Simulation-Sound Arguments for LWE and Applications to KDM-CCA2 Security

Benoît Libert, Khoa Nguyen, Alain Passelègue, Radu Titiu

Simulation-Sound Arguments for LWE and Applications to KDM-CCA2 Security

Benoît Libert[1,2], Khoa Nguyen[3], Alain Passelègue[4,2], and Radu Titiu[5,2]

[1] CNRS, Laboratoire LIP, France
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL), France
[3] Nanyang Technological University, SPMS, Singapore
[4] Inria, France
[5] Bitdefender, Bucharest, Romania

Abstract. The Naor-Yung paradigm is a well-known technique that constructs IND-CCA2-secure encryption schemes by means of non-interactive zero-knowledge proofs satisfying a notion of simulation-soundness. Until recently, it was an open problem to instantiate it under the sole Learning-With-Errors (LWE) assumption without relying on random oracles. While the recent results of Canetti *et al.* (STOC'19) and Peikert-Shiehian (Crypto'19) provide a solution to this problem by applying the Fiat-Shamir transform in the standard model, the resulting constructions are extremely inefficient as they proceed via a reduction to an NP-complete problem. In this paper, we give a direct, non-generic method for instantiating Naor-Yung under the LWE assumption outside the random oracle model. Specifically, we give a direct construction of an unbounded simulation-sound NIZK argument system which, for carefully chosen parameters, makes it possible to express the equality of plaintexts encrypted under different keys in Regev's cryptosystem. We also give a variant of our argument that provides tight security. As an application, we obtain an LWE-based public-key encryption scheme for which we can prove (tight) key-dependent message security under chosen-ciphertext attacks in the standard model.

Keywords. LWE, standard model, Naor-Yung, NIZK arguments, simulation-soundness, KDM-CCA2 security, tight security.

1 Introduction

The Fiat-Shamir transformation [48] is a well-known technique that turns any 3-move honest-verifier zero-knowledge proof system (a.k.a. Σ-protocol [41]) into a non-interactive zero-knowledge proof (NIZK) by replacing the verifier's challenge by a hash value of the transcript so far. Bellare and Rogaway [14] showed that this approach is secure if the underlying hash function is modeled as a random oracle. Since then, the Fiat-Shamir heuristic has been used in the design of countless cryptographic schemes, including digital signatures [89] and chosen-ciphertext-secure public-key encryption schemes [49]. In the standard model, however, counter-examples [56] showed that it may fail to guarantee soundness.

Until recently, it was not known to be securely instantiable without random oracles under any standard assumption. This situation drastically changed with the works of Canetti *et al.* [30] and Peikert and Shiehian [87], which imply the existence of Fiat-Shamir-based NIZK proofs for all NP languages under the sole Learning-With-Errors (LWE) assumption [90]. Their results followed a line of research [93,31,29] showing that Fiat-Shamir can provide soundness in the standard model if the underlying hash function is *correlation intractable* (CI). In short, correlation intractability for a relation R captures the infeasibility of finding an x such that $(x, H_k(x)) \in R$ given a random hashing key k. Intuitively, the reason why this property provides soundness is that a cheating prover's first message cannot be hashed into a verifier message admitting an accepting transcript, except with negligible probability.

While [30,87] resolve the challenging problem of realizing NIZK proofs for all NP under standard lattice assumptions, they leave open the question of building more efficient instantiations of Fiat-Shamir for specific languages, such as those arising in the context of chosen-ciphertext security [86,91,49].

In order to instantiate the Naor-Yung paradigm of CCA2-secure encryption [86] in the lattice setting, the only known solution is to proceed via a general NP reduction to graph Hamiltonicity and apply the Σ-protocol of Feige, Lapidot and Shamir [47] with the modifications suggested by Canetti *et al.* [30,34]. In addition, a direct application of [30,34,87] to CCA2 security requires to apply the generic compiler of [44] that turns any NIZK proof system into simulation-sound [91] proofs. Here, we consider the problem of more efficiently instantiating Naor-Yung in the standard model under lattice assumptions. Using correlation intractable hash functions, our goal is to directly construct simulation-sound arguments of plaintext equality without using generic techniques.

1.1 Our Contributions

We describe the most efficient post-quantum realization of the Naor-Yung paradigm so far and its first non-trivial instantiation under lattice assumptions. As an application, we obtain the most efficient public-key encryption scheme providing key-dependent message security under chosen-ciphertext attacks (or KDM-CCA2 security for short) under the standard Learning-With-Errors (LWE) assumption [90]. Our scheme is *not* the result of merely combining generic NIZK techniques [91,44] with the results [30,34,87] on NIZK proofs based on correlation intractable hash functions. In particular, we bypass the use of a Karp reduction to the graph Hamiltonicity language [47,30,34]. Instead, as a key building block, we directly build a simulation-sound NIZK proof system showing that two dual Regev ciphertexts [52] are encryptions of the same plaintext.

As a result of independent interest, we also obtain a multi-theorem NIZK argument system without using the Feige-Lapidot-Shamir (FLS) transformation [47]. Recall that the FLS compiler constructs a multi-theorem NIZK proof system for an NP language from a single-theorem NIZK proof system by using the latter to prove OR statements of the form "either element x is in the language OR some CRS component is in the range of a pseudorandom generator". Unlike FLS, our multi-theorem NIZK argument avoids the non-black-box use of a PRG. Another advantage is that it provides multi-theorem statistical NIZK in the common *random* string model while proving soundness under the LWE assumption. In contrast, achieving statistical multi-theorem NIZK by applying FLS to [87,34] requires a common reference string sampled from a non-uniform distribution.

