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 [66] 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 [67]. 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 [15,66] 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 gap 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 [55, 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 [72,4], 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 [47] 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.* [55] 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 [55], 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 [55] 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 [55], 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 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 [66], we construct a variant of GK with an exponentially large challenge space and where 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 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 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.* [50] recently gave a technique allowing to address the combinatorial explosion of bad challenges induced by parallel repetitions. In the full version of the paper [56], 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 $\ell_1^\star \ldots \ell_r^\star \in \{0,1\}^r$ 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 $\ell_1^\star \ldots \ell_r^\star$. 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 $\ell_1^\star \ldots \ell_r^\star \in \{0,1\}^r$ 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 [71,16,49,14,26,29,59,10,52] 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 [66] subsequently achieved the same result under the standard LWE assumption [67].

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 [14,26,59,52] 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 [68]. 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 [68], a number of solutions were given under various assumptions [12,1,70,11,46,65,2]. Bender *et al.* [6] gave stronger definitions and constructions from general assumptions. In the standard model, more efficient schemes were given [70,8] in groups with a bilinear map. Brakerski and Tauman [11] gave the first constructions from lattice assumptions.

In early realizations [68,12,70,8], 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 [46,54,57] 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 ([67]). *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** $(\zeta\text{-}\mathsf{DCR})$ *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 [66] 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 [18,52].

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 $X^{(1)}, \ldots, X^{(Q)}, X^\star \in \{0,1\}^\ell$ such that $X^\star \notin \{X^{(1)}, \ldots, X^{(Q)}\}$, we have*

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

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

Theorem 2.6 ([51, 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.*

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 [55], 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 [53] 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 [55] 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, [55] 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 [55] 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 $D_{\mathrm{inj}} = \{(\mathsf{pk}, \mathsf{tk}) \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)\}$ and $D_{\mathrm{loss}} = \{(\mathsf{pk}, \mathsf{tk}) \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ are computationally indistinguishable. For any PPT adversary $\mathcal{A}$, the following advantage function $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{indist\text{-}LPKE}}(\lambda)$ is negligible:

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

(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* 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* 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 $\{\overline{r} \leftarrow \mathsf{LOpener}(\mathsf{pk}, \mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r) \mid r \hookleftarrow D_R\}$ is statistically close to $\{\overline{r} \leftarrow \mathsf{Opener}(\mathsf{pk}, \mathsf{tk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r) \mid r \hookleftarrow D_R\}$.*

The above definition is slightly weaker than the one of [55] in the property of equivocation under lossy keys. Here, we do not require that the output of 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 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 the full version [56] of the paper) and will allow us to obtain a construction from the DCR assumption.

Definition 2.9 also differs from [55, Definition 2.10] in that the equivocation algorithms (Opener, 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* (Par-Gen, Keygen, LKeygen, Encrypt, 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* $\mathsf{LKeygen}$ *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* $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{weak\text{-}indist\text{-}LPKE}}(\lambda)$*, 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

A ring signature [68] 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 the full version of the paper. In particular, we consider unforgeability with respect to insider corruption and statistical anonymity.

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

Libert *et al.* [55] 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 [55] is recalled in the full version [56] of the paper, 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 [55], 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

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

$\textsf{Keygen}(\Gamma, K)$: 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 $l(\lambda) > L(\lambda)$ 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

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

$\textsf{LKeygen}(\Gamma, K)$: is identical to $\textsf{Keygen}$ 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 $\mathsf{pk} := \big(N, \zeta, g, \{v_{i,b}\}_{i\in[L], b\in\{0,1\}}\big)$, $\mathsf{tk} = \big(\{\alpha_{i,b}\}_{i\in[L], b\in\{0,1\}}, K\big)$.

$\textsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg})$: 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

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

$\textsf{Decrypt}(\mathsf{sk}, t, \mathsf{ct})$: 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 $\beta_g = \frac{(g^{\lambda(N)} \bmod N^{\zeta+1}) - 1}{N}$, where $\lambda(N) = \mathrm{lcm}(p-1, q-1)$ and return $\perp$ if $\beta_g = 0$ or $\gcd(\beta_g, N^\zeta) > 1$.
2. Otherwise, compute $\mathsf{Msg} = \frac{(\mathsf{ct}^{\lambda(N)} \bmod N^{\zeta+1}) - 1}{N} \cdot \beta_g^{-1} \bmod N^\zeta$, where the division is computed over $\mathbb{Z}$, and output $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$.

