One-Shot Fiat-Shamir-based NIZK Arguments of Composite Residuosity and Logarithmic-Size Ring Signatures in the Standard Model

Benoît Libert[1,2], Khoa Nguyen[3] *, Thomas Peters[4], and Moti Yung[5]

[1] CNRS, Laboratoire LIP, France
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL), France
[3] Institute of Cybersecurity and Cryptology, School of Computing and Information Technology, University of Wollongong, Australia
[4] FNRS and UCLouvain (ICTEAM), Belgium
[5] Google and Columbia University, USA

Abstract. The standard model security of the Fiat-Shamir transform has been an active research area for many years. In breakthrough results, Canetti *et al.* (STOC'19) and Peikert-Shiehian (Crypto'19) showed that, under the Learning-With-Errors (LWE) assumption, it provides soundness by applying correlation-intractable (CI) hash functions to so-called *trapdoor Σ*-protocols. In order to be compatible with CI hash functions based on standard LWE assumptions with polynomial approximation factors, all known such protocols have been obtained via parallel repetitions of a basic protocol with binary challenges. In this paper, we consider languages related to Paillier's composite residuosity assumption (DCR) for which we give the first trapdoor Σ-protocols providing soundness in one shot, via exponentially large challenge spaces. This improvement is analogous to the one enabled by Schnorr over the original Fiat-Shamir protocol in the random oracle model. Using the correlation-intractable hash function paradigm, we then obtain simulation-sound NIZK arguments showing that an element of $\mathbb{Z}_{N^2}^*$ is a composite residue, which opens the door to space-efficient applications in the standard model. As a concrete example, we build logarithmic-size ring signatures (assuming a common reference string) with the shortest signature length among schemes based on standard assumptions in the standard model. We prove security under the DCR and LWE assumptions, while keeping the signature size comparable with that of random-oracle-based schemes.

Keywords. NIZK arguments, compactness, simulation-soundness, composite residuosity, Fiat-Shamir, ring signatures, standard model.

1 Introduction

The Fiat-Shamir transform [40] is a famous technique that collapses interactive protocols into non-interactive proof systems by computing the verifier's challenges as hash values of the transcript so far. Since its introduction, it enabled

* This work was done while this author was at Nanyang Technological University, SPMS, Singapore.

a wide range of applications in the random oracle model (ROM) although it may fail to preserve soundness in general [43]. In the standard model, it was not known to be safely instantiable under standard assumptions until recently. The beautiful work of Canetti *et al.* [15] and Peikert and Shiehian [67] changed this state-of-affairs by showing the existence of Fiat-Shamir-based non-interactive zero-knowledge (NIZK) proofs for all NP languages under the Learning-With-Errors (LWE) assumption [68]. Their results followed the methodology of *correlation intractable* (CI) hash functions [17], which can sometimes emulate the properties of random oracles in the standard model.

In short, correlation intractability for a relation R requires the infeasibility of finding x such that $(x, H_k(x)) \in R$ given a random hashing key k. This property provides soundness because, with high probability, it prevents a cheating prover's first message from being hashed into a challenge admitting a valid response. Canetti *et al.* [18] formalized this intuition by observing that it suffices to build CI hash functions for efficiently searchable relations as long as Fiat-Shamir is applied to *trapdoor* Σ-protocols. These are like standard Σ-protocols with two differences. First, they assume a common reference string (CRS). Second, there exists an efficiently computable function $\mathsf{BadChallenge}$ that inputs a trapdoor τ_Σ together with a false statement $x \notin \mathcal{L}$ and a first prover message a in order to compute the only challenge Chall such that an accepting transcript (a, Chall, z) exists for some response z. If $\mathsf{BadChallenge}$ is efficiently computable, soundness can be achieved using CI hash functions for any efficiently computable relation, which covers the case of the relation R such that $(x, y) \in R$ if and only if $y = \mathsf{BadChallenge}(\tau_\Sigma, x, a)$.

While the results of [15,67] resolve the long-standing problem of realizing NIZK proofs for all NP under standard lattice assumptions, they raise the natural open question of whether LWE-based correlation-intractable hash functions can lead to compact proofs/arguments for specific languages like subgroup membership. In this paper, we consider this problem for Paillier's composite residuosity assumption [64] for which we obtain NIZK arguments that are roughly as short as those obtained from the Fiat-Shamir heuristic in the ROM. In particular, we aim at trapdoor Σ-protocols that ensure soundness in one shot, without going through $\Theta(\lambda)$ parallel repetitions to achieve negligible soundness error.

Our Contribution. We provide space-efficient NIZK arguments showing that an element is a composite residue in the group $\mathbb{Z}_{N^2}^*$, for an RSA modulus $N = pq$. In particular, we can argue that Paillier [64] or Damgård-Jurik [34] ciphertexts decrypt to 0. These arguments extend to handle multiplicative relations between Paillier ciphertexts. We achieve this by showing that several natural Σ-protocols for Paillier-related languages can be extended into trapdoor Σ-protocols with an exponentially large challenge space, which achieve negligible soundness error in a single protocol execution. To our knowledge, we thus obtain the first trapdoor Σ-protocols that guarantee soundness without parallel repetitions.

Our constructions provide multi-theorem statistical NIZK and their soundness can be proved under the Learning-With-Errors (LWE) assumption. In addition, we show how to upgrade them into unbounded simulation-sound NIZK arguments based on the LWE and DCR assumptions. In their single-theorem version, our arguments of composite residuosity are as short as their random-oracle-based counterpart obtained from the Fiat-Shamir heuristic. Their multi-theorem and simulation-sound extensions are only longer by a small constant factor. In particular, we can turn any trapdoor Σ-protocol into an unbounded simulation-sound NIZK argument for the same language while only lengthening the transcript by the size of a Paillier ciphertext and its randomness.

As a main application, we obtain logarithmic-size ring signatures with concretely efficient signature length in the standard model. Recall that ring signatures allow a signer to sign messages while hiding in an *ad hoc* set of users called a *ring*. To this end, the signer only needs to know the public keys of all ring members and its own secret key. So far, the only known logarithmic-size realizations in the standard model under standard assumptions [3,24] incur very large signatures due to the use of witness indistinguishable proofs for NP. In contrast, we obtain fairly short signatures comprised of a small number of Paillier ciphertexts while retaining security under well-studied assumptions. For rings of $R = 2^r$ users, each signature fits within the equivalent of $15r + 7$ RSA moduli, which is only 3 times as large as in a Fiat-Shamir-like construction in the random oracle model under the DCR assumption. The unforgeability of our scheme is proved under the DCR and LWE assumptions while its anonymity holds for unbounded adversaries.

To our knowledge, our NIZK arguments for DCR-related languages give the first examples where, under standard assumptions, Fiat-Shamir-based arguments in the standard model can be almost as short as those in the random oracle model. We believe they can find many other applications than ring signatures. For example, they easily extend to prove multiplicative relations among Paillier ciphertexts, which is a common task in MPC [30] or voting protocols [34]. The trapdoor Σ-protocol of our DCR-based ring signature can also be used in other applications of compact 1-out-of-R proofs [46,45].

Technical Overview. Ciampi *et al.* [27] recently showed that any Σ-protocol can be turned into a trapdoor Σ-protocol with small (i.e., binary) challenge space, which requires many repetitions to achieve negligible soundness error. In order to obtain an exponentially large challenge space in one shot, we rely on earlier an observation by Chaidos and Groth [21] who noticed that a certain family of encryption schemes with linearly homomorphic properties over their message *and* randomness spaces admit a trapdoor Σ-protocol for the language $\mathcal{L}^0 = \{x \mid \exists w \in \mathcal{R} : x = \mathcal{E}_{pk}(0; w)\}$ of encryptions of 0. At a high level, if the prover's first message is an encryption $a = \mathcal{E}_{pk}(0; r)$ of 0 and the verifier sends a challenge Chall, the response $z = r + \mathsf{Chall} \cdot w$ satisfies $a \cdot x^{\mathsf{Chall}} = \mathcal{E}_{pk}(0; z)$. If $x \notin \mathcal{L}^0$, the special soundness property ensures that, for any given a, there is at most one Chall such that $a \cdot x^{\mathsf{Chall}} = \mathcal{E}_{pk}(0; z)$ for some $z \in \mathcal{R}$. Moreover, the secret key sk can serve as a trapdoor τ_Σ to compute $\mathsf{BadChallenge}(\tau_\Sigma, x, a)$ for any element a of the ciphertext space. Indeed, if Chall lives a polynomial-size set (say $\{0,1\}^{\log \lambda}$), the bad challenge is efficiently computable by outputting the first $\mathsf{Chall} \in \{0,1\}^{\log \lambda}$ for which $\mathcal{D}_{sk}(a \cdot x^{\mathsf{Chall}}) = 0$. The above construction thus decreases the number of parallel repetitions by a factor $O(\log \lambda)$. Using the Okamoto-Uchiyama cryptosystem [63], Chaidos and Groth [21] apply the above technique to identify bad challenges within an exponentially large challenge space. A follow-up work by Lipmaa [58] shows that, although the plaintext space of Paillier's cryptosystem [64] has non-prime order $N = pq$, bad challenges are still computable using the factorization of N as long as the challenge space is contained in $\{0, \ldots, \min(p, q) - 1\}$. We actually identify a subtlety in [58], which adapts the Chaidos-Groth technique [21] to build designated verifier NIZK proofs that an Elgamal-Paillier ciphertext [13] encrypts 0. The proof of soundness of [58, Theorem 2] implicitly constructs a trapdoor Σ-protocol showing that $(C_0, C_1) = (g^r \bmod N^2, (1 + N)^b \cdot h^r \bmod N^2)$ encrypts $b = 0$. We actually show that, for false statements, the extractor may fail to extract the bad challenge when a maliciously generated first prover message is outside the range of the encryption algorithm. Our trapdoor Σ-protocol for DCR proceeds like the extractor of [58, Theorem 2] but avoids this problem as it only relies on the Paillier/Damgård-Jurik encryption scheme, which has the property that all elements of the ciphertext space encrypt something.

In order to obtain a multi-theorem NIZK argument of composite residuosity, we can then apply the construction of [56, Appendix B], which compiles any trapdoor Σ-protocol into a NIZK argument for the same language using a lossy encryption scheme with equivocable lossy mode. As considered [73,4], lossy encryption is a primitive where ciphertexts encrypted under lossy public keys – which are computationally indistinguishable from injective ones – statistically hide the underlying plaintexts. Moreover, the equivocation property (a.k.a. "efficient opening" [4]) makes it possible to trapdoor open any lossy ciphertext exactly as in a trapdoor commitment. It is known [48] that Paillier's cryptosystem [64] provides these properties under the DCR assumption.

However, in the context of the signature-of-knowledge paradigm [23], we need NIZK arguments with unbounded simulation-soundness [35]. Libert *et al.* [56] showed that any trapdoor Σ-protocol can be turned into an USS argument for the same language using a generalization of the $\mathcal{R}$-lossy encryption primitive introduced by Boyle *et al.* [9]. In [56], they introduced two distinct equivocation properties and gave a candidate based on the LWE assumption. In order to optimize the signature length, we give an efficient equivocable $\mathcal{R}$-lossy encryption candidate under the DCR assumption. This task is non-trivial since injective keys have to be indistinguishable from lossy keys, even when one of the equivocation trapdoors is given. Yet, our candidate only uses the DCR assumption while [56] used fairly powerful tools (i.e., lattice trapdoors [41]) to equivocate lossy ciphertexts. Although our DCR-based realization satisfies slightly weaker properties than those of [56], we prove it sufficient to obtain simulation-soundness. It thus allows compiling trapdoor Σ-protocols into unbounded simulation-sound NIZK arguments without using lattice trapdoors.

Armed with a DCR-based construction of USS arguments, we then build a simulation-sound NIZK argument that one-out-of-many elements of $\mathbb{Z}_{N^2}^*$ is a composite residue. To this end, we provide a DCR-based variant of the Groth-

Kohlweiss (GK) [46] Σ-protocol, which allows proving that one out of R commitments contains 0 with communication cost $O(\log R)$. The reason why DCR is the most promising assumption towards trapdooring [46] is that, in its original version, the GK protocol cannot immediately be turned into a trapdoor Σ-protocol by applying the transformation of Ciampi *et al.* [27]. The main difficulty is that it only yields $(r+1)$-special-soundness for $r = O(\log R)$, so that up to r bad challenges may exist for a false statement and a given first prover message. Even if $\mathsf{BadChallenge}$ can identify them all for a given protocol iteration, over κ repetitions, we end up with up to r^κ combinations, which are not enumerable in polynomial time for non-constant κ and r.[6] In order to apply the LWE-based CI hash function of [67], we construct a variant of GK with an exponentially large challenge space and where $\mathsf{BadChallenge}$ can efficiently enumerate all bad challenges after a single protocol iteration. We achieve this by extending our trapdoor Σ-protocol showing composite residuosity, using a $\mathsf{BadChallenge}$ function that computes the roots of a degree-r (instead of a degree-1) polynomial.

Adapting [46] to Paillier-based commitments raises several difficulties if we want to apply it in the context of ring signatures. In our security proofs, we need the Σ-protocol to be statistically honest-verifier zero-knowledge. In the protocol of [46] and our DCR-based variant, this requires that users' public keys be computed as statistically hiding commitments to 0. A first idea is to apply Paillier, where ciphertexts $C = g^m \cdot r^N \bmod N^2$ are perfectly hiding commitments when g is an N-th residue (and extractable commitments when N divides the order of g). Unfortunately, as shown in [57, Section 2.6], using a statistically hiding commitment is not sufficient to ensure statistical anonymity when the adversary can introduce maliciously generated public keys in the ring. In the case of Paillier, when g is an N-th residue, so is any honestly generated commitment. However, in the anonymity game, the adversary can choose a ring containing malformed public keys that are *not* N-th residues in $\mathbb{Z}_{N^2}^*$. This affects the statistical ZK property since the simulator cannot fully randomize commitments by multiplying them with a random commitment to 0. To address this issue, we need a statistically hiding commitment which is "dense" in that commitments to 0 are uniformly distributed over $\mathbb{Z}_{N^2}^*$. In order to obtain trapdoor Σ-protocols, we also need the commitment to be dual-mode as the $\mathsf{BadChallenge}$ function should be able to efficiently extract committed messages in the perfectly binding setting. We thus use commitments (suggested in [20] for their online/offline property) of the form $C = (1+N)^m \cdot h^y \cdot w^N \bmod N^2$, for randomness (y, w), which are perfectly binding if h is an N-th residue and perfectly hiding if N divides the order of h. Moreover, the latter configuration provides dense statistically hiding commitments since commitments to 0 are uniformly distributed over $\mathbb{Z}_{N^2}^*$.

A second difficulty arises when we adapt the proof of unforgeability of the

[6] Holmgren *et al.* [51] recently gave a technique allowing to address the combinatorial explosion of bad challenges induced by parallel repetitions. In Supplementary Material G.2, we discuss the applicability of their approach to our setting. Although it allows instantiations under the DDH assumption, these are considerably more expensive that our DCR-based candidate.

Groth-Kohlweiss ring signature, which relies on the extractability property of their Σ-protocol. They apply the forking lemma to extract an opening of a perfectly hiding commitment by replaying the adversary $O(r)$ times. In the standard model, our reduction does not have the degree of freedom of replaying the adversary with a different random oracle. Instead, we proceed with a sequence of hybrid games that exploits the dual-mode property of our DCR-based commitment and moves to a setting where the signer's identity is only computationally hidden. In one game, the commitment is switched to its extractable mode so as to extract the committed bits $\ell_1^\star \dots \ell_r^\star \in \{0,1\}^r$ of the signer's position $\ell^\star$ in the ring. In the next game, the reduction guesses which honestly generated public key $vk^{(i^\star)}$ will be in the ring position $\ell^\star$ and fails if this guess is incorrect. Finally, we modify the key generation oracle and replace the expected target user's public key $vk^{(i^\star)}$ by a random element of $\mathbb{Z}_{N^2}^*$ in order to force the forgery to prove a false statement. In the last game transition, the problem is that we cannot immediately rely on the DCR assumption to change the distribution of $vk^{(i^\star)}$ while using the factorization of N to extract $\ell_1^\star \dots \ell_r^\star$. We thus involve two distinct moduli in our DCR-based adaptation of GK. The use of distinct moduli N and $\bar{N}$ requires to adjust our Σ-protocol and force some equality to hold over the integers (and thus modulo both N and $\bar{N}$) between values $a, \ell \in \mathbb{Z}_{\bar{N}}$ that our $\mathsf{BadChallenge}$ function extracts from the commitments in the first prover message. We enforce this condition by imposing an unusual range restriction to some component of the response $z = a + \mathsf{Chall} \cdot \ell \in \mathbb{Z}$: Instead of only checking an upper bound for z, the verifier also checks a lower bound to ensure that no implicit modular reduction occurs when homomorphically computing $a + \mathsf{Chall} \cdot \ell$ over commitments sent by a malicious prover.

Using the above ideas, the proof of unforgeability requires reliable erasures. The reason is that the security proof appeals to the NIZK simulator to answer all signing queries. Hence, if the adversary corrupts some user i after a signing query involving $sk^{(i)}$, the challenger has to pretend that the random coins of user i's past signatures have been erased as it cannot efficiently compute randomness that explain the simulated NIZK arguments as real arguments. In a second step, we modify the scheme to get rid of the erasure assumption.

A first idea to avoid erasures is to adapt the proof of unforgeability in such a way that the NIZK simulator is only used to simulate signatures on behalf of the expected target user (whose index $i^\star$ is guessed upfront), while all other users' signatures are faithfully generated. If the guess is correct, user $i^\star$ is never corrupted and the reduction never gets stuck when it comes to explaining the generation of signatures created by adaptively corrupted users. However, this strategy raises a major difficulty since decoding the signer's position $\ell^\star$ in the ring is only possible when the bits $\ell_1^\star \dots \ell_r^\star \in \{0,1\}^r$ of $\ell^\star$ are committed using extractable commitments $\{L_i^\star\}_{i=1}^r$. At the same time, our security proof requires the guessed index $i^\star$ to be statistically independent of the adversary's view until the forgery stage. In turn, this requires to simulate user $i^\star$'s signatures via *statistical* NIZK arguments. Indeed, computational NIZK proofs would information-theoretically leak the index $i^\star$ of the only user for which the NIZK simulator is used in signing queries. Unfortunately, perfectly binding commitments are not compatible with statistical ZK in our setting. To resolve this tension, we need a commitment which is perfectly hiding in all signing queries and extractable in the forgery. Moreover, for anonymity purposes, the perfectly hiding mode should make it possible to perfectly randomize adversarially-chosen commitments when we multiply them with commitments to 0. We instantiate this commitment using a variant (called "dense $\mathcal{R}$-lossy PKE" hereafter) of our DCR-based $\mathcal{R}$-lossy PKE scheme. Like our original $\mathcal{R}$-lossy PKE system, it can be programmed to be statistically hiding in all signing queries and extractable in the forgery, but it features different properties: It does not have to be equivocable, but we need its lossy mode to be dense in $\mathbb{Z}_{N^2}^*$ (a property not met by our equivocable $\mathcal{R}$-lossy PKE) in order to use it in a statistically HVZK Σ-protocol.

Related Work. The negative results (e.g., [43,17]) on the standard-model soundness of Fiat-Shamir did not rule out the existence of secure instantiations when specific protocols are compiled using concrete hash functions. A large body of work [72,16,50,14,26,29,59,10,53] investigated the circumstances under which CI hash functions [17] lead to secure standard model instantiations of the paradigm. Canetti *et al.* [15] showed that correlation intractability for *efficiently searchable* relations suffices to remove interaction from any trapdoor Σ-protocol. This includes their variant of [39] for the language of Hamiltonian graphs, which enables Fiat-Shamir-based proofs for all NP. They also gave candidates assuming the existence of fully homomorphic encryption (FHE) with circular security [18]. Peikert and Shiehian [67] subsequently achieved the same result under the standard LWE assumption [68].

Canetti *et al.* [18,15] gave trapdoor Σ-protocols for the languages of Hamiltonian graphs and quadratic residues in $\mathbb{Z}_N^*$ [42]. Like the generic trapdoor Σ-protocol of [27], they proceed with parallel repetitions of a Σ-protocol with challenge space $\{0, 1\}$. CI hash functions were also used to compress protocols with multiple interaction rounds [14,26,59,53] and larger challenges. Lombardi and Vaikuntanathan [59] notably extended the CI paradigm beyond the class of protocols where the $\mathsf{BadChallenge}$ function is efficiently computable. In this case, however, evaluating the hash function in polynomial time requires a fairly strong LWE assumption to ensure correlation intractability. Brakerski *et al.* [10] considered a stronger notion of correlation intractability which allows handling relations where the $\mathsf{BadChallenge}$ function can only be approximated by a distribution over constant-degree polynomials. They thus obtained Fiat-Shamir-based NIZK arguments from standard assumptions that are not known to imply FHE.

In the following, we consider 3-message protocols where bad challenges are efficiently (and exactly) computable – and thus enable the use of polynomial-time-computable CI hash functions based on standard lattice assumptions – in an exponentially large set after a single protocol run.

Ring signatures were coined by Rivest, Shamir and Tauman [69]. They enable unconditional anonymity and involve no registration phase nor any tracing authority. Whoever has a public key can be appointed as a ring member without being asked for his consent or even being aware of it. The original motivation of ring signatures was to enable the anonymous leakage of secrets, by concealing the identity of a source (e.g., a whistleblower in a political scandal) while simultaneously providing reliability guarantees. Recently, the primitive also found applications in the context of cryptocurrencies [62].

After the work of Rivest, Shamir and Tauman [69], a number of solutions were given under various assumptions [12,1,71,11,46,65,2]. Bender *et al.* [6] gave stronger definitions and constructions from general assumptions. In the standard model, more efficient schemes were given [71,8] in groups with a bilinear map. Brakerski and Tauman [11] gave the first constructions from lattice assumptions.

In early realizations [69,12,71,8], the size of signatures was linear in the number of ring members. Dodis *et al.* [36] suggested constant-size ring signatures in the random oracle model. Chase and Lysyanskaya [23] took a similar approach while using simulation-extractable NIZK proofs in the standard model. However, it is not clear how to adapt their approach without using generic NIZK. Assuming a common reference string, constructions with sub-linear-size signatures in the standard model were given in [22,44,28]. Malavolta and Schröder [60] used SNARKs (and thus non-falsifiable assumptions) to obtain constant-size signatures. In the random oracle model, Groth and Kohlweiss [46] obtained an elegant construction with logarithmic-size ring signatures under the discrete logarithm assumption. Lattice-based analogues of [46] were given in [37,38].

The log-size signatures of [46,55,57] are obtained by applying Fiat-Shamir to Σ-protocols that are not immediately compatible with the $\mathsf{BadChallenge}$ function paradigm. In their settings, it would require to iterate a basic Σ-protocol (with small challenge space) a super-constant number of times, thus leading to a combinatorial explosion in the total number of bad challenges as each iteration would tolerate more than one bad challenge. Backes *et al.* [3] and Chatterjee *et al.* [24] eliminated the need for a CRS while retaining logarithmic signature size. However, they did not provide concrete signature sizes and, due to the use of general NIWI/ZAPs techniques, their constructions would require much longer signatures than ours for any realistic ring cardinality. For instance, even for very small rings, the construction of [24] would incur signatures comprised several hundreds of Megabytes to represent $O(\lambda^3)$ FHE ciphertexts. In stark contrast with earlier solutions, our signatures would still fit within ≈ 1.5Mb (using 3072-bit RSA moduli) for rings as large as the number of atoms in the universe.

While our construction relies on a common reference string, it features (to our knowledge) the first logarithmic-size signatures with concretely efficient signature length *and* security under standard assumptions in the standard model.

2 Background and Definitions

For any $t \geq 2$, we denote by $\mathbb{Z}_t$ the ring of integers with addition and multiplication modulo t. For a finite set S, $U(S)$ stands for the uniform distribution over S. If X and Y are distributions over the same domain, $\Delta(X,Y)$ denotes their statistical distance. For a distribution D, $x \sim D$ means that x is distributed according to D, while $x \hookleftarrow D$ denotes the explicit action of sampling x from D.

2.1 Hardness Assumptions

We first recall the Learning-With-Errors (LWE) assumption.

Definition 2.1 ([68]). *Let $m \geq n \geq 1$, $q \geq 2$ be functions of a security parameter λ and let a distribution χ over $\mathbb{Z}$. The LWE problem consists in distinguishing between the distributions $(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e})$ and $U(\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m)$, where $\mathbf{A} \sim U(\mathbb{Z}_q^{m\times n})$, $\mathbf{s} \sim U(\mathbb{Z}_q^n)$ and $\mathbf{e} \sim \chi^m$.*

When χ is the discrete Gaussian distribution $D_{\mathbb{Z}^m,\alpha q}$ with standard deviation αq for some $\alpha \in (0,1)$, this problem is as hard as worst-case instances of well-studied lattice problems. We now recall the Composite Residuosity assumption.

Definition 2.2 ([64,34]). *Let integers $N = pq$ and $\zeta > 1$ for primes p, q. The* ζ**-Decision Composite Residuosity** (ζ-DCR) *assumption states that the distributions $\{x = w^{N^\zeta} \bmod N^{\zeta+1} \mid w \leftarrow U(\mathbb{Z}_N^\star)\}$ and $\{x \mid x \leftarrow U(\mathbb{Z}_{N^{\zeta+1}}^\star)\}$ are computationally indistinguishable.*

It is known [34] that the ζ-DCR assumption is equivalent to 1-DCR for any $\zeta > 1$.

2.2 Correlation Intractable Hash Functions

We consider efficiently enumerable [15] relations $R \subseteq \mathcal{X} \times \mathcal{Y}$ where, for each $x \in \mathcal{X}$, there is a polynomial number of elements $y \in \mathcal{Y}$ satisfying $R(x,y) = 1$. Moreover, these are efficiently enumerable.

Definition 2.3. *A relation $R \subseteq \mathcal{X} \times \mathcal{Y}$ is* **enumerable** *in time T if there exists a function $f_R : \mathcal{X} \to 2^{\mathcal{Y}}$ computable in time T such that, for each $x \in \mathcal{X}$, $f_R(x) = \{y_x \in \mathcal{Y} \mid (x, y_x) \in R\}$. If $\max_{x\in\mathcal{X}} |f_R(x)| \leq 1$, it is called* **searchable**.

Let $\lambda \in \mathbb{N}$ a security parameter. A hash family with input length $n(\lambda)$ and output length λ is a collection $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^\lambda\}$ of keyed functions induced by efficient algorithms $(\mathsf{Gen}, \mathsf{Hash})$, where $\mathsf{Gen}(1^\lambda)$ outputs a key $k \in \{0,1\}^{s(\lambda)}$ and $\mathsf{Hash}(k,x)$ computes $h_\lambda(k,x) \in \{0,1\}^\lambda$.

Definition 2.4. *For a relation ensemble $\{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^\lambda\}$, a hash function family $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ is* R**-correlation intractable** *if, for any probabilistic polynomial time (PPT) adversary $\mathcal{A}$, we have $\Pr\left[k \leftarrow \mathsf{Gen}(1^\lambda)),\ x \leftarrow \mathcal{A}(k) : (x, h_\lambda(k,x)) \in R\right] = \mathsf{negl}(\lambda)$.*

Peikert and Shiehian [67] described a CI hash family for any searchable relation defined by functions f of bounded depth. Their construction relies on the standard LWE assumption with polynomial approximation factors. Their proof was given for efficiently searchable relations. However, it also implies correlation intractability for efficiently enumerable relations, as observed in [18,53].

2.3 Admissible Hash Functions

Admissible hash functions were introduced in [7] as a combinatorial tool for partitioning-based security proofs.

Definition 2.5 ([7]). *Let $\ell(\lambda), L(\lambda) \in \mathbb{N}$ be functions of $\lambda \in \mathbb{N}$. Let an efficiently computable function $\mathsf{AHF} : \{0,1\}^\ell \to \{0,1\}^L$. For each $K \in \{0,1,\perp\}^L$, let the partitioning function $F_{\mathsf{ADH}}(K,\cdot) : \{0,1\}^\ell \to \{0,1\}$ such that*

$$F_{\mathsf{ADH}}(K,X) := \begin{cases} 0 & \textit{if} \quad \forall i \in [L] \quad (\mathsf{AHF}(X)_i = K_i) \ \vee \ (K_i = \perp) \\ 1 & \textit{otherwise} \end{cases}$$

We say that AHF is an **admissible hash function** *if there exists an efficient algorithm $\mathsf{AdmSmp}(1^\lambda, Q, \delta)$ that takes as input $Q \in \mathsf{poly}(\lambda)$ and a non-negligible $\delta(\lambda) \in (0,1]$ and outputs a key $K \in \{0,1,\perp\}^L$ such that, for all $X^{(1)}, \dots, X^{(Q)}, X^\star \in \{0,1\}^\ell$ such that $X^\star \notin \{X^{(1)}, \dots, X^{(Q)}\}$, we have*

$$\Pr_K \left[F_{\mathsf{ADH}}(K, X^{(1)}) = \dots = F_{\mathsf{ADH}}(K, X^{(Q)}) = 1 \ \wedge \ F_{\mathsf{ADH}}(K, X^\star) = 0 \right] \geq \delta(Q(\lambda)) \ .$$

It is known that admissible hash functions exist for $\ell, L = \Theta(\lambda)$.

Theorem 2.6 ([52, Theorem 1]). *Let $(C_\ell)_{\ell \in \mathbb{N}}$ be a family of codes $C_\ell : \{0,1\}^\ell \to \{0,1\}^L$ with minimal distance cL for some constant $c \in (0,1/2)$. Then, $(C_\ell)_{\ell \in \mathbb{N}}$ is a family of admissible hash functions. Furthermore, $\mathsf{AdmSmp}(1^\lambda, Q, \delta)$ outputs a key $K \in \{0,1,\perp\}^L$ for which $\eta = O(\log \lambda)$ components are not $\perp$ and $\delta(Q(\lambda))$ is a non-negligible function of λ.*

Jager proved [52] Theorem 2.6 for *balanced* admissible hash functions, which provide both a lower bound and a close upper bound for the probability in Definition 2.5. However, Theorem 2.6 applies to standard admissible hash functions.

2.4 Trapdoor Σ-protocols

Canetti *et al.* [18] defined a trapdoor variant of the notion of Σ-protocols [31].

Definition 2.7 (Adapted from [18]). *Let a language $\mathcal{L}$ associated with an NP relations R. A 3-move interactive proof system $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ in the common reference string model is a Σ-protocol for $\mathcal{L}$ if it satisfies the following:*

- **3-Move Form:** *P and V both input $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$, with $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ and $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$, and a statement x. They proceed as follows: (i) P inputs $w \in R(x)$, computes $(\mathbf{a}, st) \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$ and sends $\mathbf{a}$ to V; (ii) V sends back a random challenge Chall; (iii) P finally sends a response $\mathbf{z} = \mathsf{P}(\mathsf{crs}, x, w, \mathbf{a}, \mathsf{Chall}, st)$ to V; (iv) On input of $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$, V outputs 1 or 0.*
- **Completeness:** *If $(x,w) \in R$ and P honestly computes $(\mathbf{a}, \mathbf{z})$ for a challenge Chall, then $\mathsf{V}(\mathsf{crs}, x, (\mathbf{a}, \mathsf{Chall}, \mathbf{z}))$ outputs 1 with probability $1 - \mathsf{negl}(\lambda)$.*

– **Special zero-knowledge:** *There is a PPT simulator* ZKSim *that inputs* crs, $x \in \mathcal{L}$ *and a challenge* $\mathsf{Chall} \in \mathcal{C}$. *It outputs* $(\mathbf{a}, \mathbf{z}) \leftarrow \mathsf{ZKSim}(\mathsf{crs}, x, \mathsf{Chall})$ *such that* $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *is indistinguishable from a real transcript (for* $w \in R(x)$*) with challenge* Chall.
– $(r+1)$**-Special soundness:** *For any CRS* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_\mathcal{L})$ *obtained as* $\mathsf{par} \leftarrow \mathsf{Gen}_\mathsf{par}(1^\lambda)$, $\mathsf{crs}_\mathcal{L} \leftarrow \mathsf{Gen}_\mathcal{L}(\mathsf{par}, \mathcal{L})$, *any* $x \notin \mathcal{L}$, *and any first message* $\mathbf{a}$ *sent by* P, *the set of challenges* $\mathcal{BADC} = f(\mathsf{crs}, x, \mathbf{a})$ *for which an accepting transcript* $(\mathsf{crs}, x, \mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *exists for some third message* $\mathbf{z}$ *has cardinality* $|\mathcal{BADC}| \leq r$. *The function* f *is called the "bad challenge function" of* Π. *That is, if* $x \notin \mathcal{L}$ *and* $\mathsf{Chall} \notin \mathcal{BADC}$, *the verifier never accepts.*

Canetti *et al.* [18] define *trapdoor* Σ-protocols as Σ-protocols where the bad challenge function is efficiently computable using a trapdoor. They also define instance-dependent trapdoor Σ-protocol where the trapdoor τ_Σ should be generated as a function of some instance $x \notin \mathcal{L}$. Here, we use a definition where x need not be known in advance and the trapdoor does not depend on a specific x. However, the CRS and the trapdoor may depend on the language in our setting. The CRS $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_\mathcal{L})$ consists of a fixed part par and a language-dependent part $\mathsf{crs}_\mathcal{L}$ which is generated as a function of par and a language description $\mathcal{L}$.

