in [l]\}$. Recall that, the CDS OR technique works by simulating “fake” proof for the branch $j \in [t] \setminus \{j^*\}$ with prover’s chosen challenge c_j , and generating the “real” proof for the branch j^* from the witness and a challenge c_{j^*} derived from $\{c_j\}_{j \neq j^*}$ and verifier’s challenge c . Assuming the challenge space has an abelian group structure with an $\oplus$ operation, c_{j^*} can be computed as the (unique) challenge such that $c_{j^*} \oplus (\oplus_{j \neq j^*} c_j) = c$. In the trapdoor Σ -protocol of Section 3, the challenge space **Chall** (4) naturally admits an abelian group structure, which is sufficient for applying the CDS OR technique.

We describe the algorithms of the OR trapdoor Σ -protocol as follows:

- **Gen_{par}**(1^λ) on input a security parameter $\lambda \in \mathbb{N}$ chooses a power-of-2 d defining the number field $\mathcal{R}$; choose moduli $q_0, \dots, q_{l-1} \in \text{poly}(\lambda)$; choose dimensions $l_{\mathbf{m}}, n_0, \dots, n_{l-1} \in \text{poly}(\lambda)$ and choose an integer $t \in \text{poly}(\lambda)$. The algorithm outputs

$$\text{par} = \{d, \{q_i\}_{i \in [l]}, l_{\mathbf{m}}, \{n_i\}_{i \in [l]}\}.$$

- **Gen_L**(**par**, **info_L**) takes as input public parameters **par**, a language-dependent information **info_L** specifying a norm bound $B_{\mathbf{m}} > 0$ and a collection of tl matrices $\{\mathbf{M}_i^{(j)}\}_{i \in [l], j \in [t]}$ such that $\mathbf{M}_i^{(j)} \in \mathcal{R}_{q_i}^{r_i \times n_0}, \forall i \in [l]$. It then performs the below steps:

1. Choose a positive integer B_{Chall} and a small integer $0 < d_{\text{Chall}} \leq d$ that together define the challenge space

$$\text{Chall} = \{c_0 + c_1X + \dots + c_{d_{\text{Chall}}-1}X^{d_{\text{Chall}}-1} \in \mathcal{R} : 0 \leq c_i \leq B_{\text{Chall}}\}.$$

We denote by $\overline{\text{Chall}}$ the set $\{\mathbf{c} - \mathbf{c}' : \mathbf{c}, \mathbf{c}' \in \text{Chall} \text{ and } \mathbf{c} \neq \mathbf{c}'\} \subset \mathcal{R}$. Additionally, define a binary operation $\oplus$ over Chall such that for all $\mathbf{c}_0, \mathbf{c}_1 \in \text{Chall}$

$$\mathbf{c}_0 \oplus \mathbf{c}_1 = \mathbf{c} \in \text{Chall} \text{ iff } \forall i \in [l] : \mathbf{c}[i] = \mathbf{c}_0[i] + \mathbf{c}_1[i] \bmod (B_{\text{Chall}} + 1).$$

It is easy to see that Chall together with $\oplus$ is an abelian group. We require that $\#\text{Chall} = (B_{\text{Chall}} + 1)^{d_{\text{Chall}}}$ is a polynomial in λ .

2. Generate public parameters of the extractable commitment scheme of Section A.4 in *extraction mode* (by running ExtSetup), for committing to messages in the space $\cup_{1 \leq i \leq l_{\mathbf{m}}} \mathcal{C}_{B_{\mathbf{m}}}^i$. Let the public parameters be

$$\text{pp} = (d, q, n, l_{\mathbf{m}}, l_{\mathbf{r}}, B_{\mathbf{m}}, B_{\mathbf{r}}, \mathbf{A}, \mathbf{B}),$$

and $(\mathbf{S}, p)$ be the extraction trapdoor. We note that $\mathbf{S} \in \mathcal{R}_q^{l_{\max} \times n}$ and p is a prime such that $\mathbf{S}\mathbf{A} + p\mathbf{E} = \mathbf{B} \bmod q$ where $\|\mathbf{E}\|_{\infty} \leq B_{\tau}$.

3. Choose a Gaussian width $\alpha > 0$ and a small parameter $\gamma \geq 1$ such that

$$\alpha = \sqrt{d} \cdot B_{\text{Chall}} \cdot \sqrt{B_{\mathbf{m}}^2 + dl_{\mathbf{r}}B_{\mathbf{r}}^2} \cdot \gamma \quad (28)$$

and set $M = \exp(14/\gamma + 1/(2\gamma^2))$. The parameter M dictates the rejection rate of the protocol.

The following constraints on the parameters are required

$$\begin{aligned} \alpha\sqrt{dl_{\mathbf{m}}} &< 1/2 \min\{q, q_0, \dots, q_{l-1}\}, & \alpha\sqrt{dl_{\mathbf{r}}} &< q/2 \\ p \cdot 2dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\mathbf{m}}} + 2\alpha\sqrt{dl_{\mathbf{m}}} &< q/2, & 2\alpha\sqrt{dl_{\mathbf{m}}} &< p/2 \end{aligned} \quad (29)$$

Let $\text{crs}_{\mathcal{L}} = \left(\{\mathbf{M}_i^{(j)}\}_{i \in [l], j \in [t]}, B_{\text{Chall}}, d_{\text{Chall}}, \alpha \right)$, the algorithm outputs $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$. The string $\text{crs}_{\mathcal{L}}$ defines the language

$$\mathcal{L}^{\text{OR}} = \left(\mathcal{L}_{\text{zk}}^{\text{OR}} := \bigcup_{j \in [t]} \mathcal{L}_{\text{zk}}^{(j)}, \mathcal{L}_{\text{snd}}^{\text{OR}} := \bigcup_{j \in [t]} \mathcal{L}_{\text{snd}}^{(j)} \right) \quad (30)$$

The language $\mathcal{L}_{\text{zk}}^{(j)}$ is specified by

$$\begin{aligned} \mathcal{L}_{\text{zk}}^{(j)} &= \{(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM} : \exists (\mathbf{m}, \mathbf{r}) \in \mathcal{S}_{B_{\mathbf{m}}}^{l_{\mathbf{m}}} \times \mathcal{C}_{B_{\mathbf{r}}}^{l_{\mathbf{r}}} \text{ s.t.} \\ \mathbf{c} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix} \bmod q \text{ and } \mathbf{t}_i = \mathbf{M}_i^{(j)} \cdot \mathbf{m} \bmod q_i, \forall i \in [l]\}, \end{aligned} \quad (31)$$

where $\mathcal{STM} = \mathcal{R}_q^{n+l_{\mathbf{m}}} \times \otimes_{i \in [l]} \mathcal{R}_{q_i}^{n_i}$. The language $\mathcal{L}_{\text{snd}}^{(j)}$ is specified by

$$\begin{aligned} \mathcal{L}_{\text{snd}}^{(j)} &= \{(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM} : \exists (\bar{\mathbf{c}}, \bar{\mathbf{m}}, \bar{\mathbf{e}}) \in \overline{\text{Chall}} \times \mathcal{S}_{B_{\mathbf{m}}}^{l_{\mathbf{m}}} \times \mathcal{S}_{B_{\mathbf{e}}}^{l_{\mathbf{r}}} \text{ s.t.} \\ (\mathbf{S} \ \mathbf{I}_{l_{\mathbf{m}}}) (\bar{\mathbf{c}} \cdot \mathbf{c}) &= p \cdot \bar{\mathbf{e}} + \bar{\mathbf{m}} \bmod q; \bar{\mathbf{c}} \cdot \mathbf{t}_i = \mathbf{M}_i^{(j)} \cdot \bar{\mathbf{m}} \bmod q_i, \forall i \in [l]\}, \end{aligned} \quad (32)$$

where $\bar{B}_{\mathbf{m}} = 2\alpha\sqrt{dl_{\mathbf{m}}}$, $\bar{B}_{\mathbf{e}} = 2dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\mathbf{m}}}$.

- $P(\text{crs}, (\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}), (\mathbf{m}, \mathbf{r}))$ and $V(\text{crs}, (\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}))$ take as common input a CRS $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, a statement $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM}$. The prover P takes as private input a witness $(\mathbf{m}, \mathbf{r}) \in \mathcal{R}^{l_m} \times \mathcal{R}^{l_r}$ such that $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{L}_{\text{zk}}^{i^*}$ for some $j^* \in [t]$, and generates the first message as follows:
 1. For $j \in [t]$ and $j \neq j^*$, sample $c_j \xleftarrow{\$} \text{Chall}$ and execute algorithm ZKSim in the proof of Proposition 1 on the challenge c_j , let the pair of first message and response be $(\mathbf{a}^{(j)}, \mathbf{z}^{(j)})$
 2. Using witness $(\mathbf{m}, \mathbf{r})$, execute the prover of the trapdoor Σ -protocol for $(\mathcal{L}_{\text{zk}}^{(j^*)}, \mathcal{L}_{\text{snd}}^{(j^*)})$ (Figure 1) to obtain a first message $\mathbf{a}^{(j^*)}$. Let the final first message be $(\mathbf{a}^{(j)})_{j \in [t]}$.
 Upon receiving V 's challenge $c \in \text{Chall}$, P computes the unique $c_{j^*} \in \text{Chall}$ such that $\bigoplus_{j \in [t]} c_j = c$, then runs the protocol of Figure 1 to obtain the response $\mathbf{z}^{(j^*)}$ with respect to challenge c_{j^*} . Let the final response be

$$\left((\mathbf{z}^{(j)}, c_j) \right)_{j \in [t]}.$$

Receiving a response $((\mathbf{z}^{(j)}, c_j))_{j \in [t]}$, V checks whether $\bigoplus_{j \in [t]} c_j = c$, and executes the verifier in the protocol of Figure 1 to check that if for all $j \in [t]$, $(\mathbf{a}^{(j)}, c_j, \mathbf{z}^{(j)})$ is a valid transcript of the trapdoor Σ -protocol proving $(\mathcal{L}_{\text{zk}}^{(j)}, \mathcal{L}_{\text{snd}}^{(j)})$. Returns 1 if and only if all checks succeed.

- $\text{TrapGen}(\text{par}, \mathcal{L})$ is identical to $\text{Gen}_{\mathcal{L}}$, the difference is that it retains the trapdoor $\tau = (\mathbf{S}, p)$ generated by ExtSetup , where $\mathbf{S} \in \mathcal{R}_q^{l_m \times n}$. It sets the trapdoor $\tau_{\Sigma} = (\mathbf{S}, p)$.
- $\text{BadChallenge}_{\text{OR}}(\tau_{\Sigma}, \text{crs}, (\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}), \mathbf{a})$ takes as input a trapdoor $\tau_{\Sigma} = (\mathbf{S}, p)$ where $\mathbf{S} \in \mathcal{R}_q^{l_m \times n}$, a CRS $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, a statement $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM}$ and a prover's first message $\{\mathbf{a}^{(j)} = (\mathbf{a}_{\mathbf{c}}^{(j)}, (\mathbf{a}_i^{(j)})_{i \in [l]})\}_{j \in [t]}$ such that $(\mathbf{a}_{\mathbf{c}}^{(j)}, (\mathbf{a}_i^{(j)})_{i \in [l]}) \in \mathcal{R}_q^{n+l_m} \times \bigotimes_{i \in [l]} \mathcal{R}_{q_i}^{r_i}$. It then proceeds as follows:
 1. For all $j \in [t]$, execute the BadChallenge algorithm of the trapdoor Σ -protocol of Section 3 for the language $(\mathcal{L}_{\text{zk}}^{(j)}, \mathcal{L}_{\text{snd}}^{(j)})$, on input $\tau_{\Sigma} = (\mathbf{S}, p)$, reference string crs , statement $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1})$ and first message $\mathbf{a}^{(j)} = (\mathbf{a}_{\mathbf{c}}^{(j)}, (\mathbf{a}_i^{(j)})_{i \in [l]})$.
 2. If BadChallenge outputs a challenge c_j for all $j \in [t]$, outputs $c = \bigoplus_{j \in [t]} c_j$. Otherwise, output $\perp$.

Proposition 4. *The above protocol is a trapdoor Σ -protocol for the language $\mathcal{L}^{\text{OR}} = (\mathcal{L}_{\text{zk}}^{\text{OR}}, \mathcal{L}_{\text{snd}}^{\text{OR}})$ specified by (30), (31) and (32).*

Proof. The protocol is complete and special zero-knowledge since each atomic protocol for $(\mathcal{L}_{\text{zk}}^{(j)}, \mathcal{L}_{\text{snd}}^{(j)})$ is complete and admits a statistical special zero-knowledge simulator. It remains to argue special soundness and correctness of bad challenge. For special soundness, assume that for a statement $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM}$ and a first message $(\mathbf{a}^{(j)} = (\mathbf{a}_{\mathbf{c}}^{(j)}, (\mathbf{a}_i^{(j)})_{i \in [l]}))_{j \in [t]}$, there are two valid pairs of challenge-response

$$c \in \text{Chall}, (\mathbf{z}^{(j)}, c_j)_{j \in [t]},$$

and

$$c' \in \text{Chall}, (\mathbf{z}^{(j)'}, c'_j)_{j \in [t]},$$

such that $c \neq c'$. As $c = \oplus_{j \in [t]} c_j$ and $c' = \oplus_{j \in [t]} c'_j$, there exists an index $j^* \in [t]$ such that $c_{j^*} \neq c'_{j^*}$. It follows that for statement $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{STM}$, and first message $\mathbf{a}^{(j^*)}$, there are two valid pairs of challenge-response $(c_{j^*}, \mathbf{z}^{(j^*)})$ and $(c'_{j^*}, \mathbf{z}^{(j^*)'})$ with respect to the trapdoor Σ -protocol of Section 3 for the language $(\mathcal{L}_{\text{zk}}^{(j^*)}, \mathcal{L}_{\text{snd}}^{(j^*)})$. Therefore, using the argument for special soundness in the proof of Proposition 1, we have that $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \in \mathcal{L}_{\text{snd}}^{(j^*)}$ and is a statement of $\mathcal{L}_{\text{snd}}^{\text{OR}}$. This in turn implies that if $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \notin \mathcal{L}_{\text{snd}}^{\text{OR}}$, for any first message $(\mathbf{a}^{(j)})_{j \in [t]}$ there can be at most one challenge admitting a valid response.