We further show that our argument system provides *unbounded* (as opposed to one-time [91]) simulation-soundness (USS) [44], meaning that the adversary remains unable to prove a false statement, even after having seen simulated arguments for polynomially many (possibly false) statements. This makes our argument system suitable to prove KDM-CCA2 security by applying the Naor-Yung technique to the KDM-CPA system of Applebaum, Cash, Peikert, and Sahai (ACPS) [7], which is known to provide key-dependent message security for affine functions. In addition, we provide a variant of our USS argument that can be proved tightly secure, meaning that the reduction's advantage is not affected by the number of simulated proofs obtained by the adversary. The simulation-soundness property is indeed tightly related to the security of the underlying pseudorandom function. By exploiting a result of Lai *et al.* [75], it can be combined with a tightly secure lattice-based PRF so as to instantiate our scheme with a polynomial modulus.

Our first simulation-sound NIZK argument implies a public-key encryption (PKE) scheme providing KDM-CCA2 security under the LWE assumption with polynomial approximation factors. Our second NIZK argument yields an instantiation that enjoys *tight* KDM-CCA2 security. Until recently, this was only possible under an LWE assumption with large approximation factors for lack of a tightly secure low-depth lattice-based PRF based on an LWE assumption with polynomial inverse-error rate. Lai *et al.* [75] recently showed that many tightly secure LWE-based schemes (e.g., [21,77,22]) can actually be obtained using a PRF outside NC1 without going through Barrington's theorem [12]. Their technique [75] applies to our setting and ensure that any (possibly sequential) PRF with a tight security reduction from LWE with polynomial modulus and inverse-error rate allows instantiating the scheme under a similarly standard assumption.

Recall that KDM security is formalized by an experiment where the adversary obtains N public keys. On polynomially many occasions, it sends encryption queries (i, f), for functions $f \in \mathcal{F}$ belonging to some family, and expects to receive an encryption of $f(SK_1, \ldots, SK_N)$ under PK_i. Security requires the adversary to be unable to distinguish the real encryption oracle from an oracle that always returns an encryption of 0. Our KDM-CCA2 construction supports the same function family (namely, affine functions) as the KDM-CPA system it builds on. However, like previous LWE-based realizations [7,5], it can be bootstrapped using Applebaum's technique [6] so as to retain KDM security for arbitrary functions that are computable in a priori bounded polynomial time.

We believe our LWE-based instantiation of Naor-Yung to be of interest beyond KDM security. For example, it makes possible to publicly recognize ciphertexts that correctly decrypt, which is a rare feature among LWE-based schemes and comes in handy in the threshold decryption setting (see, e.g., [49]). It can also be used to obtain chosen-ciphertext security in settings – such as inner product functional encryption [1,4] or receiver selective-opening security [61] – for which we do not know how to apply the Canetti-Halevi-Katz technique [33].

1.2 Technical Overview

Our starting point is a trapdoor $\varSigma$-protocol [30,34] allowing to prove the well-formed of ciphertexts in the KDM-CPA system of Applebaum *et al.* [7]. Namely, it allows proving that a given vector $\mathbf{c} = (\mathbf{u}, u) \in \mathbb{Z}_q^{n+1}$ is of the form $(\mathbf{u}, \mathbf{u}^\top \mathbf{s} + \mu\lfloor q/p \rfloor + \mathsf{noise})$, where $\mu \in \mathbb{Z}_p$ is the message, $\mathbf{s} \in \mathbb{Z}^n$ is the secret key and the public key is $(\mathbf{A}, \mathbf{b} = \mathbf{A}^\top \mathbf{s} + \mathsf{noise}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$ for some $m = \varOmega(n \cdot \log q)$. Recall that a standard $\varSigma$-protocol [41,40] is a 3-move protocol with transcripts of the form $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ where Chall is the verifier's challenge and messages $\mathbf{a}$ and $\mathbf{z}$ are sent by the prover. In the common reference string model, a trapdoor $\varSigma$-protocol [30,34] has the property that, for any statement x outside the language $\mathcal{L}$ and any first message $\mathbf{a}$ sent by the prover, a trapdoor makes it possible to determine the unique challenge Chall for which a valid response $\mathbf{z}$ exists. There is an efficiently computable function $\mathsf{BadChallenge}$ that takes as input a trapdoor τ, a false statement $x \notin \mathcal{L}$, and a first prover message $\mathbf{a}$, and computes the unique Chall such that there exists an accepting transcript $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ (that is, there is no accepting transcript of the form $(\mathbf{a}, \mathsf{Chall}', \mathbf{z})$ for any $\mathsf{Chall}' \neq \mathsf{Chall}$).

Our first observation is that, in order to preserve the soundness of Fiat-Shamir, it suffices for a trapdoor $\varSigma$-protocol to have a $\mathsf{BadChallenge}$ function that outputs "if there is a bad challenge at all for $\mathbf{a}$, it can only be Chall". Indeed, false positives do not hurt soundness as we only need the CI hash function to sidestep the bad challenge whenever it exists. Based on this observation, we can build a trapdoor $\varSigma$-protocol showing that a Regev ciphertext $\mathbf{c} \in \mathbb{Z}_q^{n+1}$ encrypts 0. Letting $\bar{\mathbf{A}} = [\mathbf{A}^\top \mid \mathbf{b}]^\top \in \mathbb{Z}_q^{(n+1)\times m}$, this can be done using by showing knowledge of a short $\mathbf{r} \in \mathbb{Z}^m$ such that $\mathbf{c} = \bar{\mathbf{A}} \cdot \mathbf{r}$. In $\varSigma$-protocols like [80,81], the verifier accepts transcripts $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ such that $\mathbf{a} + \mathsf{Chall} \cdot \mathbf{c} = \bar{\mathbf{A}} \cdot \mathbf{z}$ if $\mathbf{z} \in \mathbb{Z}^m$ is short enough. Since the right-hand side member of the verification equation is an encryption of 0, the $\mathsf{BadChallenge}$ function can use the decryption key $\mathbf{s}$ to infer that no valid response exists for the challenge $\mathsf{Chall} = b$ when $\mathbf{a} + b \cdot \mathbf{c}$ does not decrypt to 0.