Definition 2.8 (Adapted from [18]). *A* Σ*-protocol* $\Pi = (\mathsf{Gen}_\mathsf{par}, \mathsf{Gen}_\mathcal{L}, \mathsf{P}, \mathsf{V})$ *with bad challenge function* f *for a trapdoor language* $\mathcal{L}$ *is a* **trapdoor** Σ**-protocol** *if it satisfies the properties of Definition 2.7 and there exist PPT algorithms* $(\mathsf{TrapGen}, \mathsf{BadChallenge})$ *with the following properties.*

- $\mathsf{Gen}_\mathsf{par}$ *inputs* $\lambda \in \mathbb{N}$ *and outputs public parameters* $\mathsf{par} \leftarrow \mathsf{Gen}_\mathsf{par}(1^\lambda)$.
- $\mathsf{Gen}_\mathcal{L}$ *is a randomized algorithm that, on input of public parameters* par, *outputs the language-dependent part* $\mathsf{crs}_\mathcal{L} \leftarrow \mathsf{Gen}_\mathcal{L}(\mathsf{par}, \mathcal{L})$ *of* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_\mathcal{L})$.
- $\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_\mathcal{L})$ *inputs public parameters* par *and (optionally) a trapdoor* $\tau_\mathcal{L}$ *allowing to test membership of* $\mathcal{L}$. *It outputs* $\mathsf{crs}_\mathcal{L}$ *and a trapdoor* τ_Σ.
- $\mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}, x, \mathbf{a})$ *takes in a trapdoor* τ_Σ, *a CRS* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_\mathcal{L})$, *an instance* x, *and a first prover message* $\mathbf{a}$. *It outputs a set* $\mathcal{BADC}$.

In addition, the following properties are required.

- **CRS indistinguishability:** *For any* $\mathsf{par} \leftarrow \mathsf{Gen}_\mathsf{par}(1^\lambda)$, *and any trapdoor* $\tau_\mathcal{L}$ *for the language* $\mathcal{L}$, *an honestly generated* $\mathsf{crs}_\mathcal{L}$ *is computationally indistinguishable from a CRS produced by* $\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_\mathcal{L})$. *Namely, for any* aux *and any PPT distinguisher* $\mathcal{A}$, *we have*

$$\begin{aligned}\mathbf{Adv}_\mathcal{A}^{\text{indist-}\Sigma}(\lambda) := |\Pr[&\mathsf{crs}_\mathcal{L} \leftarrow \mathsf{Gen}_\mathcal{L}(\mathsf{par}, \mathcal{L}) : \mathcal{A}(\mathsf{par}, \mathsf{crs}_\mathcal{L}) = 1] \\ &- \Pr[(\mathsf{crs}_\mathcal{L}, \tau_\Sigma) \leftarrow \mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_\mathcal{L}) : \mathcal{A}(\mathsf{par}, \mathsf{crs}_\mathcal{L}) = 1]| \leq \mathsf{negl}(\lambda).\end{aligned}$$

- **Correctness:** *There exists a language-specific trapdoor* $\tau_\mathcal{L}$ *such that, for any instance* $x \notin \mathcal{L}$ *and all pairs* $(\mathsf{crs}_\mathcal{L}, \tau_\Sigma) \leftarrow \mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_\mathcal{L})$, *we have* $\mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}, x, \mathbf{a}) = f(\mathsf{crs}, x, \mathbf{a})$.

Note that the $\mathsf{TrapGen}$ algorithm does not take a specific statement x as input, but only a trapdoor $\tau_\mathcal{L}$ allowing to recognize elements of $\mathcal{L}$.

2.5 $\mathcal{R}$-Lossy Public-Key Encryption With Equivocation

In [56], Libert *et al.* considered a generalization of the notion of $\mathcal{R}$-lossy encryption introduced by Boyle *et al.* [9]. The primitive is a flavor of tag-based encryption [54] where the tag space $\mathcal{T}$ is partitioned into *injective* and *lossy* tags. When ciphertexts are generated for an injective tag, the decryption algorithm recovers the plaintext. On lossy tags, ciphertexts statistically hide the plaintexts. In $\mathcal{R}$-lossy PKE schemes, the tag space is partitioned according to a binary relation $\mathcal{R} \subseteq \mathcal{K} \times \mathcal{T}$. The key generation algorithm inputs an initialization value $K \in \mathcal{K}$ and partitions $\mathcal{T}$ in such a way that injective tags $t \in \mathcal{T}$ are those for which $(K, t) \in \mathcal{R}$ (i.e., all tags t for which $(K, t) \notin \mathcal{R}$ are lossy).

The definition of [56] requires the existence of a lossy key generation algorithm $\mathsf{LKeygen}$ that outputs public keys for which all tags t are lossy (in contrast with injective keys where the only lossy tags are those for which $(K, t) \notin \mathcal{R}$). In addition, [56] also asks that a trapdoor allows equivocating lossy ciphertexts (a property called *efficient opening* [4]) using an algorithm called Opener. The application to simulation-soundness [56] involves two opening algorithms Opener and $\mathsf{LOpener}$. The former operates over injective public keys for lossy tags while the latter can equivocate ciphertexts encrypted under lossy keys for any tag.

Definition 2.9. *Let $\mathcal{R} \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ be a binary relation. An equivocable $\mathcal{R}$-lossy PKE scheme is a 7-uple of PPT algorithms* $(\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{LOpener})$ *such that:*

Parameter generation: *Given a security parameter λ, a tag length $L \in \mathsf{poly}(\lambda)$ and a message length $B \in \mathsf{poly}(\lambda)$, $\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$ outputs public parameters Γ that specify a tag space $\mathcal{T}$, a space of initialization values $\mathcal{K}$, a public key space $\mathcal{PK}$, a secret key space $\mathcal{SK}$ and a trapdoor space $\mathcal{TK}$.*

Key generation: *For an initialization value $K \in \mathcal{K}$ and public parameters Γ, algorithm $\mathsf{Keygen}(\Gamma, K)$ outputs an injective public key $\mathsf{pk} \in \mathcal{PK}$, a decryption key $\mathsf{sk} \in \mathcal{SK}$ and a trapdoor key $\mathsf{tk} \in \mathcal{TK}$. The public key specifies a ciphertext space CtSp and a randomness space R^{LPKE}.*

Lossy Key generation: *Given an initialization value $K \in \mathcal{K}$ and public parameters Γ, the lossy key generation algorithm $\mathsf{LKeygen}(\Gamma, K)$ outputs a lossy public key $\mathsf{pk} \in \mathcal{PK}$, a lossy secret key $\mathsf{sk} \in \mathcal{SK}$ and a trapdoor key $\mathsf{tk} \in \mathcal{TK}$.*

Decryption on injective tags: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, any $K \in \mathcal{K}$, any tag $t \in \mathcal{T}$ such that $(K, t) \in \mathcal{R}$, and any message $\mathsf{Msg} \in \mathsf{MsgSp}$, we have $\Pr\big[\exists r \in R^{\mathsf{LPKE}} : \mathsf{Decrypt}\big(\mathsf{sk}, t, \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}; r)\big) \neq \mathsf{Msg}\big] < \nu(\lambda)$, for some negligible function $\nu(\lambda)$, where $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ and the probability is taken over the randomness of Keygen.*

Indistinguishability: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, the key generation algorithms $\mathsf{LKeygen}$ and Keygen satisfy the following:*

(i) For any $K \in \mathcal{K}$, the distributions $D_{\text{inj}} = \{(\mathsf{pk}, \mathsf{tk}) \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)\}$ and $D_{\text{loss}} = \{(\mathsf{pk}, \mathsf{tk}) \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ are computationally indistinguishable. For any PPT adversary $\mathcal{A}$, the following advantage function $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{indist\text{-}LPKE}}(\lambda)$ is negligible:

$$|\Pr[(\mathsf{pk}, \mathsf{tk}) \hookleftarrow D_{\text{inj}} : \mathcal{A}(\mathsf{pk}, \mathsf{tk}) = 1] - \Pr[(\mathsf{pk}, \mathsf{tk}) \hookleftarrow D_{\text{loss}} : \mathcal{A}(\mathsf{pk}, \mathsf{tk}) = 1]| .$$

(ii) For any initialization values $K, K' \in \mathcal{K}$, the two distributions $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ *and* $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K')\}$ *are $2^{-\Omega(\lambda)}$-close in terms of statistical distance.*

Lossiness: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, any initialization value $K \in \mathcal{K}$ and tag $t \in \mathcal{T}$ such that $(K, t) \notin \mathcal{R}$, any* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$, *and any* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$, *the following distributions are statistically close:* $\{C \mid C \leftarrow \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0)\} \approx_s \{C \mid C \leftarrow \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_1)\}$. *For any* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$, *the above holds for any tag t.*

Equivocation under lossy tags: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, any $K \in \mathcal{K}$, any keys* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$, *let D_R the distribution, defined over R^{LPKE}, from which the random coins of* $\mathsf{Encrypt}$ *are sampled. For any message* $\mathsf{Msg} \in \mathsf{MsgSp}$ *and ciphertext C, let $D_{\mathsf{pk},\mathsf{Msg},C,t}$ denote the distribution on R^{LPKE} with support* $S_{\mathsf{pk},\mathsf{Msg},C,t} = \{\overline{r} \in R^{\mathsf{LPKE}} \mid \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}, \overline{r}) = C\}$ *and such that, for each $\overline{r} \in S_{PK,\mathsf{Msg},C,t}$, we have*

$$D_{\mathsf{pk},\mathsf{Msg},C,t}(\overline{r}) = \Pr_{r' \hookleftarrow D_R}[r' = \overline{r} \mid \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}, r') = C] \ . \qquad (1)$$

For any random coins $r \hookleftarrow D_R$, any tag $t \in \mathcal{T}_\lambda$ such that $(K, t) \notin \mathcal{R}$, and any messages $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$, *algorithm* Opener *takes as inputs* $\mathsf{pk}, C = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0, r)$, *$r$ t, and* tk. *It outputs a sample $\overline{r}$ from a distribution statistically close to $D_{\mathsf{pk},\mathsf{Msg}_1,C,t}$.*

Equivocation under lossy keys: *For any $K \in \mathcal{K}$, any keys* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$, *any randomness $r \hookleftarrow D_R$, any tag $t \in \mathcal{T}_\lambda$, and any messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$, *algorithm* $\mathsf{LOpener}$ *inputs* $C = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0, r)$, *$r$, t and* sk. *It outputs $\overline{r} \in R^{\mathsf{LPKE}}$ such that* $C = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_1, \overline{r})$. *We require that, for any tag $t \in \mathcal{T}_\lambda$ such that $(K, t) \notin \mathcal{R}$, the distribution* $\{\bar{r} \leftarrow \mathsf{LOpener}(\mathsf{pk}, \mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r) \mid r \hookleftarrow D_R\}$ *is statistically close to* $\{\bar{r} \leftarrow \mathsf{Opener}(\mathsf{pk}, \mathsf{tk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r) \mid r \hookleftarrow D_R\}$.

The above definition is slightly weaker than the one of [56] in the property of equivocation under lossy keys. Here, we do not require that the output of $\mathsf{LOpener}$ be statistically close to $D_{\mathsf{pk},\mathsf{Msg}_1,C,t}$ as defined in (1): We only require that, on lossy keys and lossy tags, Opener and $\mathsf{LOpener}$ sample random coins from statistically close distributions. Our definition turns out to be sufficient for the purpose of simulation-sound arguments (as shown in Supplementary Material B) and will allow us to obtain a construction from the DCR assumption.

Definition 2.9 also differs from [56, Definition 2.10] in that the equivocation algorithms ($\mathsf{Opener}, \mathsf{LOpener}$) can use the original random coins $r \in R^{\mathsf{LPKE}}$ of the encryption algorithm. Again, this relaxation will suffice in our setting.

In our ring signature system, we also use a variant of the above $\mathcal{R}$-lossy encryption primitive to instantiate a tag-based commitment scheme.

Definition 2.10. *A dense $\mathcal{R}$-lossy PKE scheme is a tuple* ($\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}$) *of efficient algorithms that proceed identically to Definition 2.9, except that the lossy mode is dense and the indistinguishability property is relaxed as below. Moreover, no equivocation property is required.*

Weak Indistinguishability: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, the key generation algorithms* LKeygen *and* Keygen *satisfy the following:*

(i) For any $K \in \mathcal{K}$, $D_{\text{inj}} = \{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)\}$ is indistinguishable from $D_{\text{loss}} = \{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$. For any PPT adversary $\mathcal{A}$, the advantage function $\mathbf{Adv}_{\mathcal{A}}^{\text{weak-indist-LPKE}}(\lambda)$, defined as the distance $|\Pr[\mathsf{pk} \hookleftarrow D_{\text{inj}} : \mathcal{A}(\mathsf{pk}) = 1] - \Pr[\mathsf{pk} \hookleftarrow D_{\text{loss}} : \mathcal{A}(\mathsf{pk}) = 1]|$, is negligible as a function of the security parameter.

(ii) For any initialization values $K, K' \in \mathcal{K}$, the two distributions $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ and $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K')\}$ are $2^{-\Omega(\lambda)}$-close in terms of statistical distance.

Density of Lossy Mode: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, any initialization value $K \in \mathcal{K}$, any $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$ and $\mathsf{Msg} \in \mathsf{MsgSp}$, the distribution of $\{\mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg}, r) | r \hookleftarrow D_R\}$ is statistically close to $U(\mathsf{CtSp})$.*

2.6 Ring Signatures

We now recall the syntax and the definitions of ring signatures [69]. Our definitions of unforgeability and anonmyity are identical to those of [6,22].

A ring signature [69] scheme consists of the following efficient algorithms:

CRSGen(1^λ)**:** Generates a common reference string ρ.

Keygen(ρ)**:** Generates a public key vk and the corresponding secret key sk.

Sign$(\rho, sk, M, \mathsf{R})$**:** Outputs a signature Σ on the message $M \in \{0,1\}^*$ with respect to the ring $\mathsf{R} = \{vk_0, \ldots, vk_{R-1}\}$ as long as (vk, sk) is a valid key pair produced by $\mathsf{Keygen}(\rho)$ and $vk \in \mathsf{R}$ (otherwise, it outputs $\perp$).

Verify$(\rho, M, \Sigma, \mathsf{R})$**:** Given a signature Σ on a message M w.r.t. the ring of public keys R, this algorithm outputs 1 if Σ is deemed valid and 0 otherwise.

Correctness requires that users can always sign any message on behalf of a ring they belong to. The standard security requirements for ring signatures are called *unforgeability* and *anonymity*. We use the strong definitions of [6,22], which are recalled in Section C of the Supplementary Material. In particular, we consider unforgeability with respect to insider corruption and statistical anonymity.

3 $\mathcal{R}$-Lossy Encryption Schemes from DCR

Libert *et al.* [56] gave a method that directly compiles any trapdoor Σ-protocol for a trapdoor language into an unbounded simulation-sound NIZK argument for the *same* language. As a building block, their construction uses an LWE-based equivocable $\mathcal{R}$-lossy PKE scheme for the bit-matching relation.

The construction of [56] is recalled in Supplementary Material B, where we show that it applies to trapdoor Σ-protocols with $(r+1)$-special-soundness for $r > 1$ as long as we have a CI hash function for efficiently enumerable relations.

Definition 3.1. *Let $\mathcal{K} = \{0,1,\perp\}^L$ and $\mathcal{T} = \{0,1\}^L$, for some $L \in \mathsf{poly}(\lambda)$. The* **bit-matching relation** *$\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ is defined as $\mathcal{R}_{\mathsf{BM}}(K,t) = 1$ if and only if $K = K_1 \ldots K_L$ and $t = t_1 \ldots t_L$ satisfy $\bigwedge_{i=1}^{L} (K_i = \perp) \vee (K_i = t_i)$.*

In [56], the authors described an $\mathcal{R}_{\mathsf{BM}}$-lossy PKE under the LWE assumption. In order to instantiate their construction with a better efficiency, we now describe a more efficient $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme based on the DCR assumption.

3.1 An Equivocable $\mathcal{R}_{\mathsf{BM}}$-Lossy PKE Scheme from DCR

The construction goes as follows.

Par-Gen$(1^\lambda, 1^L, 1^B)$**:** Define the spaces $\mathcal{T} = \{0,1\}^L$, $\mathcal{K} = \{0,1,\perp\}^L$ and the public parameters as $\Gamma = (1^\lambda, 1^B, \mathcal{K}, \mathcal{T})$.

Keygen(Γ, K)**:** Given public parameters Γ and an initialization value $K \in \mathcal{K}$, generate a key pair as follows.

1. Choose an RSA modulus $N = pq$ such that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > L(\lambda)$ for any sufficiently large λ, and an integer $\zeta \in \mathsf{poly}(\lambda)$ such that $N^\zeta > 2^B$.
2. Choose $g \hookleftarrow U(\mathbb{Z}^*_{N^{\zeta+1}})$ and $\alpha_{i,0}, \alpha_{i,1} \hookleftarrow U(\mathbb{Z}_N^*)$ for each $i \in [L]$. Then, for each $i \in [L]$ and $b \in \{0,1\}$, compute $v_{i,b} = g^{\delta_{b,1-K_i}} \cdot \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1}$ if $K_i \neq \perp$ and $v_{i,b} = \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1}$ if $K_i = \perp$.

Define $R^{\mathsf{LPKE}} = \mathbb{Z}_N^* \times \mathbb{Z}_{N^\zeta}$ and output $\mathsf{sk} = (p, q, K)$ as well as

$$\mathsf{pk} := \Big(N, \zeta, g, \{v_{i,b}\}_{i \in [L], b \in \{0,1\}}\Big), \qquad \mathsf{tk} = \big(\{\alpha_{i,b}\}_{i \in [L], b \in \{0,1\}}, K\big).$$

LKeygen(Γ, K)**:** is identical to Keygen except that step 2 generates g by choosing $g_0 \hookleftarrow U(\mathbb{Z}_N^*)$ and computing $g = g_0^{N^\zeta} \bmod N^{\zeta+1}$. The algorithm defines $R^{\mathsf{LPKE}} = \mathbb{Z}_N^* \times \mathbb{Z}_{N^\zeta}$ and outputs the lossy secret key $\mathsf{sk} = (g_0, \mathsf{tk})$ together with $\mathsf{pk} := \big(N, \zeta, g, \{v_{i,b}\}_{i \in [L], b \in \{0,1\}}\big)$, $\mathsf{tk} = \big(\{\alpha_{i,b}\}_{i \in [L], b \in \{0,1\}}, K\big)$.

Encrypt$(\mathsf{pk}, t, \mathsf{Msg})$**:** To encrypt $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ for the tag $t = t_1 \dots t_L \in \{0,1\}^L$, choose $r \hookleftarrow U(\mathbb{Z}_N^*)$, $s \hookleftarrow U(\mathbb{Z}_{N^\zeta})$ and compute

$$\mathsf{ct} = g^{\mathsf{Msg}} \cdot \Big(\prod_{i=1}^{L} v_{i,t_i}\Big)^s \cdot r^{N^\zeta} \bmod N^{\zeta+1} \quad . \tag{2}$$

Decrypt$(\mathsf{sk}, t, \mathsf{ct})$**:** Given $\mathsf{sk} = (p, q, K)$ and $t = t_1 \dots t_L \in \{0,1\}^L$, return $\perp$ if $R_{\mathsf{BM}}(K, t) = 0$. Otherwise, $\prod_{i=1}^L v_{i,t_i} \equiv \Big(\prod_{i=1}^L \alpha_{i,t_i}\Big)^{N^\zeta} \pmod{N^{\zeta+1}}$.

1. Compute $\beta_g \in \mathbb{Z}_{N^\zeta}$ such that $g^{\lambda(N)} \equiv (1+N)^{\beta_g} \bmod N^{\zeta+1}$, where $\lambda(N) = \mathrm{lcm}(p-1, q-1)$, by applying the algorithm of [34, Section 3]. Return $\perp$ if $\beta_g = 0$ or $\gcd(\beta_g, N^\zeta) > 1$.
2. Otherwise, compute $\mu \in \mathbb{Z}_{N^\zeta}$ such that $\mathsf{ct}^{\lambda(N)} \equiv (1+N)^\mu \pmod{N^{\zeta+1}}$. Then, compute $\mathsf{Msg} = \mu \cdot \beta_g^{-1} \bmod N^\zeta$ and output $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$.

Opener$\big(\mathsf{pk}, \mathsf{tk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, (r, s)\big)$**:** Given $\mathsf{tk} = (\{\alpha_{i,b}\}_{i,b}, K)$, $t \in \{0,1\}^L$, plaintexts $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathbb{Z}_{N^s}$ and random coins $(r, s) \in R^{\mathsf{LPKE}}$ such that $\mathsf{ct} = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0; (r, s))$, return $\perp$ if $R_{\mathsf{BM}}(K, t) = 1$. Otherwise, define

$$v_t \triangleq \prod_{i=1}^{L} v_{i,t_i} \bmod N^{\zeta+1} \;=\; g^{d_t} \cdot \Big(\prod_{i=1}^{L} \alpha_{i,t_i}\Big)^{N^\zeta} \bmod N^{\zeta+1}, \qquad (3)$$

where $d_t \in \{1, \ldots, L\}$ is the number of non-$\perp$ entries of K such that $K_i \neq t_i$. Note that $\gcd(d_t, N^\zeta) = 1$ since $p, q > L$. Then, compute and output

$$\bar{s} = s + (d_t^{-1} \bmod N^\zeta) \cdot (\mathsf{Msg}_0 - \mathsf{Msg}_1) \mod N^\zeta \qquad (4)$$

$$\bar{r} = r \cdot \prod_{i=1}^{L} \alpha_{i,t_i}^{s-\bar{s}} \cdot g^{(\mathsf{Msg}_0 - \mathsf{Msg}_1 + d_t\cdot(s-\bar{s}))/N^\zeta} \mod N,$$

where the division in the exponent above g can be computed over $\mathbb{Z}$ since we have $\mathsf{Msg}_0 + d_t \cdot s \equiv \mathsf{Msg}_1 + d_t \cdot \bar{s} \pmod{N^\zeta}$. Note that $(\bar{r}, \bar{s})$ satisfy

$$\begin{aligned} g^{\mathsf{Msg}_1} \cdot v_t^{\bar{s}} \cdot \bar{r}^{N^\zeta} &\equiv g^{\mathsf{Msg}_1} \cdot \Big(g^{d_t} \cdot \prod_{i=1}^{L} \alpha_{i,t_i}^{N^\zeta}\Big)^{\bar{s}} \cdot \bar{r}^{N^\zeta}. \\ &\equiv g^{\mathsf{Msg}_1} \cdot \Big(g^{d_t} \cdot \prod_{i=1}^{L} \alpha_{i,t_i}^{N^\zeta}\Big)^{\bar{s}} \cdot r^{N^\zeta} \cdot \prod_{i=1}^{L} \alpha_{i,t_i}^{(s-\bar{s})\cdot N^\zeta} \cdot g^{(\mathsf{Msg}_0 - \mathsf{Msg}_1 + d_t\cdot(s-\bar{s}))} \\ &\equiv g^{\mathsf{Msg}_0} \cdot \Big(g^{d_t} \cdot \prod_{i=1}^{L} \alpha_{i,t_i}^{N^\zeta}\Big)^{s} \cdot r^{N^\zeta} \equiv g^{\mathsf{Msg}_0} \cdot v_t^{s} \cdot r^{N^\zeta} \pmod{N^{\zeta+1}} \end{aligned}$$

LOpener$\big(\mathsf{pk}, \mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, (r, s)\big)$**:** Given $\mathsf{sk} = \big(g_0, \mathsf{tk} = (\{\alpha_{i,b}\}_{i,b}, K)\big)$, an arbitrary tag $t \in \{0,1\}^L$, plaintexts $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathbb{Z}_{N^\zeta}$ and randomness $(r, s) \in R^{\mathsf{LPKE}}$ such that $\mathsf{ct} = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0; (r, s))$, let $d_t \in \{0, \ldots, L\}$ the number of non-$\perp$ entries such that $K_i \neq t_i$. If $d_t \neq 0$, compute $\bar{s}$ as per (4). Otherwise, choose $\bar{s} \hookleftarrow U(\mathbb{Z}_{N^\zeta})$. In both cases, output the pair $(\bar{r}, \bar{s})$, where $\bar{r} = r \cdot \prod_{i=1}^{L} \alpha_{i,t_i}^{s-\bar{s}} \cdot g_0^{\mathsf{Msg}_0 - \mathsf{Msg}_1 + d_t\cdot(s-\bar{s})} \mod N$.

Theorem 3.2. *The above scheme is an equivocable $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme under the* DCR *assumption.*

Proof. We show that the scheme satisfies the required properties.

Decryption under injective tags. Let a tag $t \in \{0,1\}^L$ and an initialization value $K \in \{0,1,\perp\}^L$ such that $\mathcal{R}_{\mathsf{BM}}(K, t) = 1$ and let a message $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$. Consider a triple $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{KeyGen}(\mathit{\Gamma}, K)$ where $\mathsf{sk} = (p, q, K)$, $\mathsf{tk} = (\{\alpha_{i,b}\}_{i,b}, K)$ and $\mathsf{pk} = (N, \zeta, g, \{v_{i,b}\}_{i\in[L], b\in\{0,1\}})$. The encryption algorithm outputs a ciphertext

$$\mathsf{ct} = g^{\mathsf{Msg}} \cdot \Big(\prod_{i=1}^{L} v_{i,t_i}\Big)^{s} \cdot r^{N^\zeta} \equiv g^{\mathsf{Msg}} \cdot \Big(\prod_{i=1}^{L} \alpha_{i,t_i}^{s} \cdot r\Big)^{N^\zeta} \pmod{N^{\zeta+1}}, \qquad (5)$$

for some $r \in \mathbb{Z}_N^*$ and $s \in \mathbb{Z}_{N^\zeta}$, where the second equality comes from the fact that $\mathcal{R}_{\mathsf{BM}}(K,t) = 1$ (i.e., $t_i = K_i$ whenever $K_i \neq \perp$). Since $g \sim U(\mathbb{Z}_{N^{\zeta+1}}^*)$, its order is a multiple of N^ζ with overwhelming probability $\varphi(N^\zeta)/N^\zeta = \varphi(N)/N$. Since $\lambda(N^{\zeta+1}) = N^\zeta \lambda(N)$, we thus know that $g^{\lambda(N)} \bmod N^{\zeta+1}$ has order N^ζ with probability $\varphi(N)/N$, in which case it has a representation of the form $g^{\lambda(N)} \equiv (1+N)^{\beta_g} \pmod{N^{\zeta+1}}$ for some $\beta_g \in \mathbb{Z}_{N^\zeta}$ such that $\gcd(\beta_g, N^\zeta) = 1$. From (5), we find that

$$\mathsf{ct}^{\lambda(N)} \equiv (1+N)^{\beta_g \cdot \mathsf{Msg}} \pmod{N^{\zeta+1}}, \tag{6}$$

which shows that $\mathsf{Decrypt}$ outputs $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ at step 2.

Indistinguishability. We have two properties to verify.

(i) When we consider the distributions of pairs $(\mathsf{pk}, \mathsf{tk})$ produced by Keygen and $\mathsf{LKeygen}$, the only difference is the way to sample g. In the injective case, g is sampled from $U(\mathbb{Z}_{N^{\zeta+1}}^*)$ while, in the lossy case, g is sampled as $g = g_0^{N^\zeta} \bmod N^{\zeta+1}$, with $g_0 \hookleftarrow U(\mathbb{Z}_N^*)$. The DCR assumption states that these two distributions are indistinguishable and the same holds for the two distributions of pairs $(\mathsf{pk}, \mathsf{tk})$.

(ii) We claim that the distributions $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K_b)\}_{b=0,1}$ are perfectly indistinguishable for any initialization values $K_0, K_1 \in \{0,1,\perp\}^L$ since $g = g_0^{N^\zeta} \bmod N^{\zeta+1}$ is an N^ζ-th residue. Indeed, for any fixed element $g_0 \in \mathbb{Z}_N^*$, the distributions $\{(N, v_{i,b} = g_0^{N^\zeta} \cdot \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1}) \mid \alpha_{i,b} \hookleftarrow U(\mathbb{Z}_N^*))\}$ and $\{(N, v_{i,b} = \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1}) \mid \alpha_{i,b} \hookleftarrow U(\mathbb{Z}_N^*))\}$ are identical.

Lossiness under lossy tags. Let an arbitrary $K \in \mathcal{K}$ and an injective key

$$\Big(\mathsf{pk} = \big(N, \zeta, g, \{v_{i,b}\}_{i \in [L], b \in \{0,1\}}\big), \qquad \mathsf{sk} = (p, q, K), \mathsf{tk} = (\{\alpha_{i,b}\}_{i,b}, K)\Big) \leftarrow \mathsf{Keygen}(\Gamma, K),$$

where $v_{i,b} = g^{\delta_{b,1-K_i}} \cdot \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1}$. Under a tag t such that $\mathcal{R}_{\mathsf{BM}}(K,t) = 0$, we have

$$v_t \triangleq \prod_{i=1}^{L} v_{i,t_i} \bmod N^{\zeta+1} \; = \; g^{d_t} \cdot \Big(\prod_{i=1}^{L} \alpha_{i,t_i}\Big)^{N^\zeta} \bmod N^{\zeta+1},$$

where $d_t \in \{1, \ldots, L\}$ is the number of non-$\perp$ positions where K disagrees with $t \in \{0,1\}^L$. Any ciphertext ct obtained by sampling $s \hookleftarrow U(\mathbb{Z}_{N^\zeta})$, $r \hookleftarrow U(\mathbb{Z}_N^*)$ and computing

$$\mathsf{ct} = g^{\mathsf{Msg}} \cdot v_t^s \cdot r^{N^\zeta} \bmod N^{\zeta+1} = g^{\mathsf{Msg} + d_t \cdot s} \cdot \Big(\prod_{i=1}^{L} \alpha_{i,t_i}^s \cdot r\Big)^{N^\zeta} \bmod N^{\zeta+1}, \tag{7}$$

perfectly hides $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ since $\gcd(d_t, N^\zeta) = 1$.

EQUIVOCATION UNDER LOSSY TAGS. We know that, for any $K \in \mathcal{K}$, any injective keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\varGamma, K)$ any message $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ and any tag t such that $\mathcal{R}_{\mathsf{BM}}(K, t) = 0$, the distribution $\{\mathsf{ct} = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}; (r, s)) \mid (r, s) \hookleftarrow U(R^{\mathsf{LPKE}})\}$ is nothing but the uniform distribution $U(\mathbb{Z}^*_{N^{\zeta+1}})$ by (7).