For correctness of $\text{BadChallenge}_{\text{OR}}$, note that if $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \notin \mathcal{L}_{\text{snd}}^{\text{OR}}$, then for all $j \in [t]$ we have that $(\mathbf{c}, \mathbf{t}_0, \dots, \mathbf{t}_{l-1}) \notin \mathcal{L}_{\text{snd}}^{(j)}$. By the special soundness property of the trapdoor Σ -protocol of Section 3 for the language $(\mathcal{L}_{\text{zk}}^{(j)}, \mathcal{L}_{\text{snd}}^{(j)})$ and the correctness of the corresponding algorithm BadChallenge , there is at most one challenge $c_j \in \text{Chall}$ so that there is some response $\mathbf{z}^{(j)}$ making $((\mathbf{a}_{\mathbf{c}}^{(j)}, (\mathbf{a}_i^{(i)})_{i \in [l]}), c_j, \mathbf{z}^{(j)})$ an accepting transcript of the Σ -protocol for the language $(\mathcal{L}_{\text{zk}}^{(j)}, \mathcal{L}_{\text{snd}}^{(j)})$, and BadChallenge returns such challenge c_j . The correctness of $\text{BadChallenge}_{\text{OR}}$ then follows. Finally, the runtime of $\text{BadChallenge}_{\text{OR}}$ is linear in t and the runtime of BadChallenge and therefore is a polynomial in λ . $\square$

Communication cost. The prover's communication consists of t first messages and responses in the basic trapdoor Σ -protocol of Section 3. Therefore from (13), the prover's communication cost in the OR trapdoor Σ -protocol is

$$\begin{aligned} t(n + l_{\mathbf{m}})d \log q + td \sum_{i \in [\ell]} n_i \log q_i + t(l_{\mathbf{m}} + l_{\mathbf{r}})d \log(\alpha \sqrt{d(l_{\mathbf{m}} + l_{\mathbf{r}})}) \\ + td_{\text{Chall}} \log(B_{\text{Chall}} + 1). \end{aligned} \quad (33)$$

C.2 The NIZK Proof Underlying Our Ring Signature

We describe the NIZK proof system underlying the ring signature scheme of Section 5.2. The NIZK is obtained via the compiler of Section B, and the trapdoor Σ -protocol for set membership of Section 4, but with some modification to the statement defining set membership relation. Recall that at depth $\ell - 1$ of the accumulator we define the language $\mathcal{L}_{\text{set}}^{(\ell-1)} = (\mathcal{L}_{\text{set}, \text{zk}}^{(\ell-1)}, \mathcal{L}_{\text{set}, \text{snd}}^{(\ell-1)})$ ((21), (22)) with the constraints that:

- The KTX commitment $\mathbf{c}_{\mathbf{v}_{\ell-1}}$ opens to some $\mathbf{v}_{\ell-1}$;
- The KTX commitment $\mathbf{c}_{\mathbf{v}_{\ell-2}}$ opens to some $\mathbf{v}_{\ell-2}$;
- The node $\mathbf{v}_{\ell-1}$ and the witness $\mathbf{w}_{\ell-1}$ are hashed correctly into $\mathbf{v}_{\ell-2}$.

In our ring signature, the leaf node $\mathbf{v}_{\ell-1}$ has the form $(\mathbf{d}, M)$ where $\mathbf{d} \in \mathcal{C}_{\beta}^{n_{k_{\text{H}}}}$ is a signer's public key and $M \in \mathcal{R}_{\text{bin}}^{l_M}$ is a message. This requires adding linear constraint between signer's secret $\mathbf{x} \in \mathcal{R}_{\text{bin}}^{m_{\text{F}}}$ and public key $\mathbf{d} \in \mathcal{C}_{\beta}^{n_{k_{\text{H}}}}$, that is

$\mathbf{G}_B \cdot \mathbf{d} = \mathbf{F} \cdot \mathbf{x} \bmod q_H$, and modifying the linear constraints involving $\mathbf{v}_{\ell-1}$. Namely, the constraints

$$\begin{aligned} \exists j \in \{0, 1\} : \mathbf{G}_B \mathbf{v}_{\ell-2} &= \mathbf{H}_j \cdot (\mathbf{d}, M) + \mathbf{H}_{1-j} \cdot \mathbf{w}_i \bmod q_H; \\ \mathbf{C} \cdot (\mathbf{d}, M) + \mathbf{D} \cdot \mathbf{r}_d &= \mathbf{c}_{\mathbf{v}_{\ell-1}} \bmod q_C. \end{aligned}$$

We parse $\mathbf{C} = [\mathbf{C}_0 | \mathbf{C}_M]$, where $\mathbf{C}_0 \in \mathcal{R}_{q_C}^{n_C \times nk_H}$ and $\mathbf{C}_M \in \mathcal{R}_{q_C}^{n_C \times l_M}$. And for $j \in \{0, 1\}$, we parse $\mathbf{H}_j = [\bar{\mathbf{H}}_j | \mathbf{H}_{j,M}]$ where $\bar{\mathbf{H}}_j \in \mathcal{R}_{q_H}^{n \times (nk_H - l_M)}$, $\mathbf{H}_{j,M} \in \mathcal{R}_{q_H}^{n \times l_M}$. The full linear relations at depth $\ell - 1$ become

$$\begin{aligned} \exists j \in \{0, 1\} : \bar{\mathbf{H}}_j \cdot \mathbf{d} + \mathbf{H}_{1-j} \cdot \mathbf{w}_i - \mathbf{G}_B \mathbf{v}_{\ell-2} &= -\mathbf{H}_{j,M} \cdot M \bmod q_H; \\ \mathbf{G}_B \cdot \mathbf{d} - \mathbf{F} \cdot \mathbf{x} &= \mathbf{0}^{n_F} \bmod q_H; \\ \mathbf{C}_0 \cdot \mathbf{d} + \mathbf{D} \cdot \mathbf{r}_d &= \mathbf{c}_{\mathbf{v}_{\ell-1}} - \mathbf{C}_M \cdot M \bmod q_C. \end{aligned}$$

The rest of the trapdoor Σ -protocol for the relation underlying our ring signature scheme is similar to the trapdoor Σ -protocol of Section 4. For completeness, we provide the full description below:

- $\text{Gen}_{\text{par}}(1^\lambda)$: On input security parameter $\lambda \in \mathbb{N}$, choose a power-of-2 d defining number field $\mathcal{R}$; choose two prime moduli $q_C, q_H \in \text{poly}(\lambda)$ and choose:
 - Dimensions $n, n_F, m_F \in \text{poly}(\lambda)$ such that $m_F > n_F \log q_H$;
 - An integer base $B > 1$, defining a norm bound $\beta = B - 1$ and a decomposition parameter $k_H = \lceil \log_B q_H \rceil$;
 - Dimensions $l_M, m_H \in \text{poly}(\lambda)$ such that $m_H = 2nk_H = 2(n_F k_H + l_M)$;
 - Dimensions $n_C, n_R \in \text{poly}(\lambda)$ and a norm bound B_R .
The above parameters additionally define a dimension $l_{\mathbf{m}} = 3nk_H + 2n_R$, a dimension $l_{\mathbf{m}}^{(0)} = m + n_R$ and a dimension $l_{\mathbf{m}}^{(\ell-1)} = n_F k_H + 2nk_H + 2n_R + m_F$. Let $l_{\max} = \max\{l_{\mathbf{m}}, l_{\mathbf{m}}^{(0)}, l_{\mathbf{m}}^{(\ell-1)}\}$ specifying the maximum message length in the extractable commitment scheme of Section A.4. The algorithm outputs public parameters as

$$\text{par} = \{d, q_C, q_H, n, n_F, m_F, l_M, m_H, n_C, n_R, l_{\mathbf{m}}, l_{\mathbf{m}}^{(0)}, l_{\mathbf{m}}^{(\ell-1)}, B, B_R\}. \quad (34)$$

- $\text{Gen}_{\mathcal{L}}(\text{par}, \text{info}_{\mathcal{L}})$: Takes as input par and language-dependent information $\text{info}_{\mathcal{L}} = N \in \text{poly}(\lambda)$ such that $N = 2^\ell$ for some $\ell \in \mathbb{N}$, the algorithm then proceeds as follows:
 1. Choose a positive integer B_{Chall} and a small integer $0 < d_{\text{Chall}} \leq d$ that together define the challenge space

$$\text{Chall} = \{c_0 + c_1 X + \dots + c_{d_{\text{Chall}}-1} X^{d_{\text{Chall}}-1} \in \mathcal{R} : 0 \leq c_i \leq B_{\text{Chall}}\}.$$

We denote by $\overline{\text{Chall}}$ the set $\{\mathbf{c} - \mathbf{c}' : \mathbf{c}, \mathbf{c}' \in \text{Chall} \text{ and } \mathbf{c} \neq \mathbf{c}'\} \subset \mathcal{R}$. Additionally, define a binary operation $\oplus$ over Chall such that for all $\mathbf{c}_0, \mathbf{c}_1 \in \text{Chall}$

$$\mathbf{c}_0 \oplus \mathbf{c}_1 = \mathbf{c} \in \text{Chall} \text{ iff } \forall i \in [l] : \mathbf{c}[i] = \mathbf{c}_0[i] + \mathbf{c}_1[i] \bmod (B_{\text{Chall}} + 1).$$

It is easy to see that Chall together with $\oplus$ is an abelian group. We require that $\#\text{Chall} = (B_{\text{Chall}} + 1)^{d_{\text{Chall}}}$ is a polynomial in λ .

2. Let $B_{\mathbf{m}} = \max \left\{ \sqrt{d[3\beta^2 n k_{\mathbf{H}} + 2B_{\mathbf{R}}^2 n_{\mathbf{R}}]}, \sqrt{d[\beta^2(2n + n_{\mathbf{F}})k_{\mathbf{H}} + 2B_{\mathbf{R}}^2 n_{\mathbf{R}} + m_{\mathbf{F}}]} \right\}$, then generate public parameters of the extractable commitment scheme of Section A.4 in *extraction mode* (by running **ExtSetup**), for committing to messages in the space $\cup_{1 \leq i \leq l_{\max}} \mathcal{C}_{B_{\mathbf{m}}}^i$. Let the public parameters be

$$\text{pp} = (d, q, n, l_{\max}, l_{\mathbf{r}}, B_{\mathbf{m}}, B_{\mathbf{r}}, \mathbf{A}, \mathbf{B}),$$

and $(\mathbf{S}, p)$ be the extraction trapdoor. We note that $\mathbf{S} \in \mathcal{R}_q^{l_{\max} \times n}$ and p is a prime such that $\mathbf{S}\mathbf{A} + p\mathbf{E} = \mathbf{B} \bmod q$ where $\|\mathbf{E}\|_{\infty} \leq B_{\tau}$.

3. Sample $\mathbf{F} \xleftarrow{\$} \mathcal{R}_{q_{\mathbf{H}}}^{n_{\mathbf{F}} \times m_{\mathbf{F}}}$, $\mathbf{H} = [\mathbf{H}_0 | \mathbf{H}_1] \xleftarrow{\$} \mathcal{R}_{q_{\mathbf{H}}}^{n \times m_{\mathbf{H}}}$, where $\mathbf{H}_0, \mathbf{H}_1 \in \mathcal{R}_{q_{\mathbf{H}}}^{n \times n_{\mathbf{H}} k_{\mathbf{H}}}$ defining a 2-input lattice-based hash function of Section 4.1. For $j \in \{0, 1\}$, we let $\mathbf{H}_j = [\overline{\mathbf{H}}_j | \mathbf{H}_{j,M}]$ where $\overline{\mathbf{H}}_j \in \mathcal{R}_{q_{\mathbf{H}}}^{n \times (n k_{\mathbf{H}} - l_M)}$, $\mathbf{H}_{j,M} \in \mathcal{R}_{q_{\mathbf{H}}}^{n \times l_M}$;
4. Sample $\mathbf{C} \xleftarrow{\$} \mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}} \times n k_{\mathbf{H}}}$, $\mathbf{D} \xleftarrow{\$} \mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}} \times n_{\mathbf{R}}}$ that together define the public parameters of the KTX commitment scheme [32]. We let $\mathbf{C} = [\mathbf{C}_0 | \mathbf{C}_M]$, where $\mathbf{C}_0 \in \mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}} \times n_{\mathbf{F}} k_{\mathbf{H}}}$ and $\mathbf{C}_M \in \mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}} \times l_M}$;
5. Choose a Gaussian width $\alpha > 0$ and a parameter $\gamma \geq 1$ such that

$$\alpha = \sqrt{d} \cdot B_{\text{Chall}} \cdot \sqrt{B_{\mathbf{m}}^2 + dl_{\mathbf{r}} B_{\mathbf{r}}^2} \cdot \gamma$$

Set $M = \exp(14/\gamma + 1/(2\gamma^2))$.

From (29), we require that

$$\begin{aligned} \alpha \sqrt{dl_{\max}} &< 1/2 \min\{q, q_{\mathbf{H}}, q_{\mathbf{C}}\}, & \alpha \sqrt{dl_{\mathbf{r}}} &< q/2, \\ p \cdot 2dl_{\mathbf{r}} B_{\tau} \alpha \sqrt{dl_{\mathbf{r}} l_{\max}} + 2\alpha \sqrt{dl_{\max}} &< q/2, & 2\alpha \sqrt{dl_{\max}} &< p/2. \end{aligned}$$

The algorithm sets

$$\text{crs}_{\mathcal{L}} = (q, l_{\mathbf{r}}, B_{\mathbf{m}}, B_{\mathbf{r}}, \mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{H}, \mathbf{C}, \mathbf{D}, B_{\text{Chall}}, d_{\text{Chall}}, \alpha) \quad (35)$$

and outputs $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$.