The next step is to argue that $\mathbf{c}$ encrypts an arbitrary $\mu \in \mathbb{Z}_p$. To this end, we exploit the fact the KDM-CPA scheme of [7] uses a square modulus $q = p^2$ when we compute part of the response $z_\mu = r_u + \mathsf{Chall} \cdot \mu \bmod p$ over $\mathbb{Z}_p$, while using a uniform mask $r_u \in \mathbb{Z}_p$ to hide $\mu \in \mathbb{Z}_p$ as in standard Schnorr-like protocols [92]. Now, the $\mathsf{BadChallenge}$ function can output $\mathsf{Chall} = 1 - b$ if it detects that $\mathbf{a} + b \cdot \mathbf{c}$ is not of the form $(\mathbf{u}, \mathbf{u}^\top \mathbf{s} + z_\mu \cdot p + \mathsf{noise})$, for some $z_\mu \in \mathbb{Z}_p$. Indeed, this rules out the existence of a short enough $\mathbf{z} \in \mathbb{Z}^m$ such that $\mathbf{a} + b \cdot \mathbf{c} = \bar{\mathbf{A}} \cdot \mathbf{z} + z_\mu \cdot [\mathbf{0}^{n\top} | p]^\top$ with $z_\mu \in \mathbb{Z}_p$. The above technique extends into a trapdoor $\varSigma$-protocol for proving plaintext equalities in the ACPS cryptosystem [7]. Our instantiation of Naor-Yung thus requires to work with LWE over a composite modulus q and we leave it as an open problem to extend it to prime moduli.

The main difficulty, however, is to turn the aforementioned trapdoor $\varSigma$-protocol into a non-interactive proof system with unbounded simulation-soundness.

This problem is non-trivial since the Canetti *et al.* protocol [30,34] is not known to satisfy this security notion.[6] The NIZK simulator of [30,34] generates simulated proofs by "programming" the CI hash function from which the verifier's challenge is derived. In the context of unbounded simulation-soundness [91,44], we cannot proceed in the same way since the simulator would have to program the hash function for each simulated proof (and thus for each challenge ciphertext in the proof of KDM-CCA2 security). Since the number of simulated proofs is not a priori bounded, it is not clear how to do that using a hashing key of length independent of the number of adversarial queries.

Our solution to this problem is inspired by the modification introduced by Canetti *et al.* [34,30] in the Feige-Lapidot-Shamir protocol [47]. In [34, Section 5.2], the first prover message $\mathbf{a}$ is computed using a lossy encryption scheme [13] instead of an ordinary commitment. Recall that, depending on the distribution of the public key PK, a lossy encryption scheme behaves either as an extractable non-interactive commitment or a statistically-hiding commitment. The extractable mode is used to prove the soundness property (by using the secret key SK corresponding to PK to compute the $\mathsf{BadChallenge}$ function) while the statistically hiding mode allows proving zero-knowledge. Our unbounded simulation-sound proof system exploits the observation made by Bellare *et al.* [13] that specific lossy encryption schemes admit an efficient opening algorithm. Namely, ciphertexts encrypted under a lossy public key can be equivocated in the same way as a trapdoor commitment using the lossy secret key SK. This suggests that, if the protocol of Canetti *et al.* [34,30] is instantiated using a lossy encryption scheme with efficient opening, we can use a strategy introduced by Damgård [43] to simulate NIZK proofs without programming the CI hash function. Namely, the simulator can generate the first prover message as a lossy encryption of 0. When receiving the verifier's challenge Chall, it can run the HVZK simulator to obtain $(\mathbf{a}, \mathbf{z})$ before using the lossy secret key SK to explain the lossy ciphertext as an encryption of the simulated $\mathbf{a}$. By doing this, we also obtain a multi-theorem NIZK argument without using the FLS transformation [47] and without using any primitive in a non-black-box way. The language of the underlying trapdoor Σ-protocol is exactly the same as that of the multi-theorem NIZK argument, so that, if the former is efficient, so is the latter.

However, standard lossy encryption schemes with efficient opening do not suffice to prove unbounded simulation-soundness: We do not only need to equivocate lossy ciphertexts in all simulated proofs, but we should also make sure that the adversary's fake proof is generated for a statistically binding (and even extractable) commitment. For this reason, we rely on a lossy encryption flavor, called $\mathcal{R}$-lossy encryption by Boyle *et al.* [23], where a tag determines whether a ciphertext is lossy or injective. The public key is generated for a computationally hidden initialization value $K \in \mathcal{K}$ and ciphertexts are encrypted under a tag $t \in \mathcal{T}$. If $\mathcal{R} \subset \mathcal{K} \times \mathcal{T}$ is a binary relation, the syntax of $\mathcal{R}$-lossy encryption [23] is

[6] It can be generically achieved using NIZK for general NP relations [44] but our goal is to obtain a more efficient solution than generic NIZK techniques. In fact, even one-time simulation-soundness is not proven in [30,34]

that a ciphertext encrypted for a tag $t \in \mathcal{T}$ is injective if $\mathcal{R}(K,t) = 1$ and lossy otherwise. For our purposes, we need to enrich the syntax of $\mathcal{R}$-lossy encryption in two aspects. First, we require lossy ciphertexts to be efficiently equivocable (i.e., the secret key SK should make it possible to find random coins that explain a lossy ciphertext as an encryption of any target plaintext). Second, in order to simplify the description of our NIZK simulator, we need the syntax to support lossy/injective tags *and* lossy/injective keys. When the public key PK is lossy, all ciphertexts are lossy, no matter which tag is used to encrypt. In contrast, injective public keys lead to injective ciphertexts whenever $\mathcal{R}(K,t) = 1$. Our NIZK simulator actually uses lossy public keys while injective keys only show up in the proof of simulation-soundness.