From (7), we also observe that a lossy ciphertext ct uniquely determines the value $\mathsf{Msg} + d_t \cdot s \bmod N^\zeta$ obtained by running $\mathsf{Decrypt}(\mathsf{sk}, t, \mathsf{ct})$. Hence, for any pair $(\mathsf{ct}, \mathsf{Msg}) \in \mathbb{Z}^*_{N^{\zeta+1}} \times \mathbb{Z}_{N^\zeta}$, there exists only one $s \in \mathbb{Z}_{N^\zeta}$ such that $\mathsf{Decrypt}(\mathsf{sk}, t, \mathsf{ct}) = \mathsf{Msg} + d_t \cdot s \bmod N^\zeta$. In turn, $(\mathsf{ct}, \mathsf{Msg}, s)$ uniquely determine $r_{\mathsf{ct}} = (\mathsf{ct} \cdot g^{-(\mathsf{Msg}+d_t \cdot s)})^{1/N^\zeta} \bmod N$ and thus $r = r_{\mathsf{ct}} \cdot \prod_{i=1}^{L} \alpha_{i,t'_i}^{-s} \bmod N$, which yields the only pair $(r, s) \in R^{\mathsf{LPKE}}$ such that $\mathsf{ct} = \mathsf{Encryt}(\mathsf{pk}, t, \mathsf{Msg}; (r, s))$. On an injective public key and a lossy tag t, for any $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$, the support $S_{\mathsf{pk},\mathsf{Msg},\mathsf{ct},t}$ is thus a singleton. This shows that, for any messages $\mathsf{Msg}_0, \mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ and randomness $(r, s) \in R^{\mathsf{LPKE}}$, Opener thus computes the only pair $(\bar{s}, \bar{r}) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}^*_N$ such that $\mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0; (r, s)) = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_1; (\bar{r}, \bar{s}))$.

EQUIVOCATION UNDER LOSSY KEYS. We now consider the case of lossy keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\varGamma, K)$, where $g = g_0^{N^\zeta} \bmod N^{\zeta+1}$. For any $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ and arbitrary tags t, ciphertexts have the distribution

$$\left\{ \mathsf{ct} = \big(g_0^{(\mathsf{Msg}+d_t \cdot s)} \cdot \prod_{i=1}^{L} \alpha_{i,t'_i}^{s} \cdot r\big)^{N^\zeta} \bmod N^{\zeta+1} \mid s \hookleftarrow U(\mathbb{Z}^*_{N^\zeta}), r \hookleftarrow U(\mathbb{Z}^*_N) \right\},$$

which is identical to $\{r^{N^\zeta} \bmod N^{\zeta+1} \mid r \hookleftarrow U(\mathbb{Z}^*_N)\}$. In contrast with injective keys, for a given N^ζ-th residue $\mathsf{ct} \in \mathbb{Z}^*_{N^{\zeta+1}}$, the support $S_{\mathsf{pk},\mathsf{Msg},\mathsf{ct},t}$ is no longer a singleton (even for lossy tags). However, for any lossy tag t and any ciphertext $\mathsf{ct} = \mathsf{Encryt}(\mathsf{pk}, t, \mathsf{Msg}_0; (r, s))$, $\mathsf{Opener}\big(\mathsf{pk}, \mathsf{tk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, (r, s)\big)$ and $\mathsf{LOpener}\big(\mathsf{pk}, \mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, (r, s)\big)$ output the same pair $(\bar{r}, \bar{s}) \in \mathbb{Z}^*_N \times \mathbb{Z}_{N^\zeta}$. Indeed, whenever $\mathcal{R}_{\mathsf{BM}}(K, t) = 0$ (and thus $d_t \neq 0$), $\mathsf{LOpener}$ computes

$$\bar{s} = s + (d_t^{-1} \bmod N^\zeta) \cdot (\mathsf{Msg}_0 - \mathsf{Msg}_1) \bmod N^\zeta,$$

exactly like Opener. Moreover, for any $\mathsf{ct} = g^{\mathsf{Msg}_0} \cdot v_t^s \cdot r^{N^\zeta} \bmod N^{\zeta+1}$ and any pair $(\mathsf{Msg}_1, \bar{s}) \in \mathbb{Z}_{N^\zeta} \times \mathbb{Z}_{N^\zeta}$,

$$\begin{aligned}
\bar{r} &= (\mathsf{ct} \cdot g^{-\mathsf{Msg}_1} \cdot v_t^{-\bar{s}})^{1/N^\zeta} \bmod N \\
&= \Big(g^{\mathsf{Msg}_0 - \mathsf{Msg}_1 + d_t \cdot (s - \bar{s})} \cdot \big(\prod_{i=1}^{L} \alpha_{i,t'_i}^{s-\bar{s}}\big)^{N^\zeta} \cdot r^{N^\zeta}\Big)^{1/N^\zeta} \bmod N \\
&= g^{(\mathsf{Msg}_0 - \mathsf{Msg}_1 + d_t \cdot (s - \bar{s}))/N^\zeta} \cdot \prod_{i=1}^{L} \alpha_{i,t'_i}^{s-\bar{s}} \cdot r \bmod N \qquad (8) \\
&= g_0^{\mathsf{Msg}_0 - \mathsf{Msg}_1 + d_t \cdot (s - \bar{s})} \cdot \prod_{i=1}^{L} \alpha_{i,t'_i}^{s-\bar{s}} \cdot r \bmod N \qquad (9)
\end{aligned}$$

is the unique $\bar{r} \in \mathbb{Z}^*_N$ (which Opener and $\mathsf{LOpener}$ compute as per (8) and (9), respectively) such that $\mathsf{ct} = g^{\mathsf{Msg}_1} \cdot v_t^{\bar{s}} \cdot \bar{r}^{N^\zeta} \bmod N^{\zeta+1}$. $\square$

By plugging the above system in the construction in Supplementary Material B, we obtain USS arguments from the DCR and LWE assumptions. A difference with [56] is that LWE is only used in the correlation intractable hash function and lattice trapdoors are not needed. This DCR-based scheme drastically reduces the signature length of our construction. If we were to use the LWE-based $\mathcal{R}$-Lossy PKE scheme from [56], a single ciphertext would already be roughly 20 larger than an entire ring signature, as discussed in Supplementary Material G.1.

3.2 A Dense $\mathcal{R}_{\mathsf{BM}}$-Lossy PKE Scheme from DCR

In order to construct a ring signature without relying on erasures, we will also use a "downgraded" version of the scheme in Section 3.1, where we do not need equivocation properties. However, we will rely on the property that its lossy mode induces dense commitments that are uniformly distributed in $\mathbb{Z}^*_{N^{\zeta+1}}$. The scheme of Section 3.1 does not have this density property as its lossy mode induces commitments that live in the subgroup of N^ζ-th residues.

$\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$: Define the spaces $\mathcal{T} = \{0,1\}^L$, $\mathcal{K} = \{0,1,\perp\}^L$ and the public parameters as $\Gamma = (1^\lambda, 1^B, \mathcal{K}, \mathcal{T})$.

$\mathsf{Keygen}(\Gamma, K)$: Given public parameters Γ and an initialization value $K \in \mathcal{K}$, generate a key pair as follows.

1. Choose an RSA modulus $N = pq$ such that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > L(\lambda) - \lambda$ for any sufficiently large λ, and an integer $\zeta \in \mathsf{poly}(\lambda)$ such that $N^\zeta > 2^B$.
2. Choose $\alpha_{i,0}, \alpha_{i,1} \hookleftarrow U(\mathbb{Z}_N^*)$ for each $i \in [L]$. Then, for each $i \in [L]$ and $b \in \{0,1\}$, compute $v_{i,b} = (1+N)^{\delta_{b,1-K_i}} \cdot \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1}$ if $K_i \neq \perp$ and $v_{i,b} = \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1}$ if $K_i = \perp$.

Define $R^{\mathsf{LPKE}} = \mathbb{Z}_N^* \times \mathbb{Z}_{N^\zeta}$ and output the secret key $\mathsf{sk} = (p, q, K)$ together with $\mathsf{pk} := \big(N, \zeta, \{v_{i,b}\}_{i\in[L], b\in\{0,1\}}\big)$ and $\mathsf{tk} = \perp$.

$\mathsf{LKeygen}(\Gamma, K)$: proceeds identically to Keygen with the difference that step 2 chooses $\{v_{i,b}\}_{i,b}$ at random. For each $i \in [L]$, $b \in \{0,1\}$, the algorithm chooses $v_{i,b} \hookleftarrow U(\mathbb{Z}^*_{N^{\zeta+1}})$. It defines $R^{\mathsf{LPKE}} = \mathbb{Z}_N^* \times \mathbb{Z}_{N^\zeta}$ and outputs $\mathsf{sk} = \perp$ as well as $\mathsf{pk} := \Big(N, \zeta, \{v_{i,b}\}_{i\in[L], b\in\{0,1\}}\Big)$, and $\mathsf{tk} = \perp$.

$\mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg})$: To encrypt $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ for the tag $t = t_1 \ldots t_L \in \{0,1\}^L$, choose random coins $r \hookleftarrow U(\mathbb{Z}_N^*)$, $s \hookleftarrow U(\mathbb{Z}_{N^\zeta})$ and compute the ciphertext $\mathsf{ct} = (1+N)^{\mathsf{Msg}} \cdot \Big(\prod_{i=1}^L v_{i,t_i}\Big)^s \cdot r^{N^\zeta} \bmod N^{\zeta+1}$.

$\mathsf{Decrypt}(\mathsf{sk}, t, \mathsf{ct})$: Given the secret key $\mathsf{sk} = (p, q, K)$ and the tag $t \in \{0,1\}^L$, return $\perp$ if $R_{\mathsf{BM}}(K, t) = 0$. Otherwise, compute $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ such that $\mathsf{ct}^{\lambda(N)} \equiv (1+N)^{\mathsf{Msg}} \pmod{N^{\zeta+1}}$ and output $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$.

Theorem 3.3. *The above system is a dense $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme under the* DCR *assumption. Moreover, the lossy mode is dense in $\mathbb{Z}^*_{N^{\zeta+1}}$.*

Proof. The property of decryption under injective keys can be shown exactly as in the proof of Theorem 3.2. The only difference is that $g = 1 + N$, so that its order is N^ζ with probability 1 and we have $\beta_g = 1$ in equation (6).

The proof of lossiness under lossy tags carries over in the same way. From (7), we observe that $s \sim U(\mathbb{Z}_{N^\zeta})$ perfectly hides Msg in the right-hand-side member of $\mathsf{ct} = (1+N)^{\mathsf{Msg}} \cdot v_t^s \cdot r^{N^\zeta} \bmod N^{\zeta+1}$ when $d_t \neq 0$.

The first weak indistinguishability property follows directly from the semantic security of Damgård-Jurik and thus the DCR assumption. Namely, under the DCR assumption, $(1+N)^{\delta_{b,1-K_i}} \cdot \alpha_{i,b}^{N^\zeta} \bmod N^{\zeta+1}$ is computationally indistinguishable from a random element of $\mathbb{Z}^*_{N^\zeta}$, regardless of $\delta_{b,1-K_i} \in \{0,1\}$. The second weak indistinguishability property is trivially satisfied since, in public keys generated by $\mathsf{LKeygen}$, $\{v_{i,b}\}_{i,b}$ are generated independently of K.

Finally, for lossy keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$, the distribution of ciphertexts is

$$\left\{ \mathsf{ct} = (1+N)^{\mathsf{Msg}} \cdot \big(\prod_{i=1}^{L} v_{i,t'_i}\big)^s \cdot r^{N^\zeta} \bmod N^{\zeta+1} \mid s \hookleftarrow U(\mathbb{Z}_{N^\zeta}),\ r \hookleftarrow U(\mathbb{Z}_N^*) \right\},$$

which is nothing but the uniform $U(\mathbb{Z}^*_{N^{\zeta+1}})$ unless there exists $t_1 \ldots t_L \in \{0,1\}^L$ such that the order of $\prod_{i=1}^{L} v_{i,t_i}$ is not a multiple of N^ζ. This occurs with probability $\leq 2^L \cdot (1 - \varphi(N)/N)$, which is negligible as long as $\min(p,q) \geq 2^{L+1-\lambda}$. $\square$

4 Trapdoor Σ-Protocols for DCR-Related Languages

Ciampi *et al.* [27] showed that any Σ-protocol with binary challenges can be turned into a trapdoor Σ-protocol by having the prover encrypt the two possible responses and send them along with its first message. While elegant, this approach requires $\Theta(\lambda)$ repetitions to achieve negligible soundness error. In this section, we give communication-efficient protocols requiring no repetitions.

In Supplementary Material D.2, we show that the standard Σ-protocol that allows proving composite residuosity readily extends into a trapdoor Σ-protocol. By exploiting earlier observations from [46,58], we show that, for a single protocol iteration, the factorization of N allows computing bad challenges within an exponentially large challenge space. In this section, we describe trapdoor Σ-protocols that will serve as building blocks for our ring signature.

4.1 Trapdoor Σ-Protocol Showing that a Paillier Ciphertext/Commitment Contains 0 or 1

We give a trapdoor Σ-protocol allowing to prove that a (lossy) Paillier ciphertext encrypts 0 or 1. This protocol is a DCR-based adaptation of a Σ-protocol proposed in [21,46] for Elgamal-like encryption schemes. The original protocol of [21,46] assumes additively homomorphic properties in the plaintext and randomness spaces. Here, we adapt it to the DCR setting where the randomness space is a multiplicative group. We also describe a $\mathsf{BadChallenge}$ function to obtain a trapdoor Σ-protocol with a large challenge space.

The $\mathsf{BadChallenge}$ function uses observation from Lipmaa [58] showing that bad challenges are also computable when the message space has composite order $N = pq$ (instead of prime order as in [21]). We actually point out an issue in [58], which aims to identify bad challenges in a Σ-protocol showing that an Elgamal-Paillier ciphertext [13] encrypts 0 or 1. However, in the Elgamal-Paillier scheme, not all elements of $\mathbb{Z}_{N^2}^* \times \mathbb{Z}_{N^2}^*$ are in the range of the encryption algorithm. In Supplementary Material D.3, we show that a cheating prover can send maliciously generated first prover messages for which bad challenges are not efficiently computable although they may exist for false statements.

Here, to avoid this issue, we need a DCR-based dual-mode commitment where the binding mode has the property that any element of $\mathbb{Z}_{N^2}^*$ is in the range of the commitment algorithm. Moreover, even the hiding mode should be dense, meaning that honestly generated commitments to 0 should be uniformly distributed over $\mathbb{Z}_{N^2}^*$. We thus use commitments of the form $C = (1+N)^{\mathsf{Msg}} \cdot h^y \cdot w^N \bmod N^2$, where the distribution of h determines if the commitment is perfectly hiding or perfectly binding. If h is an N-th residue (resp. $h \sim U(\mathbb{Z}_{N^2}^*)$), it is perfectly binding (resp. perfectly hiding). Moreover, the density property of the hiding mode will be crucial to prove the special ZK property of the Σ-protocol.

Let an RSA modulus $N = pq$ and let a random element $h \in \mathbb{Z}_{N^2}^*$. We give a trapdoor Σ-protocol for the following language, which is parametrized by h:

$$\mathcal{L}^{0\text{-}1}(h) = \big\{ C \in \mathbb{Z}_{N^2}^* \mid \exists b \in \{0,1\}, (y,w) \in \mathbb{Z}_N \times \mathbb{Z}_N^\star : \\ C = (1+N)^b \cdot h^y \cdot w^N \bmod N^2 \big\}.$$

We include h as a language parameter because we allow the CRS to depend on N, but not on h. We note that, if N divides the order of h, the language $\mathcal{L}^{0\text{-}1}(h)$ is trivial since all elements of $\mathbb{Z}_{N^2}^*$ can be explained as a commitment to a bit. However, the language becomes non-trivial when h is an N-th residue since $C = (1+N)^b\, h^y\, w^N \bmod N^2$ is then a perfectly binding commitment to b.

While a trapdoor Σ-protocol for $\mathcal{L}^{0\text{-}1}(h)$ can be obtained from [31], the one below is useful to show that one out of many ciphertexts encrypts 0 [46]. A difference with the Σ-protocols in [21, Figure 2] and [58, Section 3.2] is that, in order to use it in Section 4.2, we need the verifier to perform a non-standard interval check for the response over the integers.

$\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$: Given the security parameter λ, define $\mathsf{par} = \{\lambda\}$.

$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}^{0\text{-}1})$: Given public parameters par and the description of a language $\mathcal{L}^{0\text{-}1}$, consisting of an RSA modulus $N = pq$ with p and q prime satisfying $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > 2\lambda$, define the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{N\}$. The global CRS is $\mathsf{crs} = (\{\lambda\}, \mathsf{crs}_{\mathcal{L}})$.

$\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}^{0\text{-}1}, \tau_{\mathcal{L}})$: Given par, a language description $\mathcal{L}^{0\text{-}1}$ that specifies an RSA modulus $N = pq$, and the membership-testing trapdoor $\tau_{\mathcal{L}} = (p,q)$, output $\mathsf{crs} = (\{\lambda\}, \mathsf{crs}_{\mathcal{L}})$ as in $\mathsf{Gen}_{\mathcal{L}}$ and the trapdoor $\tau_{\Sigma} = (p,q)$.

$\mathbf{P}\big(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w}\big) \leftrightarrow \mathbf{V}(\mathsf{crs}, \boldsymbol{x})$: Given crs, a statement $\boldsymbol{x}$ = "$C \in \mathcal{L}^{0\text{-}1}(h)$", for some $h \in \mathbb{Z}_{N^2}^*$, P (who has $\boldsymbol{w} = (b, y, w)$) and V interact as follows:

1. P chooses $a \hookleftarrow U(\{2^\lambda, \ldots, 2^{2\lambda} - 1\})$, $d, e \hookleftarrow U(\mathbb{Z}_N)$, $u, v \hookleftarrow U(\mathbb{Z}_N^*)$ and sends V the following:
$$A_1 = (1+N)^a \, h^d \, u^N \bmod N^2, \qquad A_2 = (1+N)^{-a \cdot b} \, h^e \, v^N \bmod N^2.$$
2. V sends a random challenge $\mathsf{Chall} \hookleftarrow U(\{0, \ldots 2^\lambda - 1\})$.
3. P sends V the response $(z, z_d, z_e, z_u, z_v) \in \mathbb{Z} \times (\mathbb{Z}_N)^2 \times (\mathbb{Z}_N^*)^2$, where
$$z = a + \mathsf{Chall} \cdot b, \qquad z_1 = d + \mathsf{Chall} \cdot y, \qquad z_2 = e + (z - \mathsf{Chall}) \cdot y,$$
$$z_d = z_1 \bmod N, \qquad z_u = u \cdot w^{\mathsf{Chall}} \cdot h^{\lfloor z_1/N \rfloor} \bmod N,$$
$$z_e = z_2 \bmod N, \qquad z_v = v \cdot w^{z-\mathsf{Chall}} \cdot h^{\lfloor z_2/N \rfloor} \bmod N.$$
4. V returns 1 if and only if $2^\lambda \leq z < 2^{2\lambda+1}$ and
$$\begin{aligned} A_1 &= C^{-\mathsf{Chall}} \cdot (1+N)^z \cdot h^{z_d} \cdot z_u^N \bmod N^2, \\ A_2 &= C^{\mathsf{Chall}-z} \cdot h^{z_e} \cdot z_v^N \bmod N^2. \end{aligned} \tag{10}$$

$\mathbf{BadChallenge}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, \boldsymbol{x}, \boldsymbol{a}\big)$: Given a statement $\boldsymbol{x}$ = "$C \in \mathcal{L}^{0\text{-}1}(h)$", a trapdoor $\tau_\Sigma = (p, q)$ and $\boldsymbol{a} = (A_1, A_2) \in (\mathbb{Z}_{N^2}^*)^2$, return $\perp$ if h is not an N-th residue. Otherwise, decrypt C and (A_1, A_2) to obtain $b = \mathcal{D}_{\tau_\Sigma}(C) \in \mathbb{Z}_N$ and $a_i = \mathcal{D}_{\tau_\Sigma}(A_i) \in \mathbb{Z}_N$ for each $i \in \{1, 2\}$. If $\boldsymbol{x}$ is false, we have $b \notin \{0, 1\}$. Consider the following linear system with the unknowns $(\mathsf{Chall}, z) \in \mathbb{Z}_N^2$:
$$\begin{aligned} z - b \cdot \mathsf{Chall} &\equiv a_1 \pmod{N}, \\ b \cdot (\mathsf{Chall} - z) &\equiv a_2 \pmod{N}. \end{aligned} \tag{11}$$

1. If $b(b-1) \equiv 0 \pmod{N}$, assume that $b \equiv 0 \pmod{p}$ and $b \equiv 1 \pmod{q}$. Compute $z' = a_1 \bmod p$ and $\mathsf{Chall}' = z' - a_1 \bmod q$. Then, return $\perp$ if $\mathsf{Chall}' - z' \not\equiv a_2 \pmod{q}$ or $a_2 \not\equiv 0 \pmod{p}$.
2. If $b(b-1) \not\equiv 0 \pmod{N}$, define $d_b = \gcd(b(b-1), N)$, so that we have $\gcd(b(b-1), N/d_b) = 1$. Any solution of (11) also satisfies the system
$$\begin{aligned} z - b \cdot \mathsf{Chall} &\equiv a_1 \pmod{N/d_b} \\ b \cdot z - b \cdot \mathsf{Chall} &\equiv -a_2 \pmod{N/d_b}, \end{aligned}$$
which has a unique solution $(\mathsf{Chall}', z') \in (\mathbb{Z}_{N/d_b})^2$.

In both cases, if $2^\lambda \leq z' < 2^{2\lambda+1}$ and $0 \leq \mathsf{Chall}' < 2^\lambda$, return $\mathsf{Chall} = \mathsf{Chall}'$. Otherwise, return $\perp$.

Any honest protocol execution always returns a valid transcript since we have $2^\lambda \leq a + b \cdot \mathsf{Chall} \leq 2^{2\lambda} + 2^\lambda - 2 < 2^{2\lambda+1}$ and

$$\begin{aligned} &(1+N)^z \cdot h^{z_d} \cdot z_u^N \\ &\quad \equiv (1+N)^{a+\mathsf{Chall}\cdot b} \cdot u^N \cdot w^{N\cdot\mathsf{Chall}} \cdot h^{z_d} \cdot h^{(d+\mathsf{Chall}\cdot y)-(d+\mathsf{Chall}\cdot y \bmod N)} \\ &\quad \equiv (1+N)^{a+\mathsf{Chall}\cdot b} \cdot u^N \cdot w^{N\cdot\mathsf{Chall}} \cdot h^{d+\mathsf{Chall}\cdot y} \\ &\quad \equiv (1+N)^a \cdot u^N \cdot h^d \cdot \big((1+N)^b \cdot w^N \cdot h^y\big)^{\mathsf{Chall}} \equiv A_1 \cdot C^{\mathsf{Chall}} \pmod{N^2} \end{aligned}$$

$$\begin{aligned} C^{\mathsf{Chall}-z} \cdot h^{z_e} \cdot z_v^N &\equiv \big((1+N)^b \cdot w^N \cdot h^y\big)^{\mathsf{Chall}-z} \cdot v^N \cdot w^{(z-\mathsf{Chall})N} \cdot h^{e+(z-\mathsf{Chall})\cdot y} \\ &\equiv (1+N)^{b(\mathsf{Chall}-z)} \cdot v^N \cdot h^e \equiv (1+N)^{b(-a+(1-b)\cdot\mathsf{Chall})} \cdot v^N \cdot h^e \\ &\equiv (1+N)^{-ab} \cdot v^N \cdot h^e \equiv A_2 \pmod{N^2} \end{aligned}$$

The correctness of $\mathsf{BadChallenge}$ follows from the fact that $0 \leq \mathsf{Chall} < 2^\lambda$ (so that $\mathsf{Chall} = \mathsf{Chall} \bmod p = \mathsf{Chall} \bmod q$) and the observation that the verifier never accepts when $z \geq \min(p, q)$. This ensures that a valid response exists for at most one $z \in \mathbb{Z}$ such that $z = z \bmod p = z \bmod q$.

Remark 4.1. When h is a composite residue, the condition $b \in \{0, 1\}$ implies that, over $\mathbb{Z}$, we have either $z = a + b \cdot \mathsf{Chall}$ or $z = a + b \cdot \mathsf{Chall} - N$, where $a = \mathcal{D}_{\tau_\Sigma}(A_1)$ and $b = \mathcal{D}_{\tau_\Sigma}(C)$ (recall that (10) implies $z = a + b \cdot \mathsf{Chall} \bmod N$). The latter case can only occur if $b = 1$ and $N - 2^\lambda \leq a \leq N - 1$. However, this would imply $\mathsf{Chall} - 2^\lambda \leq a + \mathsf{Chall} - N \leq \mathsf{Chall} - 1$, which is not compatible with the lower bound of the verification test $2^\lambda \leq z < 2^{2\lambda+1}$. As a result, the equation $z = a + b \cdot \mathsf{Chall}$ holds over $\mathbb{Z}$, and not only modulo N. While this property is not necessary to ensure the soundness of the above Σ-protocol, it will be crucial for the $\mathsf{BadChallenge}$ function of the trapdoor Σ-protocol in Section 4.2.[7] In order to ensure perfect completeness, the prover chooses a in a somewhat unusual interval that does not start with 0. However, we still have statistical completeness and statistical HVZK if a is sampled from $U(\{0, \ldots, 2^{2\lambda} - 1\})$.

4.2 Trapdoor Σ-Protocol Showing that One Out of Many Ciphertexts/Commitments Contains 0

We now present a DCR-based variant of the Σ-protocol of Groth and Kohlweiss [46], which allows proving that one commitment out of $R = 2^r$ contains 0.

Intuition. The Σ-protocol of [46] relies on a protocol, like the one of Section 4.1, showing that a committed b is a bit using a response of the form $z = a + b \cdot \mathsf{Chall}$. To prove that some commitment $C_\ell \in \{C_i\}_{i=0}^{R-1}$ opens to 0 without revealing the index $\ell \in \{0, \ldots, R-1\}$, the bits $\ell_1 \ldots \ell_r \in \{0,1\}^r$ of ℓ are committed and, for each of them, the prover provides evidence that $\ell_j \in \{0, 1\}$. The response $z_j = a_j + \ell_j \mathsf{Chall}$ is seen as a degree-1 polynomial in Chall and used to define polynomials $f_{j,1}[X] = a_j + \ell_j X$ and $f_{f,0}[X] = X - f_j$, which in turn define

$$P_i[X] = \prod_{j=1}^{r} f_{j,i_j}[X] = \delta_{i,\ell} \cdot X^r + \sum_{k=0}^{r-1} p_{i,k} \cdot X^k \qquad \forall i \in \{0, \ldots, R-1\},$$

where $P_i[X]$ has degree r if $i = \ell$ and degree $\leq r - 1$ otherwise. In order to prove that one of the $\{P_i[X]\}_{i=0}^{R-1}$ has degree r, Groth and Kohlweiss homomorphically compute $\prod_{i=0}^{R-1} C_i^{P_i(\mathsf{Chall})}$ and multiply it with $\prod_{k=0}^{r-1} C_{d_k}^{-\mathsf{Chall}^k}$, for auxiliary commitments $\{C_{d_k} = \prod_{i=0}^{R-1} C_i^{p_{i,k}}\}_{k=0}^{r-1}$, in order to cancel out the

[7] In contrast, the upper bound for z is crucial here in the first step of $\mathsf{BadChallenge}$.

terms of degree 0 to $r-1$ in the exponent. Then, they prove that the product $\prod_{i=0}^{R-1} C_i^{P_i(\mathsf{Chall})} \cdot \prod_{k=0}^{r-1} C_{d_k}^{-\mathsf{Chall}^k}$ is indeed a commitment to 0.

Let $N = pq$ and $\bar{N} = \bar{p}\bar{q}$ denote two RSA moduli. Let also $h \in \mathbb{Z}_{N^2}^*$ and $\bar{h} \in \mathbb{Z}_{\bar{N}^2}^*$. We give a trapdoor Σ-protocol for the language

$$\begin{aligned}\mathcal{L}_{\vee}^{1\text{-}R}(h,\bar{h}) := \Big\{ \big((C_0,\ldots,C_{R-1}),(L_1,\ldots,L_r)\big) \in (\mathbb{Z}_{N^2}^*)^R \times (\mathbb{Z}_{\bar{N}^2}^*)^r \ \mid \\ \exists y \in \mathbb{Z}_N,\ w \in \mathbb{Z}_N^*,\ \exists_{j=1}^r (\ell_j, s_j, t_j) \in \{0,1\} \times \mathbb{Z}_{\bar{N}} \times \mathbb{Z}_{\bar{N}}^* : \\ \textstyle\bigwedge_{j=1}^r L_j = (1+\bar{N})^{\ell_j}\, \bar{h}^{s_j}\, t_j^{\bar{N}} \bmod \bar{N}^2 \quad \wedge \quad C_\ell = h^y w^N \bmod N^2 \Big\}\end{aligned} \tag{12}$$

where $R = 2^r$ and $\ell = \sum_{j=1}^r \ell_j \cdot 2^{j-1}$. In (12), $h \in \mathbb{Z}_{N^2}^*$ and $\bar{h} \in \mathbb{Z}_{\bar{N}^2}^*$ are used as language parameters since we allow the CRS to depend on N and $\bar{N}$, but not on h nor $\bar{h}$. The reason is that, in our construction of Section 5, we need to generate the CRS before $\bar{h}$ is chosen.

We note that $\mathcal{L}_{\vee}^{1\text{-}R}(h,\bar{h})$ is a trivial language (i.e., it is $(\mathbb{Z}_{N^2}^*)^R \times (\mathbb{Z}_{\bar{N}^2}^*)^r$) when N and $\bar{N}$ divide the order of h and $\bar{h}$, respectively. However, the security proof of our ring signature will switch to a setting where h and $\bar{h}$ are composite residues, which turns $C_\ell = h^y \cdot w^N \bmod N^2$ into a perfectly binding commitment to 0 (since $C = (1+N)^{\mathsf{Msg}} \cdot h^y \cdot w^N \bmod N^2$ uniquely determines the underlying $\mathsf{Msg} \in \mathbb{Z}_N$) and L_j into a perfectly binding commitment to ℓ_j.

Description. Our Paillier-based adaptation $\Pi_{\vee}^{1\text{-}R} = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ of the Σ-protocol of [46] is described as follows.

Gen$_{\mathsf{par}}(1^\lambda)$: Given the security parameter λ, define $\mathsf{par} = \{\lambda\}$.

Gen$_{\mathcal{L}}(\mathsf{par}, \mathcal{L}_{\vee}^{1\text{-}R})$: Given par and the description of a language $\mathcal{L}_{\vee}^{1\text{-}R}$, consisting of RSA moduli $N = pq$, $\bar{N} = \bar{p}\bar{q}$ with primes $p, q, \bar{p}, \bar{q}$ satisfying $p, q, \bar{p}, \bar{q} > 2^{l(\lambda)}$, where $l : \mathbb{N} \to \mathbb{N}$ is a polynomial such that $l(\lambda) > 2\lambda$, define the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{N, \bar{N}\}$ and the global CRS $\mathsf{crs} = (\{\lambda\}, \mathsf{crs}_{\mathcal{L}})$.

TrapGen$(\mathsf{par}, \mathcal{L}_{\vee}^{1\text{-}R}, \tau_{\mathcal{L}})$: Given par, the description of a language $\mathcal{L}_{\vee}^{1\text{-}R}$ and a language trapdoor $\tau_{\mathcal{L}}$, it proceeds identically to $\mathsf{Gen}_{\mathcal{L}}$ except that it also outputs the trapdoor $\tau_\Sigma = (p, q, \bar{p}, \bar{q})$.