For $i \in [\ell]$ and $j \in \{0, 1\}$, the system parameters pp define the languages $\mathcal{L}_{\text{ring}} = (\mathcal{L}_{\text{ring}, \text{zk}}, \mathcal{L}_{\text{ring}, \text{snd}})$ such that

$$\begin{aligned} \mathcal{L}_{\text{ring}, \text{zk}} &= \{(M, \mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}) : M \in \mathcal{R}_{\text{bin}}^{l_M}, \mathbf{u} \in \mathcal{C}_{\beta}^{n k_{\mathbf{H}}}, \mathbf{c}_0 \in \mathcal{R}_q^{n + l_{\mathbf{m}}^{(0)}}, \\ &\quad \mathbf{c}_{\ell-1} \in \mathcal{R}_q^{n + l_{\mathbf{m}}^{(\ell-1)}}, (\mathbf{c}_i)_{i \in \{1, \dots, \ell-2\}} \in (\mathcal{R}_q^{n + l_{\mathbf{m}}})^{\ell-2}, (\mathbf{c}_i)_{i \in [\ell]} \in (\mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}}})^{\ell} \\ &\quad \text{s.t. } \forall i \in [\ell-1], i > 0 : (\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) \in \mathcal{L}_{\text{ring}, \text{zk}}^{(i)}; (\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{L}_{\text{ring}, \text{zk}}^{(0)} \\ &\quad \text{and } (M, \mathbf{c}_{\ell-1}, \mathbf{c}_{\mathbf{v}_{\ell-1}}, \mathbf{c}_{\mathbf{v}_{\ell-2}}) \in \mathcal{L}_{\text{ring}, \text{zk}}^{(\ell-1)}\}, \end{aligned} \quad (36)$$

and

$$\begin{aligned} \mathcal{L}_{\text{ring}, \text{snd}} &= \{(M, \mathbf{u}, (\mathbf{c}_i)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i})_{i \in [\ell]}) : M \in \mathcal{R}_{\text{bin}}^{l_M}, \mathbf{u} \in \mathcal{C}_{\beta}^{n k_{\mathbf{H}}}, \mathbf{c}_0 \in \mathcal{R}_q^{n + l_{\mathbf{m}}^{(0)}}, \\ &\quad \mathbf{c}_{\ell-1} \in \mathcal{R}_q^{n + l_{\mathbf{m}}^{(\ell-1)}}, (\mathbf{c}_i)_{i \in \{1, \dots, \ell-2\}} \in (\mathcal{R}_q^{n + l_{\mathbf{m}}})^{\ell-2}, (\mathbf{c}_i)_{i \in [\ell]} \in (\mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}}})^{\ell} \\ &\quad \text{s.t. } \forall i \in [\ell-1], i > 0 : (\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) \in \mathcal{L}_{\text{ring}, \text{zk}}^{(i)}; (\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{L}_{\text{ring}, \text{snd}}^{(0)} \\ &\quad \text{and } (M, \mathbf{c}_{\ell-1}, \mathbf{c}_{\mathbf{v}_{\ell-1}}, \mathbf{c}_{\mathbf{v}_{\ell-2}}) \in \mathcal{L}_{\text{ring}, \text{snd}}^{(\ell-1)}\}, \end{aligned} \quad (37)$$

The languages $\mathcal{L}_{\text{ring,zk}}^{(i)} := \cup_{j \in \{0,1\}} \mathcal{L}_{\text{ring,zk}}^{(i,j)}$ and $\mathcal{L}_{\text{ring,snd}}^{(i)} := \cup_{j \in \{0,1\}} \mathcal{L}_{\text{set,snd}}^{(i,j)}$ capture the relations at depth i of the accumulator as follows:

- For $\ell - 1 > i > 0$:

$$\begin{aligned} \mathcal{L}_{\text{ring,zk}}^{(i,j)} &= \{(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) \in \mathcal{R}_q^{n+l_{\mathbf{m}}} \times (\mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}}})^2 : \\ \exists(\mathbf{m}_i, \mathbf{r}_i) &\in (\mathcal{C}_{\beta}^{n_{k_{\mathbf{H}}}} \times \mathcal{C}_{B_{\mathbf{R}}}^{n_{\mathbf{R}}})^2 \times \mathcal{C}_{\beta}^{n_{k_{\mathbf{H}}}} \times \mathcal{C}_{B_{\mathbf{r}}}^{l_{\mathbf{r}}} \text{ s.t.} \\ \mathbf{c}_i &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B}[:l_{\mathbf{m}}] \end{bmatrix} \mathbf{r}_i + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_i \end{bmatrix} \bmod q, (\mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) = \mathbf{M}_0 \cdot \mathbf{m}_i \bmod q_{\mathbf{C}} \\ &\text{and } \mathbf{0}^n = \mathbf{M}_0^{(j)} \mathbf{m}_i \bmod q_{\mathbf{H}} \}, \end{aligned} \quad (38)$$

where

$$\begin{aligned} \mathbf{M}_0 &= \begin{bmatrix} \mathbf{C} \mathbf{D} & \mathbf{0}^{n_{\mathbf{C}} \times n_{k_{\mathbf{H}}}} \\ \mathbf{C} \mathbf{D} & \mathbf{0}^{n_{\mathbf{C}} \times n_{k_{\mathbf{H}}}} \end{bmatrix} \in \mathcal{R}_{q_{\mathbf{C}}}^{2n_{\mathbf{C}} + l_{\mathbf{m}}}, \\ \mathbf{M}_0^{(j)} &= [\mathbf{H}_j | \mathbf{0}^{n \times n_{\mathbf{R}}} | -\mathbf{G}_B | \mathbf{0}^{n \times n_{\mathbf{R}}} | \mathbf{H}_{1-j}] \in \mathcal{R}_{q_{\mathbf{H}}}^{n \times l_{\mathbf{m}}}. \end{aligned}$$

And that

$$\begin{aligned} \mathcal{L}_{\text{ring,snd}}^{(i,j)} &= \{(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) \in \mathcal{R}_q^{n+l_{\mathbf{m}}} \times (\mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}}})^2 : \\ &\exists(\bar{\mathbf{c}}_i, \bar{\mathbf{m}}_i, \bar{\mathbf{e}}_i) \in \overline{\text{Chall}} \times \mathcal{S}_{\bar{B}_{\mathbf{m}}}^{l_{\mathbf{m}}} \times \mathcal{S}_{\bar{B}_{\mathbf{e}}}^{l_{\mathbf{m}}} \text{ s.t.} \\ [\mathbf{S}[:l_{\mathbf{m}}] \quad \mathbf{I}_{l_{\mathbf{m}}}] &(\bar{\mathbf{c}}_i \cdot \mathbf{c}_i) = p \cdot \bar{\mathbf{e}}_i + \bar{\mathbf{m}}_i \bmod q, \\ \bar{\mathbf{c}}_i \cdot (\mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}}) &= \mathbf{M}_0 \cdot \bar{\mathbf{m}}_i \bmod q_{\mathbf{C}}, \\ &\text{and } \mathbf{0}^n = \mathbf{M}_0^{(j)} \bar{\mathbf{m}}_i \bmod q_{\mathbf{H}} \}, \end{aligned} \quad (39)$$

where $\bar{B}_{\mathbf{m}} = 2\alpha\sqrt{dl_{\max}}$, $\bar{B}_{\mathbf{e}} = 2dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\max}}$.

- For $i = 0$:

$$\begin{aligned} \mathcal{L}_{\text{ring,zk}}^{(0,j)} &= \{(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{C}_{\beta}^{n_{k_{\mathbf{H}}}} \times \mathcal{R}_q^{n+l_{\mathbf{m}}^{(0)}} \times \mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}}} : \\ \exists(\mathbf{m}_0, \mathbf{r}_0) &\in \mathcal{C}_{\beta}^{n_{k_{\mathbf{H}}}} \times \mathcal{C}_{B_{\mathbf{R}}}^{n_{\mathbf{R}}} \times \mathcal{C}_{\beta}^{n_{k_{\mathbf{H}}}} \times \mathcal{C}_{B_{\mathbf{r}}}^{l_{\mathbf{r}}} \text{ s.t.} \\ \mathbf{c}_0 &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B}[:l_{\mathbf{m}}^{(0)}] \end{bmatrix} \mathbf{r}_0 + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_0 \end{bmatrix} \bmod q, \mathbf{c}_{\mathbf{v}_0} = [\mathbf{C} | \mathbf{D} | \mathbf{0}^{n_{\mathbf{C}} \times n_{k_{\mathbf{H}}}}] \cdot \mathbf{m}_0 \bmod q_{\mathbf{C}} \\ &\text{and } \mathbf{G}_B \mathbf{u} = [\mathbf{H}_j | \mathbf{0}^{n \times n_{\mathbf{R}}} | \mathbf{H}_{1-j}] \cdot \mathbf{m}_0 \bmod q_{\mathbf{H}} \}. \end{aligned} \quad (40)$$

And that

$$\begin{aligned} \mathcal{L}_{\text{ring,snd}}^{(0,j)} &= \{(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0}) \in \mathcal{C}_{\beta}^{n_{k_{\mathbf{H}}}} \times \mathcal{R}_q^{n+l_{\mathbf{m}}^{(0)}} \times \mathcal{R}_{q_{\mathbf{C}}}^{n_{\mathbf{C}}} : \\ &\exists(\bar{\mathbf{c}}_0, \bar{\mathbf{m}}_0, \bar{\mathbf{e}}_0) \in \overline{\text{Chall}} \times \mathcal{S}_{\bar{B}_{\mathbf{m}}}^{l_{\mathbf{m}}^{(0)}} \times \mathcal{S}_{\bar{B}_{\mathbf{e}}}^{l_{\mathbf{m}}^{(0)}} \text{ s.t.} \\ [\mathbf{S}[:l_{\mathbf{m}}^{(0)}] \quad \mathbf{I}_{l_{\mathbf{m}}^{(0)}}] &(\bar{\mathbf{c}}_0 \cdot \mathbf{c}_0) = p \cdot \bar{\mathbf{e}}_0 + \bar{\mathbf{m}}_0 \bmod q, \\ \bar{\mathbf{c}}_0 \cdot \mathbf{c}_{\mathbf{v}_0} &= [\mathbf{C} | \mathbf{D} | \mathbf{0}^{n_{\mathbf{C}} \times n_{k_{\mathbf{H}}}}] \cdot \bar{\mathbf{m}}_0 \bmod q_{\mathbf{C}}, \\ \bar{\mathbf{c}}_0 \cdot \mathbf{G}_B \mathbf{u} &= [\mathbf{H}_j | \mathbf{0}^{n \times n_{\mathbf{R}}} | \mathbf{H}_{1-j}] \cdot \bar{\mathbf{m}}_0 \bmod q_{\mathbf{H}} \}. \end{aligned} \quad (41)$$

where $\overline{B}_{\mathbf{m}} = 2\alpha\sqrt{dl_{\max}}$, $\overline{B}_{\mathbf{e}} = 2dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\max}}$.

- For $i = \ell - 1$:

$$\begin{aligned}
\mathcal{L}_{\text{ring}, \text{zk}}^{(\ell-1, j)} &= \{(M, \mathbf{c}_{\ell-1}, \mathbf{c}_{\mathbf{v}_{\ell-1}}, \mathbf{c}_{\mathbf{v}_{\ell-2}}) \in \mathcal{R}_{\text{bin}}^{l_M} \times \mathcal{R}_q^{n+l_{\mathbf{m}}^{(\ell-1)}} \times (\mathcal{R}_{q_C}^{n_C})^2 : \\
\exists(\mathbf{m}_{\ell-1}, \mathbf{r}_{\ell-1}) &\in (\mathcal{C}_{\beta}^{n_{\mathbf{F}}k_{\mathbf{H}}} \times \mathcal{C}_{B_R}^{n_R}) \times \mathcal{C}_{\beta}^{nk_{\mathbf{H}}} \times \mathcal{C}_{B_R}^{n_R} \times \mathcal{C}_{\beta}^{nk_{\mathbf{H}}} \times \mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}} \times \mathcal{C}_{B_r}^{l_r} \text{ s.t.} \\
\mathbf{c}_{\ell-1} &= \begin{bmatrix} \mathbf{A} \\ \mathbf{B}[:l_{\mathbf{m}}^{(\ell-1)}] \end{bmatrix} \mathbf{r}_{\ell-1} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_{\ell-1} \end{bmatrix} \bmod q, \\
(\mathbf{c}_{\mathbf{v}_{\ell-1}} - \mathbf{C}_M \cdot M, \mathbf{c}_{\mathbf{v}_{\ell-2}}) &= \mathbf{M}_0^{(\ell-1)} \cdot \mathbf{m}_{\ell-1} \bmod q_C, \\
(-\mathbf{H}_{j, M} \cdot M, \mathbf{0}^{n_{\mathbf{F}}}) &= \mathbf{M}_0^{(\ell-1, j)} \mathbf{m}_{\ell-1} \bmod q_{\mathbf{H}},
\end{aligned} \tag{42}$$

where

$$\begin{aligned}
\mathbf{M}_0^{(\ell-1)} &= \begin{bmatrix} \mathbf{C}_0 \mathbf{D} & \mathbf{0}^{n_C \times (nk_{\mathbf{H}} + \mathbf{F})} \\ \mathbf{C} \mathbf{D} & \mathbf{0}^{n_C \times (nk_{\mathbf{H}} + m_{\mathbf{F}})} \end{bmatrix} \in \mathcal{R}_{q_C}^{2n_C \times l_{\mathbf{m}}^{(\ell-1)}}, \\
\mathbf{M}_0^{(\ell-1, j)} &= \begin{bmatrix} \overline{\mathbf{H}}_j | \mathbf{0}^{n \times n_R} | -\mathbf{G}_B | \mathbf{0}^{n \times n_R} | \mathbf{H}_{1-j} | \mathbf{0}^{n \times m_{\mathbf{F}}} \\ \mathbf{G}_B | \mathbf{0}^{n \times n_R} | \mathbf{0}^{n \times nk_{\mathbf{H}}} | \mathbf{0}^{n \times n_R} | \mathbf{0}^{n \times nk_{\mathbf{H}}} | -\mathbf{F} \end{bmatrix} \in \mathcal{R}_{q_{\mathbf{H}}}^{(n+n_{\mathbf{F}}) \times l_{\mathbf{m}}^{(\ell-1)}}.
\end{aligned}$$