We then construct an $\mathcal{R}$-lossy encryption scheme for the bit-matching relation (i.e., $\mathcal{R}_{\mathsf{BM}}(K,t) = 1$ if and only if K and t agree in all positions where K does not contain a "don't care entry") under the LWE assumption. The scheme can be viewed as a combination of the primal Regev cryptosystem [90] – which is known [88] to be a lossy PKE scheme and is easily seen to support efficient openings as defined in [13] – with the lattice trapdoors of Micciancio and Peikert [85]. An injective public key consists of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ with short vectors in its row space. In order to encrypt $\boldsymbol{\mu} \in \{0,1\}^{n_0}$ under a tag t, we sample a short Gaussian $\mathbf{r} \in \mathbb{Z}^{2m}$ and compute $\mathbf{c} = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_t + (1 - \mathcal{R}(K,t)) \cdot \mathbf{G}] \cdot \mathbf{r} + [\mathbf{0} \mid \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor]^\top$, for some small-norm $\mathbf{R}_t \in \mathbb{Z}^{m \times m}$, where $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ is the gadget matrix of [85]. In each lossy tag, we have $\mathcal{R}(K,t) = 0$, in which case the matrix $\mathbf{R}_t$ can be used as a trapdoor (using the techniques of [3,85]) to sample a Gaussian $\mathbf{r} \in \mathbb{Z}^{2m}$ that explains $\mathbf{c}$ as an encryption of any arbitrary $\boldsymbol{\mu} \in \{0,1\}^{n_0}$. In injective tags, we have $\mathcal{R}(K,t) = 1$, so that the gadget matrix vanishes from the matrix $\mathbf{A}_t = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_t + (1 - \mathcal{R}(K,t)) \cdot \mathbf{G}]$. Since $\mathbf{A}$ has short vectors in its row space, so does $\mathbf{A}_t$ and we can thus use these short vectors to recover $\boldsymbol{\mu}$ from $\mathbf{c}$ exactly as in the primal Regev cryptosystem. When the public key PK is lossy, the matrix $\mathbf{A}$ is replaced by a statistically uniform matrix over $\mathbb{Z}_q^{n \times m}$. We can then use a trapdoor for $\varLambda^\perp(\mathbf{A})$ to equivocate lossy ciphertexts for any arbitrary tag.

Our USS argument system uses our $\mathcal{R}$-lossy encryption scheme – with the standard trick of using the verification key of a one-time signature as a tag – to compute the first prover message $\mathbf{a}$ by encrypting the first message $\mathbf{a}'$ of a basic trapdoor $\varSigma$-protocol. In the security proof, we have a noticeable probability that: (i) For all adversarially-chosen statements, proofs can be simulated by equivocating lossy ciphertexts; (ii) When the adversary comes up with a proof of its own, the underlying commitment is an injective ciphertext. If these conditions are fulfilled, we can annihilate the adversary's chance of proving a false statement by using a hash function which is statistically CI for the relation that evaluates the $\mathsf{BadChallenge}$ function on input of the decryption of an $\mathcal{R}$-lossy ciphertext.

At a high-level, our simulation-sound proof system bears similarities with interactive proof systems described by MacKenzie and Yang [83]. Our extension of $\mathcal{R}$-lossy encryption resembles their notion of simulation-sound trapdoor commitments. The difference is that, while [83] only requires commitments to be computationally binding for tags that have never been equivocated, we need adversarially-chosen tags to be extractable.

Our first USS argument system does not provide tight security because it relies on admissible hash functions [17] to partition the tag space of the $\mathcal{R}$-lossy PKE scheme into two disjoint subspaces (which contain equivocable and extractable tags, respectively). In order to obtain tight simulation-soundness, our second USS argument partitions the tag space of an $\mathcal{R}$-lossy PKE scheme using a pseudorandom function instead of an admissible hash function. For this purpose, we build an $\mathcal{R}$-lossy PKE scheme for a relation $\mathcal{R}_{\mathsf{PRF}}$ induced by a PRF family. Analogously to [63], we consider tags $t = (t_c, t_a)$ consisting of an auxiliary component t_a (which can be an arbitrary string) and core component t_c. The PRF-induced relation $\mathcal{R}_{\mathsf{PRF}}$ is then defined as $\mathcal{R}_{\mathsf{PRF}}(K, (t_c, t_a)) = 1$ if and only if $t_c \neq \mathsf{PRF}_K(t_a)$, where K is the PRF secret key. Our $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE then proceeds as in [77] and uses a public key containing Gentry-Sahai-Waters encryptions [53] $\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i + k_i \cdot \mathbf{G}$ of the bits of K. To encrypt $\boldsymbol{\mu} \in \{0,1\}^{n_0}$ under a tag $t = (t_c, t_a)$, the encryptor first homomorphically computes $\mathbf{A}_{F,t} = \mathbf{A} \cdot \mathbf{R}_t + (1 - \mathcal{R}_{\mathsf{PRF}}(K, t)) \cdot \mathbf{G}$ before sampling a short Gaussian $\mathbf{r} \in \mathbb{Z}^{2m}$ and computing $\mathbf{c} = [\mathbf{A} \mid \mathbf{A}_{F,t}] \cdot \mathbf{r} + [\mathbf{0} \mid \boldsymbol{\mu} \cdot \lfloor q/2 \rfloor]^{\top}$. In the proof of simulation-soundness, the reduction simulates all arguments by "adaptively programming" all tags $t = (\mathsf{PRF}_K(t_a), t_a)$ to ensure equivocability. At the same time, the adversary can only output an argument on an extractable tag $t^\star = (t_c^\star, t_a^\star)$, where $\mathcal{R}_{\mathsf{PRF}}(K, t^\star) = 1$, unless it can predict $t_c^\star = \mathsf{PRF}_K(t_a^\star)$.