P$\big(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w}\big) \leftrightarrow$ **V**(crs, x) : P has the witness $\boldsymbol{w} = (y, w, \{(\ell_j, s_j, t_j)\}_{j=1}^r)$ to the statement $\boldsymbol{x} =$ "$\big((C_0,\ldots,C_{R-1}),(L_1,\ldots,L_r)\big) \in \mathcal{L}_{\vee}^{1\text{-}R}(h,\bar{h})$" and interacts with the verifier V in the following way:

1. For each $j \in [r]$, P chooses $\bar{a}_j \hookleftarrow U(\{2^\lambda, \ldots, 2^{2\lambda}-1\})$, $\bar{d}_j, \bar{e}_j, \hookleftarrow U(\mathbb{Z}_{\bar{N}})$, $\bar{u}_j, \bar{v}_j \hookleftarrow U(\mathbb{Z}_{\bar{N}}^*)$ and computes

$$\begin{cases} \bar{A}_j = (1+\bar{N})^{\bar{a}_j} \cdot \bar{h}^{\bar{d}_j} \cdot \bar{u}_j^{\bar{N}} \bmod \bar{N}^2, \\ \bar{B}_j = (1+\bar{N})^{-\bar{a}_j \cdot \ell_j} \cdot \bar{h}^{\bar{e}_j} \cdot \bar{v}_j^{\bar{N}} \bmod \bar{N}^2. \end{cases} \tag{13}$$

It then defines degree-1 polynomials $F_{j,1}[X] = \bar{a}_j + \ell_j X \in \mathbb{Z}_N[X]$, $F_{j,0}[X] = X - F_{j,1}[X] \in \mathbb{Z}_N[X]$. For each index $i \in \{0,\ldots,R-1\}$ of binary expansion $i_1 \ldots i_r \in \{0,1\}^r$, it computes the polynomial

$$P_i[X] = \prod_{j=1}^r F_{j,i_j}[X] = \delta_{i,\ell} \cdot X^r + \sum_{k=0}^{r-1} p_{i,k} \cdot X^k \ \in \mathbb{Z}_N[X], \tag{14}$$

which has degree $\leq r-1$ if $i \neq \ell$ and degree r if $i = \ell$. Then, using the coefficients $p_{i,0}, \ldots, p_{i,r-1} \in \mathbb{Z}_N$ of (14), P computes commitments

$$C_{d_k} = \prod_{i=0}^{R-1} C_i^{p_{i,k}} \cdot h^{\mu_k} \cdot \rho_k^N \bmod N^2 \qquad 0 \leq k \leq r-1, \tag{15}$$

where $\mu_0, \ldots, \mu_{r-1} \hookleftarrow U(\mathbb{Z}_N)$, $\rho_0, \ldots, \rho_{r-1} \hookleftarrow U(\mathbb{Z}_N^*)$. Finally, P sends V the message $\boldsymbol{a} = \big(\{(\bar{A}_j, \bar{B}_j)\}_{j=1}^r,\ \{C_{d_k}\}_{k=0}^{r-1}\big)$.

2. V sends a random challenge $\mathsf{Chall} \hookleftarrow U(\{0, \ldots, 2^\lambda - 1\})$.
3. P sends the response $\big(z_y, z_w, \{(\bar{z}_j, \bar{z}_{d,j}, \bar{z}_{e,j}, \bar{z}_{u,j}, \bar{z}_{v,j})\}_{j=1}^r\big)$, where

$$\begin{cases} \bar{z}_{d,j} = \bar{d}_j + \mathsf{Chall} \cdot s_j \bmod \bar{N} \qquad \bar{z}_j = \bar{a}_j + \mathsf{Chall} \cdot \ell_j \\ \bar{z}_{e,j} = \bar{e}_j + (\bar{a}_j + \mathsf{Chall} \cdot (\ell_j - 1)) \cdot s_j \bmod \bar{N} \\ \bar{z}_{u,j} = \bar{u}_j \cdot \bar{t}_j^{\mathsf{Chall}} \cdot \bar{h}^{\lfloor (\bar{d}_j + \mathsf{Chall} \cdot s_j)/\bar{N} \rfloor} \bmod \bar{N} \\ \bar{z}_{v,j} = \bar{v}_j \cdot \bar{t}_j^{\bar{a}_j + \mathsf{Chall} \cdot (\ell_j - 1)} \cdot \bar{h}^{\lfloor (\bar{e}_j + (\bar{a}_j + \mathsf{Chall} \cdot (\ell_j - 1)) \cdot s_j)/\bar{N} \rfloor} \bmod \bar{N} \end{cases} \tag{16}$$

and, letting $P'[X] = y \cdot X^r - \sum_{k=1}^{r-1} \mu_k \cdot X^k \in \mathbb{Z}[X]$,

$$\begin{aligned} z_y &= y \cdot \mathsf{Chall}^r - \sum_{k=0}^{r-1} \mu_k \cdot \mathsf{Chall}^k \bmod N = P'(\mathsf{Chall}) \bmod N, \\ z_w &= w^{\mathsf{Chall}^r} \prod_{k=0}^{r-1} \rho_k^{-\mathsf{Chall}^k} \prod_{i=0}^{R-1} C_i^{-\lfloor P_i(\mathsf{Chall})/N \rfloor} \cdot h^{\lfloor P'(\mathsf{Chall})/N \rfloor} \bmod N, \end{aligned} \tag{17}$$

where $P_i(\mathsf{Chall})$ and $P'(\mathsf{Chall})$ are evaluated over $\mathbb{Z}$ in the exponent.
4. V defines $f_{j,1} = \bar{z}_j$ and $f_{j,0} = \mathsf{Chall} - \bar{z}_j \bmod N$ for each $j \in [r]$. Then, it accepts if $2^\lambda \leq \bar{z}_j < 2^{2\lambda+1}$ for all $j \in [r]$,

$$\forall j \in [r] : \begin{cases} \bar{A}_j = L_j^{-\mathsf{Chall}} \cdot (1 + \bar{N})^{\bar{z}_j} \cdot \bar{h}^{\bar{z}_{d,j}} \cdot \bar{z}_{u,j}^{\bar{N}} \bmod \bar{N}^2 \\ \bar{B}_j = L_j^{\mathsf{Chall} - \bar{z}_j} \cdot \bar{h}^{\bar{z}_{e,j}} \cdot \bar{z}_{v,j}^{\bar{N}} \quad \bmod \bar{N}^2 \end{cases} \tag{18}$$

and, parsing each $i \in \{0, \ldots, R-1\}$ into bits $i_1 \ldots i_r \in \{0,1\}^r$,

$$\prod_{k=0}^{r-1} C_{d_k}^{-\mathsf{Chall}^k} \cdot \prod_{i=0}^{R-1} C_i^{(\prod_{j=1}^r f_{j,i_j} \bmod N)} \equiv h^{z_y} \cdot z_w^N \pmod{N^2}. \tag{19}$$

It returns 0 if any of these conditions are not satisfied.

BadChallenge$\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, \boldsymbol{x}, \boldsymbol{a}\big)$: On input of a trapdoor $\tau_\Sigma = (p, q, \bar{p}, \bar{q})$, a statement $\boldsymbol{x}$ = "$((C_0, \ldots, C_{R-1}), (L_1, \ldots, L_r)) \in \mathcal{L}_\vee^{1\text{-}R}(h, \bar{h})$" and a first prover message $\boldsymbol{a} = \big(\{(\bar{A}_j, \bar{B}_j)\}_{j=1}^r,\ \{C_{d_k}\}_{k=0}^{r-1}\big)$, return $\perp$ if h is not an N-th residue in $\mathbb{Z}_{N^2}^*$ or $\bar{h}$ is not an $\bar{N}$-th residue in $\mathbb{Z}_{\bar{N}^2}^*$. Otherwise, compute $\ell_j = \mathcal{D}_{\tau_\Sigma}(L_j) \in \mathbb{Z}_{\bar{N}}$ and decrypt $\boldsymbol{a}$ so as to obtain $\bar{a}_j = \mathcal{D}_{\tau_\Sigma}(\bar{A}_j) \in \mathbb{Z}_{\bar{N}}$, $\bar{b}_j = \mathcal{D}_{\tau_\Sigma}(\bar{B}_j) \in \mathbb{Z}_{\bar{N}}$, for each $j \in [r]$, and $c_{d_k} = \mathcal{D}_{\tau_\Sigma}(C_{d_k}) \in \mathbb{Z}_N$ for each k. Let also $c_i = \mathcal{D}_{\tau_\Sigma}(C_i) \in \mathbb{Z}_N$ for each $i = 0$ to $R-1$. Since $\boldsymbol{x}$ is false, we have either: (i) $\ell_j \notin \{0,1\}$, for some $j \in [r]$; or (ii) $\forall j \in [r] : \ell_j \in \{0,1\}$ but $c_\ell \neq 0 \bmod N$, where $\ell = \sum_{j=1}^r \ell_j \cdot 2^{j-1}$. We consider two cases:

1. If there exists $j \in [r]$ such that $\ell_j \notin \{0,1\}$, then run the $\mathsf{BadChallenge}^{0\text{-}1}$ function of Sec. 4.1 on input of elements $\big(\mathsf{par}, (\bar{p}, \bar{q}), \{\bar{N}\}, L_j, (\bar{A}_j, \bar{B}_j)\big)$ and return whatever it outputs.
2. Otherwise, we have $\ell_j \in \{0,1\}$ for all $j \in [r]$. Define degree-1 polynomials $F_{j,1}[X] = \bar{a}_j + \ell_j X$, $F_{j,0}[X] = X - F_{j,1}[X] \in \mathbb{Z}_N[X]$ and compute $\{P_i[X]\}_{i=0}^{R-1}$ as per (14). For each $i \in \{0, \ldots, R-1\}$, parse the polynomial $P_i[X] \in \mathbb{Z}_N[X]$ as $P_i[X] = \delta_{i,\ell} \cdot X^r + \sum_{k=0}^{r-1} p_{i,k} \cdot X^k$ for some $p_{i,0}, \ldots, p_{i,r-1} \in \mathbb{Z}_N$. Define the polynomial

$$Q[X] \triangleq c_\ell \cdot X^r + \sum_{k=0}^{r-1} \left(\Big(\sum_{i=0}^{R-1} c_i \cdot p_{i,k} \Big) - c_{d_k} \right) \cdot X^k \ \in \mathbb{Z}_N[X],$$

which has degree r since $c_\ell \neq 0 \bmod N$. Define $Q_p[X] \triangleq Q[X] \bmod p$ and $Q_q[X] \triangleq Q[X] \bmod q$ over $\mathbb{Z}_p[X]$ and $\mathbb{Z}_q[X]$, respectively. Since at least one of them has degree r, we assume w.l.o.g. that $\deg(Q_p[X]) = r$. Then, compute the roots[8] $\mathsf{Chall}_{p,1}, \ldots, \mathsf{Chall}_{p,r}$ of $Q_p[X]$ over $\mathbb{Z}_p[X]$ in lexicographical order (if it has less than r roots, the non-existing roots are replaced by $\mathsf{Chall}_{p,i} = \perp$). For each $i \in [r]$, do the following:
   a. If $\mathsf{Chall}_{p,i} \notin \{0, \ldots, 2^\lambda - 1\}$, set $\mathsf{Chall}_i = \perp$.
   b. If $\mathsf{Chall}_{p,i} \in \{0, \ldots, 2^\lambda - 1\}$ and $Q_q(\mathsf{Chall}_{p,i}) \equiv 0 \pmod q$, then set $\mathsf{Chall}_i = \mathsf{Chall}_{p,i}$. Otherwise, set $\mathsf{Chall}_i = \perp$.

Correctness. To see that honestly generated proofs are always accepted by the verifier, we first note that $2^\lambda \leq \bar{a}_j \leq \bar{z}_j = \bar{a}_j + \mathsf{Chall} \cdot \ell_j \leq 2^{2\lambda} + 2^\lambda < 2^{2\lambda+1}$, for all $j \in [r]$, and that the equations (18) are satisfied for the same reasons as in Section 4.1. As for equation (19), we observe that, if the witnesses $y \in \mathbb{Z}_N$ and $w \in \mathbb{Z}_N^*$ satisfy $C_\ell = h^y \cdot w^N \bmod N^2$, we have

$$\begin{aligned}
h^{z_y} \cdot z_w^N \cdot \prod_{i=0}^{R-1} C_i^{-(\prod_{j=1}^{r} f_{j,i_j} \bmod N)} &\equiv h^{z_y} \cdot z_w^N \cdot \prod_{i=0}^{R-1} C_i^{-P_i(\mathsf{Chall}) \bmod N} \\
\equiv h^{z_y} \cdot w^{\mathsf{Chall}^r \cdot N} \cdot \prod_{k=0}^{r-1} \rho_k^{-\mathsf{Chall}^k \cdot N} \cdot \prod_{i=0}^{R-1} & C_i^{-P_i(\mathsf{Chall}) + (P_i(\mathsf{Chall}) \bmod N)} \\
& \cdot h^{P'(\mathsf{Chall}) - z_y} \cdot \prod_{i=0}^{R-1} C_i^{-P_i(\mathsf{Chall}) \bmod N} \\
\equiv h^{P'(\mathsf{Chall})} \cdot w^{\mathsf{Chall}^r \cdot N} \cdot \prod_{k=0}^{r-1} \rho_k^{-\mathsf{Chall}^k \cdot N} \cdot \prod_{i=0}^{R-1} & C_i^{-P_i(\mathsf{Chall})}
\end{aligned}$$

[8] This can be efficiently achieved using the Cantor-Zassenhaus algorithm [19], which is a probabilistic algorithm with small failure probability. The CI hash function of [67] is compatible with $\mathsf{BadChallenge}$ functions failing with negligible probability.

$$\equiv h^{\mathsf{Chall}^r \cdot y} \cdot w^{\mathsf{Chall}^r \cdot N} \cdot \prod_{k=0}^{r-1} (h^{-\mathsf{Chall}^k \mu_k} \cdot \rho_k^{-\mathsf{Chall}^k \cdot N}) \cdot \prod_{i=0}^{R-1} C_i^{-\delta_{i,\ell} \cdot \mathsf{Chall}^r - \sum_{k=0}^{r-1} p_{i,k} \cdot \mathsf{Chall}^k}$$

$$\equiv (h^y \cdot w^N)^{\mathsf{Chall}^r} \cdot \prod_{k=0}^{r-1} (h^{\mu_k} \cdot \rho_k^N)^{-\mathsf{Chall}^k} \cdot C_\ell^{-\mathsf{Chall}^r} \cdot \prod_{i=0}^{R-1} C_i^{-\sum_{k=0}^{r-1} p_{i,k} \cdot \mathsf{Chall}^k}$$

$$\equiv \prod_{k=0}^{r-1} (h^{\mu_k} \cdot \rho_k^N)^{-\mathsf{Chall}^k} \cdot \prod_{k=0}^{r-1} \prod_{i=0}^{R-1} C_i^{-p_{i,k} \cdot \mathsf{Chall}^k} \equiv \prod_{k=0}^{r-1} C_{d_k}^{-\mathsf{Chall}^k} \pmod{N^2}.$$

Lemma 4.2. *The above construction is a trapdoor Σ-protocol for $\mathcal{L}_\vee^{1\text{-}R}$.*

Proof. Lemma 4.3 shows that, on a CRS generated by $\mathsf{Gen}_{\mathcal{L}}$, the Σ-protocol is statistically special honest-verifier zero-knowledge when the orders of $h \in \mathbb{Z}_{N^2}^*$ and $\bar{h} \in \mathbb{Z}_{\bar{N}^2}^*$ are multiples of N and $\bar{N}$, respectively.

When h and $\bar{h}$ are composite residues (thus making the commitments perfectly binding), we show that $\mathsf{BadChallenge}$ correctly identifies all the possible bad challenges for any first message sent by the prover. We only consider the case where $\ell_j \in \{0,1\}$ since, otherwise, we can simply rely on the correctness of $\mathsf{BadChallenge}^{0\text{-}1}$. At step 2, the polynomials $\{P_i[X]\}_{i=0}^{R-1}$ that $\mathsf{BadChallenge}$ computes from $\{\ell_j, \bar{a}_j\}_{j=1}^r$ are of the form $P_i[X] = \delta_{i,\ell} \cdot X^r + \sum_{k=0}^{r-1} p_{i,k} \cdot X^k \in \mathbb{Z}_N[X]$, by construction. By the verification equations (18), any valid challenge-response pair $\big(\mathsf{Chall}, \big(z_y, z_w, \{(\bar{z}_j, \bar{z}_{d,j}, \bar{z}_{e,j}, \bar{z}_{u,j}, \bar{z}_{v,j})\}_{j=1}^r\big)\big)$ must involve elements $\{\bar{z}_j\}_{j=1}^r$ that satisfy the conditions $\bar{z}_j = \bar{a}_j + \mathsf{Chall} \cdot \ell_j$ over the integers, and not only modulo $\bar{N}$, due to the lower bound on $\bar{z}_j$ and since ℓ_j is a bit (see Remark 4.1). Hence, the polynomials $\{F_{j,1}[X]\}_{j=1}^r$ computed at step 2 of $\mathsf{BadChallenge}$ satisfy $\bar{z}_j = F_{j,1}(\mathsf{Chall}) = F_{j,1}(\mathsf{Chall}) \bmod N$. Moreover, from equation (19), we have

$$h^{z_y} \cdot z_w^N \equiv \prod_{k=0}^{r-1} C_{d_k}^{-\mathsf{Chall}^k} \cdot \prod_{i=0}^{R-1} C_i^{P_i(\mathsf{Chall}) \bmod N} \tag{20}$$

$$\equiv \prod_{k=0}^{r-1} C_{d_k}^{-\mathsf{Chall}^k} \cdot \prod_{i=0}^{R-1} C_i^{(\delta_{i,\ell} \cdot \mathsf{Chall}^r + \sum_{k=0}^{r-1} p_{i,k} \cdot \mathsf{Chall}^k \bmod N)} \pmod{N^2}.$$

By decrypting all members of (20), we see that a bad challenge must be a root of $Q[X]$ over $\mathbb{Z}_N$ since $\mathcal{D}_{\tau_\Sigma}(h) = 0$. Since a bad challenge Chall satisfies

$$Q(\mathsf{Chall}) = c_\ell \cdot \mathsf{Chall}^r + \sum_{k=0}^{r-1} \left(\Big(\sum_{i=0}^{R-1} c_i \cdot p_{i,k} \Big) - c_{d_k} \right) \cdot \mathsf{Chall}^k \equiv 0 \pmod N,$$

it also satisfies $Q(\mathsf{Chall}) \equiv 0 \pmod p$ and $Q(\mathsf{Chall}) \equiv 0 \pmod q$. Given that $\gcd(c_\ell, N) \in \{1, p, q\}$ and $\gcd(c_\ell, N/\gcd(c_\ell, N)) = 1$, we have either $c_\ell \not\equiv 0 \pmod p$ or $c_\ell \not\equiv 0 \pmod q$, meaning that at least one the polynomials $Q_p[X]$

and $Q_q[X]$ is a non-zero degree-r polynomial over a prime field. Since $p, q > 2^\lambda$, we know that any $\mathsf{Chall} \in \{0, \ldots, 2^\lambda - 1\}$ fits in both $\mathbb{Z}_p$ and $\mathbb{Z}_q$. The condition $Q(\mathsf{Chall}) \equiv 0 \pmod N$ then implies that a bad challenge $\mathsf{Chall} \in \{0, \ldots, 2^\lambda - 1\}$ necessarily satisfies $Q_p(\mathsf{Chall}) = 0 \bmod p$ and $Q_q(\mathsf{Chall}) = 0 \bmod q$. This shows that $\mathsf{BadChallenge}$ always identifies all the bad challenges at step 2. $\square$

Following [46] and standard Σ-protocols over the integers, the above Σ-Protocol $\Pi_\vee^{1\text{-}R} = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ is statistically special honest-verifier zero-knowledge. Although the adversary can choose Paillier commitments $\{C_i\}_{i=0}^{R-1}$ of its own (which may be N-th residues or not), we can rely on the fact that h has a component of order N to perfectly randomize commitments $\{C_{d_k}\}_{k=0}^{r-1}$ over the full group $\mathbb{Z}_{N^2}^*$ even if some of the $\{C_i\}_{i=0}^{R-1}$ are maliciously generated.

Lemma 4.3. *For any language $\mathcal{L}_\vee^{1\text{-}R}(h, \bar{h})$ such that N divides the order of $h \in \mathbb{Z}_{N^2}^*$ and $\bar{N}$ divides the order of $\bar{h} \in \mathbb{Z}_{\bar{N}^2}^*$, $\Pi_\vee^{1\text{-}R}(h, \bar{h})$ is statistically special honest-verifier zero-knowledge.* (The proof is in Supplementary Material D.4.)

5 Logarithmic-Size Ring Signatures in the Standard Model from DCR and LWE

In Supplementary Material E, we give a simplified version of the scheme where the signer erases its random coins after each signature generation.

The proof of unforgeability departs from [46] in that we cannot replay the adversary with a different random oracle. Instead, we use Paillier as a dual-mode commitment, which is made extractable at some step to enable the extraction of bits $\ell_1^\star \ldots \ell_r^\star \in \{0,1\}^r$ from the commitments $\{L_j^\star\}_{j=1}^r$ contained in the forgery $\boldsymbol{\Sigma}^\star = ((L_1^\star, \ldots, L_r^\star), \boldsymbol{\pi}^\star)$. The next step is to have the reduction guess which honestly generated public key $vk^{(i^\star)}$ will belong to the signer identified by decoding the forgery. Then, $vk^{(i^\star)}$ is replaced by a random element of $\mathbb{Z}_{N^2}^*$ in order to force the adversary to break the simulation-soundness of Π^{uss} by arguing that $vk^{(i^\star)}$ is a commitment to 0, which it is not. The use of two distinct moduli allows us to decode $\ell_1^\star, \ldots, \ell_r^\star \in \{0,1\}^r$ from $\{L_j^\star\}_{j=1}^r$ (which is necessary to check that $\ell^\star = \ell_1^\star \ldots \ell_r^\star$ still identifies the expected verification key $vk^{(i^\star)}$) even when we rely on the DCR assumption to modify the distribution of $vk^{(i^\star)}$.

The security proof of our simplified scheme relies on erasures because the NIZK simulator is used in all signing queries. If the adversary makes a corruption query $\mathsf{Corrupt}(i)$ after a signing query involving $sk^{(i)}$, the challenger's loophole is to claim that it erased the signer's randomness in signing queries of the form $(i, \cdot, \cdot)$.

To avoid erasures, we adapt the security proof in such a way that the NIZK simulator only simulates signatures on behalf of the expected target user $i^\star$. All other users' signatures are faithfully generated, thus allowing the challenger to reveal consistent randomness explaining their generation. Since user $i^\star$ is not corrupted with noticeable probability, the challenger never has to explain the generation of a simulated signature. This strategy raises a major difficulty since decoding $\ell_1^\star \dots \ell_r^\star$ from $\{L_j^\star\}_{j=1}^r$ is only possible when these are extractable commitments. Unfortunately, the NIZK simulator cannot answer signing queries $(i^\star, \cdot, \cdot)$ by computing $\{L_j\}_{j=1}^r$ as perfectly binding commitments as this would not preserve the statistical ZK property of the Σ-protocol of Section 4.2. Moreover, relying on computational ZK does not work because we need the guessed index $i^\star$ to be statistically independent of the adversary's view until the forgery stage. If we were to simulate signatures using computational NIZK proofs, they would information-theoretically leak the index $i^\star$ of the only user for which the NIZK simulator is used in signing queries $(i^\star, \cdot, \cdot)$. To resolve this problem, we use a tag-based commitment scheme which is perfectly hiding in all signing queries and extractable in the forgery (with noticeable probability).

We thus commit to the string $\ell \in \{0,1\}^r$ using the dense $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme of Section 3.2. We use the property that, depending on which tag is used to generate a commitment, it either behaves as perfectly hiding or extractable commitment. In the perfectly hiding mode, we also exploit its density property to ensure the statistical ZK property.

The construction uses the trapdoor Σ-protocol of Section 4.2 to prove membership of the parametrized language

$$\begin{aligned}\mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h}_{\mathsf{VK}}) := \big\{ \big((C_0, \dots, C_{R-1}), (L_1, \dots, L_r) \big) \in (\mathbb{Z}_{N^2}^*)^R \times (\mathbb{Z}_{\bar{N}^2}^*)^r \quad | \\ \exists y \in \mathbb{Z}_N,\ w \in \mathbb{Z}_N^*,\ s_1, \dots, s_r \in \mathbb{Z}_{\bar{N}},\ t_1, \dots, t_r \in \mathbb{Z}_{\bar{N}}^*, \\ (\ell_1, \dots, \ell_r) \in \{0,1\}^r \quad : \quad C_\ell = h^y \cdot w^N \bmod N^2 \\ \wedge \quad L_j = (1+\bar{N})^{\ell_j} \cdot \bar{h}_{\mathsf{VK}}^{s_j} \cdot t_j^{\bar{N}} \bmod \bar{N}^2 \qquad \forall j \in [r] \ \big\},\end{aligned} \tag{21}$$

with $R = 2^r$ and $\ell = \sum_{j=1}^r \ell_j \cdot 2^{j-1}$, where $\bar{h}_{\mathsf{VK}}$ changes in each signature.

The construction relies on the following ingredients:

- A trapdoor Σ-protocol $\Pi' = (\mathsf{Gen}'_{\mathsf{par}}, \mathsf{Gen}'_{\mathcal{L}}, \mathsf{P}', \mathsf{V}')$ for the parametrized language $\mathcal{L}_{\vee}^{1\text{-}R}$ defined in (21).
- A strongly unforgeable one-time signature scheme $\mathsf{OTS} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys of length $\ell_v \in \mathsf{poly}(\lambda)$.
- An admissible hash function $\mathsf{AHF} : \{0,1\}^{\ell_v} \to \{0,1\}^L$, for some $L \in \mathsf{poly}(\lambda)$.
- A dense $\mathcal{R}$-lossy PKE scheme $\mathcal{R}\text{-}\mathsf{LPKE} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ for $\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$, where $\mathcal{K} = \{0,1,\perp\}^L$ and $\mathcal{T} = \{0,1\}^L$.

Our erasure-free ring signature goes as follows.

CRSGen(1^λ) : Given a security parameter λ, conduct the following steps.

1. Generate $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ for the trapdoor Σ-protocol of Section 4.2.
2. Generate an RSA modulus $N = pq$ and choose an element $h \hookleftarrow U(\mathbb{Z}_{N^2}^*)$, which has order divisible by N w.h.p.
3. Choose an admissible hash function $\mathsf{AHF} : \{0,1\}^{\ell_v} \to \{0,1\}^L$. Generate public parameters $\Gamma \hookleftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^{|N|})$ for the dense $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme of Section 3.2 with $\zeta = 1$, which is associated with the bit-matching relation $\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$. Choose a random initialization

value $K \hookleftarrow U(\mathcal{K})$ and generate lossy keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$. Parse pk as $\mathsf{pk} := \big(\bar{N}, \{\bar{v}_{i,b}\}_{i \in [L], b \in \{0,1\}}\big)$, for an RSA modulus $\bar{N} = \bar{p}\bar{q}$, where $\bar{v}_{i,b} \sim U(\mathbb{Z}^*_{\bar{N}^2})$ for each $i \in [L]$, $b \in \{0,1\}$.

4. Generate a pair $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}^{1\text{-}R}_{\vee})$ comprised of the CRS crs of an USS argument Π^{uss} (recalled in Supplementary Material B) for the language $\mathcal{L}^{1\text{-}R}_{\vee}$ defined in (21) with a simulation trapdoor τ_{zk}. The common reference string crs contains $\mathsf{crs}'_{\mathcal{L}} = \{N, \bar{N}\}$, which is part of a CRS $\mathsf{crs}' = (\{\lambda\}, \mathsf{crs}'_{\mathcal{L}})$ for the Σ-protocol of Section 4.2.

Output the common reference string $\rho = (\mathsf{crs}, h, \mathsf{AHF}, \mathsf{pk}, \Gamma, \mathsf{OTS})$, where OTS is the specification of a one-time signature scheme.

Keygen(ρ) : Pick $w \hookleftarrow U(\mathbb{Z}^*_N)$, $y \hookleftarrow U(\mathbb{Z}_N)$ and compute $C = h^y \cdot w^N \bmod N^2$. Output (sk, vk), where $sk = (w, y)$ and $vk = C$.

Sign$(\rho, sk, M, \mathsf{R})$: Given a ring $\mathsf{R} = \{vk_0, \ldots, vk_{R-1}\}$ (we assume that $R = 2^r$ for some $r \in \mathbb{N}$), a message M and a secret key $sk = (w, y) \in \mathbb{Z}^*_N \times \mathbb{Z}_N$, let $\ell \in \{0, \ldots, R-1\}$ the index such that $vk_\ell = h^y \cdot w^N \bmod N^2$.

1. Generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathsf{OTS}.\mathcal{G}(1^\lambda)$ and let $\mathsf{VK}' = \mathsf{AHF}(\mathsf{VK}) \in \{0,1\}^L$. Compute $\bar{h}_{\mathsf{VK}} = \prod_{j=1}^{L} \bar{v}_{j,\mathsf{VK}'[j]} \bmod \bar{N}^2$.
2. For each $j \in [r]$, choose $s_j \hookleftarrow U(\mathbb{Z}_{\bar{N}})$, $t_j \hookleftarrow U(\mathbb{Z}^*_{\bar{N}})$ and compute a commitment $L_j = (1 + \bar{N})^{\ell_j} \cdot \bar{h}_{\mathsf{VK}}^{s_j} \cdot t_j^{\bar{N}} \bmod \bar{N}^2$.
3. Define $\mathsf{lbl} = \mathsf{VK}$ and compute a NIZK argument $\boldsymbol{\pi} \leftarrow \mathsf{P}\big(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w}, \mathsf{lbl}\big)$ that $\boldsymbol{x} \triangleq ((vk_0, \ldots, vk_{R-1}), (L_1, \ldots, L_r)) \in \mathcal{L}^{1\text{-}R}_{\vee}(h, \bar{h}_{\mathsf{VK}})$ by running the prover P of Supplementary Material B with the Σ-protocol of Section 4.2 using the witness $\boldsymbol{w} = ((\ell_1, \ldots, \ell_r), w, (s_1, \ldots, s_r), (t_1, \ldots, t_r))$.
4. Generate a one-time signature $sig \leftarrow \mathsf{OTS}.\mathcal{S}(\mathsf{SK}, (\boldsymbol{x}, M, \mathsf{R}, \boldsymbol{\pi}))$.

Output the signature $\boldsymbol{\Sigma} = (\mathsf{VK}, (L_1, \ldots, L_r), \boldsymbol{\pi}, sig)$.

Verify$(\rho, M, \boldsymbol{\Sigma}, \mathsf{R})$: Given a signature $\boldsymbol{\Sigma} = (\mathsf{VK}, (L_1, \ldots, L_r), \boldsymbol{\pi}, sig)$, a message M and a ring $\mathsf{R} = \{vk_0, \ldots, vk_{R-1}\}$, return 0 if these do not parse properly. Otherwise, let $\mathsf{lbl} = \mathsf{VK}$ and return 0 if $\mathsf{OTS}.\mathcal{V}(\mathsf{VK}, (\boldsymbol{x}, M, \mathsf{R}, \boldsymbol{\pi}), sig) = 0$. Otherwise, run $\mathsf{V}(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{\pi}, \mathsf{lbl})$ which outputs 1 iff $\boldsymbol{\pi}$ is a valid argument that $\big((vk_0, \ldots, vk_{R-1}), (L_1, \ldots, L_r)\big) \in \mathcal{L}^{1\text{-}R}_{\vee}(h, \bar{h}_{\mathsf{VK}})$.

In Supplementary Material F, we provide concrete efficiency estimations showing that, in terms of signature length, the above realization competes with its random-oracle-model counterpart. We now state our main security results.