And that

$$\begin{aligned}
\mathcal{L}_{\text{ring}, \text{snd}}^{(\ell-1, j)} &= \{(M, \mathbf{c}_{\ell-1}, \mathbf{c}_{\mathbf{v}_{\ell-1}}, \mathbf{c}_{\mathbf{v}_{\ell-2}}) \in \mathcal{R}_{\text{bin}}^{l_M} \times \mathcal{R}_q^{n+l_{\mathbf{m}}^{(\ell-1)}} \times (\mathcal{R}_{q_C}^{n_C})^2 : \\
\exists(\bar{\mathbf{c}}_{\ell-1}, \bar{\mathbf{m}}_{\ell-1}, \bar{\mathbf{e}}_{\ell-1}) &\in \overline{\text{Chall}} \times \mathcal{S}_{\overline{B}_{\mathbf{m}}}^{l_{\mathbf{m}}^{(\ell-1)}} \times \mathcal{S}_{\overline{B}_{\mathbf{e}}}^{l_{\mathbf{m}}^{(\ell-1)}} \text{ s.t.} \\
\begin{bmatrix} \mathbf{S} & \mathbf{I}_{l_{\mathbf{m}}^{(\ell-1)}} \end{bmatrix} (\bar{\mathbf{c}}_{\ell-1} \cdot \mathbf{c}_{\ell-1}) &= p \cdot \bar{\mathbf{e}}_{\ell-1} + \bar{\mathbf{m}}_{\ell-1} \bmod q, \\
\bar{\mathbf{c}}_{\ell-1} \cdot (\mathbf{c}_{\mathbf{v}_{\ell-1}} - \mathbf{C}_M \cdot M, \mathbf{c}_{\mathbf{v}_{\ell-2}}) &= \mathbf{M}_0^{(\ell-1)} \cdot \bar{\mathbf{m}}_{\ell-1} \bmod q_C, \\
\bar{\mathbf{c}}_{\ell-1} \cdot (-\mathbf{H}_{j, M} \cdot M, \mathbf{0}^{n_{\mathbf{F}}}, \mathbf{0}^{2n}) &= \mathbf{M}_0^{(\ell-1, j)} \bar{\mathbf{m}}_{\ell-1} \bmod q_{\mathbf{H}},
\end{aligned} \tag{43}$$

where $\overline{B}_{\mathbf{m}} = 2\alpha\sqrt{dl_{\max}}$, $\overline{B}_{\mathbf{e}} = 2dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\max}}$.

- The prover $\mathbf{P}$ and verifier $\mathbf{V}$ take as common input a CRS crs , a statement $(M, \mathbf{u}, \mathbf{c}_0, \dots, \mathbf{c}_{\ell-1}, \mathbf{c}_{\mathbf{v}_0}, \dots, \mathbf{c}_{\mathbf{v}_{\ell-1}}) \in \mathcal{R}_{\text{bin}}^{l_M} \times \mathcal{C}_{\beta}^{nk_{\mathbf{H}}} \times \mathcal{R}_q^{n+l_{\mathbf{m}}^{(0)}} \times (\mathcal{R}_q^{n+l_{\mathbf{m}}})^{\ell-2} \times \mathcal{R}_q^{n+l_{\mathbf{m}}^{(\ell-1)}} \times (\mathcal{R}_{q_C}^{n_C})^{\ell}$. The prover $\mathbf{P}$ takes as private input $((\mathbf{m}_i, \mathbf{r}_i))_{i \in [\ell]}$ as witness that $(M, \mathbf{u}, \mathbf{c}_0, \dots, \mathbf{c}_{\ell-1}, \mathbf{c}_{\mathbf{v}_0}, \dots, \mathbf{c}_{\mathbf{v}_{\ell-1}}) \in \mathcal{L}_{\text{ring}, \text{zk}}$. Then for $i \in [\ell]$, $\mathbf{P}$ runs the prover in the trapdoor Σ -protocol Π_i of Section C.1 for $\mathcal{L}_{\text{ring}, \text{zk}}^{(i)}$ as follows:
 - For $i = 0$, $\mathbf{P}$ runs the prover of the Σ -protocol Π_0 for $\mathcal{L}_{\text{ring}}^{(0)}$ (see (40), (41)), with input CRS crs , statement $(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0})$ and witness $(\mathbf{m}_0, \mathbf{r}_0)$;
 - For $\ell - 1 > i > 0$, $\mathbf{P}$ runs the prover of the Σ -protocol Π_i for $\mathcal{L}_{\text{ring}}^{(i)}$ (see (38), (39)), with input CRS crs , statement $(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}})$ and witness $(\mathbf{m}_i, \mathbf{r}_i)$;
 - For $i = \ell - 1$, $\mathbf{P}$ runs the prover of the Σ -protocol $\Pi_{\ell-1}$ for $\mathcal{L}_{\text{ring}}^{(\ell-1)}$ (see (42), (43)), with input CRS crs , statement $(M, \mathbf{c}_{\ell-1}, \mathbf{c}_{\mathbf{v}_{\ell-1}}, \mathbf{c}_{\mathbf{v}_{\ell-2}})$ and witness $(\mathbf{m}_{\ell-1}, \mathbf{r}_{\ell-1})$;

Let the prover's first message in Π_i be MSG_i . The prover P sends $(\text{MSG}_i)_{i \in [\ell]}$ to the verifier V .

Upon receiving V 's challenge $c \xleftarrow{\$} \text{Chall}$, P executes the provers of Π_i to obtain the corresponding response RSP_i . Let the final response be $(\text{RSP}_i)_{i \in [\ell]}$.

Receiving P 's response $(\text{RSP}_i)_{i \in [\ell]}$, V runs the verification algorithm of each protocol Π_i to check that $(\text{MSG}_i, c, \text{RSP}_i)$ is a valid transcript of Π_i . It outputs 1 if and only if all checks pass.

- $\text{TrapGen}(\text{par}, \mathcal{L})$ is identical to $\text{Gen}_{\mathcal{L}}$, the difference is that it retains the trapdoor $\tau = (\mathbf{S}, p)$ generated by ExtSetup , where $\mathbf{S} \in \mathcal{R}_q^{l_{\max} \times n}$. It sets the trapdoor $\tau_{\Sigma} = (\mathbf{S}, p)$.
- $\text{BadChallenge}_{\text{Ring}}(\tau_{\Sigma}, \text{crs}, \text{STM}, (\text{MSG}_i)_{i \in [\ell]})$ takes input a trapdoor $\tau_{\Sigma} = \mathbf{S} \in \mathcal{R}_q^{l_m \times n}$, a CRS $\text{crs} = (\text{par}, \text{crs}_{\mathcal{L}})$, a statement STM such that

$$\begin{aligned} \text{STM} &= (M, \mathbf{u}, \mathbf{c}_0, \dots, \mathbf{c}_{\ell-1}, \mathbf{c}_{\mathbf{v}_0}, \dots, \mathbf{c}_{\mathbf{v}_{\ell-1}}) \\ &\in \mathcal{R}_{\text{bin}}^{l_M} \times \mathcal{C}_{\beta}^{n_{kH}} \times \mathcal{R}_q^{n+l_m^{(0)}} \times (\mathcal{R}_q^{n+l_m})^{\ell-2} \times \mathcal{R}_q^{n+l_m^{(\ell-1)}} \times (\mathcal{R}_{qc}^{n_c})^{\ell} \end{aligned}$$

and a first message $(\text{MSG}_i)_{i \in [\ell]}$. It then proceeds as follows.

1. For $i \in [\ell]$, execute the $\text{BadChallenge}_{\text{OR}}$ algorithm underlying the trapdoor protocol Π^i for the language $(\mathcal{L}_{\text{set}, \text{zk}}^{(i)}, \mathcal{L}_{\text{set}, \text{snd}}^{(i)})$. Namely
 - When $i = 0$, execute the $\text{BadChallenge}_{\text{OR}}$ algorithm underlying the trapdoor protocol Π^0 on input the trapdoor τ_{Σ} , the string crs , the statement $(\mathbf{u}, \mathbf{c}_0, \mathbf{c}_{\mathbf{v}_0})$ and first message MSG_0 ;
 - When $\ell - 1 > i > 0$, execute the $\text{BadChallenge}_{\text{OR}}$ algorithm underlying the trapdoor protocol Π^i on input the trapdoor τ_{Σ} , the string crs , the statement $(\mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}})$ and first message MSG_i .
 - When $i = \ell - 1$, execute the $\text{BadChallenge}_{\text{OR}}$ algorithm underlying the trapdoor protocol $\Pi^{\ell-1}$ on input the trapdoor τ_{Σ} , the string crs , the statement $(M, \mathbf{c}_i, \mathbf{c}_{\mathbf{v}_i}, \mathbf{c}_{\mathbf{v}_{i-1}})$ and first message MSG_i ;
2. For $i \in [\ell]$, let the output of each execution above be c_i . If $c_0 = \dots = c_{\ell-1} := c \in \text{Chall}$, output c . Else, output $\perp$.

Similar to Proposition 2, the above Σ -protocol is a trapdoor Σ -protocol for language $\mathcal{L}_{\text{ring}}$ defined via (36) and (37). And the prover communication cost in a single execution has a bit size of order $\mathcal{O}(\ell \cdot \lambda^2 \log^2 \lambda)$.

We remark that in the above Σ -protocol, the language $\mathcal{L}_{\text{ring}, \text{zk}}$ (36) is well-defined *regardless of how the extractable commitment is instantiated*. Our ring signature scheme in Section 5 benefits greatly from this observation: setting up the extractable commitment in normal mode provides signers with statistical anonymity.

C.3 Parameter Selection

We now provide an explicit asymptotic parameter selection for the constructions above. The purpose is not to optimize concrete parameters, but to show that all parameter requirements used throughout the paper can be satisfied simultaneously. In particular, the assignment below satisfies the constraints of the basic

multi-moduli trapdoor Σ -protocol in (6), the ring-signature construction in (25), the OR protocol in (29), as well as the auxiliary hiding, extraction, invertibility, and norm conditions required by the underlying building blocks and security proofs.

Let $d = \Theta(\lambda)$ be a power of 2, and set $n = 2$, $n_{\mathbf{F}} = n_{\mathbf{C}} = 1$, $B = 2$, $\beta = 1$, and $B_{\text{Chall}} = 2$. Choose $d_{\text{Chall}} = \Theta(\log \lambda)$. Then

$$\#\text{Chall} = (B_{\text{Chall}} + 1)^{d_{\text{Chall}}} = 3^{\Theta(\log \lambda)} = \text{poly}(\lambda).$$

Choose prime moduli $q_{\mathbf{H}}, q_{\mathbf{C}} = \Theta(\lambda^5)$, with $q_{\mathbf{H}} \equiv 5 \pmod{8}$. Then $k_{\mathbf{H}} = \lceil \log_B q_{\mathbf{H}} \rceil = \Theta(\log \lambda)$. Set

$$l_{\mathbf{M}} = k_{\mathbf{H}}, \quad m_{\mathbf{F}} = 2 \lceil \log q_{\mathbf{H}} \rceil.$$

Hence $m_{\mathbf{F}} > n_{\mathbf{F}} \log q_{\mathbf{H}}$ and

$$m_{\mathbf{H}} = 2nk_{\mathbf{H}} = 2(n_{\mathbf{F}}k_{\mathbf{H}} + l_{\mathbf{M}}).$$

Moreover, $B_{\text{Chall}} < \sqrt{q_{\mathbf{H}}}$ for sufficiently large λ , as required for the invertibility argument (Lemma 5).

Choose

$$n_{\mathbf{R}} = c_{\mathbf{R}} \log \lambda, \quad B_{\mathbf{R}} = \Theta(\lambda),$$

where the constant $c_{\mathbf{R}} > 0$ is sufficiently large. Then

$$\sqrt{\left(1 + \frac{q_{\mathbf{C}}^{n_{\mathbf{C}}}}{(2B_{\mathbf{R}} + 1)^{n_{\mathbf{R}}}}\right)^d} - 1 \in \text{negl}(\lambda),$$

so the statistical hiding requirement of the KTX commitment is satisfied. With these choices,

$$l_{\mathbf{m}}, l_{\mathbf{m}}^{(0)}, l_{\mathbf{m}}^{(\ell-1)}, l_{\max} = \Theta(\log \lambda),$$

and, from the definition of $B_{\mathbf{m}}$,

$$B_{\mathbf{m}} = O(\lambda^{3/2} \sqrt{\log \lambda}).$$

In particular, $B_{\mathbf{m}} = o(q_{\mathbf{H}})$ and $B_{\mathbf{m}} = o(q_{\mathbf{C}})$.

For the extractable commitment, choose

$$l_{\mathbf{r}} = c_{\mathbf{r}} \log \lambda, \quad B_{\mathbf{r}} = \Theta(\lambda), \quad B_{\tau} = \Theta(\lambda),$$

where $c_{\mathbf{r}} > 0$ is sufficiently large. Choose primes $p = \Theta(\lambda^5)$ and $q = \Theta(\lambda^{12})$ with $p < q$. Then

$$pdl_{\mathbf{r}}B_{\tau}B_{\mathbf{r}} + B_{\mathbf{m}} = O(\lambda^8 \log \lambda) = o(q)$$

and $B_{\mathbf{m}} = o(p)$, satisfying the correctness/extractability conditions of the commitment scheme. By choosing $c_{\mathbf{r}}$ sufficiently large, we also have

$$\sqrt{\left(1 + \frac{q^{n+l_{\max}}}{(2B_{\mathbf{r}} + 1)^{l_{\mathbf{r}}}}\right)^d} - 1 \in \text{negl}(\lambda),$$

which gives statistical hiding in normal mode.