$\textsf{Opener}\big(\mathsf{pk}, \mathsf{tk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, (r, s)\big)$: 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

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

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

$$\begin{aligned} \bar{s} &= s + (d_t^{-1} \bmod N^\zeta) \cdot (\mathsf{Msg}_0 - \mathsf{Msg}_1) \mod N^\zeta \\ \bar{r} &= r \cdot \prod_{i=1}^{L} \alpha_{i,t_i}^{s-\bar{s}} \cdot g^{(\mathsf{Msg}_0 - \mathsf{Msg}_1 + d_t \cdot (s - \bar{s}))/N^\zeta} \mod N, \end{aligned} \tag{4}$$

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

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

$\mathsf{LOpener}\big(\mathsf{pk}, \mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, (r, s)\big)$: 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.* (The proof is given in the full version of the paper.)

By plugging the above system in the construction described in the full version of the paper, we obtain USS arguments from the DCR and LWE assumptions. A difference with [55] is that LWE is only used in the correlation intractable hash function and lattice trapdoors are not needed anywhere. 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 [55], a single ciphertext would already be roughly 20 larger than an entire ring signature, as discussed in the full version of the paper.

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.

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 $l(\lambda) > L(\lambda) - \lambda$ 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 $\mathsf{tk} = \perp$.

LKeygen(Γ, 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 $\mathsf{tk} = \perp$.

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

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 $\mathsf{Msg} = \frac{(\mathsf{ct}^{\lambda(N)} \bmod N^{\zeta+1}) - 1}{N} \bmod N^\zeta$, where the division is computed over $\mathbb{Z}$, 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 $\mathbb{Z}_{N^{\zeta+1}}^*$.* (The proof is given in the full version of the paper.)

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 the full version of the paper, 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 BadChallenge function to obtain a trapdoor Σ-protocol with a large challenge space.

The 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]. Lipmaa 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 the full version of the paper, 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 $\tau_{\Sigma} = (p, q)$.

$\mathbf{P}(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w}) \leftrightarrow \mathbf{V}(\mathsf{crs}, \boldsymbol{x})$: 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

$$\begin{aligned} z &= a + \mathsf{Chall} \cdot b, & z_1 &= d + \mathsf{Chall} \cdot y, & z_2 &= e + (z - \mathsf{Chall}) \cdot y, \\ z_d &= z_1 \mod N, & z_u &= u \cdot w^{\mathsf{Chall}} \cdot h^{\lfloor z_1/N \rfloor} \mod N, & & \\ z_e &= z_2 \mod N, & z_v &= v \cdot w^{z - \mathsf{Chall}} \cdot h^{\lfloor z_2/N \rfloor} \mod N. & & \end{aligned}$$

4. V returns 1 if and only if $2^{\lambda} \leq z < 2^{2\lambda+1}$ and

$$\begin{aligned} A_1 &= C^{-\mathsf{Chall}} \cdot (1+N)^z \cdot h^{z_d} \cdot z_u^N \bmod N^2, \\ A_2 &= C^{\mathsf{Chall}-z} \cdot h^{z_e} \cdot z_v^N \mod N^2. \end{aligned} \tag{5}$$

$\mathsf{BadChallenge}(\mathsf{par}, \tau_{\Sigma}, \mathsf{crs}, \boldsymbol{x}, \boldsymbol{a})$: 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 $(\mathsf{Chall}, z) \in \mathbb{Z}_N^2$:

$$\begin{aligned} z - b \cdot \mathsf{Chall} &\equiv a_1 \pmod N, \\ b \cdot (\mathsf{Chall} - z) &\equiv a_2 \pmod N. \end{aligned} \tag{6}$$

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 (6) 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 (5) 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,

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

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 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

$$\begin{aligned}\mathcal{L}_{\vee}^{1\text{-}R}(h,\bar{h}) := \big\{ \big((C_0,\ldots,C_{R-1}),(L_1,\ldots,L_r)\big) \in (\mathbb{Z}_{N^2}^*)^R \times (\mathbb{Z}_{\bar{N}^2}^*)^r \ \mid \\ \exists y \in \mathbb{Z}_N,\ w \in \mathbb{Z}_N^*,\ \exists_{j=1}^{r}(\ell_j, s_j, t_j) \in \{0,1\} \times \mathbb{Z}_{\bar{N}} \times \mathbb{Z}_{\bar{N}}^* : \\ \textstyle\bigwedge_{j=1}^{r} L_j = (1+\bar{N})^{\ell_j}\, \bar{h}^{s_j}\, t_j^{\bar{N}} \bmod \bar{N}^2 \quad \wedge \quad C_\ell = h^y w^N \bmod N^2 \big\}\end{aligned} \tag{7}$$