Theorem 5.1. *The above scheme provides unforgeability assuming that: (i) The one-time signature* OTS *is strongly unforgeable; (ii) The scheme of Section 3.2 is a secure dense* $\mathcal{R}_{\mathsf{BM}}$*-lossy PKE scheme; (iii) The* DCR *assumption holds; (iv)* Π^{uss} *is an unbounded simulation-sound NIZK argument for the parametrized language* $\mathcal{L}^{1\text{-}R}_{\vee}$*.*

Proof. To prove the result, we consider a sequence of games starting with the real unforgeability experiment and ending with a game where we give a direct reduction from the simulation-soundness of Π^{uss}. In each game, we call W_i the event that the challenger outputs 1.

Game_0: This is the real unforgeability experiment. The adversary $\mathcal{A}$ receives a CRS ρ and is granted access to a key generation oracle Keygen, a signing oracle Sign and a corruption oracle $\mathsf{Corrupt}$. At the i-th query to Keygen, the challenger returns a verification key of the form $vk^{(i)} = h^{y_i} \cdot w_i^N \bmod N^2$ for some $w_i \hookleftarrow U(\mathbb{Z}_N^*)$, $y_i \hookleftarrow U(\mathbb{Z}_N)$ and keeps $sk^{(i)} = (w_i, y_i)$ for later use. If $\mathcal{A}$ makes a corruption query $\mathsf{Corrupt}(i)$, the challenger reveals $sk^{(i)}$. At each signing query (i, M, R), the challenger returns $\perp$ if R contains a key $vk \notin \mathbb{Z}_{N^2}^*$. Otherwise, it runs $\boldsymbol{\Sigma} \leftarrow \mathsf{Sign}(\rho, sk, M, \mathsf{R})$ and returns $\boldsymbol{\Sigma}$ to $\mathcal{A}$. When $\mathcal{A}$ halts, it outputs a triple $(M^\star, \mathsf{R}^\star, \boldsymbol{\Sigma}^\star)$, where $\mathsf{R}^\star = \{vk_0^\star, \ldots, vk_{R-1}^\star\}$ and $\boldsymbol{\Sigma}^\star = (\mathsf{VK}^\star, (L_1^\star, \ldots, L_r^\star), \boldsymbol{\pi}^\star, sig^\star)$, and the challenger outputs 1 if: (i) $\mathsf{Verify}(\rho, M^\star, \boldsymbol{\Sigma}^\star, \mathsf{R}^\star) = 1$; (ii) $\mathsf{R}^\star$ only consists of uncorrupted keys produced by the Keygen oracle; (iii) No signing query $(\cdot, M^\star, \mathsf{R}^\star)$ was made. By definition, we have $\Pr[W_0] = \mathbf{Adv}_{\mathcal{A}}^{\mathrm{unforge}}(\lambda)$.

Game_1: This is like Game_0 except that we introduce a failure event. The challenger initially chooses a random index $i^\star \hookleftarrow U([Q_V])$, where Q_V is the number of Keygen-queries. The challenger $\mathcal{B}$ outputs 0 if $\mathcal{A}$ outputs a forgery $\boldsymbol{\Sigma}^\star = (\mathsf{VK}^\star, (L_1^\star, \ldots, L_r^\star), \boldsymbol{\pi}^\star, sig^\star)$ containing a $\mathsf{VK}^\star$ that coincides with a verification key contained in the output of a signing query of the form $(i^\star, \cdot, \cdot)$. If the failure event does not occur, the challenger outputs the same result as in Game_0. The strong unforgeability of OTS implies that $\Pr[W_1]$ cannot noticeably differ from $\Pr[W_0]$. We can easily turn $\mathcal{B}$ into a one-time-signature forger such that $|\Pr[W_1] - \Pr[W_0]| \leq Q_V \cdot Q_S \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathrm{ots}}(\lambda)$.

Game_2: This is like Game_1 with one change at step 3 of CRSGen. Instead of sampling $K \hookleftarrow U(\mathcal{K})$ uniformly, the challenger runs $K \leftarrow \mathsf{AdmSmp}(1^\lambda, Q_S, \delta)$ to generate a key $K \in \{0,1,\perp\}^L$ for an admissible hash function $\mathsf{AHF} : \{0,1\}^{\ell_v} \to \{0,1\}^L$, where Q_S is an upper bound on the number of signing queries. Then, the sampled key K is used as an initialization value to generate $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$. By the second indistinguishability property of the dense $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme (which holds in the statistical sense), changing the initialization value does not impact $\mathcal{A}$'s view. In particular, the scheme of Section 3.2 has the property that the distribution of lossy public keys is perfectly independent of K. It follows that $\Pr[W_2] = \Pr[W_1]$.

Game_3: This game is like Game_2 with one change. When $\mathcal{A}$ halts and outputs $\boldsymbol{\Sigma}^\star = (\mathsf{VK}^\star, (L_1^\star, \ldots, L_r^\star), \boldsymbol{\pi}^\star, sig^\star)$, the challenger checks if the conditions

$$F_{\mathsf{ADH}}(K, \mathsf{VK}^{(1)}) = \cdots = F_{\mathsf{ADH}}(K, \mathsf{VK}^{(Q_S)}) = 1 \ \wedge \ F_{\mathsf{ADH}}(K, \mathsf{VK}^\star) = 0 \quad (22)$$

are satisfied, where $\mathsf{VK}^\star$ is the one-time verification key in the forgery and $\mathsf{VK}^{(1)}, \ldots, \mathsf{VK}^{(Q_S)}$ are those involved in signing queries of the form $(i^\star, \cdot, \cdot)$. If these conditions do not hold, the challenger aborts and returns 0. For simplicity, we assume that $\mathcal{B}$ aborts at the beginning of the game if it detects that there exists $j \in [Q_S]$ such that $F_{\mathsf{ADH}}(K, \mathsf{VK}^{(j)}) = 0$ (recall that the verification keys $\{\mathsf{VK}^{(j)}\}_{j=1}^{Q_S}$ used in signing queries $(i^\star, \cdot, \cdot)$ can be chosen at the outset of the game by $\mathcal{B}$). If conditions (22) are satisfied, the challenger returns 1 whenever the challenger of Game_2 does. Letting Fail denote the event that $\mathcal{B}$ aborts because (22) does not hold, we have $W_3 = W_2 \wedge \neg\mathsf{Fail}$. Since K is perfectly independent of the distribution of keys produced by $\mathsf{LKeygen}$, we can apply Theorem 2.6 to argue that there is a noticeable function $\delta(\lambda)$ such that $\Pr[\neg\mathsf{Fail}] \geq \delta(\lambda)$. This implies

$$\Pr[W_3] = \Pr[W_2 \wedge \neg\mathsf{Fail}] \geq \delta(\lambda) \cdot \Pr[W_2] \ , \tag{23}$$

where the inequality stems from the fact that Fail is independent of W_1 since K is statistically independent of $\mathcal{A}$'s view.

We note that, if conditions (22) are satisfied in Game_3, the sequence of one-time verification keys $(\mathsf{VK}^{(1)}, \ldots, \mathsf{VK}^{(Q_S)}, \mathsf{VK}^\star)$ satisfies $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{VK}^\star) = 1$ and $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{VK}^{(j)}) = 0$ for all $j \in [Q_S]$.

Game_4: We modify the distribution of crs. At step 3 of CRSGen, we generate the keys for $\mathcal{R}$-LPKE as injective keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ instead of lossy keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$. The weak indistinguishability property (i) of $\mathcal{R}$-LPKE (see Definition 2.10) ensures that $\Pr[W_4]$ and $\Pr[W_3]$ are negligibly far apart. Indeed, we can immediately build a reduction $\mathcal{B}$ from this property of $\mathcal{R}$-LPKE in order to transition from Game_3 to Game_4. We thus have $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{indist\text{-}LPKE\text{-}1}}(\lambda) \leq \mathbf{Adv}^{\mathsf{DCR}}(\lambda)$.

Game_5: In this game, the challenger uses the secret key $\mathsf{sk} = (\bar{p}, \bar{q}, K)$ produced by $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ (which contains the factors of $\bar{N} = \bar{p}\bar{q}$) to extract the content of commitments $\{(L_j^\star, \bar{A}_j^\star)\}_{j=1}^r$ contained in $\mathcal{A}$'s forgery. Note that, if the conditions (22) introduced in Game_3 are satisfied, $\mathcal{A}$'s output $\boldsymbol{\Sigma}^\star = (\mathsf{VK}^\star, (L_1^\star, \ldots, L_r^\star), \boldsymbol{\pi}^\star, sig^\star)$ involves an injective tag $\mathsf{VK}^\star$, which allows extracting the content of all commitments generated for this tag. If the challenger does not fail, it thus obtains $\{(\ell_j^\star, \bar{a}_j^\star)\}_{j=1}^r$ and also outputs 0 if there exists $j \in [r]$ such that $\ell_j^\star \not\in \{0,1\}$. Otherwise, it obtains a string $\ell_1^\star \ldots \ell_r^\star \in \{0,1\}^r$ and reconstructs the index $\ell^\star = \sum_{k=1}^r \ell_k^\star \cdot 2^{k-1} \in \mathbb{Z}_R$ of the verification key $vk_{\ell^\star}^\star = C_{\ell^\star}^\star$ in the ring $\mathsf{R}^\star = \{vk_0^\star, \ldots, vk_{R-1}^\star\}$. If $\ell_j^\star \not\in \{0,1\}$ for some $j \in [r]$, the soundness of Π^{uss} (and thus its simulation-soundness) is broken and we have $|\Pr[W_5] - \Pr[W_4]| \leq 2^{-\Omega(\lambda)} + \mathbf{Adv}^{\mathrm{uss}}(\lambda)$.

Game_6: This game is like Game_5 with the following change. Recall that, in Game_1 and subsequent games, the challenger initially chooses a random index $i^\star \hookleftarrow U([Q_V])$. Now, we let the challenger abort and output 0 if the verification key $vk_{\ell^\star}^\star$ contained in $\mathsf{R}^\star$ does not coincide with the verification key $vk^{(i^\star)}$ returned by the challenger at the $i^\star$-th query to the Keygen oracle. In addition, the challenger aborts and outputs 0 if $\mathcal{A}$ makes the corruption query $\mathsf{Corrupt}(i^\star)$. Otherwise (i.e., if $vk_{\ell^\star}^\star = vk^{(i^\star)}$ and $sk^{(i^\star)}$ is not corrupted), the challenger outputs the same bit as in Game_5. Since $i^\star$ is chosen independently of $\mathcal{A}$'s view, it is correct with probability $1/Q_V$, where Q_V is the number of Keygen-queries. We thus have $\Pr[W_6] = \Pr[W_5]/Q_V$.

Game_7: This game is identical to Game_6 except that, in all signing queries (i, M, R), such that $i = i^\star$,[9] the challenger simulates the Sign oracle by running the NIZK simulator of Π^{uss} instead of using the real witness. Namely, the commitments $\{L_j\}_{j=1}^r$ of each signature $\boldsymbol{\Sigma} = (\mathsf{VK}, (L_1, \ldots, L_r), \boldsymbol{\pi}, sig)$ still commit to the binary decomposition $(\ell_1, \ldots, \ell_r)$ of $vk^{(i^\star)}$'s location in R but $\boldsymbol{\pi}$ is simulated without using $sk^{(i^\star)}$ nor $((\ell_1, \ldots, \ell_r), t_1, \ldots, t_r)$. Recall that, when $h \in \mathbb{Z}_{N^2}^\star$ and $\bar{h} = \bar{h}_{\mathsf{VK}} \in \mathbb{Z}_{\bar{N}^2}^*$ have order at least N and $\bar{N}$, respectively, the trapdoor Σ-protocol of Section 4.2 is statistically special zero-knowledge. Also, all commitments generated using $\mathcal{R}$-LPKE are perfectly hiding when $\bar{h}_{\mathsf{VK}}$ has order at least $\bar{N}$. Moreover, the latter condition is met when (22) holds since, in this case, the zero-knowledge simulator computes the commitments $(L_1, \ldots, L_r)$ and $\{(\bar{A}_j, \bar{B}_j)\}_{j=1}^r$ on lossy tags $\mathsf{VK}^{(j)}$. The statistical ZK properties of Π^{uss} and the underlying trapdoor Σ-protocol then ensure that $|\Pr[W_7] - \Pr[W_6]| \leq Q_S \cdot 2^{-\Omega(\lambda)}$.

Game_8: This game is like Game_7 except that we change the distribution of the CRS $\rho = (\mathsf{crs}, h, \mathsf{pk}, \Gamma, \mathsf{OTS})$. Instead of sampling h uniformly in $\mathbb{Z}_{N^2}^*$, we now choose it as a random N-th residue. Namely, the challenger now sets $h = h_0^N \bmod N^2$, where $h_0 \hookleftarrow U(\mathbb{Z}_N^*)$. Under the DCR assumption in $\mathbb{Z}_{N^2}^*$, this changes has no noticeable impact on $\mathcal{A}$'s forging probability and a straightforward reduction shows that $|\Pr[W_8] - \Pr[W_7]| \leq \mathbf{Adv}^{\mathsf{DCR}}(\lambda)$.

We note that, due to the modification introduced in Game_8, all public keys produced by the Keygen oracle live in the subgroup of N-th residues in $\mathbb{Z}_{N^2}^*$.

Game_9: We change the distribution of $vk^{(i^\star)} = C^{(i^\star)}$ and sample $C^{(i^\star)} \hookleftarrow U(\mathbb{Z}_{N^2}^*)$ uniformly instead of sampling it as an N-th residue in $\mathbb{Z}_{N^2}^*$. Since the secret key $sk^{(i^\star)}$ is not used in Game_8, we can rely on the DCR assumption in $\mathbb{Z}_{N^2}^*$ and argue that $|\Pr[W_9] - \Pr[W_8]| \leq \mathbf{Adv}^{\mathsf{DCR}}(\lambda)$.

In Game_9, we claim that $\Pr[W_9] \leq \mathbf{Adv}^{\mathrm{uss}}(\lambda) + 2^{-\Omega(\lambda)}$ as, except with probability $1/N < 2^{-\Omega(\lambda)}$, the challenger can only output 1 if $\mathcal{A}$ breaks the simulation-soundness of Π^{uss}. Indeed, W_9 only occurs if $vk_{\ell^\star}^\star = vk^{(i^\star)}$ (which implies that no query $\mathsf{Corrupt}(i^\star)$ was made); $\boldsymbol{\pi}^\star$ is a valid argument for the statement $((vk_0^\star, \ldots, vk_{R-1}^\star), (L_1^\star, \ldots, L_r^\star)) \in \mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h}_{\mathsf{VK}^\star})$; and no signing query $(i, M^\star, \mathsf{R}^\star)$ has been made for any $vk^{(i)} \in \mathsf{R}^\star$ (in particular for $i = i^\star$). Since $vk_{\ell^\star}$ was sampled uniformly in $\mathbb{Z}_{N^2}^*$, it is *not* an N-th residue except with probability $1/N$. This shows that, except with probability $2^{-\Omega(\lambda)}$, W_9 only occurs when $\boldsymbol{\pi}^\star$ is an accepting argument for a false statement $\boldsymbol{x}^\star \in \mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h}_{\mathsf{VK}^\star})$ on an unqueried label $\mathsf{lbl}^\star \triangleq \mathsf{VK}^\star$.

[9] Signing queries (i, M, R) with $i \neq i^\star$ are still faithfully answered in such a way that we do not need to rely on erasures when users $i \neq i^\star$ are corrupted after a signing query of the form $(i, ., .)$.

Putting the above altogether, we can bound the adversary's advantage as

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{unforge}}(\lambda) \leq \frac{1}{\delta} \cdot (2Q_V + 1) \cdot \Big(\mathbf{Adv}^{\mathrm{uss}}(\lambda) + \mathbf{Adv}^{\mathsf{DCR}}(\lambda) + Q_S \cdot 2^{-\Omega(\lambda)}\Big) + Q_S \cdot Q_V \cdot \mathbf{Adv}^{\mathrm{ots}}(\lambda)$$

where Q_V is the number of Keygen-queries and Q_S is the number of signing queries. □

The proof of anonymity follows from the fact that all commitments are perfectly hiding when the CRS ρ is configured as in the real scheme. The proof of Theorem 5.2 is identical to that of Theorem E.2 in Supplementary Material E.

Theorem 5.2. *The above construction instantiated with the trapdoor Σ-protocol of Section 4.2 provides full anonymity under key exposure provided Π^{uss} is a statistical NIZK argument for the language $\mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h}_{\mathsf{VK}})$ of (21) when the order of h is a multiple of N and the order of $\bar{h}_{\mathsf{VK}}$ is a multiple of $\bar{N}$.*

Acknowledgements

We thank Zesheng Li for pointing out a mistake in the scheme of Section 3 in an earlier version of the paper. This research was partially funded by the French ANR ALAMBIC project (ANR-16-CE39-0006). Khoa Nguyen was supported in part by the Gopalakrishnan - NTU PPF 2018, by A*STAR, Singapore under research grant SERC A19E3b0099, and by Vietnam National University HoChiMinh City (VNU-HCM) under grant number NCM2019-18-01. Thomas Peters is a research associate of the Belgian Fund for Scientific Research (F.R.S.-FNRS).

References

1. M. Abe, M. Ohkubo, K. Suzuki. 1-out-of-n signatures from a variety of keys. *Asiacrypt*, 2002.
2. M. Abe, M. Ambrona, A. Bogdanov, M. Ohkubo, A. Rosen. Non-Interactive Composition of Sigma-Protocols via Share-then-Hash. *Asiacrypt*, 2020.
3. M. Backes, N. Döttling, L. Hanzlik, K. Kluczniak, J. Schneider. Ring signatures: Logarithmic-size, no setup — from standard assumptions. *Eurocrypt*, 2019.
4. M. Bellare, D. Hofheinz, S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. *Eurocrypt*, 2009.
5. M. Bellare, S. Yilek. Encryption Schemes Secure under Selective Opening Attack. Cryptology ePrint Archive: Report 2009/101.
6. A. Bender, J. Katz, R. Morselli. Ring Signatures: Stronger Definitions, and Constructions without Random Oracles. *J. Cryptology*, 22(1), 2009.
7. D. Boneh, X. Boyen. Secure identity based encryption without random oracles. *Crypto*, 2004.
8. X. Boyen. Mesh signatures. *Eurocrypt*, 2007.
9. E. Boyle, G. Segev, D. Wichs. Fully leakage-resilient signatures. *Eurocrypt*, 2011.

10. Z. Brakerski, V. Koppula, T. Mour. NIZK from LPN and trapdoor hash via correlation intractability for approximable relations. *Crypto*, 2020.
11. Z. Brakerski, Y. Tauman-Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. Cryptology ePrint Archive: Report 2010/086, 2010.
12. E. Bresson, J. Stern, M. Szydlo. Threshold ring signatures and applications to ad-hoc groups. *Crypto*, 2002.
13. J. Camenisch, V. Shoup. Practical Verifiable Encryption and Decryption of Discrete Logarithms. *Crypto*, 2003.
14. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. Rothblum, R. Rothblum. Fiat-Shamir from simpler assumptions. Cryptology ePrint Archive: Report 2018/1004.
15. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. Rothblum, R. Rothblum, D. Wichs. Fiat-Shamir: From practice to theory. *STOC*, 2019.
16. R. Canetti, Y. Chen, L. Reyzin, and R. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. *Eurocrypt*, 2018.
17. R. Canetti, O. Goldreich, S. Halevi. The random oracle methodology, revisted. *J. of the ACM*, 51(4), 2004.
18. R. Canetti, A. Lombardi, D. Wichs. Fiat-Shamir: From Practice to Theory, Part II (NIZK and Correlation Intractability from Circular-Secure FHE). Cryptology ePrint Archive: Report 2018/1248.
19. D. Cantor, H. Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, 1981.
20. D. Catalano, R. Gennaro, N. Howgrave-Graham, P. Nguyen. Paillier's cryptosystem revisited. *ACM-CCS*, 2001.
21. P. Chaidos, J. Groth. Making Sigma-protocols non-interactive without random oracles. *PKC*, 2015.
22. N. Chandran, J. Groth, A. Sahai. Ring Signatures of Sub-linear Size Without Random Oracles. *ICALP*, 2007.
23. M. Chase, A. Lysyanskaya. On signatures of knowledge. *Crypto*, 2006.
24. R. Chatterjee, S. Garg, M. Hajiabadi, D. Khurana, X. Liang, G. Malavolta, O. Pandey, S. Shiehian. Compact Ring Signatures from Learning With Errors. *Crypto*, 2021.
25. D. Chaum, T. Pedersen. Wallet databases with observers. *Crypto*, 1992.
26. A. Choudhuri, P. Hubacek, K. C., K. Pietrzak, A. Rosen, G. Rothblum. Finding a Nash equilibrium is no easier than breaking Fiat-Shamir. *STOC*, 2019.
27. M. Ciampi, R. Parisella, D. Ventury. On adaptive security of delayed-input Sigma protocols and Fiat-Shamir NIZKs. *SCN*, 2020.
28. G. Couteau, D. Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. *Crypto*, 2020.
29. G. Couteau, S. Katsumata, B. Ursu. Non-interactive zero-knowledge in pairing-free groups from weaker assumptions. *Eurocrypt*, 2020.
30. R. Cramer, I. Damgård, J.-B. Nielsen. Multiparty computation from threshold homomorphic encryption. *Eurocrypt*, 2001.
31. R. Cramer, I. Damgård, B. Schoenmaekers. Proofs of partial knowledge and simplified design of witness hiding protocols. *Crypto*, 1994.
32. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. *Eurocrypt*, 2000.
33. I. Damgård, N. Fazio, A. Nicolosi. Non-interactive Zero-Knowledge from Homomorphic Encryption. *TCC*, 2006.
34. I. Damgård, M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. *PKC*, 2001.

35. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, A. Sahai. Robust non-interactive zero-knowledge. *Crypto*, 2001.
36. Y. Dodis, A. Kiayias, A. Nicolosi, V. Shoup. Anonymous identification in ad hoc groups. *Eurocrypt*, 2004.
37. M. Esgin, R. Steinfeld, J. Liu, D. Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. *Crypto*, 2019.
38. M. Esgin, R. Zhao, R. Steinfeld, J. Liu, D. Liu. MatRiCT: Efficient, scalable and post-quantum blockchain confidential transactions protocol. *ACM-CCS*, 2019.
39. U. Feige, D. Lapidot, A. Shamir. Multiple non-interactive zero-knowledge under general assumptions. *SIAM J. of Computing*, 29(1), 1999.
40. A. Fiat, A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. *Crypto*, 1986.
41. C. Gentry, C. Peikert, V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. *STOC*, 2008.
42. S. Goldwasser, S. Micali, C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 1989.
43. S. Goldwasser, Y. Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. *FOCS*, 2003.
44. A. González. Shorter ring signatures from standard assumptions. *PKC*, 2019.
45. M. Green, B.-W. Ladd, I. Miers. A Protocol for Privately Reporting Ad Impressions at Scale. *ACM-CCS*, 2016.
46. J. Groth, M. Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. *Eurocrypt*, 2015.
47. L. Guillou, J.-J. Quisquater. A "paradoxical" indentity-based signature scheme resulting from zero-knowledge. *Crypto*, 1988.
48. B. Hemenway, B. Libert, R. Ostrovsky, D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. *Asiacrypt*, 2011.
49. D. Hofheinz, T. Jager, A. Rupp. Public-Key Encryption with Simulation-Based Selective-Opening Security and Compact Ciphertexts. *TCC*, 2016-B.
50. J. Holmgren, A. Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). *FOCS*, 2018.
51. J. Holmgren, A. Lombardi, R. Rothblum. Fiat-Shamir via List-Recoverable Codes (or: Parallel Repetition of GMW is not Zero-Knowledge). *STOC*, 2021.
52. T. Jager. Verifiable random functions from weaker assumptions. *TCC*, 2015.
53. R. Jawale, Y. Tauman-Kalai, D. Khurana, R. Zhang. SNARGs for bounded depth computations and PPAD hardness from sub-exponential LWE. *STOC*, 2021.
54. E. Kiltz. Chosen-ciphertext security from tag-based encryption. *TCC*, 2006.
55. B. Libert, S. Ling, K. Nguyen, H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. *Eurocrypt*, 2016.
56. B. Libert, K. Nguyen, A. Passelègue, R. Titiu. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. *Asiacrypt*, 2020.
57. B. Libert, T. Peters, C. Qian. Logarithmic-size ring signatures with tight security from the DDH assumption. *ESORICS*, 2018.
58. H. Lipmaa. Optimally Sound Sigma Protocols Under DCRA. *FC*, 2017.
59. A. Lombardi, V. Vaikuntanathan. PPAD-hardness and VDFs based on iterated squaring, in the standard model. *Crypto*, 2020.
60. G. Malavolta, D. Schröder. Efficient ring signatures in the standard model. *Asiacrypt*, 2017.

61. P. Mohassel. One-time signatures and chameleon hash functions. *SAC*, 2010.
62. S. Noether. Ring signature confidential transactions for monero. Cryptology ePrint Archive Report 2015/1098, 2015.
63. T. Okamoto, S. Uchiyama. A New Public-Key Cryptosystem as Secure as Factoring. *Eurocrypt*, 1998.
64. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Eurocrypt*, 1999.
65. S. Park, A. Sealfon. It wasn't me! repudiability and unclaimability of ring signatures. *Crypto*, 2019.
66. R. Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. *TCC*, 2013.
67. C. Peikert, S. Shiehian. Non-interactive zero knowledge for NP from (plain) Learning With Errors. *Crypto*, 2019.
68. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *STOC*, 2005.
69. R. Rivest, A. Shamir, Y. Tauman. How to Leak a Secret. *Asiacrypt*, 2001.
70. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. *FOCS*, 1999.
71. H. Shacham, B. Waters. Efficient ring signatures without random oracles. *PKC*, 2007.
72. Y. Tauman Kalai, G. Rothblum, R. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. *Crypto*, 2017.
73. A. Young, M. Yung. Questionable encryption and its applications. *Mycrypt*, 2005.

Supplementary Material

A Non-Interactive Zero-Knowledge and Simulation-Sound Arguments

We recall the definitions of NIZK proofs. Since it is sufficient for our applications, we allow the common reference string to be generated as a function of the language $\mathcal{L}$.

In addition, we consider NIZK argument systems where each argument comes with a label lbl taken as input by both the prover and the verifier.

Definition A.1. *A non-interactive zero-knowledge (NIZK) argument system* Π *for a language* $\mathcal{L}$ *associated with an NP relations* R *consists of four PPT algorithms* $(\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ *with the following syntax:*

- $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ *takes as input a security parameter* λ *and outputs public parameters* par*.*
- $\mathsf{Gen}_{\mathcal{L}}(1^\lambda, \mathcal{L}, \tau_{\mathcal{L}})$ *takes as input a security parameter* λ*, the description of* $\mathcal{L}$ *which specifies a statement length* N*, and a membership testing trapdoor* $\tau_{\mathcal{L}}$ *for* $\mathcal{L}$*. It outputs the language-dependent part* $\mathsf{crs}_{\mathcal{L}}$ *of the common reference string* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$*.*
- $\mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$ *is a proving algorithm taking as input the common reference string* crs*, a statement* $x \in \{0,1\}^N$*, a witness* w *such that* $(x, w) \in R$ *and a label* lbl*. It outputs a proof* π*.*
- $\mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl})$ *is a verification algorithm taking as input a common reference string* crs*, a statement* $x \in \{0,1\}^N$*, and a proof* π*. It outputs* 1 *or* 0*.*

Moreover, Π *should satisfy the following properties. For simplification we denote below by* Setup *an algorithm that runs successively* $\mathsf{Gen}_{\mathsf{par}}$ *and* $\mathsf{Gen}_{\mathcal{L}}$ *to generate a common reference string.*

- **Completeness:** *For any* $(x, w) \in R$ *and any* $\mathsf{lbl} \in \{0,1\}^*$*, we have*

$$\Pr\big[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{L}),\\ \pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl}) : \mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl}) = 1\big] \geq 1 - \mathsf{negl}(\lambda).$$

- **Soundness:** *For any* $x \in \{0,1\}^N \setminus \mathcal{L}$ *and any PPT prover* P^**, we have*

$$\Pr\big[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{L}),\ (\pi, \mathsf{lbl}) \leftarrow P^*(\mathsf{crs}, x) : \mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl}) = 1\big] \leq \mathsf{negl}(\lambda).$$

- **Zero-Knowledge:** *There is a PPT simulator* $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ *such that, for any PPT adversary* $\mathcal{A}$*, we have*

$$|\Pr[\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{L})\ :\ 1 \leftarrow \mathcal{A}^{\mathsf{P}(\mathsf{crs},\cdot,\cdot)}(\mathsf{crs})]\\ - \Pr[(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Sim}_0(1^\lambda, \mathcal{L})\ :\ 1 \leftarrow \mathcal{A}^{\mathcal{O}(\mathsf{crs},\tau_{\mathsf{zk}},\cdot,\cdot)}(\mathsf{crs})]| \leq \mathsf{negl}(\lambda).$$

Here, $\mathsf{P}(\mathsf{crs},\cdot,\cdot)$ *is an oracle that outputs* $\perp$ *on input of* $(x,w,\mathsf{lbl}) \notin R$ *and outputs a valid proof* $\pi \leftarrow \mathsf{P}(\mathsf{crs},x,w,\mathsf{lbl})$ *otherwise;* $\mathcal{O}(\mathsf{crs},\tau_{\mathsf{zk}},\cdot,\cdot)$ *is an oracle that outputs* $\perp$ *on input of* (x,w,lbl) *such that* $(x,w) \notin R$ *and outputs a simulated argument* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs},\tau_{\mathsf{zk}},x,\mathsf{lbl})$ *on input of* (x,w,lbl) *such that* $(x,w) \in R$*. Note that this simulated proof* π *is generated independently of the witness* w *provided as input.*[10]

The notion of soundness captured by Definition A.1 is *non-adaptive* in that the statement is given as input to the dishonest prover and chosen independently of the common reference string. The stronger notion of *adaptive soundness* allows the target statement to be chosen by the adversary after having received the common reference string. It is known (see, e.g., [66]) that perfect or statistical NIZK arguments cannot provide adaptive soundness under falsifiable assumptions. The reason lies in the impossibility of recognizing when the adversary wins and outputs a proof for a false statement. One way to bypass the impossibility results is to consider *trapdoor languages*, where a trapdoor can be used to recognize false statements. In our application to ring signatures, we will consider a notion of adaptive soundness for trapdoor languages.

Definition A.1 captures a notion of multi-theorem zero-knowledge, which allows the adversary to obtain proofs for multiple statements. Feige *et al.* [39] gave a generic transformation of a multi-theorem NIZK argument system from a single-theorem one (where the adversary can only invoke the oracle once).

Simulation-Soundness. We now recall the definition of simulation-soundness introduced in [70], which informally captures the adversary's inability to create a new proof for a false statement $x^\star$ even after having seen simulated proofs for possibly false statements $\{x_i\}_i$ of its choice.

In the following, in order to allow a challenger to efficiently check the winning condition (ii) in the security experiment, we restrict ourselves to *trapdoor languages*, where a language-specific trapdoor $\tau_{\mathcal{L}}$ makes it possible to determine if a given statement $x^\star \in \{0,1\}^N$ belongs to the language $\mathcal{L}$ with overwhelming probability. This restriction has no impact on our applications where we always have a membership testing trapdoor $\tau_{\mathcal{L}}$ at our disposal.