For soundness amplification, choose $\kappa = \Theta(\lambda/\log \lambda)$ and $\gamma = \Theta(1)$. Then the expected number of attempts in the rejection-sampling procedure is

$$M = \exp\left(\frac{14}{\gamma} + \frac{1}{2\gamma^2}\right) = O(1).$$

For the κ parallel executions, set the Gaussian width to

$$\alpha = B_{\text{Chall}} \sqrt{d\kappa} \sqrt{B_{\mathbf{m}}^2 + dl_{\mathbf{r}} B_{\mathbf{r}}^2} \gamma = O(\lambda^{5/2}).$$

Since $\#\text{Chall} = \text{poly}(\lambda)$, this choice of κ reduces the soundness error to $2^{-\Omega(\lambda)}$. Furthermore,

$$\alpha \sqrt{dl_{\max}}, \alpha \sqrt{dl_{\mathbf{r}}} = O(\lambda^3 \sqrt{\log \lambda}) = o(q_{\mathbf{H}}), o(q_{\mathbf{C}}), o(q).$$

Thus, the first two inequalities in (6), (25), and (29) are satisfied.

We also have

$$2\alpha \sqrt{dl_{\max}} = O(\lambda^3 \sqrt{\log \lambda}) = o(p),$$

and

$$p \cdot 2dl_{\mathbf{r}} B_{\tau} \alpha \sqrt{dl_{\mathbf{r}} l_{\max}} + 2\alpha \sqrt{dl_{\max}} = O(\lambda^{10} \log^2 \lambda) = o(q).$$

Hence the remaining extraction constraints in (6), (25), and (29) are also satisfied.

Finally, the norm bounds appearing in the unforgeability reduction satisfy

$$\beta_0 = 4\sqrt{2}dB_{\text{Chall}}\alpha \sqrt{dl_{\max}} = O(\lambda^4 \sqrt{\log \lambda}) = o(q_{\mathbf{H}}), o(q_{\mathbf{C}}),$$

and

$$\beta_1 = 2\alpha \sqrt{dl_{\max}} + dB_{\text{Chall}} = O(\lambda^3 \sqrt{\log \lambda}) = o(q_{\mathbf{H}}).$$

Therefore, this parameter assignment satisfies all the required parameter conditions of the construction, including (6), (25), and (29).

C.4 Deferred Security Proofs of Our Ring Signature

We prove Theorem 3 via Theorem 5 and Theorem 6.

Theorem 5 (Anonymity). *Assume that the parameters are chosen such that*

$$\begin{aligned} \sqrt{(1 + q^{n+l_{\max}}/(2B_{\mathbf{r}} + 1)^{l_{\mathbf{r}}})^d - 1} &\in \text{negl}(\lambda); \\ \sqrt{(1 + q_{\mathbf{C}}^{n_{\mathbf{C}}}/(2B_{\mathbf{R}} + 1)^{n_{\mathbf{R}}})^d - 1} &\in \text{negl}(\lambda), \end{aligned}$$

and the NIZK argument system for $\mathcal{L}_{\text{ring}} = (\mathcal{L}_{\text{ring,zk}}, \mathcal{L}_{\text{ring,snd}})$ is statistically zero-knowledge. Then the ring signature of Section 5.2 is statistically anonymous under full key exposure.

Proof. We prove the above theorem via a sequence of games. The first game is the experiment defining anonymity per Definition 6. The last game is designed so that a (possibly unbounded) adversary $\mathcal{A}$ wins with advantage exactly 0. We show that the games are indistinguishable.

Game 0: This is the experiment defining anonymity in Definition 6, where $b \in \{0, 1\}$ is a uniformly random bit chosen by the challenger, w.l.o.g. we can assume that $b \in \{0, 1\}$ is chosen at the start of the game, and generates all keys $\{pk_i = \mathbf{d}_i, sk_i = \mathbf{x}_i\}_{i \in [N]}$ by executing **KeyGen**. The adversary $\mathcal{A}$ is given all public keys and access to a signing oracle. In the challenge phase, it returns a message M^* , two user indices j_0, j_1 , and a ring R^* containing the public keys $\mathbf{d}_{j_0}, \mathbf{d}_{j_1}$ of users j_0 and j_1 . The challenger then picks a uniformly random bit b and generates a challenge ring signature Σ^* using the secret key of user j_b . We remark that by (26), the signature Σ^* has the form

$$\Sigma^* = (\mathbf{u}^*, (\mathbf{c}_i^*)_{i \in [\ell]}, \dots, (\mathbf{c}_{\mathbf{v}_i}^*)_{i \in [\ell]}, \pi^*),$$

where $\mathbf{u}^* = \text{TAcc}(R^*)$ and $R^* = \{(\mathbf{d}^*, M) : \mathbf{d}^* \in R^*\}$. The adversary is then given a challenge signature Σ^* and the randomnesses $\{r_i\}_{i \in [N]}$ used by **KeyGen** for generating pk_i , and try to guess the bit b chosen by the challenger. Define W_0 the event that $\mathcal{A}$ outputs $b' = b$ at the end of the experiment, the advantage $\text{Adv}_0 := |\Pr[W_0] - 1/2|$ is the advantage of $\mathcal{A}$ against the anonymity of the ring signature.

Game 1: In this game, we make a change of how the challenge signature is computed. During the setup phase, the challenger retains the lossy secret key sk of the lossy encryption scheme LPKE, and uses sk to simulate the NIZK proof π^* in Σ^* . The statistical zero-knowledge property of the NIZK system (follows from Theorem 4 and from the statistical special zero-knowledge property of the trapdoor Σ -protocol of Section C.1), implies that $\mathcal{A}$'s view is affected up to a negligible amount and $|\Pr[W_1] - \Pr[W_0]| \leq 2^{-\Omega(\lambda)}$.

Game 2: In this game, we make a change of how the challenge signature is computed. Note that in **Game-1**, we have that $\mathbf{c}_{\mathbf{v}_{\ell-1}}^*, \mathbf{c}_{\mathbf{v}_{\ell-2}}^*, \dots, \mathbf{c}_{\mathbf{v}_0}^*$ are KTX commitments to the nodes on the path from $(\mathbf{d}_{j_b}, M)$ to the accumulated root $\mathbf{u}^*$. In this game, we sample the commitments $\mathbf{c}_{\mathbf{v}_{\ell-1}}^*, \mathbf{c}_{\mathbf{v}_{\ell-2}}^*, \dots, \mathbf{c}_{\mathbf{v}_0}^*$ independently, uniformly random from $\mathcal{R}_{qc}^{nc}$. By Lemma 2, the condition

$$\sqrt{(1 + q_C^{nc} / (2B_R + 1)^{n_R})^d - 1} \in \text{negl}(\lambda)$$

implies that the distributions of $\mathbf{c}_{\mathbf{v}_{\ell-1}}^*, \mathbf{c}_{\mathbf{v}_{\ell-2}}^*, \dots, \mathbf{c}_{\mathbf{v}_0}^*$ in **Game-1** are statistically close to uniform and thus $|\Pr[W_2] - \Pr[W_1]| = \text{negl}(\lambda)$.

Game 3: In this game, we make a change of how the challenge signature is computed. In this game, we sample the commitments $\mathbf{c}_0^*, \mathbf{c}_1^*, \dots, \mathbf{c}_{\ell-1}^*$ independently, uniformly random over their respective space. Since the parameters of

the commitment scheme of Section A.4 are generated in normal mode and that $\sqrt{(1 + q^{n+l_{\max}}/(2B_{\mathbf{r}} + 1)^{l_{\mathbf{r}}})^d - 1} \in \text{negl}(\lambda)$, the commitments $\mathbf{c}_0^*, \mathbf{c}_1^*, \dots, \mathbf{c}_{\ell-1}^*$ in **Game-2** are distributed statistically close to uniform and thus we have that $|\Pr[W_3] - \Pr[W_2]| = \text{negl}(\lambda)$.

Now, $\Pr[W_3] = 1/2$ since the challenge signature Σ^* is generated independently of the challenger's bit $b \in \{0, 1\}$. In particular, the commitments $\mathbf{c}_{\mathbf{v}_i}^*$ and $\mathbf{c}_i^*$ are uniformly random vectors, and π^* is a simulated proof. $\square$

Theorem 6 (Unforgeability). *Assume that $q_{\mathbf{H}} = 5 \bmod 8$, $B_{\text{Chall}} < \sqrt{q_{\mathbf{H}}}$ and there exists a prime modulus $p < q$ and an ℓ_{∞} -norm bound B_{τ} such that $p \cdot 2dl_{\mathbf{r}}B_{\tau}\alpha\sqrt{dl_{\mathbf{r}}l_{\max}} + 2\alpha\sqrt{dl_{\max}} < q/2$ and $2\alpha\sqrt{dl_{\max}} < p/2$. Furthermore, assume that:*

- The NIZK system is sound;
- $\text{MLWE}_{l_{\max}, l_{\mathbf{r}}, \chi}$ assumption holds, where χ is uniform over $\mathcal{C}_{B_{\tau}}$;
- $\text{MSIS}_{n_{\mathbf{c}}, nk_{\mathbf{H}} + n_{\mathbf{R}}, q_{\mathbf{C}}, \beta_0}^{(2)}$; $\text{MSIS}_{n, m_{\mathbf{H}}, q_{\mathbf{H}}, \beta_0}^{(2)}$ and $\text{MSIS}_{n_{\mathbf{F}}, m_{\mathbf{F}}, q_{\mathbf{H}}, \beta_1}^{(\infty)}$ assumptions hold, where $\beta_0 = 4\sqrt{2dB_{\text{Chall}}}\alpha\sqrt{dl_{\max}}$ and $\beta_1 = 2\alpha\sqrt{dl_{\max}} + dB_{\text{Chall}}$.

Then the ring signature scheme of Section 5.2 is unforgeable against insider corruption.

The proof of Theorem 6 makes use of the following lemma from [40].

Lemma 6 (Adapted from [40, Lemma 8]). *For any matrix $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ and a randomly chosen $\mathbf{x} \xleftarrow{\$} \mathcal{R}_{\text{bin}}^m$, the probability that there exists another $\mathbf{x}' \in \mathcal{R}_{\text{bin}}^m$ such that $\mathbf{x} \neq \mathbf{x}'$ and that $\mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x}' \bmod q$ is at least $1 - 2^{dn \log q - dm}$.*

Message binding. We first make explicit how the message is bound to the accumulated ring. For a ring $R = \{\mathbf{d}_0, \dots, \mathbf{d}_{N-1}\}$ and a message M , the verifier forms

$$R_M = \{(\mathbf{d}_0, M), \dots, (\mathbf{d}_{N-1}, M)\}$$

and recomputes $\text{TAcc}(R_M)$. Hence, a valid signature $\Sigma = (\mathbf{u}, \dots)$ on (M, R) satisfies $\mathbf{u} = \text{TAcc}(R_M)$. If the same signature is used for a different message $M' \neq M$ with the same ring, the verifier instead forms

$$R_{M'} = \{(\mathbf{d}_0, M'), \dots, (\mathbf{d}_{N-1}, M')\}$$

and checks $\mathbf{u} = \text{TAcc}(R_{M'})$. Since $R_M \neq R_{M'}$, passing both root checks would give $\text{TAcc}(R_M) = \text{TAcc}(R_{M'})$, contradicting the security of the accumulator except with negligible probability. This argument is used at the beginning of the unforgeability proof below: for an accepted forgery (M^*, R^*, Σ^*) , the verifier's root check implies $\mathbf{u}^* = \text{TAcc}(R^*)$, where $R^* = \{(\mathbf{d}_0^*, M^*), \dots, (\mathbf{d}_{N-1}^*, M^*)\}$.

For readability, we outline the reduction strategy in the proof of Theorem 6 as follows. Assume for the moment that the NIZK is sound for the language $\mathcal{L}_{\text{ring}, \text{zk}}$ (36) (i.e, the NIZK provides *exact* soundness), and that we can extract a

witness from any valid proof. Consider an adversary $\mathcal{A}$ producing a valid forgery (M^*, R^*, Σ^*) , where $\Sigma^* = (\mathbf{u}^*, (\mathbf{c}_i^*)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i}^*)_{i \in [\ell]}, \pi^*)$ and $R^* = \{\mathbf{d}_0^*, \dots, \mathbf{d}_{N-1}^*\}$ only contains uncorrupted keys. Note that we have $\mathbf{u}^* = \text{TAcc}(R^{*\prime})$, where $R^{*\prime} = \{(\mathbf{d}_0^*, M^*), \dots, (\mathbf{d}_{N-1}^*, M^*)\}$.