where $R = 2^r$ and $\ell = \sum_{j=1}^{r} \ell_j \cdot 2^{j-1}$. In (7), $h \in \mathbb{Z}_{N^2}^*$ 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

$$\begin{cases} \bar{A}_j = (1 + \bar{N})^{\bar{a}_j} \cdot \bar{h}^{\bar{d}_j} \cdot \bar{u}_j^{\bar{N}} \bmod \bar{N}^2, \\ \bar{B}_j = (1 + \bar{N})^{-\bar{a}_j \cdot \ell_j} \cdot \bar{h}^{\bar{e}_j} \cdot \bar{v}_j^{\bar{N}} \bmod \bar{N}^2. \end{cases} \tag{8}$$

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

$$P_i[X] = \prod_{j=1}^{r} F_{j,i_j}[X] = \delta_{i,\ell} \cdot X^r + \sum_{k=0}^{r-1} p_{i,k} \cdot X^k \ \in \mathbb{Z}_N[X], \tag{9}$$

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 (9), P computes commitments

$$C_{d_k} = \prod_{i=0}^{R-1} C_i^{p_{i,k}} \cdot h^{\mu_k} \cdot \rho_k^N \bmod N^2 \qquad 0 \leq k \leq r-1, \tag{10}$$

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

$$\begin{cases} \bar{z}_{d,j} = \bar{d}_j + \mathsf{Chall} \cdot s_j \ \bmod \bar{N} \qquad\qquad \bar{z}_j = \bar{a}_j + \mathsf{Chall} \cdot \ell_j \\ \bar{z}_{e,j} = \bar{e}_j + (\bar{a}_j + \mathsf{Chall} \cdot (\ell_j - 1)) \cdot s_j \ \bmod \bar{N} \\ \bar{z}_{u,j} = \bar{u}_j \cdot \bar{t}_j^{\mathsf{Chall}} \cdot \bar{h}^{\lfloor (\bar{d}_j + \mathsf{Chall} \cdot s_j)/\bar{N} \rfloor} \ \bmod \bar{N} \\ \bar{z}_{v,j} = \bar{v}_j \cdot \bar{t}_j^{\bar{a}_j + \mathsf{Chall} \cdot (\ell_j - 1)} \cdot \bar{h}^{\lfloor (\bar{e}_j + (\bar{a}_j + \mathsf{Chall} \cdot (\ell_j - 1)) \cdot s_j)/\bar{N} \rfloor} \ \bmod \bar{N} \end{cases} \tag{11}$$

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

$$\begin{aligned} z_y &= y \cdot \mathsf{Chall}^r - \sum_{k=0}^{r-1} \mu_k \cdot \mathsf{Chall}^k \ \bmod N = P'(\mathsf{Chall}) \ \bmod N, \\ z_w &= w^{\mathsf{Chall}^r} \prod_{k=0}^{r-1} \rho_k^{-\mathsf{Chall}^k} \prod_{i=0}^{R-1} C_i^{-\lfloor P_i(\mathsf{Chall})/N \rfloor} \cdot h^{\lfloor P'(\mathsf{Chall})/N \rfloor} \ \bmod N, \end{aligned} \tag{12}$$

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 and only if $2^\lambda \leq \bar{z}_j < 2^{2\lambda+1}$ for all $j \in [r]$,

$$\forall j \in [r] : \begin{cases} \bar{A}_j = L_j^{-\mathsf{Chall}} \cdot (1 + \bar{N})^{\bar{z}_j} \cdot \bar{h}^{\bar{z}_{d,j}} \cdot \bar{z}_{u,j}^{\bar{N}} \bmod \bar{N}^2 \\ \bar{B}_j = L_j^{\mathsf{Chall} - \bar{z}_j} \cdot \bar{h}^{\bar{z}_{e,j}} \cdot \bar{z}_{v,j}^{\bar{N}} \quad \bmod \bar{N}^2 \end{cases} \tag{13}$$