Definition A.2 ([70,35]). *Let a language* $\mathcal{L}$*. A NIZK argument system for* $\mathcal{L}$ *provides* **unbounded simulation soundness** *if no PPT adversary has noticeable advantage in this game.*

1. *The challenger chooses a membership testing trapdoor* $\tau_{\mathcal{L}}$ *that allows recognizing elements of* $\mathcal{L}$*. Let* $\mathsf{Sim} = (\mathsf{Sim}_0, \mathsf{Sim}_1)$ *be an efficient NIZK simulator for* $\mathcal{L}$*. The challenger runs* $(\mathsf{crs},\tau_{\mathsf{zk}}) \leftarrow \mathsf{Sim}_0(1^\lambda,\mathcal{L})$ *and gives* $(\mathsf{crs},\tau_{\mathcal{L}})$ *to the adversary* $\mathcal{A}$*.*
2. *The adversary* $\mathcal{A}$ *is given oracle access to* $\mathsf{Sim}_1(\mathsf{crs},\tau_{\mathsf{zk}},\cdot,\cdot)$*. At each query,* $\mathcal{A}$ *chooses a statement* $x \in \{0,1\}^N$ *and a label* $\mathsf{lbl} \in \{0,1\}^*$*. It obtains a simulated argument* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs},\tau_{\mathsf{zk}},x,\mathsf{lbl})$*.*

[10] In particular, Sim_1 can be run on any statement x, even $x \notin \mathcal{L}$.

3. $\mathcal{A}$ outputs $(x^\star, \mathsf{lbl}^\star, \pi^\star)$.

Let $\mathcal{Q}$ be the set of all simulation queries and responses $(x_i, \mathsf{lbl}_i, \pi_i)$ made by $\mathcal{A}$ to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$. The adversary $\mathcal{A}$ wins if the following conditions are satisfied: (i) $(x^\star, \mathsf{lbl}^\star, \pi^\star) \notin \mathcal{Q}$; (ii) $x^\star \notin \mathcal{L}$; and (iii) $\mathsf{V}(\mathsf{crs}, x^\star, \pi^\star, \mathsf{lbl}^\star) = 1$. The adversary's advantage $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{uss}}(\lambda)$ is its probability of success taken over all coin tosses.

B Simulation-Sound NIZK Arguments from Trapdoor Σ-Protocols

In [56], the authors construct unbounded simulation-sound NIZK arguments by combining a trapdoor Σ-protocol and an $\mathcal{R}$-lossy public-key encryption scheme.

B.1 The Argument System

In order to apply the construction of [56] to trapdoor Σ-protocols with r bad challenges, we need a CI hash function for efficiently enumerable relations $R \subseteq \mathcal{X} \times \mathcal{Y}$ where, for each $x \in \mathcal{Y}$, the set $\mathcal{Y}_x = \{y \in \mathcal{Y} \mid (x, y) \in R\}$ has cardinality at most $|\mathcal{Y}_x| \leq r$ and is efficiently computable from x.

The hash function of Peikert and Shiehian [67] provides this property. They provide a correlation intractable hash function for efficiently searchable relations. The bootstrapping theorem of [67] actually implies the existence of such a hash family under the LWE assumption with polynomial approximation factors. In [18], it was further observed that any CI hash function for efficiently searchable relations is also correlation intractable for efficiently enumerable relations.

The construction hereunder thus adapts [56] using the following ingredients:

- A trapdoor Σ-protocol $\Pi' = (\mathsf{Gen}'_{\mathsf{par}}, \mathsf{Gen}'_{\mathcal{L}}, \mathsf{P}', \mathsf{V}')$ for the same language $\mathcal{L}$ with challenge space $\mathcal{C} = \{0,1\}^\lambda$ and which satisfies the properties of Definition 2.8. This language is assumed to be a trapdoor language in that its description can be sampled with a trapdoor $\tau_{\mathcal{L}}$ allowing to test membership of $\mathcal{L}$. In addition, $\mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}, x, \mathbf{a})$ should be computable within time $T \in \mathsf{poly}(\lambda)$ for any input $(\tau_\Sigma, \mathsf{crs}, x, \mathbf{a})$.
- A strongly unforgeable one-time signature scheme $\mathsf{OTS} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys of length $\ell_v \in \mathsf{poly}(\lambda)$.
- An admissible hash function $\mathsf{AHF} : \{0,1\}^{\ell_v} \to \{0,1\}^L$, for some $L \in \mathsf{poly}(\lambda)$ with $L > \ell_v$, which induces the relation $\mathcal{R}_{\mathsf{BM}} : \{0,1,\perp\}^L \times \{0,1\}^{\ell_v} \to \{0,1\}$.
- An $\mathcal{R}$-lossy PKE scheme $\mathcal{R}\text{-}\mathsf{LPKE} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{LOpener})$ for the relation $\mathcal{R}_{\mathsf{BM}} : \{0,1,\perp\}^L \times \{0,1\}^L \to \{0,1\}$ with public (resp. secret) key space $\mathcal{PK}$ (resp. $\mathcal{SK}$). We assume that $\mathsf{Decrypt}$ is computable within time T. We denote the message (resp. ciphertext) space by MsgSp (resp. CtSp) and the randomness space by R^{LPKE}. Let also D_R^{LPKE} denote the distribution from which the random coins of $\mathsf{Encrypt}$ are sampled.

- A correlation intractable hash family $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ with output length λ for the class $\mathcal{R}_{\mathsf{CI}}$ of relations that are efficiently enumerable within time T.

It is required that P' outputs a first prover message $\mathbf{a}$ which fits in the message space MsgSp of $\mathcal{R}$-LPKE.

The argument system $\Pi^{\mathrm{uss}} = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ allows P and V to input a label lbl consisting of public data that can be bound to non-interactive arguments in a non-malleable way. The construction proceeds as follows.

$\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$: Run $\mathsf{par} \leftarrow \mathsf{Gen}'_{\mathsf{par}}(1^\lambda)$ and output par.

$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$: Given public parameters par and a language $\mathcal{L} \subset \{0,1\}^N$, the CRS is generated as follows.

1. Generate a CRS $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ for the trapdoor Σ-protocol Π'.
2. Choose a random member $\mathsf{AHF} : \{0,1\}^{\ell_v} \to \{0,1\}^L$ of an admissible hash function family.
3. Generate public parameters $\Gamma \hookleftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$ for the $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme where $\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ is the bit-matching relation and $B \in \mathsf{poly}(\lambda)$ is the length of the first prover messages. Given the spaces $\mathcal{K} = \{0,1,\perp\}^L$ and $\mathcal{T} = \{0,1\}^L$ specified by Γ, choose a random initialization value $K \leftarrow \mathcal{K}$ and generate lossy keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$.
4. Generate a key $k \leftarrow \mathsf{Gen}(1^\lambda)$ for a correlation intractable hash function with output length λ.

Output the language-dependent $\mathsf{crs}_{\mathcal{L}} := \big(\mathsf{crs}'_{\mathcal{L}}, k\big)$ and the simulation trapdoor $\tau_{\mathsf{zk}} := \mathsf{sk}$, which is the lossy secret key of $\mathcal{R}$-LPKE. The global common reference string consists of $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, \mathsf{pk}, \mathsf{AHF}, \mathsf{OTS})$.

$\mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$: To prove a statement x for a label $\mathsf{lbl} \in \{0,1\}^*$ using $w \in R(x)$, generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^\lambda)$. Then,

1. Compute $\big(\mathbf{a}', st'\big) \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w)$ using the prover of Π'. Then, compute $\mathbf{a} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{AHF}(\mathsf{VK}), \mathbf{a}'; \mathbf{r})$ using random coins $\mathbf{r} \hookleftarrow D_R^{\mathsf{LPKE}}$.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK})) \in \{0,1\}^\lambda$.
3. Compute $\mathbf{z}' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, \mathbf{a}', \mathsf{Chall}, st')$. Define $\mathbf{z} = (\mathbf{z}', \mathbf{a}', \mathbf{r})$.
4. Generate $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, \mathbf{z}, \mathsf{lbl}))$ and output $\boldsymbol{\pi} = \big(\mathsf{VK}, \mathbf{z}, sig\big)$.

$\mathsf{V}(\mathsf{crs}, x, \boldsymbol{\pi}, \mathsf{lbl})$: Given a statement x, a label lbl as well as a purported proof $\boldsymbol{\pi} = \big(\mathsf{VK}, (\mathbf{a}, \mathbf{z}), sig\big)$, return 0 if $\mathcal{V}(\mathsf{VK}, (x, \mathbf{z}, \mathsf{lbl}), sig) = 0$. Otherwise,

1. Write $\mathbf{z}$ as $\mathbf{z} = (\mathbf{z}', \mathbf{a}', \mathbf{r})$ and return 0 if they do not parse properly (in particular, if $\mathbf{r} \notin R^{\mathsf{LPKE}}$). Otherwise, set $\mathbf{a} = \mathsf{Encrypt}(\mathsf{pk}, \mathsf{AHF}(\mathsf{VK}), \mathbf{a}'; \mathbf{r})$.
2. Let $\mathsf{Chall} = \mathsf{Hash}\big(k, (x, \mathbf{a}, \mathsf{VK})\big)$. If $\mathsf{V}'(\mathsf{crs}'_{\mathcal{L}}, x, (\mathbf{a}', \mathsf{Chall}, \mathbf{z}')) = 1$, return 1. Otherwise, return 0.

The NIZK simulator uses a technique due to Damgård [32], which uses a trapdoor commitment scheme to achieve a straight-line simulation of 3-move zero-knowledge proofs in the common reference string model.

Theorem B.1 ([56]). *The above argument is multi-theorem zero-knowledge if the trapdoor Σ-protocol Π' is special zero-knowledge.*

Proof. The proof was given in [56, Theorem 3.3]. For completeness, we recall the simulator $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ which uses the lossy secret key $\tau_{\mathsf{zk}} = \mathsf{sk}$ of $\mathcal{R}$-LPKE to simulate transcripts $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ without using the witnesses. Namely, on input of $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$, Sim_0 generates $\mathsf{crs}_{\mathcal{L}}$ by proceeding identically to $\mathsf{Gen}_{\mathcal{L}}$ while Sim_1 is described hereunder.

$\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, x, \mathsf{lbl})$: On input a statement $x \in \{0,1\}^N$, a label lbl and the simulation trapdoor $\tau_{\mathsf{zk}} = \mathsf{sk}$, algorithm Sim_1 proceeds as follows.

1. Generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^\lambda)$. Let $\mathbf{0}^{|\mathbf{a}'|}$ the all-zeroes string of the same length as the first prover message of Π'. Compute $\mathbf{a} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{AHF}(\mathsf{VK}), \mathbf{0}^{|\mathbf{a}'|}; \mathbf{r}_0)$ using random coins $\mathbf{r}_0 \hookleftarrow D_R^{\mathsf{LPKE}}$ independently sampled from the distribution D_R^{LPKE}.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK})) \in \{0,1\}^\lambda$.
3. Run the ZK simulator $(\mathbf{a}', \mathbf{z}') \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Chall})$ of Π' to obtain a simulated transcript $(\mathbf{a}', \mathsf{Chall}, \mathbf{z}')$ of Π' for the challenge $\mathsf{Chall} \in \{0,1\}^\lambda$.
4. Using the lossy secret key sk of $\mathcal{R}$-LPKE, compute random coins $\mathbf{r} \leftarrow \mathsf{LOpener}(\mathsf{pk}, \mathsf{sk}, \mathsf{AHF}(\mathsf{VK}), \mathbf{a}, \mathbf{0}^{|\mathbf{a}'|}, \mathbf{a}', \mathbf{r}_0)$ which explain $\mathbf{a}$ as an encryption of $\mathbf{a}'$ under the tag $\mathsf{AHF}(\mathsf{VK})$. Then, define $\mathbf{z} = \big(\mathbf{z}', \mathbf{a}', \mathbf{r}\big)$.
5. Generate a one-time signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, \mathbf{z}, \mathsf{lbl}))$ and output the proof $\boldsymbol{\pi} = \big(\mathsf{VK}, \mathbf{z}, sig\big)$.

We refer to [56, Theorem 3.3] for a proof that the simulation is indistinguishable from a real argument. In particular, the proof shows that zero-knowledge holds in the statistical sense if Π' is statistically special ZK. $\square$

Theorem B.2 states the unbounded simulation-soundness property. The proof is identical to that of [56, Theorem 3.4] with a slight modification since we consider efficiently enumerable relations (and not only unique-output efficiently searchable relations). The difference with the proof of [56] is just syntactical since we need a $\mathsf{BadChallenge}$ function that outputs an enumerable set instead of a single element of the challenge space.

As in [56], the proof uses the $\mathcal{R}$-lossy PKE scheme as a trapdoor commitment to equivocate lossy encryptions of the first prover message in Π' while forcing the adversary's fake proof to take place on an extractable commitment.

Theorem B.2. *The above non-interactive argument system provides unbounded simulation-soundness if: (i)* OTS *is a strongly unforgeable one-time signature; (ii)* $\mathcal{R}$-LPKE *is an* $\mathcal{R}_{\mathsf{BM}}$*-lossy PKE scheme; (iii)* $\mathcal{H}$ *is correlation-intractable for all relations that are enumerable within time T, where T denotes the maximal running time of algorithms* $\mathsf{BadChallenge}(\cdot,\cdot,\cdot,\cdot)$ *and* $\mathsf{Decrypt}(\cdot,\cdot,\cdot)$.

Proof. We consider a sequence of games where, for each i, we define a variable $W_i \in \{\mathtt{true}, \mathtt{false}\}$ where $W_i = \mathtt{true}$ if and only if the adversary wins in Game_i.

Game_0: This is the real game of Definition A.2. Namely, the challenger runs $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Sim}_0(\mathsf{par}, 1^N)$ and gives $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, \Gamma, \mathsf{pk}, \mathsf{AHF}, \Pi^{\mathrm{ots}})$ to the adversary $\mathcal{A}$. At the same time, the challenger generates a trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}$ in such a way that it can efficiently test if $\mathcal{A}$'s output satisfies the winning condition (ii). The adversary is granted oracle access to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$. At each query, $\mathcal{A}$ chooses a statement $x \in \{0,1\}^N$ with a label lbl and the challenger replies by returning a simulated argument $\boldsymbol{\pi} \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, x, \mathsf{lbl})$. When $\mathcal{A}$ halts, it outputs a triple $(x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star)$, where $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, \mathbf{z}^\star, sig^\star\big)$. The Boolean variable W_0 is thus set to $W_0 =$ `true` under the following three conditions: (i) $(x^\star, \mathsf{lbl}^\star, \boldsymbol{\pi}^\star) \not\in \mathcal{Q}$, where $\mathcal{Q} = \{(x_i, \mathsf{lbl}_i, \boldsymbol{\pi}_i)\}_{i=1}^{Q}$ denotes the set of queries to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$ and the corresponding responses $\boldsymbol{\pi}_i = \big(\mathsf{VK}^{(i)}, \mathbf{z}_i = (\mathbf{z}'_i, \mathbf{a}'_i, \mathbf{r}_i), sig_i\big)$; (ii) $x^\star \not\in \mathcal{L}$; and (iii) $V(\mathsf{crs}, x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star) = 1$. We may assume w.l.o.g. that the one-time verification keys $\{\mathsf{VK}^{(i)}\}_{i=1}^{Q}$ are chosen ahead of time at the beginning of the game. By definition we have $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{uss}}(\lambda) = \Pr[W_0]$.

Game_1: This is like Game_0 except that the challenger $\mathcal{B}$ sets $W_1 =$ `false` if $\mathcal{A}$ outputs a fake proof $(x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star)$, where $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, \mathbf{z}^\star, sig^\star\big)$ contains a $\mathsf{VK}^\star$ that coincides with the verification key $\mathsf{VK}^{(i)}$ contained in an output $\boldsymbol{\pi}_i = \big(\mathsf{VK}^{(i)}, \mathbf{z}_i, sig_i\big)$ of $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$. The strong unforgeability of OTS implies that $\Pr[W_1]$ cannot noticeably differ from $\Pr[W_0]$. We can easily turn $\mathcal{B}$ into a forger such that $|\Pr[W_1] - \Pr[W_0]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathrm{ots}}(\lambda)$.

Game_2: This game is like Game_1 with the following changes. At step 2 of $\mathsf{Gen}_{\mathcal{L}}$, the challenger runs $K \leftarrow \mathsf{AdmSmp}(1^\lambda, Q, \delta)$ to generate a key $K \in \{0,1,\perp\}^L$ for an admissible hash function $\mathsf{AHF} : \{0,1\}^{\ell_v} \to \{0,1\}^L$, where Q is an upper bound on the number of adversarial queries. By the second indistinguishability property of the $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme (which holds in the statistical sense), we know that changing the initialization value does not significantly affect $\mathcal{A}$'s view. It follows that $|\Pr[W_2] - \Pr[W_1]| \leq 2^{-\Omega(\lambda)}$.

Game_3: This game is identical to Game_2 with one modification. When the adversary halts and outputs $x^\star$, the challenger checks if the conditions

$$F_{\mathsf{ADH}}(K, \mathsf{VK}^{(1)}) = \cdots = F_{\mathsf{ADH}}(K, \mathsf{VK}^{(Q)}) = 1 \ \wedge \ F_{\mathsf{ADH}}(K, \mathsf{VK}^\star) = 0 \qquad (24)$$

are satisfied, where $\mathsf{VK}^\star$ is the one-time verification key in the adversary's output and $\mathsf{VK}^{(1)}, \ldots, \mathsf{VK}^{(Q)}$ are those in adversarial queries. If these conditions do not hold, the challenger aborts and sets $W_3 =$ `false`. For simplicity, we assume that $\mathcal{B}$ aborts at the very beginning of the game if it detects that there exists $i \in [Q]$ such that $F_{\mathsf{ADH}}(K, \mathsf{VK}^{(i)}) = 0$ (recall that $\{\mathsf{VK}^{(i)}\}_{i=1}^{Q}$ are chosen at the outset of the game by $\mathcal{B}$). If conditions (24) are satisfied, the challenger sets $W_3 =$ `true` whenever $W_1 =$ `true`. Letting Fail denote the event that $\mathcal{B}$ aborts because (24) does not hold, we have $W_3 = W_2 \wedge \neg\mathsf{Fail}$. Since the key K of the admissible hash function is statistically independent of the adversary's view, we can apply Theorem 2.6 to argue that there is a noticeable function $\delta(\lambda)$ such that $\Pr[\neg\mathsf{Fail}] \geq \delta(\lambda)$. This implies

$$\Pr[W_3] = \Pr[W_2 \wedge \neg\mathsf{Fail}] \geq \delta(\lambda) \cdot \Pr[W_2] \ , \qquad (25)$$

where the inequality stems from the fact that Fail is independent of W_1 since K is statistically independent of $\mathcal{A}$'s view.

We note that, if conditions (24) are satisfied in Game_3, the sequence of one-time verification keys $(\mathsf{VK}^{(1)}, \ldots, \mathsf{VK}^{(Q)}, \mathsf{VK}^\star)$ satisfies $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{AHF}(\mathsf{VK}^\star)) = 1$ and $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{AHF}(\mathsf{VK}^{(i)})) = 0$ for all $i \in [Q]$.

Game_4: We modify the oracle $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$ and by exploiting the equivocation property of $\mathcal{R}$-LPKE for lossy tags (instead of lossy keys). At the i-th query (x_i, lbl_i) to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$, we must have $F_{\mathsf{ADH}}(K, \mathsf{VK}^{(i)}) = 1$ (meaning that $\mathsf{VK}^{(i)}$ is a lossy tag as $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{VK}^{(i)}) = 0$) if $\mathcal{B}$ did not abort. This allows $\mathcal{B}$ to equivocate $\mathbf{a}$ using the trapdoor key tk instead of the lossy secret key sk of $\mathcal{R}$-LPKE. Namely, at step 4 of Sim_1, the modified $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$ oracle computes random coins

$$\mathbf{r}_i \leftarrow \mathsf{Opener}\big(\mathsf{pk}, \mathsf{tk}, \mathsf{AHF}(\mathsf{VK}^{(i)}), \mathbf{a}_i, \mathbf{0}^{|\mathbf{a}'_i|}, \mathbf{a}'_i, \mathbf{r}_{i,0}\big)$$

instead of running $\mathsf{LOpener}$ using sk. We define the variable W_4 exactly as W_3. Since Opener and $\mathsf{LOpener}$ output samples from statistically close distributions on all lossy tags $\mathsf{VK}^{(i)}$, this implies $|\Pr[W_4] - \Pr[W_3]| \leq 2^{-\Omega(\lambda)}$.

Game_5: We now modify the distribution of crs. At step 2 of Gen, we generate the keys for $\mathcal{R}$-LPKE as injective keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ instead of lossy keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$. The indistinguishability property (i) of $\mathcal{R}$-LPKE ensures that $\Pr[W_5]$ and $\Pr[W_4]$ are negligibly far apart. Recall that this indistinguishability property ensures that the distributions of pairs $(\mathsf{pk}, \mathsf{tk})$ produced by Keygen and $\mathsf{LKeygen}$ are computationally indistinguishable. We can thus easily build a distinguisher $\mathcal{B}$ against $\mathcal{R}$-LPKE that bridges between Game_4 and Game_5 (by using tk to simulate $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$ as in Game_4). It comes that $|\Pr[W_5] - \Pr[W_4]| \leq \mathbf{Adv}_{\mathcal{B}}^{\text{indist-LPKE}}(\lambda)$.

Due to the modification introduced in Game_5, if the conditions (24) are satisfied, we have $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{AHF}(\mathsf{VK}^\star)) = 1$, meaning that the adversary's fake proof $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, \mathbf{z}^\star = (\mathbf{z}'^\star, \mathbf{a}'^\star, \mathbf{r}^\star), sig^\star\big)$ involves an injective tag $\mathsf{VK}^\star$. Since pk is now an injective key, this implies that $\mathbf{a}^\star = \mathsf{Encrypt}(\mathsf{pk}, \mathsf{AHF}(\mathsf{VK}^\star), \mathbf{a}'^\star; \mathbf{r}^\star)$ is an injective encryption of $\mathbf{a}'^\star$ under the tag $\mathsf{VK}^\star$ using the randomness $\mathbf{r}^\star$.

Game_6: We change the distribution of $\mathsf{crs} = \big(\mathsf{par}, (\mathsf{crs}'_{\mathcal{L}}, k), \mathsf{pk}, \mathsf{AHF}, \mathsf{OTS}\big)$ by relying on the CRS indistinguishability property of the trapdoor Σ-protocol Π'. Namely, we use the $\mathsf{TrapGen}'$ algorithm of Definition 2.8 to generate $\mathsf{crs}'_{\mathcal{L}}$ as $(\mathsf{crs}'_{\mathcal{L}}, \tau_\Sigma) \leftarrow \mathsf{TrapGen}'(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$ instead of $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$. We immediately have $|\Pr[W_6] - \Pr[W_5]| \leq \mathbf{Adv}_{\mathcal{A}}^{\text{indist-}\Sigma}(\lambda)$.

We note that the trapdoor τ_Σ produced by $\mathsf{TrapGen}'$ in Game_6 can be used in later games to compute the $\mathsf{BadChallenge}$ function of the trapdoor Σ-protocol Π'. To evaluate $\mathsf{BadChallenge}$, we also use the secret key sk produced by $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ which allows decrypting $\mathbf{a}^\star$ when $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{AHF}(\mathsf{VK}^\star)) = 1$.

Game$_7$: In this game, we use the decryption algorithm of $\mathcal{R}$-LPKE. If $\mathcal{B}$ did not fail, we know that $\mathcal{A}$'s output $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, \mathbf{z}^\star = (\mathbf{z}'^\star, \mathbf{a}'^\star, \mathbf{r}^\star), sig^\star\big)$ involves an injective tag $\mathsf{VK}^\star$, so that $\mathbf{a}^\star = \mathsf{Encrypt}(\mathsf{pk}, \mathsf{AHF}(\mathsf{VK}^\star), \mathbf{a}'^\star; \mathbf{r}^\star)$ is a statistically binding commitment to $\mathbf{a}'^\star$. With probability $2^{-\Omega(\lambda)}$, there thus exists only one message $\mathbf{a}'^\star$ such that $\mathbf{a}^\star = \mathsf{Encrypt}(\mathsf{pk}, \mathsf{AHF}(\mathsf{VK}^\star), \mathbf{a}'^\star; \mathbf{r}^\star)$ for some $\mathbf{r}^\star \in R^{\mathsf{LPKE}}$. We thus consider the relation R_{bad} defined by

$$\begin{aligned}\big((x, \mathbf{a}, \mathsf{VK}), \mathsf{Chall}\big) \in R_{\mathsf{bad}} \quad &\Leftrightarrow \quad x \notin \mathcal{L} \quad \wedge \\ &\mathsf{Chall} \in \mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Decrypt}(\mathsf{sk}, \mathsf{AHF}(\mathsf{VK}), \mathbf{a})\big).\end{aligned} \tag{26}$$

We now set $W_7 = \mathsf{false}$ if

$$\begin{aligned}&\mathsf{Hash}(k, (x^\star, \mathbf{a}^\star, \mathsf{VK}^\star)) \\ &\qquad \notin \mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x^\star, \mathsf{Decrypt}(\mathsf{sk}, \mathsf{AHF}(\mathsf{VK}^\star), \mathbf{a}^\star)\big),\end{aligned} \tag{27}$$

where $\mathbf{a}^\star = \mathsf{Encrypt}(\mathsf{pk}, \mathsf{AHF}(\mathsf{VK}^\star), \mathbf{a}'^\star; \mathbf{r}^\star)$, and $W_7 = W_6$ otherwise. The decryption property under injective tags implies $|\Pr[W_7] - \Pr[W_6]| \leq 2^{-\Omega(\lambda)}$ since, unless $\mathbf{a}^\star$ does not decrypt to $\mathbf{a}'^\star$, $\boldsymbol{\pi}^\star$ cannot correctly verify if (27) holds.

In Game_7, we claim that $\Pr[W_7] \leq \mathbf{Adv}_{\mathcal{A}}^{\mathrm{CI}}(\lambda)$ since, if we had

$$\mathsf{Hash}(k, (x^\star, \mathbf{a}^\star, \mathsf{VK}^\star)) \in \mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x^\star, \mathsf{Decrypt}(\mathsf{sk}, \mathsf{AHF}(\mathsf{VK}^\star), \mathbf{a}^\star)\big),$$

it would break the correlation-intractability of $\mathcal{H}$ for the relation R_{bad}.

Putting the above altogether, we obtain

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}}^{\mathrm{uss}}(\lambda) \leq 2^{-\Omega(\lambda)} + \mathbf{Adv}_{\mathcal{B}}^{\mathrm{ots}}(\lambda) + \frac{1}{\delta(\lambda)} \cdot \Big(&\mathbf{Adv}_{\mathcal{B}}^{\mathsf{indist\text{-}LPKE}}(\lambda) \\ &+ \mathbf{Adv}_{\mathcal{B}}^{\mathsf{indist}\text{-}\Sigma}(\lambda) + \mathbf{Adv}_{\mathcal{B}}^{\mathrm{CI}}(\lambda) + 2^{-\Omega(\lambda)}\Big),\end{aligned}$$

which completes the proof. □

C Security Definitions for Ring Signatures

The correctness requirement is formalized as follows.

Definition C.1 (Correctness). A ring signature scheme $(\mathsf{CRSGen}, \mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ is correct if, for any $\rho \leftarrow \mathsf{CRSGen}(1^\lambda)$, any $(vk, sk) \leftarrow \mathsf{Keygen}(\rho)$, any R s.t. $vk \in \mathsf{R}$, any $M \in \{0,1\}^*$, we have $\mathsf{Verify}\big(\rho, M, \mathsf{Sign}(\rho, sk, M, \mathsf{R}), \mathsf{R}\big) = 1$.

A ring signature is unforgeable with respect to insider corruption if it is infeasible to forge a ring signature without controlling one of the ring members.

Definition C.2 (Unforgeability w.r.t. insider corruption). *A ring signature scheme* $(\mathsf{CRSGen}, \mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ *is unforgeable w.r.t. insider corruption if for any PPT adversary* $\mathcal{A}$,

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{unforge}}(\lambda) := \Pr[\rho \leftarrow \mathsf{CRSGen}(1^{\lambda}); (M^{\star}, R^{\star}, \Sigma^{\star}) \leftarrow \mathcal{A}^{\mathrm{Keygen,Sign,Corrupt}}(\rho) : \mathsf{Verify}(\rho, M^{\star}, \Sigma^{\star}, R^{\star}) = 1] \in \mathsf{negl}(\lambda),$$

where:

– Keygen *on the* i*-th query runs* $(vk^{(i)}, sk^{(i)}) \leftarrow \mathsf{Keygen}(\rho)$ *and returns* $vk^{(i)}$.
– $\mathrm{Sign}(i, M, \mathsf{R})$ *returns the output of* $\mathsf{Sign}(\rho, sk^{(i)}, M, \mathsf{R})$ *if: (i)* $(vk^{(i)}, sk^{(i)})$ *has been generated by* Keygen*; (ii)* $vk^{(i)} \in R$*. Otherwise, it returns* $\perp$.
– $\mathrm{Corrupt}(i)$ *returns* $sk^{(i)}$ *if* $(vk^{(i)}, sk^{(i)})$ *was generated by* Keygen.
– $\mathcal{A}$ *outputs* $(M^{\star}, \mathsf{R}^{\star}, \Sigma^{\star})$ *such that* $\mathrm{Sign}(\cdot, M^{\star}, \mathsf{R}^{\star})$ *has not been queried. Moreover,* $\mathsf{R}^{\star}$ *is non-empty and only contains public keys* $vk^{(i)}$ *generated by* Keygen *and such that no query* $\mathrm{Corrupt}(i)$ *was made.*

Definition C.3. A ring signature scheme $(\mathsf{CRSGen}, \mathsf{Keygen}, \mathsf{Sign}, \mathsf{Verify})$ provides statistical anonymity if, for any (possibly unbounded) adversary $\mathcal{A}$,

$$\left| \Pr\left[\begin{array}{c} \rho \leftarrow \mathsf{CRSGen}(1^{\lambda}); (M^{\star}, i_0, i_1, \mathsf{R}^{\star}) \leftarrow \mathcal{A}^{\mathsf{Keygen}(\cdot)}(\rho) \\ b \stackrel{\$}{\leftarrow} \{0,1\}; \Sigma^{*} \leftarrow \mathsf{Sign}(\rho, sk_{i_b}, M^{\star}, R^{\star}) \end{array} : \mathcal{A}(\Sigma^{\star}) = b \right] - \frac{1}{2} \right| \in \mathsf{negl}(\lambda)$$

where $vk_{i_0}, vk_{i_1} \in \mathsf{R}^{\star}$ *and were both generated by the* Keygen *oracle.*

We note that the definition of anonymity under full key exposure [6] requires that the random coins of Keygen be revealed to the adversary. In our setting, it does not make a difference since we assume unbounded adversaries. We also insist that we do not assume any upper bound on the size of a ring.

D Deferred Material for the Trapdoor Σ-Protocols of Section 4

D.1 Trapdoor Σ-Protocols With Small Challenge Space for Linearly Homomorphic Encryptions of 0

We first observe that any encryption scheme which is additively homomorphic over its message and randomness spaces has a direct trapdoor Σ-protocol proving that a ciphertext encrypts 0.

We consider additively homomorphic encryption schemes $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ where the message space $\mathcal{M}$, the randomness space $\mathcal{R}$ and the ciphertext space $\mathcal{C}$ form groups $(\mathcal{M}, +)$, $(\mathcal{R}, +)$ and $(\mathcal{C}, \cdot)$ for operations $+$ and $\cdot$, respectively. For any public key pk produced as $(pk, sk) \leftarrow \mathcal{K}(1^{\lambda})$, any messages $m_1, m_2 \in \mathcal{M}$ and randomness $r_1, r_2 \in \mathcal{R}$, we require that

$$\mathcal{E}_{pk}(m_1; r_1) \cdot \mathcal{E}_{pk}(m_2; r_2) = \mathcal{E}_{pk}(m_1 + m_2; r_1 + r_2).$$

In addition, we assume that the order $|\mathcal{R}|$ of the group $(\mathcal{R},+)$ is public and that this group is efficiently samplable. We describe a trapdoor Σ-protocol for the language $\mathcal{L} := \{c \in \mathcal{C} \mid \exists r \in \mathcal{R} : c = \mathcal{E}_{pk}(0;r)\}$.