For $i \in [\ell]$, consider the language $\mathcal{L}_{\text{ring}, \text{zk}}^{(i)}$ defining the hash chain relation at depth i of the Merkle tree in the accumulator, then

- At depth $i = 0$, from the statement $(\mathbf{u}^*, \mathbf{c}_0^*, \mathbf{c}_{\mathbf{v}_0}^*)$ and the proof π^* , we can extract a path node $\mathbf{v}_0^* \in \mathcal{C}_\beta^{nk_H}$, a node $\mathbf{w}_0^* \in \mathcal{C}_\beta^{nk_H}$ and a path index $j_0^* \in \{0, 1\}$ showing that: (i) $\mathbf{v}_0^*$ and $\mathbf{w}_0^*$ is correctly hashed into $\mathbf{u}^*$; (ii) that the KTX commitment $\mathbf{c}_{\mathbf{v}_0}^*$ opens to $\mathbf{v}_0^*$.
- At depth $i = 1$, from the statement $(\mathbf{c}_1^*, \mathbf{c}_{\mathbf{v}_1}^*, \mathbf{c}_{\mathbf{v}_0}^*)$, we can extract a path node $\mathbf{v}_1^* \in \mathcal{C}_\beta^{nk_H}$, a node $\mathbf{w}_1^* \in \mathcal{C}_\beta^{nk_H}$, a path node $\mathbf{v}_0^{*\prime} \in \mathcal{C}_\beta^{nk_H}$, and a path index $j_1^* \in \{0, 1\}$ showing that: (i) $\mathbf{v}_1^*$ and $\mathbf{w}_1^*$ is correctly hashed into $\mathbf{v}_0^{*\prime}$; (iii) that the KTX commitment $\mathbf{c}_{\mathbf{v}_1}^*$ opens to $\mathbf{v}_1^*$; (ii) that the KTX commitment $\mathbf{c}_{\mathbf{v}_0}^*$ opens to $\mathbf{v}_0^{*\prime}$. Note that $\mathbf{v}_0^* = \mathbf{v}_0^{*\prime}$, otherwise we break the binding property of the KTX commitment.
- At depth $i = 2$, from the statement $(\mathbf{c}_2^*, \mathbf{c}_{\mathbf{v}_2}^*, \mathbf{c}_{\mathbf{v}_1}^*)$ and the proof π^* , we can extract a path node $\mathbf{v}_2^* \in \mathcal{C}_\beta^{nk_H}$, a node $\mathbf{w}_2^* \in \mathcal{C}_\beta^{nk_H}$, a path node $\mathbf{v}_1^{*\prime} \in \mathcal{C}_\beta^{nk_H}$, and a path index $j_2^* \in \{0, 1\}$ showing that: (i) $\mathbf{v}_2^*$ and $\mathbf{w}_2^*$ is correctly hashed into $\mathbf{v}_1^{*\prime}$; (ii) that the KTX commitment $\mathbf{c}_{\mathbf{v}_2}^*$ opens to $\mathbf{v}_2^*$; (iii) that the KTX commitment $\mathbf{c}_{\mathbf{v}_1}^*$ opens to $\mathbf{v}_1^{*\prime}$. Similar to when $i = 1$, we have that $\mathbf{v}_1^* = \mathbf{v}_1^{*\prime}$.
- ...
- At depth $i = \ell - 1$, from the statement $(M^*, \mathbf{c}_{\ell-1}^*, \mathbf{c}_{\mathbf{v}_{\ell-1}}^*, \mathbf{c}_{\mathbf{v}_{\ell-2}}^*)$ and the proof π^* , we can extract a public key $\mathbf{d}^* \in \mathcal{C}_\beta^{n_{\mathbf{F}}^{k_H}}$, a node $\mathbf{w}_{\ell-1}^* \in \mathcal{C}_\beta^{nk_H}$, a node $\mathbf{v}_{\ell-2}^{*\prime} \in \mathcal{C}_\beta^{nk_H}$, a path index $j_{\ell-1}^* \in \{0, 1\}$ and a vector $\mathbf{x}^* \in \mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}}$ showing that: (i) $(\mathbf{d}^*, M^*)$ and $\mathbf{w}_{\ell-1}^*$ is correctly hashed into $\mathbf{v}_{\ell-1}^{*\prime}$; (ii) that the KTX commitment $\mathbf{c}_{\mathbf{v}_{\ell-1}}^*$ opens to $(\mathbf{d}^*, M^*)$; (iii) that the KTX commitment $\mathbf{c}_{\mathbf{v}_{\ell-2}}^*$ opens to $\mathbf{v}_{\ell-2}^{*\prime}$; (iv) that $\mathbf{G}_B \mathbf{d} = \mathbf{F} \mathbf{x}^* \bmod q_H$. Similar to when $i = 1$, we have that $\mathbf{v}_{\ell-2}^* = \mathbf{v}_{\ell-2}^{*\prime}$.

Essentially, we recover a path witness $(j_{\ell-1}^*, \dots, j_0^*, \mathbf{w}_{\ell-1}^*, \dots, \mathbf{w}_0^*)$ showing that $(\mathbf{d}^*, M^*)$ is accumulated in $\mathbf{u}^*$. Note that, if $\mathbf{d}^* \notin R^*$ then we break the security of the accumulator. Therefore, it must be the case that $\mathbf{d}^* \in R^*$ and thus $\mathbf{d}^*$ is an uncorrupted public key, for which we can retrieve a corresponding secret key $\mathbf{x}' \in \mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}}$. Lemma 6 implies that with noticeable probability, there are at least two secret keys corresponding to $\mathbf{d}^*$. Combining with the zero-knowledge property of the NIZK, we should have that $\mathbf{x}' \neq \mathbf{x}^*$ with noticeable probability, which in turn gives a non-trivial solution to an MSIS instance defined by $\mathbf{F}$.

Note that the generic NIZK from trapdoor Σ -protocol in Section B is not a proof-of-knowledge in general and thus it is not clear how to extract a witness

given a valid NIZK proof π^* for the relation $\mathcal{L}_{\text{ring}, \text{zk}}$. In the proof of Theorem 6, this is done via the extraction mode of the extractable commitment of Section A.4. Due to the relaxed soundness, we can only extract some witness showing that the statement of a forgery belongs to the language $\mathcal{L}_{\text{ring}, \text{snd}}$ (37). Thus, rather than reducing a forger to one that breaks the security of the building blocks (KTX commitment, accumulator, lattice-based one-way function), we reduce the forger to one that *directly* solves some MSIS instances.

Proof (Theorem 6). Consider a PPT algorithm $\mathcal{A}$ playing the unforgeability game with respect to insider corruption (Definition 5). We show that if $\mathcal{A}$ wins the game with a non-negligible probability ε , then $\mathcal{A}$ can either break the soundness of the proof system, or be converted into an MSIS solver.

The reduction $\mathcal{B}$ sets up the system faithfully except for the matrices $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $\mathbf{B} \in \mathcal{R}_q^{l_{\max} \times m}$ in the public parameters of the commitment scheme of Section A.4. Instead of generating $\mathbf{A}$ and $\mathbf{B}$ uniformly at random, $\mathcal{B}$ generates $\mathbf{A}$ and $\mathbf{B}$ in extraction mode by sampling $\mathbf{S} \xleftarrow{\$} \mathcal{R}_q^{l_{\max} \times n}$, $\mathbf{E} \xleftarrow{\$} \mathcal{C}_{B_r}^{l_{\max} \times m}$ and set $\mathbf{B} = \mathbf{S}\mathbf{A} + p\mathbf{E} \bmod q$ and retains the extraction trapdoor $(\mathbf{S}, p)$. As we assume that $\text{MLWE}_{l_{\max}, m, \chi}$ is hard against PPT adversaries, $\mathcal{A}$'s view cannot change by a non-negligible quantity. The reduction $\mathcal{B}$ also retains the underlying chosen secret key $\mathbf{x} \in \mathcal{R}_{\text{bin}}^{m_F}$ after $\mathcal{A}$'s query to PKGen. Since $\mathcal{B}$ knows all secret signing keys, $\mathcal{B}$ can answer all corruption queries as well as signing queries.

By assumption, $\mathcal{A}$ outputs forgery (M^*, R^*, Σ^*) with probability at least $\varepsilon - \text{negl}(\lambda)$, where

$$R^* = \{\mathbf{d}_0^*, \dots, \mathbf{d}_{N-1}^*\},$$

and

$$\Sigma^* = (\mathbf{u}^*, (\mathbf{c}_i^*)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i}^*)_{i \in [\ell]}, \pi^*),$$

such that R^* only contains uncorrupted keys, and $\text{RVerify}(M^*, R^*, \Sigma^*) = 1$. Note that this implies $\mathbf{u}^* = \text{TAcc}(R^{*\prime})$, where $R^{*\prime} = \{(\mathbf{d}_0^*, M^*) : i \in [N]\}$; and that π^* is a valid NIZK proof for $\text{STM} = (M^*, \mathbf{u}^*, (\mathbf{c}_i^*)_{i \in [\ell]}, (\mathbf{c}_{\mathbf{v}_i}^*)_{i \in [\ell]})$. By the soundness of the NIZK system, we have that $\text{STM} \in \mathcal{L}_{\text{ring}, \text{snd}}$ (37). This means there exist challenge differences $(\bar{c}_0, \dots, \bar{c}_{\ell-1}) \in \overline{\text{Chall}}^\ell$ and witness $(\bar{\mathbf{m}}_i, \bar{\mathbf{e}}_i)_{i \in [\ell]}$ such that:

- At intermediate depths $0 < i < \ell - 1$, $(\mathbf{c}_i^*, \mathbf{c}_{\mathbf{v}_i}^*, \mathbf{c}_{\mathbf{v}_{i-1}}^*) \in \cup_{j \in \{0,1\}} \mathcal{L}_{\text{ring}, \text{snd}}^{(i,j)}$ (39), or equivalently

$$[\mathbf{S}[: l_m] \quad \mathbf{I}_{l_m}] (\bar{c}_i \cdot \mathbf{c}_i^*) = p \cdot \bar{\mathbf{e}}_i + \bar{\mathbf{m}}_i \bmod q, \quad (44)$$

$$\bar{c}_i \cdot (\mathbf{c}_{\mathbf{v}_i}^*, \mathbf{c}_{\mathbf{v}_{i-1}}^*) = \mathbf{M}_0 \cdot \bar{\mathbf{m}}_i \bmod q_C \quad (45)$$

$$\mathbf{0} = \mathbf{M}_1^{(j_i)} \cdot \bar{\mathbf{m}}_i \bmod q_H \quad \text{for some } j_i \in \{0, 1\} \quad (46)$$

where $\bar{\mathbf{m}}_i \in \mathcal{S}_{B_m}^{l_m}$, $\bar{\mathbf{e}}_i \in \mathcal{S}_{B_e}^{l_m}$, and

$$\mathbf{M}_0 = \begin{bmatrix} \mathbf{C} \mathbf{D} & \mathbf{0}^{n_C \times n_{k_H}} \\ \mathbf{C} \mathbf{D} & \mathbf{0}^{n_C \times n_{k_H}} \end{bmatrix} \in \mathcal{R}_{q_C}^{2n_C + l_m},$$

$$\mathbf{M}_0^{(j)} = [\mathbf{H}_j | \mathbf{0}^{n \times n_R} | -\mathbf{G}_B | \mathbf{0}^{n \times n_R} | \mathbf{H}_{1-j}] \in \mathcal{R}_{q_H}^{n \times l_m},$$

and $\overline{B}_{\mathbf{m}} = 2\alpha\sqrt{dl_{\max}}$, $\overline{B}_{\mathbf{e}} = 2dl_{\mathbf{r}}B_r\alpha\sqrt{dl_{\mathbf{r}}l_{\max}}$.

In this case, the reduction $\mathcal{B}$ proceeds as follows: for all possible $\bar{c}_i \in \overline{\text{Chall}}$ (note that $\#\text{Chall} = (B_{\text{Chall}} + 1)^{d_{\text{Chall}}} = \text{poly}(\lambda)$), iterate through all possible $\bar{c}_i$ of $\overline{\text{Chall}}$ to compute $\overline{\mathbf{m}}_i = ([\mathbf{S}[:l_{\mathbf{m}}] \quad \mathbf{I}_{l_{\mathbf{m}}}] (\bar{c}_i \cdot \mathbf{c}_i^*) \bmod q) \bmod p$, and test whether $\|\overline{\mathbf{m}}_i\| \leq \overline{B}_{\mathbf{m}}$, $\bar{c}_i \cdot (\mathbf{c}_{\mathbf{v}_i}^*, \mathbf{c}_{\mathbf{v}_{i-1}}^*) = \mathbf{M}_0 \cdot \overline{\mathbf{m}}_i \bmod q_{\mathcal{C}}$ and $\mathbf{0} = \mathbf{M}_1^{(j_i)} \cdot \overline{\mathbf{m}}_i \bmod q_{\mathcal{H}}$ for some $j_i \in \{0, 1\}$. Since $p\overline{B}_{\mathbf{e}} + \overline{B}_{\mathbf{m}} < q/2$ and that $\overline{B}_{\mathbf{m}} < p/2$, $\mathcal{B}$ can recover some $(\bar{c}_i^*, \overline{\mathbf{m}}_i^*, j_i^*) \in \overline{\text{Chall}} \times \mathcal{S}_{\overline{B}_{\mathbf{m}}}^{l_{\mathbf{m}}} \times \{0, 1\}$ satisfying (44), (45) and (46). Parsing $\overline{\mathbf{m}}_i^*$ as

$$\overline{\mathbf{m}}_i^* = (\mathbf{v}_i^*(\overline{\mathbf{m}}_i^*), \mathbf{r}_{\mathbf{v}_i}^*(\overline{\mathbf{m}}_i^*), \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_i^*), \mathbf{r}_{\mathbf{v}_{i-1}}^*(\overline{\mathbf{m}}_i^*), \mathbf{w}_i^*(\overline{\mathbf{m}}_i^*)),$$

where $\mathbf{v}_i^*(\overline{\mathbf{m}}_i^*), \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_i^*) \in \mathcal{R}^{nk_{\mathcal{H}}}$; $\mathbf{r}_{\mathbf{v}_i}^*(\overline{\mathbf{m}}_i^*), \mathbf{r}_{\mathbf{v}_{i-1}}^*(\overline{\mathbf{m}}_i^*) \in \mathcal{R}^{n_{\mathcal{R}}}$ and $\mathbf{w}_i^*(\overline{\mathbf{m}}_i^*) \in \mathcal{R}^{nk_{\mathcal{H}}}$. Then (45) and (46) imply that

$$\bar{c}_i^* \cdot \mathbf{c}_{\mathbf{v}_i}^* = \mathbf{C} \cdot \mathbf{v}_i^*(\overline{\mathbf{m}}_i^*) + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_i}^*(\overline{\mathbf{m}}_i^*) \bmod q_{\mathcal{C}} \quad (47)$$

$$\bar{c}_i^* \cdot \mathbf{c}_{\mathbf{v}_{i-1}}^* = \mathbf{C} \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_i^*) + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_{i-1}}^*(\overline{\mathbf{m}}_i^*) \bmod q_{\mathcal{C}} \quad (48)$$

$$\mathbf{G}_B \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_i^*) = \mathbf{H}_{j_i^*} \mathbf{v}_i^*(\overline{\mathbf{m}}_i^*) + \mathbf{H}_{1-j_i^*} \mathbf{w}_i^*(\overline{\mathbf{m}}_i^*) \bmod q_{\mathcal{H}} \quad (49)$$