1.3 Related Work

Fiat-Shamir in the Standard Model. The Fiat-Shamir methodology was shown [56] not to be sound in the standard model in general. Known negative results (see [56,15] and references therein) nevertheless left open the existence of secure instantiations of the paradigm when specific protocols are transformed using concrete hash functions. Of particular interest is the notion of *correlation intractable* hash function [32], which rules out specific relations between an input and its hash value. It was actually shown [59] that correlation intractability for all sparse relations[7] suffices to ensure soundness as long as the underlying protocol is statistically sound. A recent line of work [93,31,66,29] focused on the design of correlation intractable hash functions leading to sound instantiation of Fiat–Shamir in the standard model. Canetti *et al.* [30] showed that it is actually sufficient to obtain correlation intractable hash families for *efficiently searchable* relations (i.e., where each x has at most one corresponding y, which is computable within some polynomial time bound). This opened the way to CI hash candidates based on more established assumptions like the circular security of fully homomorphic encryption (FHE) schemes [34]. Peikert and Shiehian [87] recently gave an elegant FHE-based solution relying on the hardness of the LWE problem [90] with polynomial approximation factors. While specific to the Gentry-Sahai-Waters (GSW) FHE [53], their construction does not require any non-standard circular security assumption. Together with the techniques of

[7] A relation $R \subset \mathcal{X} \times \mathcal{Y}$ is sparse if, for a given $x \in \mathcal{X}$, the fraction of $y \in \mathcal{Y}$ for which $(x, y) \in R$ is negligible.

[34,30], it implies NIZK for all NP languages.

In [34,30], Canetti *et al.* showed that, besides the language of Hamiltonian graphs considered in [47], trapdoor Σ-protocols also exist for other languages like that of quadratic residues modulo a composite integer [55]. Using the CI hash function of [87], they thus obtained a NIZK proof for the Quadratic Residuosity language under the LWE assumption. Choudhuri *et al.* [37] showed that the hash families of [30] make the transformation sound for the sumcheck protocol.

Multi-Theorem NIZK. Several multi-theorem NIZK constructions are available in the literature (see, e.g., [47,45,36,57]). Under the LWE assumption, all solutions so far either rely on the FLS transformation [39,87] – thus incurring proofs of OR statements via non-black-box techniques – or restrict themselves to the designated verifier setting [39,78]. While the meta-proof approach of De Santis and Yung [45] provides an alternative to FLS, it makes non-black-box use of a single-theorem proof system for an NP-complete language. Our construction uses a single-theorem argument for the same language as the one for which we need a multi-theorem argument. Hence, if the former is efficient, so is the latter.

KDM Security. This security notion was first formalized by Black, Rogaway and Shrimpton [16] and motivated by applications in anonymous credentials [27] or in disk encryption (e.g., in the BitLocker encryption utility [19]), where the key may be stored on the disk being encrypted. The first examples of KDM-secure secret-key encryption were given by Black *et al.* [16] in the random oracle model.

In the standard model, Boneh *et al.* [19] designed the first public-key scheme with provable KDM-CPA security w.r.t. all affine functions under the decisional Diffie-Hellman (DDH) assumption. Applebaum *et al.* [7] showed that a variant of Regev's system [90] is KDM-secure for all affine functions under the LWE assumption. They also gave a secret-key construction based on the hardness of the Learning Parity with Noise (LPN) problem for which Döttling gave a public-key variant [46]. Under the Quadratic Residuosity (QR) and Decisional Composite Residuosity (DCR) assumptions, Brakerski and Goldwasser [24] gave alternative constructions that additionally provide security under key leakage. Alperin-Sheriff and Peikert [5] showed that a variant of the identity-based encryption scheme of Agrawal *et al.* [3] provides KDM security for a bounded number of challenge ciphertexts.

Brakerski *et al.* [25] and Barak *et al.* [11] came up with different techniques to prove KDM security for richer function families. Malkin *et al.* [84] suggested a much more efficient scheme with ciphertexts of $O(d)$ group elements for function families containing degree d polynomials. Applebaum [6] put forth a generic technique that turns any PKE scheme with KDM security for projection functions – where each output bit only depends on a single input bit – into a scheme providing KDM security for any circuit of a priori bounded polynomial size.

KDM-CCA Security. The first PKE scheme with KDM-CCA2 security in the standard model appeared in the work of Camenisch, Chandran, and Shoup [28]. They gave a generic construction based on the Naor-Yung paradigm that combines a KDM-CPA system, a standard CPA-secure encryption scheme, and a simulation-sound NIZK proof system. For their purposes, they crucially need *unbounded* simulation-soundness since the KDM setting inherently involves many challenge ciphertexts and single-challenge security is not known to imply multi-challenge security. They instantiated their construction using the DDH-based KDM-CPA system of Boneh *et al* [19] and Groth-Sahai proofs [58]. Our scheme is an instantiation of the generic construction of [28] in the lattice setting, where we cannot simply use Groth-Sahai proofs. Hofheinz [64] subsequently obtained chosen-ciphertext circular security (i.e., for selection functions where $f(SK_1, \ldots, SK_N) = SK_i$ for some $i \in [N]$) with shorter ciphertexts.