$\mathsf{\textbf{Gen}}_{\mathsf{par}}(1^\lambda)$: Define public parameters $\mathsf{par} = \{\lambda, t\}$ consisting of a security parameter $\lambda \in \mathbb{N}$ and an integer $t = \mathcal{O}(\log \lambda)$ that defines a challenge space $\{0,1\}^t$ such that all prime divisors of $|\mathcal{M}|$ are strictly larger than 2^t.

$\mathsf{\textbf{Gen}}_{\mathcal{L}}(\mathsf{par},\mathcal{L})$: Given public parameters par as well as a description of a language $\mathcal{L}$ which specifies a public key pk produced by $\mathcal{K}$, define the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{pk\}$. The global common reference string is

$$\mathsf{crs} = \big(pk, \mathsf{crs}_{\mathcal{L}}\big).$$

$\mathsf{\textbf{TrapGen}}(\mathsf{par},\mathcal{L},\tau_{\mathcal{L}})$: Given par, the description of a language $\mathcal{L}$ that specifies a public key produced by $(pk, sk) \leftarrow \mathcal{K}(1^\lambda)$, and a membership-testing trapdoor $\tau_{\mathcal{L}} = sk$ consisting of a secret key underlying pk, output $\mathsf{crs} = \big(pk, \mathsf{crs}_{\mathcal{L}}\big)$, which defines $\mathsf{crs} = \big(pk, \mathsf{crs}_{\mathcal{L}}\big)$, and the trapdoor $\tau_{\Sigma} = sk$.

$\mathsf{\textbf{P}}\big(\mathsf{crs}, x, w\big) \leftrightarrow \mathsf{\textbf{V}}(\mathsf{crs}, x)$: Given crs, a statement $x = \mathcal{E}_{pk}(0;w)$ for some $w \in \mathcal{R}$, the prover P and the verifier V interact in the following way.

1. P chooses $r \leftarrow \mathcal{R}$ and sends $a = \mathcal{E}_{pk}(0;r)$ to V.
2. V sends a random challenge $\mathsf{Chall} \in \{0,1\}^t$, which is interpreted as an integer in $\{0,1,\ldots,2^t-1\}$.
3. P computes the response $z = r + \mathsf{Chall} \cdot w \in \mathcal{R}$ and sends it to V.
4. V checks if $z \in \mathcal{R}$ and $a \cdot x^{\mathsf{Chall}} = \mathcal{E}_{pk}(0;z)$. If these conditions do not both hold, V halts and returns $\perp$.

$\mathsf{\textbf{BadChallenge}}\big(\mathsf{par}, \tau_{\Sigma}, \mathsf{crs}, x, a\big)$: Given $\tau_{\Sigma} = sk$, parse the first prover message as $a \in \mathcal{C}$ and return $\perp$ if it does not parse properly. Otherwise, if there exists $j \in \{0,1,\ldots,2^t-1\}$ such that $\mathcal{D}_{sk}(a \cdot x^j) = 0$, return $\mathsf{Chall} = j$. If no such j exists, return $\mathsf{Chall} = \perp$.

When the above construction is instantiated using the Elgamal encryption scheme, we obtain a variant of the Chaum-Pedersen protocol [25] that allows proving the equality of discrete logarithms. The latter protocol was previously shown [27] to be a trapdoor Σ-protocol with binary challenges. Here, we observe that the challenge space can actually be of size $\mathcal{O}(\log \lambda)$ while keeping $\mathsf{BadChallenge}$ efficient.

Lemma D.1. *The above construction is a trapdoor Σ-protocol for $\mathcal{L}$.*

Proof. The CRS indistinguishability follows immediately from the fact that both the real CRS produced by $\mathsf{Gen}_{\mathcal{L}}$ and the one generated by $\mathsf{TrapGen}$ have exactly the same distribution.

We next prove the special ZK property by providing a transcript simulator. Given as inputs crs, a statement $x \in \mathcal{C}$ and a challenge $\mathsf{Chall} \in \{0,1\}$, the simulator first samples $z \hookleftarrow U(\mathcal{R})$ uniformly, computes $a = \mathcal{E}(0;z) \cdot x^{-\mathsf{Chall}}$ and outputs (a,z). Since we assumed that $(\mathcal{R},+)$ is efficiently samplable and has public order, z is distributed as in the real protocol: in both the real protocol and the simulation, a is uniquely determined by the challenge Chall, the statement x and the response z. Hence, if $x \in \mathcal{L}$, the simulated transcript (a, Chall, z) is statistically close to a real transcript with challenge Chall.

Soundness follows from the homomorphism. The verification equations of two valid transcripts $(a, \mathsf{Chall}_0, z_0)$, $(a, \mathsf{Chall}_1, z_1)$ imply $x^{\mathsf{Chall}_1 - \mathsf{Chall}_0} = \mathcal{E}(0; z_1 - z_0)$, where we assume w.l.o.g. that $\mathsf{Chall}_1 > \mathsf{Chall}_0$. Since $\mathsf{Chall}_1 - \mathsf{Chall}_0$ is co-prime with the group order $|\mathcal{M}|$, this implies that $x \in \mathcal{L}$.

We finally show that $\mathsf{BadChallenge}$ provides the correct result. Let $x \notin \mathcal{L}$ be a false statement. Note that we cannot simultaneously have $\mathcal{D}_{\mathsf{sk}}(a \cdot x^i) = 0$ and $\mathcal{D}_{\mathsf{sk}}(a \cdot x^j) = 0$ for distinct $i, j \in \{0, 1, \ldots, 2^t - 1\}$ as the additively homomorphic property would imply $\mathcal{D}_{\mathsf{sk}}(x^{j-i}) = 0$, which would contradict the hypothesis that $x \notin \mathcal{L}$ since $\gcd(j - i, |\mathcal{M}|) = 1$.

Therefore, $\mathsf{BadChallenge}$ can always compute the only $\mathsf{Chall} \in \{0, 1, \ldots, 2^t - 1\}$ for which a valid response exists after $2^t = \mathsf{poly}(\lambda)$ executions of $\mathcal{D}_{\mathsf{sk}}(\cdot)$. □

D.2 Trapdoor Σ-Protocol Showing Composite Residuosity

The trapdoor Σ-protocol in Supplementary Material D.1 supports polynomial-size challenge spaces and thus requires $\Theta(\lambda / \log \lambda)$ repetitions to ensure soundness. In the case of Paillier [64] and Damgård-Jurik [34] systems, we give similar trapdoor Σ-protocol with an *exponentially large* challenge space. In Damgård-Jurik, for an integer $\zeta > 1$ and an RSA modulus $N = pq$, proving that a ciphertext encrypts 0 is equivalent to proving that an element of $\mathbb{Z}^*_{N^{\zeta+1}}$ is an N^{ζ}-th residue. Paillier [64] is a special case with $\zeta = 1$.

The trapdoor Σ-protocol below can be seen as an extension (based on standard Σ-protocols from [47,34]) with a large challenge space of the trapdoor Σ-protocol given in [18, Section 6.2] for the Quadratic Residuosity language. However, the challenge space is now $\{0, \ldots, 2^{\lambda} - 1\}$, so that we obtain (to our knowledge) the first trapdoor Σ-protocol with exponentially large challenges.

Let an RSA modulus $N = pq$ and an integer $\zeta > 1$. We give a trapdoor Σ-protocol for $\mathcal{L}^{\mathsf{DCR}} := \{x \in \mathbb{Z}^*_{N^{\zeta+1}} \mid \exists w \in \mathbb{Z}^{\star}_N : x = w^{N^{\zeta}} \bmod N^{\zeta+1}\}$. In order for $\mathsf{BadChallenge}$ to identify bad challenges for any $x \notin \mathcal{L}^{\mathsf{DCR}}$, one difficulty is the case of instances $x \in \mathbb{Z}^*_{N^{\zeta+1}}$ that decrypt to $\alpha_x \in \mathbb{Z}_{N^{\zeta}}$ such that $\gcd(\alpha_x, N^{\zeta}) > 1$ since we cannot identify a unique bad challenge by inverting α_x in $\mathbb{Z}_{N^{\zeta}}$. However, an observation from [58, Theorem 2] shows that, even in this case, the bad challenge is efficiently computable since $\min(p, q) > 2^{\lambda}$. A difference with [58] is that Lipmaa applied this idea to the Elgamal-Paillier cryptosystem [13], where not all elements of the ciphertext space are in the range of the encryption algorithm. In Supplementary Material D.3, we point out that the ability to identify bad challenges crucially relies on all elements of $\mathbb{Z}^*_{N^{\zeta+1}}$ being Damgård-Jurik encryptions of some message $m \in \mathbb{Z}_{N^{\zeta}}$.

$\mathsf{Gen}_{\mathsf{par}}(1^{\lambda})$: Given the security parameter λ, define $\mathsf{par} = \{\lambda\}$.
$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}^{\mathsf{DCR}})$: Given public parameters par and the description of a language $\mathcal{L}^{\mathsf{DCR}}$, consisting of an RSA modulus $N = pq$ with p and q prime satisfying

$p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > \lambda$, define the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{N\}$. The global CRS is $\mathsf{crs} = (\{\lambda\}, \mathsf{crs}_{\mathcal{L}})$.

$\textbf{TrapGen}(\mathsf{par}, \mathcal{L}^{\mathsf{DCR}}, \tau_{\mathcal{L}})$: Given the same inputs as $\mathsf{Gen}_{\mathcal{L}}$ *and* a membership-testing trapdoor $\tau_{\mathcal{L}} = (p, q)$, output $\mathsf{crs} = \big(\{\lambda\}, \mathsf{crs}_{\mathcal{L}} = \{N\}\big)$ and the trapdoor $\tau_{\Sigma} = (p, q)$.

$\mathbf{P}\big(\mathsf{crs}, x, w\big) \leftrightarrow \mathbf{V}(\mathsf{crs}, x)$: Given a crs, a statement $x = w^{N^{\zeta}} \bmod N^{\zeta+1}$, P (who has the witness $w \in \mathbb{Z}_N^{\star}$) and V interact as follows:

1. P chooses a random $r \hookleftarrow U(\mathbb{Z}_N^*)$ and sends $a = r^{N^{\zeta}} \bmod N^{\zeta+1}$ to V.
2. V sends a random challenge $\mathsf{Chall} \hookleftarrow U(\{0, \ldots, 2^{\lambda} - 1\})$ to P.
3. P computes the response $z = r \cdot w^{\mathsf{Chall}} \bmod N$ and sends it to V.
4. V returns 1 if $a \cdot x^{\mathsf{Chall}} \equiv z^{N^{\zeta}} \pmod{N^{\zeta+1}}$ and 0 otherwise.

$\textbf{BadChallenge}\big(\mathsf{par}, \tau_{\Sigma}, \mathsf{crs}, x, a\big)$: Given $\tau_{\Sigma} = (p, q)$, decrypt x and a to obtain $\alpha_x = \mathcal{D}_{\tau_{\Sigma}}(x) \in \mathbb{Z}_{N^{\zeta}}$, $\alpha_a = \mathcal{D}_{\tau_{\Sigma}}(a) \in \mathbb{Z}_{N^{\zeta}}$. If $\alpha_x = 0$, return $\mathsf{Chall} = \perp$. Otherwise, let $d_x = \gcd(\alpha_x, N^{\zeta})$.

a. If $1 < d_x < N^{\zeta}$, return $\perp$ if d_x does not divide $N^{\zeta} - \alpha_a$.
b. Otherwise, the congruence $\alpha_a + \mathsf{Chall} \cdot \alpha_x \equiv 0 \pmod{\frac{N^{\zeta}}{d_x}}$ has a unique solution $\mathsf{Chall}' = -\alpha_x^{-1} \cdot \alpha_a \in \mathbb{Z}_{N^{\zeta}/d_x}$ since $\gcd(\alpha_x, N^{\zeta}/d_x) = 1$. If $\mathsf{Chall}' \in \mathbb{Z}_{N^{\zeta}/d_x} \setminus \{0, \ldots, 2^{\lambda} - 1\}$, return $\perp$. Else, return $\mathsf{Chall} = \mathsf{Chall}'$.

Lemma D.2 now shows that the above construction is a trapdoor Σ-protocol with large challenge space. By applying [67], this implies relatively short NIZK arguments (i.e., without using parallel repetitions to achieve negligible soundness error) for the language $\mathcal{L}^{\mathsf{DCR}}$ assuming that the LWE assumption holds.

Lemma D.2. *The above protocol is a trapdoor Σ-protocol for $\mathcal{L}^{\mathsf{DCR}}$.*

Proof. The CRS indistinguishability property is straightforward. We now construct a simulator for the special ZK property, which is identical to [34, Lemma 2] but we give it for completeness. Given crs, a statement $x \in \mathcal{L}^{\mathsf{DCR}}$ and a challenge $\mathsf{Chall} \in \{0, \ldots, 2^{\lambda} - 1\}$, the simulator first samples $z \hookleftarrow U(\mathbb{Z}_N^*)$, computes $a = z^{N^{\zeta}} \cdot x^{-\mathsf{Chall}} \bmod N^{\zeta+1}$ and outputs (a, z). Since $x \in \mathcal{L}^{\mathsf{DCR}}$, the simulated transcript (a, Chall, z) produced by the simulator is identical to that of a real conversation when the challenge is Chall. This follows from the fact that: (i) For any $x \in \mathcal{L}^{\mathsf{DCR}}$, $\{a = z^{N^{\zeta}} \cdot x^{-\mathsf{Chall}} \bmod N^{\zeta+1} \in \mathbb{Z}_{N^{\zeta+1}}^* \mid z \hookleftarrow U(\mathbb{Z}_N^*)\}$ is the uniform distribution over the subgroup of N^{ζ}-th residues in $\mathbb{Z}_{N^{\zeta+1}}^*$; (ii) $a \in \mathbb{Z}_{N^{\zeta+1}}^*$ and $r = a^{1/N^{\zeta}} \bmod N$ are uniquely determined by x, Chall and z in the simulation while $z = (a \cdot x^{\mathsf{Chall}})^{1/N^{\zeta}} \bmod N$ is uniquely determined by r, x and Chall in the real protocol. For a given challenge Chall, we thus obtain the same distribution of (a, Chall, z) by first sampling $r \hookleftarrow U(\mathbb{Z}_N^*)$ before defining z as a function of r and Chall as when we first sample $z \hookleftarrow U(\mathbb{Z}_N^*)$ before defining r as a function of z and Chall.

Soundness follows from standard arguments used in [34, Lemma 2]. By combining two accepting transcripts (a, Chall, z) and (a, Chall', z') for the same first message $a \in \mathbb{Z}_{N^{\zeta+1}}^*$, we obtain $x^{\mathsf{Chall}'-\mathsf{Chall}} = (z' \cdot z^{-1})^{N^\zeta} \bmod N^{\zeta+1}$. Since $p, q > 2^\lambda$, we have $\gcd(\mathsf{Chall}' - \mathsf{Chall}, N^\zeta) = 1$, which implies that $x \in \mathcal{L}^{\mathsf{DCR}}$.

Finally, we have to show that $\mathsf{BadChallenge}$ provides the correct result. If $\alpha_a = \mathcal{D}(a) = 0$, the product $a \cdot x^{\mathsf{Chall}}$ cannot be an encryption of 0 if $\mathsf{Chall} \neq 0$ since $\alpha_x \neq 0$. Then, the only possible bad challenge is $\mathsf{Chall} = 0$. If $\alpha_a \neq 0$ and $\alpha_x = \mathcal{D}(x)$ is invertible in $\mathbb{Z}_{N^\zeta}$, having $a \cdot x^{\mathsf{Chall}}$ be an encryption of 0 implies that $\alpha_a + \mathsf{Chall} \cdot \alpha_x \equiv 0 \pmod{N^\zeta}$, so that $\mathsf{Chall} = -\alpha_a \cdot \alpha_x^{-1} \bmod N^\zeta$ is uniquely determined.

We are left with the case where $d_x = \gcd(\alpha_x, N^\zeta) > 1$, so that α_x is non-invertible. Note that d_x lives in the set

$$\{p^i q^j \mid 0 \leq i < \zeta,\ 0 \leq j < \zeta\} \cup \{p^i q^\zeta \mid 0 \leq i < \zeta\} \cup \{p^s q^j \mid 0 \leq j < \zeta\}$$

and we know that the congruence $\alpha_x \cdot \mathsf{Chall} \equiv -\alpha_a \pmod{N^\zeta}$ has solutions if and only if d_x divides $N^\zeta - \alpha_a$. Moreover, in this case, there are exactly d_x solutions, which are given by

$$\left\{ \mathsf{Chall}_i = \mathsf{Chall}_0 + i \cdot \frac{N^\zeta}{d_x} \bmod N^\zeta \;\mid\; i \in \{0, \ldots, d_x - 1\} \right\},$$

where Chall_0 is an arbitrary solution. However, since $N^\zeta / d_x > \min(p, q) > 2^\lambda$, at most one of these solutions can fit in $\{0, \ldots, 2^\lambda - 1\}$. This solution is efficiently obtained at step 2 of $\mathsf{BadChallenge}$ when it exists. We conclude that $\mathsf{BadChallenge}$ always outputs the only challenge $\mathsf{Chall} \in \{0, \ldots, 2^\lambda - 1\}$ for which a valid response exists. □

D.3 On the Importance of Dense Ciphertext Spaces to Identify Bad Challenges in the DCR Setting

Chaidos and Groth [21] previously used the $\mathsf{BadChallenge}$ function methodology to build non-interactive designated verifier proofs by applying a technique introduced by Damgård, Fazio and Nicolosi (DFN) [33]. In the designated-verifier setting, proofs are not publicly verifiable. The verifier has a public key pk and a secret verification key sk.

The constructions of [33,21] rely on an additively homomorphic encryption scheme with key pair (ek, dk). The verifier's public key $pk = (ek, c)$ consists of ek and a homomorphic encryption $c = \mathcal{E}_{ek}(\mathsf{Chall})$ of a random challenge Chall while the secret verification key consists of $sk = (dk, \mathsf{Chall})$. Given a Σ-protocol Π, the DFN transformation proceeds by having the prover generate a first prover message a for Π (using some internal randomness r). The NIZK proof $\boldsymbol{\pi} = (a, c_z)$ is then obtained by computing $c_z = \mathcal{E}_{ek}(z)$ as an encryption of the prover's response z for Π. Verification then proceeds by decrypting c_z and running the verification procedure of Π on input of (a, Chall, z). The transformation exploits the property that, in many Σ-protocols, the response z is an affine function of the challenge Chall. Hence, as long as the encryption scheme $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ is additively homomorphic, the encrypted response $c_z = \mathcal{E}_{ek}(z)$ is computable from

$c = \mathcal{E}_{ek}(\mathsf{Chall})$, the witness w and the prover's randomness r.

Chaidos and Groth [21] used the above transformation to obtain a designated-verifier NIZK proof that a ciphertext (computed using an additively homomorphic cryptosystem) encrypts 0 or 1. Assuming that $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ provides IND-CPA security, they prove that the transformation satisfies the notion of adaptive *culpable* soundness, where the adversary wins if it outputs a proof for a false statement x of its choice, as long as it also outputs a witness w_{guilt} for the relation R_{guilt}, where $(x, w_{\text{guilt}}) \in R_{\text{guilt}}$ implies that x is a false statement. The proof assumes that Π has *identifiable bad challenges*, which is somewhat similar to the property of trapdoor Σ-protocols.

Definition D.3 ([21]). *A Σ-protocol Π has a* **unique identifiable challenge** *using NP-relation R_{guilt} if there exists a polynomial time algorithm E that inputs a statement x, a witness w_{guilt} and an initial message a. It returns the unique challenge* Chall *that can be answered. Formally, for all statements x, and witnesses w_{guilt} such that $(x, w_{\text{guilt}}) \in R_{\text{guilt}}$, algorithm $E(x, w_{\text{guilt}}, a)$ outputs the only* Chall *for which there exists z such that $V(x, a, \mathsf{Chall}, z) = 1$, where V is the verifier's algorithm in Π.*

The proof of culpable soundness [21, Theorem 2] goes as follows. The reduction obtains an encryption key ek from its IND-CPA challenger. It chooses a random Chall in the challenge space of Π and sends it to the challenger. The latter flips a coin b and returns either an encryption $c^\star = \mathcal{E}_{ek}(\mathsf{Chall})$ of Chall (if $b = 1$) or an encryption $c^\star = \mathcal{E}_{ek}(R)$ of a random message R (if $b = 0$). The reduction then runs the culpable soundness adversary $\mathcal{A}$ on input of the public key $pk = (ek, c^\star)$. When $\mathcal{A}$ outputs a triple $(x, w_{\text{guilt}}, \boldsymbol{\pi} = (a, c_z))$, the reduction uses the extractor of Definition D.3 to compute $\mathsf{Chall}' = E(x, w_{\text{guilt}}, a)$. If $\mathsf{Chall} = \mathsf{Chall}'$, the reduction outputs $b' = 1$. Otherwise, it outputs $b' = 0$. If $b = 1$, the probability that the reduction outputs $b' = 1$ is exactly (up to the negligible probability that E fails to extract the unique bad challenge from (x, w_{guilt}, a)) the probability that $\mathcal{A}$ breaks the culpable soundness of the designated verifier NIZK proof. If $b = 0$, the probability that $b' = 1$ is negligible since Chall is independent of $\mathcal{A}$'s view.

The proof of [21, Theorem 2] actually assumes that, in the Σ-protocol Π, the extraction algorithm E works for any (possibly maliciously generated) first message a contained in $\boldsymbol{\pi} = (a, c_z)$. This is the case in the instantiation of [21], where a is an Okamoto-Uchiyama encryption [63] of some message (in the Okamoto-Uchiyama cryptosystem, all elements of the ciphertext space are in the range of the encryption algorithm). However, in Lipmaa's Σ-protocol [58], the prover's first message is a ciphertext of the Elgamal-Paillier encryption scheme [13], where valid ciphertexts are sparse in the ciphertext space. The extractors of [58] (Figure 2 and Figure 4) can only compute bad challenges when the statement and the first prover message are both Elgamal-Paillier encryptions of something. They may fail when a malicious prover sends a first message $a = (g^u \bmod N^2, (1+N)^r \cdot h^v \bmod N^2)$ for arbitrary $r \in \mathbb{Z}_N$, $u, v \in \mathbb{Z}$. Consider the statement that $C = (g^r \bmod N^2, h^s \bmod N^2)$ encrypts 0 or 1, which is false when $r \neq s \bmod p'q'$, where $N = pq$ is a product of safe primes $p = 2p' + 1$, $q = 2q' + 1$. When the prover sends $a = (a_1, a_2) = (g^u \bmod N^2, h^v \bmod N^2)$ with $u \neq v \bmod p'q'$, the bad challenge is uniquely determined modulo $p'q'$ (and can possibly live in the challenge space $\{0, \ldots, 2^\lambda - 1\}$). However, neither $w_{\text{guilt}} = \log_g(h)$ nor $w_{\text{guilt}} = (p, q)$ allows computing the only challenge that satisfies the verification equations without knowing (r, s, u, v).

In our setting, this issue does not arise since all elements of $\mathbb{Z}_{N^2}^*$ are Paillier encryptions of something. A similar property is satisfied by Okamoto-Uchiyama. When the ciphertext space is not dense, the result of [21, Theorem 1] requires that valid ciphertexts be efficiently recognizable and always decrypt to some plaintext. In the case of Elgamal-Paillier, while invalid ciphertexts can be detected using $w_{\text{guilt}} = \log_g(h)$ (by observing that $a_2 \cdot a_1^{-w_{\text{guilt}}} \bmod N^2$ does not have order N in $\mathbb{Z}_{N^2}^*$), there is no efficient way to compute the bad challenge.

This invalidates the claim [58] that the DFN transform yields a culpably sound designated-verifier NIZK argument about Elgamal-Paillier ciphertexts encrypting 0 or 1. On the other hand, it does provide culpable soundness when instantiated with Paillier's cryptosystem if we use our Σ-protocol of Section 4.1.

D.4 Proof of Lemma 4.3

Proof. Let crs be a CRS generated by $\mathsf{Gen}_{\mathcal{L}}$. By the hypothesis on their order, $h \in \mathbb{Z}_{N^2}^*$ and $\bar{h} \in \mathbb{Z}_{\bar{N}^2}^*$ can be written as $h = (1+N)^\alpha \cdot \beta^N$ and $\bar{h} = (1+\bar{N})^{\bar{\alpha}} \cdot \bar{\beta}^{\bar{N}}$, for some $\alpha \in \mathbb{Z}_N^*$, $\bar{\alpha} \in \mathbb{Z}_{\bar{N}}^*$. We can thus assume that the commitments are perfectly hiding. To prove the claim, we describe a simulator ZKSim which, on input $(\mathsf{crs}, \boldsymbol{x}, \mathsf{Chall})$, where $\boldsymbol{x} \in \mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h})$ and $\mathsf{Chall} \in \{0, \ldots, 2^\lambda - 1\}$, builds a transcript $(\boldsymbol{a}, \mathsf{Chall}, \boldsymbol{z})$ by computing a pair $(\boldsymbol{a}, \boldsymbol{z})$ as follows.

1. Given $\big((C_0, \ldots, C_{R-1}), (L_1, \ldots, L_r)\big)$, for each $j \in [r]$, pick uniformly random $\bar{z}_j \hookleftarrow U(\{2^\lambda, \ldots, 2^{2\lambda} - 1\})$, $\bar{z}_{d,j}, \bar{z}_{e,j} \hookleftarrow U(\mathbb{Z}_{\bar{N}})$ and $\bar{z}_{u,j}, \bar{z}_{v,j} \hookleftarrow U(\mathbb{Z}_{\bar{N}}^*)$, Then, compute
$$\forall j \in [r] : \begin{cases} \bar{A}_j = L_j^{-\mathsf{Chall}} \cdot (1+\bar{N})^{\bar{z}_j} \cdot \bar{h}^{\bar{z}_{d,j}} \cdot \bar{z}_{u,j}^{\bar{N}} \bmod \bar{N}^2 \\ \bar{B}_j = L_j^{\mathsf{Chall} - \bar{z}_j} \cdot \bar{h}^{\bar{z}_{e,j}} \cdot \bar{z}_{v,j}^{\bar{N}} \quad \bmod \bar{N}^2 \end{cases} \tag{28}$$
2. Select $z_y \hookleftarrow U(\mathbb{Z}_N)$ and $z_w \hookleftarrow U(\mathbb{Z}_N^*)$ as well as $C_{d_k} \hookleftarrow U(\mathbb{Z}_{N^2}^*)$, for each $k \in \{1, \ldots, r-1\}$. Then, define $f_{j,1} = \bar{z}_j$ and $f_{j,0} = \mathsf{Chall} - \bar{z}_j \bmod N$ for each $j \in [r]$ and compute
$$C_{d_0} = h^{z_y} \cdot z_w^N \cdot \prod_{k=1}^{r-1} C_{d_k}^{\mathsf{Chall}^k} \cdot \prod_{i=0}^{R-1} C_i^{-(\prod_{j=1}^r f_{j,i_j} \bmod N)} \bmod N^2, \tag{29}$$
where $i_1 \ldots i_r \in \{0,1\}^r$ is the binary expansion of $i \in \mathbb{Z}_R$.
3. Return the first-message $\boldsymbol{a} = \big(\{(\bar{A}_j, \bar{B}_j)\}_{j=1}^r, \{C_{d_k}\}_{k=0}^{r-1}\big)$ and the response $\boldsymbol{z} = \big(z_y, z_w, \{(\bar{z}_j, \bar{z}_{d,j}, \bar{z}_{e,j}, \bar{z}_{u,j}, \bar{z}_{v,j})\}_{j=1}^r\big)$.

Now, let $(\boldsymbol{x}, \boldsymbol{w}, \mathsf{Chall})$ be the output of an unbounded adversary $\mathcal{A}(\mathsf{crs})$ such that $(\boldsymbol{x}, \boldsymbol{w}) \in \mathcal{R}_{\vee}^{1\text{-}R}(h, \bar{h})$. We analyze both distributions of $(\boldsymbol{a}, \mathsf{Chall}, \boldsymbol{z})$, where either $(\boldsymbol{a}, \boldsymbol{z}) \leftarrow \mathsf{ZKSim}(\mathsf{crs}, x, \mathsf{Chall})$ is simulated or $(\boldsymbol{a}, st) \leftarrow P(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w})$ and $\boldsymbol{z} \leftarrow P(\mathsf{crs}, st, \mathsf{Chall})$ come from the real execution of the protocol. Note that both transcripts are valid in that case.

First, since $\bar{N}$ divides the order of $\bar{h}$ in $\mathbb{Z}_{\bar{N}^2}^*$, the triples $(\bar{A}_j, \bar{z}_{d,j}, \bar{z}_{u,j})$ and $(\bar{B}_j, \bar{z}_{e,j}, \bar{z}_{v,j})$ are all identically distributed for each $j \in [r]$. Indeed, in both cases, equations (18) and (28) imply that, given L_j, Chall and $\bar{z}_j$, there is a one-to-one correspondence between $\bar{A}_j$ and $(\bar{z}_{d,j}, \bar{z}_{u,j})$, and $\bar{B}_j$ and $(\bar{z}_{e,j}, \bar{z}_{v,j})$. This is because $(\bar{d}_j, \bar{e}_j)$ in the real distribution and $(\bar{z}_{d,j}, \bar{z}_{e,j})$ in the simulated distribution are uniformly random pairs in $\mathbb{Z}_{\bar{N}} \times \mathbb{Z}_{\bar{N}}$. Then, as long as all the $\bar{z}_j$'s of both distributions are statistically indistinguishable, so are the transcripts parts related to $\bar{N}$. We postpone the analysis of the distribution of $\bar{z}_j$ to the end of the proof.

Second, assuming that N divides the order of h modulo N^2, $C_{d_1}, \ldots, C_{d_{r-1}}$ are uniformly random elements in $\mathbb{Z}_{N^2}^*$ in both transcripts. Indeed, in the real distribution, $\mu_1, \ldots, \mu_{r-1} \sim U(\mathbb{Z}_N)$ thus fully randomize the $\mathbb{Z}_N$-components while $\rho_1, \ldots, \rho_{r-1} \sim U(\mathbb{Z}_N^*)$ fully randomize the $\mathbb{Z}_N^*$-component over the group $\mathbb{Z}_{N^2}^* \simeq \mathbb{Z}_N \times \mathbb{Z}_N^*$. Moreover, in the simulated distribution, those elements are directly drawn from $U(\mathbb{Z}_{N^2}^*)$. As for the triple (z_y, z_w, C_{d_0}), for a fixed choice of $\{C_{d_k}\}_{k=1}^{r-1}$, $\{\bar{z}_j\}_{j=1}^{r}$ and Chall, equations (29) and (19) give a one-to-one relation between $(z_y, z_w) \in \mathbb{Z}_N \times \mathbb{Z}_N^*$ and C_{d_0} in both transcripts whose distributions are the same since μ_0 of the real distribution and z_y of the simulated distribution are uniformly random element over $\mathbb{Z}_N$.

Finally, we show that the statistical distance between the distributions of $\bar{z}_j$ in the real case and the simulated case is negligible for all $j \in [r]$. For $j \in [r]$, the statistical distance is 0 if $\ell_j = 0$. If $\ell_j = 1$, we have to bound the statistical distance between $U([\mathsf{Chall}+2^{\lambda}, \mathsf{Chall}+B_\lambda])$ and $U([2^\lambda, B_\lambda])$, where $B_\lambda = 2^{2\lambda}-1$. We compute

$$\begin{aligned}\textstyle\sum_{i=0}^{\infty} \big| \Pr[z_j \hookleftarrow U([2^\lambda, B_\lambda] + \mathsf{Chall}) : z_j = i] - \Pr[z_j \hookleftarrow U([2^\lambda, B_\lambda]) : z_j = i] \big| \\ \le \textstyle\sum_{i=2^\lambda}^{2^\lambda+\mathsf{Chall}-1} 2^{1-2\lambda} + \sum_{i=B_\lambda+1}^{\mathsf{Chall}+B_\lambda} 2^{1-2\lambda} \le 2 \cdot \mathsf{Chall} \cdot 2^{1-2\lambda} \le 2^{2-\lambda}.\end{aligned}$$

When the order of h is a multiple of N and the order of $\bar{h}$ is a multiple of $\bar{N}$, this shows that both distributions are within distance $r \cdot 2^{2-\lambda}$, with $r = \mathcal{O}(\log \lambda)$. □

E Simpler Ring Signatures in the Erasure Setting

In this section, we describe a simpler variant of our main scheme where the signer is required to erase its random coins after each signature generation. The main difference is the commitment scheme that allows computing $\{L_j\}_{j=1}^{r}$. Instead of computing each L_j using an $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme, the simplified construction uses the Paillier-based commitment of [20].