– At depth $i = 0$, $(\mathbf{u}^*, \mathbf{c}_0^*, \mathbf{c}_{\mathbf{v}_0}^*) \in \cup_{j \in \{0, 1\}} \mathcal{L}_{\text{ring, snd}}^{(0, j)}$ (41), or equivalently

$$[\mathbf{S}[:l_{\mathbf{m}}^{(0)}] \quad \mathbf{I}_{l_{\mathbf{m}}^{(0)}}] (\bar{c}_0 \cdot \mathbf{c}_0^*) = p \cdot \bar{\mathbf{e}}_0 + \overline{\mathbf{m}}_0 \bmod q, \quad (50)$$

$$\bar{c}_0 \cdot \mathbf{c}_{\mathbf{v}_0}^* = [\mathbf{C}|\mathbf{D}|\mathbf{0}^{n_{\mathcal{C}} \times nk_{\mathcal{H}}}] \cdot \overline{\mathbf{m}}_0 \bmod q_{\mathcal{C}}, \quad (51)$$

$$\bar{c}_0 \cdot \mathbf{G}_B \mathbf{u}^* = [\mathbf{H}_j|\mathbf{0}^{n \times n_{\mathcal{R}}|\mathbf{H}_{1-j}}] \cdot \overline{\mathbf{m}}_0 \bmod q_{\mathcal{H}} \text{ for some } j_0 \in \{0, 1\}. \quad (52)$$

Similar to the case $0 < i < \ell - 1$, the reduction $\mathcal{B}$ can use $\mathbf{S}$ to recover $(\bar{c}_0^*, \overline{\mathbf{m}}_0^*, j_0^*) \in \overline{\text{Chall}} \times \mathcal{S}_{\overline{B}_{\mathbf{m}}}^{l_{\mathbf{m}}^{(0)}} \times \{0, 1\}$ satisfying (50), (51) and (52). By parsing

$$\overline{\mathbf{m}}_0^* = (\mathbf{v}_0^*(\overline{\mathbf{m}}_0^*), \mathbf{r}_{\mathbf{v}_0}^*(\overline{\mathbf{m}}_0^*), \mathbf{w}_0^*(\overline{\mathbf{m}}_0^*)),$$

where $\mathbf{v}_0^*(\overline{\mathbf{m}}_0^*) \in \mathcal{R}^{nk_{\mathcal{H}}}$; $\mathbf{r}_{\mathbf{v}_0}^*(\overline{\mathbf{m}}_0^*) \in \mathcal{R}^{n_{\mathcal{R}}}$ and $\mathbf{w}_0^*(\overline{\mathbf{m}}_0^*) \in \mathcal{R}^{nk_{\mathcal{H}}}$. Then (51) and (52) imply that

$$\bar{c}_0^* \cdot \mathbf{c}_{\mathbf{v}_0}^* = \mathbf{C} \cdot \mathbf{v}_0^*(\overline{\mathbf{m}}_0^*) + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_0}^*(\overline{\mathbf{m}}_0^*) \bmod q_{\mathcal{C}} \quad (53)$$

$$\bar{c}_0^* \cdot \mathbf{G}_B \mathbf{u}^* = \mathbf{H}_{j_0^*} \mathbf{v}_0^*(\overline{\mathbf{m}}_0^*) + \mathbf{H}_{1-j_0^*} \mathbf{w}_0^*(\overline{\mathbf{m}}_0^*) \bmod q_{\mathcal{H}} \quad (54)$$

– At depth $i = \ell - 1$, $(M^*, \mathbf{c}_{\ell-1}^*, \mathbf{c}_{\mathbf{v}_{\ell-1}}^*, \mathbf{c}_{\mathbf{v}_{\ell-2}}^*) \in \cup_{j \in \{0, 1\}} \mathcal{L}_{\text{ring, snd}}^{(\ell-1, j)}$ (43), or equivalently

$$[\mathbf{S} \quad \mathbf{I}_{l_{\mathbf{m}}^{(\ell-1)}}] (\bar{c}_{\ell-1} \cdot \mathbf{c}_{\ell-1}^*) = p \cdot \bar{\mathbf{e}}_{\ell-1} + \overline{\mathbf{m}}_{\ell-1} \bmod q \quad (55)$$

$$\bar{c}_{\ell-1} \cdot (\mathbf{c}_{\mathbf{v}_{\ell-1}}^* - \mathbf{C}_M \cdot M^*, \mathbf{c}_{\mathbf{v}_{\ell-2}}^*) = \mathbf{M}_0^{(\ell-1)} \cdot \overline{\mathbf{m}}_{\ell-1} \bmod q_{\mathcal{C}} \quad (56)$$

$$(-\mathbf{H}_{j_{\ell-1}, M} \cdot M^*, \mathbf{0}^{n_{\mathcal{F}}}) = \mathbf{M}_0^{(j)} \overline{\mathbf{m}}_{\ell-1} \bmod q_{\mathcal{H}} \text{ for some } j_{\ell-1} \in \{0, 1\}, \quad (57)$$

where $[\mathbf{C}_0|\mathbf{C}_M] = \mathbf{C}$, $[\overline{\mathbf{H}}_0|\mathbf{H}_{0,M}] = \mathbf{H}_0$, $[\overline{\mathbf{H}}_1|\mathbf{H}_{1,M}] = \mathbf{H}_1$ and

$$\mathbf{M}_0^{(\ell-1)} = \begin{bmatrix} \mathbf{C}_0 & \mathbf{D} & \mathbf{0}^{n_C \times (nk_H + m_F)} \\ \mathbf{C} & \mathbf{D} & \mathbf{0}^{n_C \times (nk_H + m_F)} \end{bmatrix} \in \mathcal{R}_{q_C}^{2n_C \times l_m^{(\ell-1)}},$$

$$\mathbf{M}_0^{(\ell-1,j)} = \begin{bmatrix} \overline{\mathbf{H}}_j | \mathbf{0}^{n \times n_R} | -\mathbf{G}_B | \mathbf{0}^{n \times n_R} | \mathbf{H}_{1-j} | \mathbf{0}^{n \times m_F} \\ \mathbf{G}_B | \mathbf{0}^{n \times n_R} | \mathbf{0}^{n \times nk_H} | \mathbf{0}^{n \times n_R} | \mathbf{0}^{n \times nk_H} | -\mathbf{F} \end{bmatrix} \in \mathcal{R}_{q_H}^{(n+n_F) \times l_m^{(\ell-1)}}.$$

Similar to the case $0 < i < \ell - 1$, the reduction $\mathcal{B}$ can use $\mathbf{S}$ to recover $(\overline{c}_{\ell-1}^*, \overline{\mathbf{m}}_{\ell-1}^*, j_{\ell-1}^*) \in \overline{\text{Chall}} \times \mathcal{S}_{\overline{\mathbf{B}}_m}^{l_m^{\ell-1}} \times \{0, 1\}$ satisfying (55), (56) and (57). By parsing

$$\overline{\mathbf{m}}_{\ell-1}^* = (\overline{\mathbf{d}}^*, \mathbf{r}_{\mathbf{v}_{\ell-1}}^*(\overline{\mathbf{m}}_{\ell-1}^*), \mathbf{v}_{\ell-2}^*(\overline{\mathbf{m}}_{\ell-1}^*), \mathbf{r}_{\mathbf{v}_{\ell-2}}^*(\overline{\mathbf{m}}_{\ell-1}^*), \mathbf{w}_{\ell-1}^*(\overline{\mathbf{m}}_{\ell-1}^*), \overline{\mathbf{x}}^*),$$

where $\overline{\mathbf{d}}^* \in \mathcal{R}^{n_F k_H}$, $\mathbf{v}_{\ell-2}^*(\overline{\mathbf{m}}_{\ell-1}^*) \in \mathcal{R}^{nk_H}$; $\mathbf{r}_{\mathbf{v}_{\ell-1}}^*(\overline{\mathbf{m}}_{\ell-1}^*), \mathbf{r}_{\mathbf{v}_{\ell-2}}^*(\overline{\mathbf{m}}_{\ell-1}^*) \in \mathcal{R}^{n_R}$, $\mathbf{w}_{\ell-1}^*(\overline{\mathbf{m}}_{\ell-1}^*) \in \mathcal{R}^{nk_H}$ and $\overline{\mathbf{x}}^* \in \mathcal{R}^{m_F}$. Then (56) and (57) imply that

$$\overline{c}_{\ell-1}^* \cdot \mathbf{c}_{\mathbf{v}_{\ell-1}}^* = \mathbf{C} \cdot (\overline{\mathbf{d}}^*, \overline{c}_{\ell-1}^* \cdot M^*) + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_{\ell-1}}^*(\overline{\mathbf{m}}_{\ell-1}^*) \bmod q_C, \quad (58)$$

$$\overline{c}_{\ell-1}^* \cdot \mathbf{c}_{\mathbf{v}_{\ell-2}}^* = \mathbf{C} \cdot \mathbf{v}_{\ell-2}^*(\overline{\mathbf{m}}_{\ell-1}^*) + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_{\ell-2}}^*(\overline{\mathbf{m}}_{\ell-1}^*) \bmod q_C, \quad (59)$$

$$\mathbf{G}_B \mathbf{v}_{\ell-2}^*(\overline{\mathbf{m}}_{\ell-1}^*) = \mathbf{H}_{j_{\ell-1}^*}(\overline{\mathbf{d}}^*, \overline{c}_{\ell-1}^* \cdot M^*) + \mathbf{H}_{1-j_{\ell-1}^*} \mathbf{w}_{\ell-1}^*(\overline{\mathbf{m}}_{\ell-1}^*) \bmod q_H, \quad (60)$$

$$\mathbf{F} \cdot \overline{\mathbf{x}}^* = \mathbf{G}_B \cdot \overline{\mathbf{d}}^* \bmod q_H. \quad (61)$$

Observe that (48), (47) and (59) imply that

$$\overline{c}_i \cdot \mathbf{c}_{\mathbf{v}_{i-1}}^* = \mathbf{C} \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_i^*) + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_{i-1}}^*(\overline{\mathbf{m}}_i^*) \bmod q_C,$$

$$\overline{c}_{i-1} \cdot \mathbf{c}_{\mathbf{v}_{i-1}}^* = \mathbf{C} \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_{i-1}^*) + \mathbf{D} \cdot \mathbf{r}_{\mathbf{v}_{i-1}}^*(\overline{\mathbf{m}}_{i-1}^*) \bmod q_C;$$

for all $i \in [\ell]$ and $i > 0$. If there is some i such that $\overline{c}_i \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_{i-1}^*) \neq \overline{c}_{i-1} \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_i^*)$, we obtain that

$$\mathbf{C} \cdot \mathbf{s}_C^* - \mathbf{D} \cdot \mathbf{s}_D^* = \mathbf{0}^{n_C} \bmod q_C.$$

where $\mathbf{s}_C^* = \overline{c}_i^* \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_{i-1}^*) - \overline{c}_{i-1}^* \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_i^*)$ and $\mathbf{s}_D^* = \overline{c}_i^* \cdot \mathbf{r}_{\mathbf{v}_{i-1}}^*(\overline{\mathbf{m}}_{i-1}^*) - \overline{c}_{i-1}^* \cdot \mathbf{r}_{\mathbf{v}_{i-1}}^*(\overline{\mathbf{m}}_i^*)$. In this case we obtain a non-trivial MSIS solution $(\mathbf{s}_C^*, \mathbf{s}_D^*)$ of $[\mathbf{C}|\mathbf{D}] \in \mathcal{R}_{q_C}^{n_C \times (nk_H + n_R)}$, such that

$$\begin{aligned} \|(\mathbf{s}_C^*, \mathbf{s}_D^*)\| &= \sqrt{\|\mathbf{s}_C^*\|^2 + \|\mathbf{s}_D^*\|^2} \\ &\leq \sqrt{4d^2 B_{\text{Chall}}^2 \overline{B}_m^2 + 4d^2 B_{\text{Chall}}^2 \overline{B}_m^2} \\ &= 2\sqrt{2}d B_{\text{Chall}} \overline{B}_m = 4\sqrt{2}d B_{\text{Chall}} \alpha \sqrt{dl_{\max}} = \beta_0. \end{aligned}$$

Otherwise, we have that for all $i \in [\ell]$ and $i > 0$

$$\overline{c}_i^* \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_{i-1}^*) = \overline{c}_{i-1}^* \cdot \mathbf{v}_{i-1}^*(\overline{\mathbf{m}}_i^*). \quad (62)$$

Let $\mathbf{v}_{\ell-1}^*(\mathbf{m}_{\ell-1}^*) = (\bar{\mathbf{d}}^*, \bar{c}_{\ell-1}^* \cdot M^*)$. From (49), (54) and (60)

$$\begin{aligned}\mathbf{G}_B \mathbf{v}_{i-1}^*(\bar{\mathbf{m}}_i^*) &= \mathbf{H}_{j_i^*} \cdot \mathbf{v}_i^*(\bar{\mathbf{m}}_i^*) + \mathbf{H}_{1-j_i^*} \cdot \mathbf{w}_i^*(\bar{\mathbf{m}}_i^*) \bmod q_H, \forall i \in [\ell] \setminus \{0\}; \\ \bar{c}_0^* \cdot \mathbf{G}_B \mathbf{u}^* &= \mathbf{H}_{j_0^*} \cdot \mathbf{v}_0^*(\bar{\mathbf{m}}_0^*) + \mathbf{H}_{1-j_0^*} \cdot \mathbf{w}_0^*(\bar{\mathbf{m}}_0^*) \bmod q_H.\end{aligned}$$

Combining with (62), we obtain

$$\begin{aligned}\bar{c}_i^* \mathbf{G}_B \mathbf{v}_{i-1}^*(\bar{\mathbf{m}}_{i-1}^*) &= \mathbf{H}_{j_i^*} \cdot \bar{c}_{i-1}^* \mathbf{v}_i^*(\bar{\mathbf{m}}_i^*) + \mathbf{H}_{1-j_i^*} \cdot \bar{c}_{i-1}^* \mathbf{w}_i^*(\bar{\mathbf{m}}_i^*) \bmod q_H, \forall i \in [\ell] \setminus \{0\}; \\ \bar{c}_0^* \cdot \mathbf{G}_B \mathbf{u}^* &= \mathbf{H}_{j_0^*} \cdot \mathbf{v}_0^*(\bar{\mathbf{m}}_0^*) + \mathbf{H}_{1-j_0^*} \cdot \mathbf{w}_0^*(\bar{\mathbf{m}}_0^*) \bmod q_H.\end{aligned}$$