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

$$\prod_{k=0}^{r-1} C_{d_k}^{-\mathsf{Chall}^k} \cdot \prod_{i=0}^{R-1} C_i^{(\prod_{j=1}^{r} f_{j,i_j} \bmod N)} \equiv h^{z_y} \cdot z_w^N \pmod{N^2}. \qquad (14)$$

$\mathsf{BadChallenge}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, \boldsymbol{x}, \boldsymbol{a}\big)$: 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 (9). 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 (13) are satisfied for the same reasons as in Section 4.1. As for equation (14), we observe that, if the witnesses $y \in \mathbb{Z}_N$

[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 [66] is compatible with $\mathsf{BadChallenge}$ functions failing with negligible probability.

and $w \in \mathbb{Z}_N^*$ satisfy $C_\ell = h^y \cdot w^N \bmod N^2$, we have

$$\begin{aligned}
h^{z_y} \cdot z_w^N \cdot \prod_{i=0}^{R-1} C_i^{-(\prod_{j=1}^{r} f_{j,i_j} \bmod N)} &\equiv h^{z_y} \cdot z_w^N \cdot \prod_{i=0}^{R-1} C_i^{-P_i(\mathsf{Chall}) \bmod N} \\
&\equiv h^{z_y} \cdot w^{\mathsf{Chall}^r \cdot N} \cdot \prod_{k=0}^{r-1} \rho_k^{-\mathsf{Chall}^k \cdot N} \cdot \prod_{i=0}^{R-1} C_i^{-P_i(\mathsf{Chall}) + (P_i(\mathsf{Chall}) \bmod N)} \\
&\qquad\qquad\qquad\qquad \cdot h^{P'(\mathsf{Chall}) - z_y} \cdot \prod_{i=0}^{R-1} C_i^{-P_i(\mathsf{Chall}) \bmod N} \\
&\equiv h^{P'(\mathsf{Chall})} \cdot w^{\mathsf{Chall}^r \cdot N} \cdot \prod_{k=0}^{r-1} \rho_k^{-\mathsf{Chall}^k \cdot N} \cdot \prod_{i=0}^{R-1} C_i^{-P_i(\mathsf{Chall})} \\
&\equiv h^{\mathsf{Chall}^r \cdot y} \cdot w^{\mathsf{Chall}^r \cdot N} \cdot \prod_{k=0}^{r-1} (h^{-\mathsf{Chall}^k \mu_k} \cdot \rho_k^{-\mathsf{Chall}^k \cdot N}) \\
&\qquad\qquad\qquad\qquad \cdot \prod_{i=0}^{R-1} C_i^{-\delta_{i,\ell} \cdot \mathsf{Chall}^r - \sum_{k=0}^{r-1} p_{i,k} \cdot \mathsf{Chall}^k} \\
&\equiv (h^y \cdot w^N)^{\mathsf{Chall}^r} \cdot \prod_{k=0}^{r-1} (h^{\mu_k} \cdot \rho_k^N)^{-\mathsf{Chall}^k} \cdot C_\ell^{-\mathsf{Chall}^r} \cdot \prod_{i=0}^{R-1} C_i^{-\sum_{k=0}^{r-1} p_{i,k} \cdot \mathsf{Chall}^k} \\
&\equiv \prod_{k=0}^{r-1} (h^{\mu_k} \cdot \rho_k^N)^{-\mathsf{Chall}^k} \cdot \prod_{k=0}^{r-1} \prod_{i=0}^{R-1} C_i^{-p_{i,k} \cdot \mathsf{Chall}^k} \equiv \prod_{k=0}^{r-1} C_{d_k}^{-\mathsf{Chall}^k} \pmod{N^2}.
\end{aligned}$$

Lemma 4.2. *The above construction is a trapdoor Σ-protocol for $\mathcal{L}_{\vee}^{1\text{-}R}$.* (The proof is available in the full version of the paper.)

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 given in the full version of the paper.)