CRSGen(1^λ) : Given a security parameter λ, conduct the following steps.

1. Generate $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ for the trapdoor Σ-protocol of Section 4.2.
2. Generate RSA moduli $N = pq$ and $\bar{N} = \bar{p}\bar{q}$ such that $p, q, \bar{p}, \bar{q} > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$, and choose $h \hookleftarrow U(\mathbb{Z}_{N^2}^*)$, $\bar{h} \hookleftarrow U(\mathbb{Z}_{\bar{N}^2}^*)$.
3. Generate a pair $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}_\vee^{1\text{-}R})$ consisting of the common reference string $\mathsf{crs} = \big(\mathsf{par}, (\mathsf{crs}'_{\mathcal{L}}, \mathsf{pk}_{\mathsf{LPKE}}, k, \mathsf{AHF}, \mathsf{OTS})\big)$ of a simulation-sound argument Π^{uss} for the language $\mathcal{L}_\vee^{1\text{-}R}(h, \bar{h})$ defined in (12) together with a simulation trapdoor $\tau_{\mathsf{zk}} := \mathsf{sk}_{\mathsf{LPKE}}$. In crs, the language-dependent $\mathsf{crs}'_{\mathcal{L}} = \{N, \bar{N}\}$ is part of a common reference string $\mathsf{crs}' = (\{\lambda\}, \mathsf{crs}'_{\mathcal{L}})$ for the Σ-protocol of Section 4.2.

Output the common reference string $\rho = \mathsf{crs}$.

Keygen(ρ) : Pick $w \hookleftarrow U(\mathbb{Z}_N^*)$, $y \hookleftarrow U(\mathbb{Z}_N)$ and compute $C = h^y \cdot w^N \bmod N^2$. Output (sk, vk), where $sk = (w, y)$ and $vk = C$.

Sign$(\rho, sk, M, \mathsf{R})$: Given a ring $\mathsf{R} = \{vk_0, \ldots, vk_{R-1}\}$ (we assume that $R = 2^r$ for some $r \in \mathbb{N}$), a message M and a secret key $sk = (w, y) \in \mathbb{Z}_N^* \times \mathbb{Z}_N$, let $\ell_1 \cdots \ell_r = \ell \in \{0, \ldots, R-1\}$ be the index such that $vk_\ell = h^y \cdot w^N \bmod N^2$.

1. For each $j \in [r]$, choose $s_j \hookleftarrow U(\mathbb{Z}_{\bar{N}})$ and $t_j \hookleftarrow U(\mathbb{Z}_{\bar{N}}^*)$, and compute the commitment $L_j = (1 + \bar{N})^{\ell_j} \cdot \bar{h}^{s_j} \cdot t_j^{\bar{N}} \bmod \bar{N}^2$ to the bit ℓ_j.
2. Define the label $\mathsf{lbl} = (M, \mathsf{R})$ and generate a NIZK argument $\boldsymbol{\pi} \leftarrow \mathsf{P}\big(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w}, \mathsf{lbl}\big)$ that $\boldsymbol{x} \triangleq ((vk_0, \ldots, vk_{R-1}), (L_1, \ldots, L_r)) \in \mathcal{L}_\vee^{1\text{-}R}(h, \bar{h})$ by running the prover P of Supplementary Material B with the Σ-protocol of Section 4.2 using $\boldsymbol{w} = (y, w, \{(\ell_j, s_j, t_j)\}_{j=1}^r)$.

Output the signature $\boldsymbol{\Sigma} = ((L_1, \ldots, L_r), \boldsymbol{\pi})$ and erase all random coins.

Verify$(\rho, M, \boldsymbol{\Sigma}, \mathsf{R})$: Given a signature $\boldsymbol{\Sigma} = ((L_1, \ldots, L_r), \boldsymbol{\pi})$, a message M and a ring $\mathsf{R} = \{vk_0, \ldots, vk_{R-1}\}$, return 0 if any of these does not parse properly. Otherwise, let $\mathsf{lbl} = (M, \mathsf{R})$ and return $\mathsf{V}(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{\pi}, \mathsf{lbl})$ which outputs 1 iff $\boldsymbol{\pi}$ is a valid argument that $\big((vk_0, \ldots, vk_{R-1}), (L_1, \ldots, L_r)\big) \in \mathcal{L}_\vee^{1\text{-}R}(h, \bar{h})$.

Theorem E.1. *The above ring signature instantiated using the trapdoor Σ-protocol of Section 4.2 provides unforgeability assuming reliable erasures and under the assumptions that: (i) The* DCR *assumption holds; (ii) Π^{uss} is an unbounded simulation-sound NIZK argument for the language $\mathcal{L}_\vee^{1\text{-}R}(h, \bar{h})$.*

Proof. We use a sequence of games starting with the real experiment and ending with a game where we give a direct reduction from the simulation-soundness of Π^{uss}. For each i, W_i is the event that the challenger outputs 1 in Game_i.

Game_0: This is the real experiment. The adversary $\mathcal{A}$ receives a CRS ρ and has access to a key generation oracle Keygen, a signing oracle Sign and a corruption oracle $\mathsf{Corrupt}$. At the i-th query to Keygen, the challenger returns a verification key $vk^{(i)} = h^{y_i} \cdot w_i^N \bmod N^2$ for some $w_i \hookleftarrow U(\mathbb{Z}_N^*)$, $y_i \hookleftarrow U(\mathbb{Z}_N)$ and keeps $sk^{(i)} = (w_i, y_i)$ for later use. If $\mathcal{A}$ makes a corruption query $\mathsf{Corrupt}(i)$, the challenger reveals $sk^{(i)}$ to $\mathcal{A}$. At each signing query (i, M, R), the challenger returns $\perp$ if R contains a key $vk \notin \mathbb{Z}_{N^2}^*$. Otherwise, it runs $\boldsymbol{\Sigma} \leftarrow \mathsf{Sign}(\rho, sk, M, \mathsf{R})$ and returns $\boldsymbol{\Sigma}$ to $\mathcal{A}$. When $\mathcal{A}$ halts, it outputs a triple $(M^\star, \mathsf{R}^\star, \boldsymbol{\Sigma}^\star)$, where $\mathsf{R}^\star = \{vk_0^\star, \ldots, vk_{R^\star-1}^\star\}$ and $\boldsymbol{\Sigma}^\star = ((L_1^\star, \ldots, L_r^\star), \boldsymbol{\pi}^\star)$, and the challenger outputs 1 if: (i) $\mathsf{Verify}(\rho, M^\star, \boldsymbol{\Sigma}^\star, \mathsf{R}^\star) = 1$; (ii) $\mathsf{R}^\star$ only contains uncorrupted keys produced by Keygen; (iii) No signing query $(\cdot, M^\star, \mathsf{R}^\star)$ was made. By definition, we have $\Pr[W_0] = \mathbf{Adv}_{\mathcal{A}}^{\text{unforge}}(\lambda)$.

Game_1: This game is like Game_0 except that, in all signing queries (i, M, R), the challenger simulates the Sign oracle by running the NIZK simulator of Π^{uss} instead of using the real witness. Namely, the commitments $\{L_j\}_{j=1}^r$ of each signature $\boldsymbol{\Sigma} = ((L_1, \ldots, L_r), \boldsymbol{\pi})$ still commit to the bits $(\ell_1, \ldots, \ell_r)$ of $vk^{(i)}$'s location in R but $\boldsymbol{\pi}$ is simulated without using $sk^{(i)} = (w_i, y_i)$ nor $\{(\ell_j, s_j, t_j)\}_{j=1}^r$. Recall that the trapdoor Σ-protocol of Section 4.2 is statistically special ZK when $h \sim U(\mathbb{Z}_{N^2}^\star)$ and $\bar{h} \sim U(\mathbb{Z}_{\bar{N}^2}^\star)$. The statistical NIZK property of Π^{uss}, ensures that $|\Pr[W_1] - \Pr[W_0]| \leq Q_S \cdot 2^{-\Omega(\lambda)}$.

Game_2: This game is like Game_1 except that we change the distribution of crs in $\rho = \mathsf{crs}$. For the parameters, we now sample $h_0 \hookleftarrow U(\mathbb{Z}_N^*)$ and $\bar{h}_0 \hookleftarrow U(\mathbb{Z}_{\bar{N}}^*)$ and compute $h = h_0^N \bmod N^2$ and $\bar{h} = \bar{h}_0^{\bar{N}} \bmod \bar{N}^2$ as uniformly random composite residues. Under the DCR assumption in $\mathbb{Z}_{N^2}^*$ and $\mathbb{Z}_{\bar{N}^2}^*$, these changes have no noticeable impact on $\mathcal{A}$'s forging probability and a straightforward reduction shows that $|\Pr[W_2] - \Pr[W_1]| \leq 2 \cdot \mathbf{Adv}^{\mathsf{DCR}}(\lambda)$.

Game_3: In this game, the challenger uses the factorization of $\bar{N} = \bar{p}\bar{q}$ to decrypt the Paillier ciphertexts $\{L_j^\star\}_{j=1}^r$ contained in $\mathcal{A}$'s forgery. The challenger obtains $\{\ell_j^\star\}_{j=1}^r$ and outputs 0 if there exists $j \in [r]$ such that $\ell_j^\star \notin \{0, 1\}$. Otherwise, it obtains a string $\ell_1^\star \ldots \ell_r^\star \in \{0,1\}^r$ and reconstructs the index $\ell^\star = \sum_{k=1}^r \ell_k^\star \cdot 2^{k-1} \in \mathbb{Z}_R$ of the verification key $vk_{\ell^\star}^\star = C_{\ell^\star}^\star$ in the ring $\mathsf{R}^\star = \{vk_0^\star, \ldots, vk_{R-1}^\star\}$. If $\ell_j^\star \notin \{0,1\}$ for some $j \in [r]$, the soundness of Π^{uss} is broken and we have $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\text{uss}}(\lambda)$.

Game_4: This game is like Game_3 with one change. At the outset of the game, the challenger draws $i^\star \hookleftarrow U([Q_V])$ as a guess that $vk_{\ell^\star}^\star$ coincides with the verification key $vk^{(i^\star)}$ returned by the challenger at the $i^\star$-th query to the Keygen oracle. If the guess eventually turns out to be wrong or if $\mathcal{A}$ makes the corruption query $\mathsf{Corrupt}(i^\star)$, the challenger aborts and outputs 0. Otherwise (i.e., if $vk_{\ell^\star}^\star = vk^{(i^\star)}$), it outputs the same bit as in Game_3. Since $i^\star$ is chosen independently of $\mathcal{A}$'s view, it is correct with probability $1/Q_V$, where Q_V is the number of Keygen-queries. We thus have $\Pr[W_4] = \Pr[W_3]/Q_V$.

Game_5: We change the distribution of $vk^{(i^\star)} = C^{(i^\star)}$ and sample $C^{(i^\star)} \hookleftarrow U(\mathbb{Z}_{N^2}^*)$ uniformly instead of sampling it as an N-th residue in $\mathbb{Z}_{N^2}^*$. As a result, the signing oracle may now return simulated arguments for false statements. However, since the challenger uses neither the factorization of N nor the secret key $sk^{(i^\star)}$ in Game_4, we can rely on the DCR assumption in $\mathbb{Z}_{N^2}^*$ to argue that $|\Pr[W_5] - \Pr[W_4]| \leq \mathbf{Adv}^{\mathsf{DCR}}(\lambda)$.

In Game_5, we claim that $\Pr[W_5] \leq \mathbf{Adv}^{\text{uss}}(\lambda) + 2^{-\Omega(\lambda)}$ as, except with probability $1/N < 2^{-\Omega(\lambda)}$, the challenger only outputs 1 if $\mathcal{A}$ manages to break the simulation-soundness of Π^{uss}. Indeed, W_5 only occurs if $vk_{\ell^\star}^\star = vk^{(i^\star)}$ (which implies that $sk^{(i^\star)}$ was not corrupted); $\boldsymbol{\pi}^\star$ is a valid argument for the statement $((vk_0^\star, \ldots, vk_{R-1}^\star), (L_1^\star, \ldots, L_r^\star)) \in \mathcal{L}_{\vee}^{1\text{-}R}$; and no signing query $(i, M^\star, \mathsf{R}^\star)$ has been made for any $vk^{(i)} \in \mathsf{R}^\star$ (in particular for $i = i^\star$). Since $vk_{\ell^\star}$ was sampled uniformly in $\mathbb{Z}_{N^2}^*$, it is *not* an N-th residue except with probability $1/N$. If W_5 occurs, this necessarily implies that $\boldsymbol{\pi}^\star$ is an accepting argument for a false statement $\boldsymbol{x}^\star \in \mathcal{L}_{\mathsf{V}}^{1\text{-}R}$ on an unqueried label $\mathsf{lbl}^\star \triangleq (M^\star, \mathsf{R}^\star)$.

Putting the above altogether, we can bound the adversary's advantage as

$$\mathbf{Adv}_{\mathcal{A}}^{\text{unforge}}(\lambda) \leq (2 + Q_V) \cdot \Big(\mathbf{Adv}^{\text{uss}}(\lambda) + \mathbf{Adv}^{\mathsf{DCR}}(\lambda)\Big) + (Q_S + Q_V) \cdot 2^{-\Omega(\lambda)}$$

where Q_V and Q_S are the number of Keygen-queries and signing queries. $\square$

The proof of anonymity follows from the fact that the Paillier-based commitments are perfectly hiding when the parameters contained in ρ are configured in such a way that the order of h (*resp.* $\bar{h}$) in $\mathbb{Z}_{N^2}^*$ (*resp.* in $\mathbb{Z}_{\bar{N}^2}^*$) is at least N (*resp.* $\bar{N}$). Then, the statistical NIZK property of Π^{uss} ensures that the signatures produced by any two distinct signers have statistically indistinguishable distributions.

Theorem E.2. *The above scheme instantiated with the trapdoor Σ-protocol of Section 4.2 provides full anonymity under key exposure provided Π^{uss} is a statistical NIZK argument for the language $\mathcal{L}_{\mathsf{V}}^{1\text{-}R}(h, \bar{h})$.*

Proof. We consider a sequence of statistically indistinguishable games that ends with a game where the adversary has no advantage. In each game, we call W_i the even that the adversary outputs 1 in Game_i.

Game_0: This is the real anonymity game. Namely, the adversary is granted access to a Keygen oracle that returns a pair $(vk^{(i)}, sk^{(i)})$ at each query. In the challenge phase, the adversary $\mathcal{A}$ chooses a message $M^\star$ together with a ring $\mathsf{R}^\star = \{vk_0^\star, \ldots, vk_{R^\star-1}^\star\}$ and two indices $i_0, i_1 \in \mathbb{Z}_R$ such that $vk_{i_0}^\star$ and $vk_{i_1}^\star$ were both produced by the Keygen oracle. The challenger then flips a fair coin $b \hookleftarrow U(\{0,1\})$ and computes $\boldsymbol{\Sigma}^\star \leftarrow \mathsf{Sign}(\rho, sk_{i_b}^\star, M^\star, \mathsf{R}^\star)$. At the end of the game, the adversary outputs a bit $b' \in \{0,1\}$ and the challenger outputs 1 if and only if $b' = b$.

Game_1: In this game, the challenger generates $\boldsymbol{\Sigma}^\star = ((L_1^\star, \ldots, L_r^\star), \boldsymbol{\pi}^\star)$ without using $sk_{i_b}^\star$. Namely, $(L_1^\star, \ldots, L_r^\star)$ are still generated as Paillier commitments to the binary representation of i_b but $\boldsymbol{\pi}^\star$ is obtained by running the NIZK simulator of Π^{uss}. The latter's statistical NIZK property guarantees that $|\Pr[W_1] - \Pr[W_0]| \leq 2^{-\Omega(\lambda)}$. More precisely, when all dual-mode commitments are computed in their perfectly hiding mode, the trapdoor Σ-protocol of Section 4.2 provides statistical special zero-knowledge. In this setting, Π^{uss} is statistically ZK.

In Game_1, we have $\Pr[W_1] = 1/2$ since $(L_1^\star, \ldots, L_r^\star)$ are perfectly hiding commitments and $\boldsymbol{\pi}^\star$ is generated independently of $b \in \{0,1\}$. $\square$

F Optimized Ring Signature Size

In this section, we assess the size of signatures $\boldsymbol{\Sigma} = (\mathsf{VK}, (L_1, \ldots, L_r), \boldsymbol{\pi}, sig)$ in the erasure-free scheme of Section 5. If we use a DCR-based instantiation of the generic one-time signature suggested by Mohassel [61], we can re-use a Paillier modulus N_{ots} from the CRS of our ring signature to generate many one-time key pairs. In this case, VK has the size of 3 group elements modulo N_{ots} and the signature sig has the size of 2 group elements (as described at the end of this section). The size of $(L_1, \ldots, L_r)$ amounts to r elements modulo $\bar{N}^2$. We now focus on the size of the NIZK argument $\boldsymbol{\pi} \leftarrow \mathsf{P}\big(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w}, \mathsf{lbl}\big)$.

Recall that the instance $\boldsymbol{x} \triangleq ((vk_0, \ldots, vk_{R-1}), (L_1, \ldots, L_r))$ is in the language $\mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h}_{\mathsf{VK}})$ given in equation (21). While verifying a signature requires $\boldsymbol{x}$ and $\bar{h}_{\mathsf{VK}}$, the ring $\mathsf{R} = (vk_0, \ldots, vk_{R-1})$ is not part of the signature and $\bar{h}_{\mathsf{VK}}$ can be recomputed from VK and the CRS. Moreover, $\mathsf{lbl} = \mathsf{VK}$, so that we are left with evaluating the size of $\boldsymbol{\pi}$. Now, we parse $\boldsymbol{\pi} = \big(\mathsf{VK}', \mathbf{z}, sig'\big)$ as in Section B, where (VK', sig') is another one-time key pair. It is actually easy to avoid the need for a second pair (VK', sig') as the ring signature can actually recycle the one-time pair (VK, sig) from the simulation-sound argument $\boldsymbol{\pi}$ without affecting the security. We decided to use separate one-time pairs in the description for the sake of clarity and modularity.

In the optimized version, we are left with evaluating the size of the response $\mathbf{z} = (\mathbf{z}', \mathbf{a}', \mathbf{r})$, where $\mathbf{a}' = \big(\{(\bar{A}_j, \bar{B}_j)\}_{j=1}^{r},\ \{C_{d_k}\}_{k=0}^{r-1}\big)$ is the first message; $\mathbf{z}' = \big(z_y, z_w, \{(\bar{z}_j, \bar{z}_{d,j}, \bar{z}_{e,j}, \bar{z}_{u,j}, \bar{z}_{v,j})\}_{j=1}^{r}\big)$ the response of the 1-out-of-R trapdoor Σ-protocol of Section 4.2; and $\mathbf{r}$ is the random coin used to encrypt $\mathbf{a}'$ with the $\mathcal{R}$-lossy PKE scheme of Section 3.1. We can further optimize the signature by not including $\{(\bar{A}_j, \bar{B}_j)\}_{j=1}^{r}$ and C_{d_0} in $\boldsymbol{\pi}$ and replacing them by the challenge Chall in the argument system of Section B since they can be recomputed given $\mathbf{z}'$, Chall and $\{C_{d_k}\}_{k=1}^{r-1}$, as done in equations (28)-(29). The resulting ring signature is easily seen to be equivalent to its original version.

Eventually, we assume that all moduli have the same bitlength and that the modulus $\tilde{N}$ of the $\mathcal{R}$-lossy PKE is the largest one. This allows encoding $\mathbf{a}'$ as a single plaintext modulo $\tilde{N}^{6r}$ in the $\tilde{N}^2$-adic representation since we have $3r$ elements modulo a square modulus. Therefore, $\mathbf{r} = (\tilde{r}, \tilde{s}) \in \mathbb{Z}_{\tilde{N}}^{*} \times \mathbb{Z}_{\tilde{N}^{6r}}$ and we can estimate the global length as follows, where we count an element modulo the ζ-th power of a modulus N as ζ elements of $\mathbb{Z}_N$.

– One λ-bit string for Chall and 5 elements for (VK, sig);
– $2r$ elements modulo N: $r-1$ for $\{C_{d_k}\}_{k=1}^{r-1}$ and 2 for (z_y, z_w);
– $6r$ elements modulo $\bar{N}$: r for $\{L_j\}_{j=1}^{r}$ and $4r$ for $\{(\bar{z}_{d,j}, \bar{z}_{e,j}, \bar{z}_{u,j}, \bar{z}_{v,j})\}_{j=1}^{r}$;
– r strings of $2\lambda + 1$ bits each for $\{\bar{z}_j\}_{j=1}^{r}$;
– $6r+1$ elements modulo $\tilde{N}$ for the random coins $\mathbf{r}$ of the $\mathcal{R}$-lossy PKE scheme.

This amounts to a total size bounded by the size of $15r + 7$ elements modulo $\tilde{N}$. Interestingly, the signature length is only roughly three times as large as in a ring signature obtained by applying the standard Fiat-Shamir heuristic under the DCR assumption in the random oracle model. In a ROM-based version of our scheme, we can apply Fiat-Shamir to our Σ-protocol of Section 4.2 and simulation-soundness comes for free without using tag-based commitments or even one-time signatures. Even though we can apply exactly the same proof strategy as Groth and Kohlweiss [46] using more efficient DCR-based commitments of the form $L_j = g^{\ell_j} \cdot r^N \bmod N$, the signature size only drops to $5r+1$ elements of $\mathbb{Z}_N$. At the 128-bit security level, we may choose moduli of size $|N| = 3072$ and end up with signatures of size smaller than 1.57Mb for rings of size $|R| < 10^{82}$ (which is an estimation on the number of atoms in the universe).

Short Keys. Each user's public key consists of a single element $vk = C$ over $\mathbb{Z}_{N^2}^*$ and the secret key is a pair $(w, y) \in \mathbb{Z}_N^* \times \mathbb{Z}_N$. In both cases, the size is equivalent to that of two elements of $\mathbb{Z}_N$.

A Concrete DCR*-based One-Time Signature.* For completeness, we describe a strongly unforgeable one-time signature based on a chameleon hash function (or, equivalently, a trapdoor commitment) under the DCR assumption, which follows the blueprint of [61]. Using a DCR-based variant of Mohassel's generic construction allows recycling the modulus N across different one-time key generations. Since the chameleon hashing trapdoors are N-th roots, the factorization of N can remain unknown to signers. For this reason, we give a construction that does not rely on the factoring-based instantiation of [61] since we already have RSA moduli in the CRS and signers can re-use them. The construction hereunder can be proven strongly unforgeable under the RSA assumption with public exponent N, which is implied by the DCR assumption, as shown in [64].

CRSGen(1^λ): Given a security parameter λ, generate an RSA modulus $N = pq$ satisfying $p, q > 2^{l(\lambda)}$, where $l : \mathbb{N} \to \mathbb{N}$ is a polynomial, choose $H \hookleftarrow \mathcal{H}$ from a family $\mathcal{H}$ of collision-resistant hash functions. Return $pp = (N, H)$.

Keygen(pp): Choose random $u, v, w \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $g = u^N \bmod N$, $h = v^N \bmod N$ and $c = w^N \bmod N$ (which is seen as a commitment to 0 with commitment key g). Return $osk = (u, v, w)$ and $ovk = (g, h, c)$.

Sign(ovk, osk, m): Given a secret key $osk = (u, v, w)$, pick $s \hookleftarrow U(\mathbb{Z}_N^*)$ and compute $d = h^m \cdot s^N \bmod N$ and $e = H(g, h, d)$. Then, use w and u to equivocate c for the message e. Namely, compute $r = w \cdot u^{-e} \bmod N$ and output the signature $\sigma = (r, s)$.

Verify(ovk, m, σ): Given $\sigma = (r, s)$ and a verification key $ovk = (g, h, c)$, output 1 if $c = g^e \cdot r^N \bmod N$, where $d = h^m \cdot s^N \bmod N$ and $e = H(g, h, d)$, and 0 otherwise.

Since it is well-known that $c = g^e \cdot r^N \bmod N$ and $d = h^m \cdot s^N \bmod N$ are trapdoor commitments, the strong unforgeability directly follows from [61].

G Comparison with Other Instantiations

G.1 Comparison with an LWE-based $\mathcal{R}$-Lossy PKE Scheme

We note that our ring signatures would be much longer if we were to use the equivocable $\mathcal{R}$-lossy PKE scheme of [56] instead of the DCR-based construction of Section 3.1. A single ciphertext would contain $n \cdot \log q$ bits, where n is larger than the length of the plaintext (i.e., the first prover message in the Σ-protocol of Section 4.2, where $r = 1$ already implies $n > 6 \cdot 3072 = 18432$ when $|N| = 3072$ at the 128-bit security level) and $\log q > 50$. For $r = 1$, this would be over 20 times larger than our entire ring signature and it would only get worse for $r > 1$.

G.2 Comparison with Possible DDH-based Instantiations

As mentioned in the introduction, the discrete-logarithm-based Σ-protocol of Groth and Kohlweiss [46] is not immediately compatible with the bad-challenge-function methodology. The main obstacle is that, in order to compile it into a trapdoor Σ-protocol, adapting the transformation of Ciampi *et al.* [27] would require the prover to send encrypted responses to all possible challenges along with its first message. This requires to proceed with a challenge space of polynomial size λ^c, for some constant c, and $\kappa = O(\lambda / \log \lambda)$ repetitions. In order to identify the bad challenges for a given first prover message, we run into a problem since a single protocol execution has $(r+1)$-special-soundness (so that up to $r > 1$ bad challenges may exist). Over κ iterations, this leads to a super-polynomial number r^κ of bad challenges, which is no longer compatible with correlation intractable hash functions for efficiently enumerable relations.

Recently, Holmgren *et al.* [51] introduced a technique to address this combinatorial blow-up using list-recoverable codes. If $S = S_1 \times \ldots \times S_\kappa$ denotes the product set of bad challenges at each iteration (with size $|S_i| \leq r$ for each $i \in [\kappa]$), they construct a list-recoverable code $(\mathsf{Encode}, \mathsf{Recover})$ such that the set $\mathcal{M}_S$ of input messages m for which $\mathsf{Encode}(m) \in S = S_1 \times \ldots \times S_\kappa$ has cardinality smaller than $L \in \mathsf{poly}(\lambda)$. Moreover, there is an efficient algorithm $\mathsf{Recover}(S_1, \ldots, S_\kappa)$ that ouputs M_S on input of $S_1, \ldots, S_\kappa$. As shown in [51], this allows instantiating Fiat-Shamir by applying correlation-intractable hash functions to Σ-protocols with more than one bad challenge at each repetition. To this end, they use a CI hash function of the form $H'(a) = \mathsf{Encode}(H(a))$, where H is a correlation-intractable hash function for efficiently enumerable relations (e.g., [67]). Intuitively, the corresponding bad challenge function first computes the product set $S = S_1 \times \ldots S_\kappa$ of bad challenges determined by the first prover's message a. It then runs the decoding algorithm to output the polynomial-size set $M_S = \mathsf{Decode}(S_1, \ldots, S_\kappa)$ of challenges that $H'(a)$ should avoid.

The technique of [51][11] suggests possible instantiations of our high-level approach under the DDH assumption (we insist that these do not directly follow from [46,51] as they rely on our proof technique, which departs from that of Groth and Kohlweiss [46]). Nevertheless, they would be more expensive than in the DCR setting. Indeed, in the discrete-log-based Σ-protocol of [46], the prover's first message consists of $r = O(\log R)$ commitments for one iteration. In order

[11] It was proven to soundly instantiate Fiat-Shamir for a wide family of Commit-and-Open protocols repeated in parallel. While the original Σ-protocol of [46] does not quite fit the Commit-and-Open abstraction of [51], it does when we apply the transformation of [27] to obtain a trapdoor Σ-protocol.

to obtain a trapdoor Σ-protocol, the transformation of [27] requires to send $2r$ encrypted responses[12] (one for each possible challenge value) together with the first prover message of [46]. If we then apply the result of Holmgren *et al.* [51, Theorem 5.1] and proceed with $\kappa = \tilde{O}(\lambda)$ repetitions, we obtain a first prover message comprised of $\tilde{O}(\lambda^2)$ bits. However, our security proof requires to instantiate our USS argument system of Supplementary Material B under the DDH assumption. To this end, we need to encrypt the first prover message using a DDH-based analogue of our equivocable $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme in Section 3.1.

We are only aware of two DDH-based $\mathcal{R}_{\mathsf{BM}}$-lossy PKE schemes that suit our purposes. The first one encrypts messages bit-by-bit and can be obtained from the scheme of Boyle *et al.* [9, Section 4.2] with slight modifications that notably require to encode the plaintext in the exponent so as to achieve equivocation (see [5, Section 5.4] for details). We note that this construction additionally requires the simulation-sound argument to contain a proof that each $\mathcal{R}_{\mathsf{BM}}$-lossy PKE ciphertext encrypts a bit (whereas, in our DCR-based construction, all ciphertexts encrypt a valid message). Since the message to be encrypted has length $\tilde{O}(\lambda^2)$, this construction incurs a communication cost $\tilde{O}(\lambda^3)$ in our setting, which is asymptotically worse than the signature size $O(\lambda^3/\mathsf{polylog}(\lambda))$ enabled by DCR.

The second possible DDH-based $\mathcal{R}_{\mathsf{BM}}$-lossy PKE candidate is achievable from the equivocable lossy PKE scheme of Hofheinz *et al.* [49].[13] The latter provides short ciphertexts at the expense of a public key size $O(|m|^2 \cdot \lambda)$, where $|m|$ is the number of bits to encrypt, and randomness of length $|m| \cdot \lambda$. In order to turn [49] into an $\mathcal{R}_{\mathsf{BM}}$-lossy system, we need to combine it with an admissible hash function and increase the public key size by another factor $O(\lambda)$, thus leading to a total public key size $O(|m|^2 \cdot \lambda^2)$. To avoid an a priori upper bound on the ring size, we should allow for $|m| = \lambda^2 \cdot \omega(\log \lambda)$, thus resulting in a communication cost $\tilde{O}(\lambda^3)$ (since the random encryption coins of size $|m| \cdot \lambda = \tilde{O}(\lambda^3)$ should be part of the prover's response in our USS argument of Section B) and a CRS of size $\tilde{O}(\lambda^6)$. In all cases, the signature size and even the CRS size remain significantly larger than in the DCR setting.

The above comparisons only hold in the asymptotic regime. For concrete parameters $\lambda = 128$, we may set $|N| = 3072$ and more drastically outperform DDH-based instantiations. If we apply [51, Theorem 5.4] with $t > \lambda$ repetitions and a 256-bit group order, the concrete signature size exceeds $150 \cdot 10^6 \cdot r^2$ bits if the underlying $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme is instantiated from [49]. Under the DCR assumption, our signature length is strictly less than $60000 \cdot r$ bits.

[12] In order to keep the communication logarithmic in R, it is better to use a challenge space of size $2r$ (so as to have a fraction $\rho = 1/2$ of bad challenges) at each repetition.

[13] Due to the use of a universal hash function, this scheme is not compatible with efficient Σ-protocols but this is not necessary in our USS argument. We also need a "Σ-protocol-friendly" DDH analogue of our dense lossy PKE in Section 3.2. Since the latter construction does not have to support equivocation, it can be instantiated from the scheme of Boyle *et al.* [9, Section 4.2].