A first attempt of KDM-CCA security without pairings was made by Lu *et al.* [79]. Han *et al.* [60] identified a bug in [79] and gave a patch using the same methodology. They obtained KDM-CCA security for bounded-degree polynomial functions under the DDH and DCR assumptions. Kitigawa and Tanaka [74] described a framework for the design of KDM-CCA systems under a single number theoretic assumption (i.e., DDH, QR, or DCR). Their results were extended by Kitigawa *et al.* [73] so as to prove tight KDM-CCA2 security under the DCR assumption. Since the framework of [74] relies on hash proof systems [42], it is not known to provide LWE-based realizations (indeed, hash proof systems do not readily enable chosen-ciphertext security from LWE), let alone with tight security. To our knowledge, our scheme is thus the first explicit solution with tight KDM-CCA2 security under the LWE assumption. Before [73], the only pathway to tight KDM-CCA security was to instantiate the construction of Camenisch *et al.* [28] using a tightly secure USS proof/argument (e.g., [65]), which tends to incur very large ciphertexts. Our system also follows this approach with the difference that ciphertexts are not much longer than in its non-tight variant.

Kitigawa and Matsuda [72] generically obtained KDM-CCA security for bounded-size circuits from any system providing KDM-CPA security for projection functions. While their result shows the equivalence between KDM-CPA and KDM-CCA security, our scheme is conceptually simpler and significantly more efficient than an LWE-based instantiation of the construction in [72]. In particular, such an instantiation requires both garbling schemes and $\Omega(\lambda)$ designated-verifier proofs of plaintext equalities with negligible soundness error. While these proofs seem realizable by applying the techniques of [78] to specific Σ-protocols, each of them would cost $\Omega(\lambda^2)$ public-key encryptions. Our scheme is much simpler and only requires one argument of plaintext equality, thus compressing ciphertexts by a factor at least $\Omega(\lambda)$.

1.4 Organization

Section 2 first recalls the the building blocks of our constructions. Our first simulation-sound argument is presented in Section 3 together with the underlying $\mathcal{R}$-lossy PKE scheme. Its tightly secure variant is described in Section 4. In Section 5, we give a trapdoor Σ-protocol allowing to apply the Naor-Yung transformation to the ACPS cryptosystem. The resulting (tightly secure) KDM-CCA2 system is then detailed in Section F of the Supplementary Material. As written, our security proof only shows tightness in the number of challenge ciphertexts, but not in the number of users. In Section F of the Supplementary Material, we explain how to also obtain tightness w.r.t. the number of users.

2 Background

We recall the main tools involved in our constructions. Additional standard tools, such as NIZK proofs, are defined in Section A of the Supplementary Material.

2.1 Lattices

For any $q \geq 2$, $\mathbb{Z}_q$ denotes the ring of integers with addition and multiplication modulo q. If $\mathbf{x} \in \mathbb{R}^n$ is a vector, $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2}$ denotes its Euclidean norm and $\|\mathbf{x}\|_\infty = \max_i |x_i|$ its infinity norm. If $\mathbf{M}$ is a matrix over $\mathbb{R}$, then $\|\mathbf{M}\| := \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{M}\mathbf{x}\|}{\|\mathbf{x}\|}$ and $\|\mathbf{M}\|_\infty := \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{M}\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty}$ denote its induced norms. 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.

Let $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ be a symmetric positive-definite matrix, and $\mathbf{c} \in \mathbb{R}^n$. We define the Gaussian function on $\mathbb{R}^n$ by $\rho_{\mathbf{\Sigma},\mathbf{c}}(\mathbf{x}) = \exp(-\pi(\mathbf{x} - \mathbf{c})^\top \mathbf{\Sigma}^{-1}(\mathbf{x} - \mathbf{c}))$ and if $\mathbf{\Sigma} = \sigma^2 \cdot \mathbf{I}_n$ and $\mathbf{c} = \mathbf{0}$ we denote it by ρ_σ. For an n dimensional lattice $\Lambda \subset \mathbb{R}^n$ and for any lattice vector $\mathbf{x} \in \Lambda$ the discrete Gaussian is defined by $\rho_{\Lambda,\mathbf{\Sigma},\mathbf{c}}(\mathbf{x}) = \frac{\rho_{\mathbf{\Sigma},\mathbf{c}}}{\rho_{\mathbf{\Sigma},\mathbf{c}}(\Lambda)}$.

For an n-dimensional lattice Λ, we define $\eta_\varepsilon(\Lambda)$ as the smallest $r > 0$ such that $\rho_{1/r}(\widehat{\Lambda} \setminus \mathbf{0}) \leq \varepsilon$ with $\widehat{\Lambda}$ denoting the dual of Λ, for any $\varepsilon \in (0, 1)$.

For a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$, we define $\Lambda^\perp(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{0} \bmod q\}$ and $\Lambda(\mathbf{A}) = \mathbf{A}^\top \cdot \mathbb{Z}^n + q\mathbb{Z}^m$. For an arbitrary vector $\mathbf{u} \in \mathbb{Z}_q^n$, we also define the shifted lattice $\Lambda^{\mathbf{u}}(\mathbf{A}) = \{\mathbf{x} \in \mathbb{Z}^m : \mathbf{A} \cdot \mathbf{x} = \mathbf{u} \bmod q\}$.

Definition 2.1 (LWE)**.** *Let $m \geq n \geq 1$, $q \geq 2$ and $\alpha \in (0, 1)$ be functions of a security parameter λ. 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 $\mathbf{e} \sim D_{\mathbb{Z}^m,\alpha q}$. For an algorithm $\mathcal{A} : \mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m \to \{0, 1\}$, we define:*

$$\mathbf{Adv}^{\mathsf{LWE}}_{q,m,n,\alpha}(\lambda) = |\Pr[\mathcal{A}(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}) = 1| \quad ,$$

where the probabilities are over $\mathbf{A} \sim U(\mathbb{Z}_q^{m\times n})$, $\mathbf{s} \sim U(\mathbb{Z}_q^n)$, $\mathbf{u} \sim U(\mathbb{Z}_q^m)$ and $\mathbf{e} \sim D_{\mathbb{Z}^m,\alpha q}$ and the internal randomness of $\mathcal{A}$. We say that $\mathsf{LWE}_{q,m,n,\alpha}$ is hard if, for any PPT algorithm $\mathcal{A}$, the advantage $\mathbf{Adv}^{\mathsf{LWE}}_{q,m,n,\alpha}(\mathcal{A})$ is negligible.