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

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 $\ell_1^\star \dots \ell_r^\star \in \{0,1\}^r$ from the commitments $\{L_j^\star\}_{j=1}^r$ contained in the forgery $\boldsymbol{\Sigma}^\star = ((L_1^\star, \dots, L_r^\star), \boldsymbol{\pi}^\star)$. 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 $\mathsf{\Pi}^{\mathrm{uss}}$ 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 $\ell_1^\star, \dots, \ell_r^\star \in \{0,1\}^r$ from $\{L_j^\star\}_{j=1}^r$ (which is necessary to check that $\ell^\star = \ell_1^\star \dots \ell_r^\star$ 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

$$\begin{aligned} \mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h}_{\mathsf{VK}}) := \big\{ \big((C_0, \dots, C_{R-1}), (L_1, \dots, L_r) \big) \in (\mathbb{Z}_{N^2}^*)^R \times (\mathbb{Z}_{\bar{N}^2}^*)^r \quad | \\ \exists y \in \mathbb{Z}_N,\ w \in \mathbb{Z}_N^*,\ s_1, \dots, s_r \in \mathbb{Z}_{\bar{N}},\ t_1, \dots, t_r \in \mathbb{Z}_{\bar{N}}^*, \\ (\ell_1, \dots, \ell_r) \in \{0,1\}^r \quad : \quad C_\ell = h^y \cdot w^N \bmod N^2 \\ \wedge \quad L_j = (1+\bar{N})^{\ell_j} \cdot \bar{h}_{\mathsf{VK}}^{s_j} \cdot t_j^{\bar{N}} \bmod \bar{N}^2 \qquad \forall j \in [r] \ \big\}, \end{aligned} \tag{15}$$

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 (15).
- 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 $\mathcal{R}$-$\mathsf{LPKE} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ 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 $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}_{\vee}^{1\text{-}R})$ comprised of the CRS crs of an USS argument Π^{uss} (recalled in the full version of the paper) for the language $\mathcal{L}_{\vee}^{1\text{-}R}$ defined in (15) 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 $\boldsymbol{x} \triangleq ((vk_0, \ldots, vk_{R-1}), (L_1, \ldots, L_r)) \in \mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h}_{\mathsf{VK}})$ by running the prover P 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 $\boldsymbol{\Sigma} = (\mathsf{VK}, (L_1, \ldots, L_r), \boldsymbol{\pi}, sig)$.

$\textsf{\textbf{Verify}}(\rho, M, \boldsymbol{\Sigma}, \mathsf{R})$: 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 $\big((vk_0, \ldots, vk_{R-1}), (L_1, \ldots, L_r)\big) \in \mathcal{L}_{\vee}^{1\text{-}R}(h, \bar{h}_{\mathsf{VK}})$.

In the full version of the paper, 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 ring signature provides unforgeability if: (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* $\mathcal{L}_{\vee}^{1\text{-}R}$. (The proof is in the full version of the paper.)

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 given in the full version of the paper.

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* (15) *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

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 senior 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. B. Hemenway, B. Libert, R. Ostrovsky, D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. *Asiacrypt*, 2011.
48. D. Hofheinz, T. Jager, A. Rupp. Public-Key Encryption with Simulation-Based Selective-Opening Security and Compact Ciphertexts. *TCC*, 2016-B.
49. J. Holmgren, A. Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). *FOCS*, 2018.
50. J. Holmgren, A. Lombardi, R. Rothblum. Fiat-Shamir via List-Recoverable Codes (or: Parallel Repetition of GMW is not Zero-Knowledge). *STOC*, 2021.
51. T. Jager. Verifiable random functions from weaker assumptions. *TCC*, 2015.
52. R. Jawale, Y. Tauman-Kalai, D. Khurana, R. Zhang. SNARGs for bounded depth computations and PPAD hardness from sub-exponential LWE. *STOC*, 2021.
53. E. Kiltz. Chosen-ciphertext security from tag-based encryption. *TCC*, 2006.
54. 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.
55. B. Libert, K. Nguyen, A. Passelègue, R. Titiu. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. *Asiacrypt*, 2020.
56. B. Libert, K. Nguyen, T. Peters, M. Yung. One-Shot Fiat-Shamir-based NIZK Arguments of Composite Residuosity and Logarithmic-Size Ring Signatures in the Standard Model. Cryptology ePrint Archive: Report 2020/1334.

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. C. Peikert, S. Shiehian. Non-interactive zero knowledge for NP from (plain) Learning With Errors. *Crypto*, 2019.
67. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *STOC*, 2005.
68. R. Rivest, A. Shamir, Y. Tauman. How to Leak a Secret. *Asiacrypt*, 2001.
69. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. *FOCS*, 1999.
70. H. Shacham, B. Waters. Efficient ring signatures without random oracles. *PKC*, 2007.
71. Y. Tauman Kalai, G. Rothblum, R. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. *Crypto*, 2017.
72. A. Young, M. Yung. Questionable encryption and its applications. *Mycrypt*, 2005.