Let $\bar{c}_{-1}^* = 1$ and $\mathbf{v}_{-1}^*(\bar{\mathbf{m}}_{-1}^*) = \mathbf{u}^*$, the above can be rewritten as

$$\bar{c}_i^* \mathbf{G}_B \mathbf{v}_{i-1}^*(\bar{\mathbf{m}}_{i-1}^*) = \mathbf{H}_{j_i^*} \cdot \bar{c}_{i-1}^* \mathbf{v}_i^*(\bar{\mathbf{m}}_i^*) + \mathbf{H}_{1-j_i^*} \cdot \bar{c}_{i-1}^* \mathbf{w}_i^*(\bar{\mathbf{m}}_i^*) \bmod q_H, \forall i \in [\ell].$$

Recall that $\mathbf{u}^* = \text{TAcc}_H(R^{*'})$, where $R^{*'} = \{(\mathbf{d}_i^*, M^*) : i \in [N]\}$. By tracing the path $(j_0^*, \dots, j_{\ell-1}^*)$ of the Merkle tree in the accumulator, we can find $\mathbf{v}_{\ell-1}^*, \mathbf{w}_{\ell-1}^*, \dots, \mathbf{v}_0^*, \mathbf{w}_0^* \in \mathcal{C}_\beta^{nk_H}$ such that

$$\mathbf{G}_B \mathbf{v}_{i-1}^* = \mathbf{H}_{j_i^*} \cdot \mathbf{v}_i^* + \mathbf{H}_{1-j_i^*} \cdot \mathbf{w}_i^* \bmod q_H, \forall i \in [\ell];$$

where $\mathbf{v}_{-1}^* = \mathbf{u}^*$ and $\mathbf{v}_{\ell-1}^* \in R^{*'}$. Define $S = \{i \in [\ell] : \bar{c}_i^* \mathbf{v}_i^* \neq \mathbf{v}_i^*(\bar{\mathbf{m}}_i^*)\}$. We consider two cases:

– *S is nonempty*: let $i' = \min S \geq 0$. By definition of S we have that $\bar{c}_{i'-1}^* \mathbf{v}_{i'-1}^* = \mathbf{v}_{i'-1}^*(\bar{\mathbf{m}}_{i'-1}^*)$. Then by the equations

$$\begin{aligned}\bar{c}_{i'}^* \mathbf{G}_B \mathbf{v}_{i'-1}^*(\bar{\mathbf{m}}_{i'-1}^*) &= \mathbf{H}_{j_{i'}^*} \cdot \bar{c}_{i'-1}^* \mathbf{v}_{i'}^*(\bar{\mathbf{m}}_{i'}^*) + \mathbf{H}_{1-j_{i'}^*} \cdot \bar{c}_{i'-1}^* \mathbf{w}_{i'}^*(\bar{\mathbf{m}}_{i'}^*) \bmod q_H, \\ \mathbf{G}_B \mathbf{v}_{i'-1}^* &= \mathbf{H}_{j_{i'}^*} \cdot \mathbf{v}_{i'}^* + \mathbf{H}_{1-j_{i'}^*} \cdot \mathbf{w}_{i'}^* \bmod q_H,\end{aligned}$$

we have that

$$\begin{aligned}\bar{c}_{i'}^* \mathbf{G}_B \mathbf{v}_{i'-1}^*(\bar{\mathbf{m}}_{i'-1}^*) &= \mathbf{H}_{j_{i'}^*} \cdot \bar{c}_{i'-1}^* \mathbf{v}_{i'}^*(\bar{\mathbf{m}}_{i'}^*) + \mathbf{H}_{1-j_{i'}^*} \cdot \bar{c}_{i'-1}^* \mathbf{w}_{i'}^*(\bar{\mathbf{m}}_{i'}^*) \bmod q_H, \\ \bar{c}_{i'}^* \mathbf{G}_B \mathbf{v}_{i'-1}^*(\bar{\mathbf{m}}_{i'-1}^*) &= \mathbf{H}_{j_{i'}^*} \cdot (\bar{c}_{i'}^* \bar{c}_{i'-1}^*) \mathbf{v}_{i'}^* + \mathbf{H}_{1-j_{i'}^*} \cdot (\bar{c}_{i'}^* \bar{c}_{i'-1}^*) \mathbf{w}_{i'}^* \bmod q_H,\end{aligned}$$

If $d = 1$, then $\mathcal{R} = \mathbb{Z}$ and since $0 < \|\bar{c}_{i'-1}^*\|_\infty \leq B_{\text{Chall}} < \sqrt{q_H}$, we have that $\bar{c}_{i'-1}^*$ is invertible modulo q_H as q_H is a prime. If $d > 1$, then the conditions $q_H = 5 \bmod 8$ and $0 < \|\bar{c}_{i'-1}^*\|_\infty \leq B_{\text{Chall}} < \sqrt{q_H}$, implies that $\bar{c}_{i'-1}^*$ is invertible modulo q_H (Lemma 5). Therefore

$$\mathbf{H}_{j_{i'}^*} \cdot (\mathbf{v}_{i'}^*(\bar{\mathbf{m}}_{i'}^*) - \bar{c}_{i'}^* \mathbf{v}_{i'}^*) + \mathbf{H}_{1-j_{i'}^*} \cdot (\mathbf{w}_{i'}^*(\bar{\mathbf{m}}_{i'}^*) - \bar{c}_{i'}^* \mathbf{w}_{i'}^*) = \mathbf{0}^n \bmod q_H.$$

By assumption $\mathbf{v}_{i'}^*(\bar{\mathbf{m}}_{i'}^*) - \bar{c}_{i'}^* \mathbf{v}_{i'}^* \neq \mathbf{0}^{nk_H}$ and we obtain a non-trivial MSIS solution of $\mathbf{H} \in \mathcal{R}_{q_H}^{n \times m_H}$, with ℓ_2 -norm also upper bounded by $2\sqrt{d}B_{\text{Chall}}\bar{B}_{\mathbf{m}} = \beta_0$.

– *S is empty*: In this case we have that $\bar{c}_{\ell-1}^* \mathbf{v}_{\ell-1}^* = \mathbf{v}_{\ell-1}^*(\bar{\mathbf{m}}_{\ell-1}^*)$. Recall that $\mathbf{v}_{\ell-1}^*(\mathbf{m}_{\ell-1}^*) = (\bar{\mathbf{d}}^*, \bar{c}_{\ell-1}^* \cdot M^*)$ and $\mathbf{v}_{\ell-1}^* \in \{(\mathbf{d}_i^*, M^*) : i \in [N]\}$. Therefore

there exists a public key $\mathbf{d}^* \in R^*$ that $\bar{\mathbf{d}}^* = \bar{c}_{\ell-1}^* \cdot \mathbf{d}^*$. Since the ring R^* in the forgery only consists of uncorrupted keys, we can retrieve a secret key $\mathbf{x}^* \in \mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}}$ such that $\mathbf{F}\mathbf{x}^* = \mathbf{G}_B \cdot \mathbf{d}^* \bmod q_H$. Combine this with (61), we obtain

$$\mathbf{F}(\bar{c}_{\ell-1}^* \mathbf{x}^* - \bar{\mathbf{x}}^*) = \mathbf{0}^n \bmod q_H.$$

Condition on the existence of such an $\mathbf{x}'$. Let $\mathcal{X}_{\mathbf{d}^*} = \{\mathbf{x} \in \mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}} : \mathbf{F}\mathbf{x} = \mathbf{G}_B \mathbf{d}^* \bmod q_H\}$ be the set of valid secret keys corresponding to $\mathbf{d}^*$. Since the signing key is sampled uniformly from $\mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}}$, conditioned on the public key $\mathbf{d}^*$ it is uniformly distributed over $\mathcal{X}_{\mathbf{d}^*}$; moreover, the existence of $\mathbf{x}' \neq \mathbf{x}^*$ implies that $|\mathcal{X}_{\mathbf{d}^*}| \geq 2$.

Recall that at the beginning of the reduction, the commitment of Appendix A.4 is switched from normal mode to extraction mode. We argue that the adversary's view is computationally indistinguishable regardless of which secret key in $\mathcal{X}_{\mathbf{d}^*}$ is used to answer signing queries. Consider the following sequence of hybrids. First switch the commitment parameters from extraction mode to normal mode, while continuing to answer signing queries using the same secret key. By Proposition 3, this changes the adversary's view by at most a negligible amount. In normal mode, the commitment is statistically hiding. Moreover, by Theorem 4, the NIZK obtained from the compiler is computationally zero-knowledge. Thus, we may first replace the proofs in the signing queries by simulated proofs, then replace the commitments to one valid secret key by commitments to another valid secret key in $\mathcal{X}_{\mathbf{d}^*}$, and finally replace the simulated proofs by real proofs for the new secret key; these changes alter the adversary's view by at most a negligible amount. Finally, switch the commitment parameters back from normal mode to extraction mode, which again changes the adversary's view by at most a negligible amount by Proposition 3. Hence, in extraction mode, the adversary's view, and therefore the distribution of the extracted $\bar{\mathbf{x}}^*$, is computationally indistinguishable regardless of which secret key in $\mathcal{X}_{\mathbf{d}^*}$ is used to answer the signing queries.

Since $\bar{c}_{\ell-1}^*$ is invertible modulo q_H , for any fixed extracted $\bar{\mathbf{x}}^*$, the equality $\bar{\mathbf{x}}^*/\bar{c}_{\ell-1}^* = \mathbf{x}$ can hold for at most one $\mathbf{x} \in \mathcal{X}_{\mathbf{d}^*}$. Therefore, $\bar{\mathbf{x}}^*/\bar{c}_{\ell-1}^* = \mathbf{x}^*$ with probability at most $1/2 + \text{negl}(\lambda)$, and hence $\bar{\mathbf{x}}^*/\bar{c}_{\ell-1}^* \neq \mathbf{x}^*$ with probability at least $1/2 - \text{negl}(\lambda)$. Taking into account the probability of existence of $\mathbf{x}'$, we obtain a non-trivial MSIS solution with probability at least $(1 - 2^{dn \log q_H - dm_{\mathbf{F}}})(1/2 - \text{negl}(\lambda))$, which is noticeable. $\square$

C.5 Asymptotic Efficiency Analysis

We analyze the asymptotic efficiency of the ring signature in Section 5, in terms of key sizes and signature sizes. A user's public key is an element of $\mathcal{C}_{\beta}^{n_{\mathbf{F}} k_H}$, which has bit size $dn_{\mathbf{F}} k_H \log B = \mathcal{O}(\lambda \log \lambda)$. A secret key is a short vector of $\mathcal{R}_{\text{bin}}^{m_{\mathbf{F}}}$, which has bit size of order $dm_{\mathbf{F}} = \mathcal{O}(\lambda \log \lambda)$. By (26), a ring signature Σ consists of an accumulated value $\mathbf{u} \in \mathcal{C}_{\beta}^{n_{\mathbf{F}} k_H}$, ℓ extractable commitments $(\mathbf{c}_i)_{i \in [\ell]} \subset \mathcal{R}_q^{n+L_{\max}}$ and ℓ KTX commitments $(\mathbf{c}_i)_{i \in [\ell]} \subset \mathcal{R}_{q^c}^{n_c}$. All of those together have a total bit size of order $\mathcal{O}(\ell \cdot \lambda \log^2 \lambda)$. For the NIZK proof π in

Σ , observe that in a single execution of the trapdoor Σ -protocol for $\mathcal{L}_{\text{ring}}$ (Section C.2), the prover's first message has bit size $\mathcal{O}(\ell \cdot \lambda \log^2 \lambda)$. From the NIZK compiler in Section B, the most dominant part of π is the randomness of the lossy encryption for encrypting the prover's first message (viewed as a bit string). Using a lattice-based lossy encryption scheme such as [52], this requires randomness of bit length at most $\mathcal{O}(\ell \cdot \lambda \log^3 \lambda)$. Since the trapdoor Σ -protocol for $\mathcal{L}_{\text{ring}}$ is run in parallel $\kappa = \Theta(\lambda / \log \lambda)$ times, the bit size of final signature is of order $\mathcal{O}(\ell \cdot \lambda^2 \log^2 \lambda) = \mathcal{O}(\log R \cdot \lambda^2 \log^2 \lambda)$.

Comparison with known lattice-based standard model ring signatures.

We explain the asymptotic analysis in Table 1.

- In the ring signature of [10], for signing a message of bit length l , a user's public key consists of $2l + 1$ SIS matrix $\{\mathbf{A}, \{\mathbf{A}_j^{(i)}\}_{j \in \{0,1\}, i \in [l]}\} \subset \mathbb{Z}_q^{n \times m}$ and secret key is a trapdoor matrix $\mathbf{S} \in \mathbb{Z}^{m \times m}$ for Gaussian sampling over any coset of the q -ary lattice defined by $\mathbf{A}$. Therefore the public key bit size is of order at least $\Omega(\lambda^2 \log^2 \lambda)$ and the secret key bit size is of order $\mathcal{O}(\lambda^2 \log^2 \lambda)$. A signature consists of $(l+1) \cdot R$ short vectors, each of dimension $m = \lambda \log \lambda$. This gives a signature of bit size $\Omega(R \cdot \lambda \log \lambda)$.
- In the ring signature of [17], user's secret key and public key contains a verification key of a digital signature, and an encryption key of a PKE which is used for encrypting the verification key. This results in a bit size of order $\Omega(\lambda^2)$ of both user's secret and public keys. A ring signature in the construction of [17] contains a ZAP in which there is a witness-extractable commitment consisting of $\mathcal{O}(\log R \cdot \lambda)$ FHE ciphertexts. The ZAP further needs to be repeated $\Omega(\lambda)$ times to achieve soundness error $2^{-\Omega(\lambda)}$, yielding a signature size of order $\Omega(\log R \cdot \lambda^4)$.