Micciancio and Peikert [85] described a trapdoor mechanism for LWE. Their technique uses a "gadget" matrix $\mathbf{G} \in \mathbb{Z}_q^{n\times w}$, with $w = n \log q$, for which anyone can publicly sample short vectors $\mathbf{x} \in \mathbb{Z}^w$ such that $\mathbf{G} \cdot \mathbf{x} = \mathbf{0}$.

Lemma 2.2 (**[85, Section 5]**)**.** *Assume that $\bar{m} \geq n \log q + O(\lambda)$ and $m = \bar{m} + n\lceil \log q\rceil$. There exists a probabilistic polynomial time (PPT) algorithm* GenTrap *that takes as inputs matrices $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n\times \bar{m}}$, $\mathbf{H} \in \mathbb{Z}_q^{n\times n}$ and outputs* *matrices* $\mathbf{R} \in \{-1,1\}^{\bar{m} \times n \cdot \lceil \log q \rceil}$ *and* $\mathbf{A} = [\bar{\mathbf{A}} \mid \bar{\mathbf{A}}\mathbf{R} + \mathbf{H} \cdot \mathbf{G}] \in \mathbb{Z}_q^{n \times m}$ *such that if* $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$ *is invertible, then* $\mathbf{R}$ *is a* $\mathbf{G}$*-trapdoor for* $\mathbf{A}$ *with tag* $\mathbf{H}$*; and if* $\mathbf{H} = \mathbf{0}$*, then* $\mathbf{R}$ *is a punctured trapdoor.*

Further, in case of a $\mathbf{G}$*-trapdoor, one can efficiently compute from* $\mathbf{A}, \mathbf{R}$ *and* $\mathbf{H}$ *a basis* $(\mathbf{t}_i)_{i \leq m}$ *of* $\Lambda^{\perp}(\mathbf{A})$ *such that* $\max_i \|\mathbf{t}_i\| \leq O(m^{3/2})$*.*

Lemma 2.3 ([52, Theorem 4.1]). *There is a PPT algorithm that, given a basis* $\mathbf{B}$ *of an* n*-dimensional* $\Lambda = \Lambda(\mathbf{B})$*, a parameter* $s > \|\tilde{\mathbf{B}}\| \cdot \omega(\sqrt{\log n})$*, and a center* $\mathbf{c} \in \mathbb{R}^n$*, outputs a sample from a distribution statistically close to* $D_{\Lambda,s,\mathbf{c}}$*.*

2.2 Correlation Intractable Hash Functions

We consider unique-output searchable binary relations [30]. These are binary relations such that, for every x, there is at most one y such that $R(x, y) = 1$ and y is efficiently computable from x.

Definition 2.4. *A relation* $R \subseteq \mathcal{X} \times \mathcal{Y}$ *is* **searchable** *in time* T *if there exists a function* $f : \mathcal{X} \to \mathcal{Y}$ *which is computable in time* T *and such that, if there exists* y *such that* $(x, y) \in R$*, then* $f(x) = y$*.*

Letting $\lambda \in \mathbb{N}$ denote a security parameter, a hash family with input length $n(\lambda)$ and output length $m(\lambda)$ is a collection $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ of keyed hash functions implemented 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 a hash value $h_\lambda(k, x) \in \{0,1\}^{m(\lambda)}$.

Definition 2.5. *For a relation ensemble* $\{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$*, a hash function family* $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ *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 [87] described a correlation-intractable hash family for any searchable relation (in the sense of Definition 2.4) defined by functions f of bounded depth. Their construction relies on the standard Short Integer Solution assumption (which is implied by LWE) with polynomial approximation factors.

2.3 Admissible Hash Functions

Admissible hash functions were introduced in [17] as a combinatorial tool for partitioning-based security proofs. A simplified definition was given in [50].

Definition 2.6 ([17,50]). *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*

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

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 2.7 ([68, 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* $c \cdot L$ *for some constant* $c \in (0, 1/2)$*. Then,* $(C_\ell)_{\ell \in \mathbb{N}}$ *is a family of admissible hash functions. Furthermore,* $\mathsf{AdmSmp}(1^\lambda, Q, \delta)$ *outputs a key* $K \in \{0,1,\perp\}^L$ *for which* $\eta = O(\log \lambda)$ *components are not* $\perp$ *and* $\delta(Q(\lambda))$ *is a non-negligible function of* λ*.*

Jager proved [68] Theorem 2.7 for *balanced* admissible hash functions, which provide both a lower bound and a close upper bound for the probability in Definition 2.6. Here, we only need the standard definition of admissible hash functions since we use them in a game where the adversary aims at outputting a hard-to-compute result (instead of breaking an indistinguishability property). However, the result of Theorem 2.7 applies to standard admissible hash functions.

2.4 Trapdoor Σ-protocols

Canetti *et al.* [34] considered a definition of Σ-protocols that slightly differs from the usual formulation [41,40].

Definition 2.8 (Adapted from [34,9]). *Let a language* $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$ *associated with two NP relations* $R_{\mathsf{zk}}, R_{\mathsf{sound}}$*. 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 Gap* Σ*-protocol for* $\mathcal{L}$ *if it satisfies the following conditions:*

- **3-Move Form:** *The prover and the verifier both take as 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 *and proceed as follows: (i)* P *takes in* $w \in R_{\mathsf{zk}}(x)$*, computes* $(\mathbf{a}, st) \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$ *and sends* $\mathbf{a}$ *to the verifier; (ii)* V *sends back a random challenge* Chall *from the challenge space* $\mathcal{C}$*; (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* $(x, w) \in R_{\mathsf{zk}}$ *and* P *honestly computes* $(\mathbf{a}, \mathbf{z})$ *for a challenge* Chall*,* $\mathsf{V}(\mathsf{crs}, x, (\mathbf{a}, \mathsf{Chall}, \mathbf{z}))$ *outputs* 1 *with probability* $1 - \mathsf{negl}(\lambda)$*.*
- **Special zero-knowledge:** *There is a PPT simulator* ZKSim *that, on input of* crs*,* $x \in \mathcal{L}_{\mathsf{zk}}$ *and a challenge* $\mathsf{Chall} \in \mathcal{C}$*, outputs* $(\mathbf{a}, \mathbf{z}) \leftarrow \mathsf{ZKSim}(\mathsf{crs}, x, \mathsf{Chall})$ *such that* $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *is computationally indistinguishable from a real transcript with challenge* Chall *(for* $w \in R_{zk}(x)$*).*
- **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* $x \notin \mathcal{L}_{\mathsf{sound}}$*, and any first message* $\mathbf{a}$ *sent by* P*, there is at most one challenge* $\mathsf{Chall} = f(\mathsf{crs}, x, \mathbf{a})$ *for which an*

accepting transcript $(\mathsf{crs}, x, \mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *exists for some third message* $\mathbf{z}$*. The function* f *is called the "bad challenge function" of* Π*. That is, if* $x \notin \mathcal{L}_{\mathsf{sound}}$ *and the challenge differs from the bad challenge, the verifier never accepts.*

Definition 2.8 is taken from [34,9] and relaxes the standard special soundness property in that extractability is not required. Instead, it considers a bad challenge function f, which may not be efficiently computable. Canetti *et al.* [34] 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 $x \notin \mathcal{L}_{\mathsf{sound}}$. Here, we use a definition where x need not be known in advance (which is not possible in applications to chosen-ciphertext security, where x is determined by a decryption query) and the trapdoor does not depend on a specific x. However, the common reference string and the trapdoor may depend on the language (which is determined by the public key in our application).

The common reference string $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$ 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 parameter $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$.

Definition 2.9 (Adapted from [34]). *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* $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$ *is a* **trapdoor** Σ**-protocol** *if it satisfies the properties of Definition 2.8 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* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$.
- $\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$ *takes as input public parameters* par *and a membership-testing trapdoor* $\tau_{\mathcal{L}}$ *for the language* $\mathcal{L}_{\mathsf{sound}}$*. It outputs a common reference string* $\mathsf{crs}_{\mathcal{L}}$ *and a trapdoor* $\tau_\Sigma \in \{0,1\}^{\ell_\tau}$*, for some* $\ell_\tau(\lambda)$.
- $\mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}, x, \mathbf{a})$ *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 challenge* Chall.

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* $x \notin \mathcal{L}_{\mathsf{sound}}$ *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 $\mathcal{L}_{\mathsf{sound}}$.

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

We generalize the notion of $\mathcal{R}$-lossy public-key encryption introduced by Boyle *et al.* [23]. As defined in [23], it is a tag-based encryption scheme [70] where the tag space $\mathcal{T}$ is partitioned into a set of *injective* tags and a set of *lossy* tags. When ciphertexts are generated for an injective tag, the decryption algorithm correctly recovers the underlying plaintext. When messages are encrypted under lossy tags, the ciphertext is statistically independent of the plaintext. 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 takes as input an initialization value $K \in \mathcal{K}$ and partitions $\mathcal{T}$ in such a way that injective tags $t \in \mathcal{T}$ are exactly those for which $(K, t) \in \mathcal{R}$ (i.e., all tags t for which $(K, t) \not\in \mathcal{R}$ are lossy).

From a security standpoint, the definitions of [23] require the initialization value K to be computationally hidden by the public key. For our purposes, we need to introduce additional requirements.

First, we require the existence of a lossy key generation algorithm $\mathsf{LKeygen}$ which outputs public keys with respect to which all tags t are lossy (in contrast with injective keys where the only lossy tags are those for which $(K, t) \not\in \mathcal{R}$). Second, we also ask that the secret key makes it possible to equivocate lossy ciphertexts (a property called *efficient opening* by Bellare *et al.* [13]) using an algorithm called Opener. Finally, we use two distinct opening algorithms Opener and $\mathsf{LOpener}$. The former operates over (lossy and injective) public keys for lossy tags while the latter can equivocate ciphertexts encrypted under lossy keys for any tag.

Definition 2.10. *Let $\mathcal{R} \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ be an efficiently computable binary relation. An $\mathcal{R}$-lossy PKE scheme with efficient opening 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: *On input a security parameter λ, $\mathsf{Par\text{-}Gen}(1^\lambda)$ outputs public parameters Γ.*

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

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

Decryption under injective tags: *For any initialization value $K \in \mathcal{K}$, any tag $t \in \mathcal{T}$ such that $(K, t) \in \mathcal{R}$, and any message $\mathsf{Msg} \in \mathsf{MsgSp}$, we have*

$$\Pr\left[\exists r \in R^{\mathsf{LPKE}} : \mathsf{Decrypt}\big(sk, t, \mathsf{Encrypt}(pk, t, \mathsf{Msg}; r)\big) \neq \mathsf{Msg}\right] < \nu(\lambda) \ ,$$

for some negligible function $\nu(\lambda)$, where $(pk, sk, tk) \leftarrow \mathsf{Keygen}(\Gamma, K)$ and the probability is taken over the randomness of Keygen.

Indistinguishability: *Algorithms $\mathsf{LKeygen}$ and Keygen satisfy the following:*

(i) For any $K \in \mathcal{K}_\lambda$, the distributions $D_{\mathrm{inj}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{Keygen}(\varGamma, K)\}$ and $D_{\mathrm{loss}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, K)\}$ are computationally indistinguishable. Namely, for any PPT adversary $\mathcal{A}$, we have $\mathbf{Adv}_{\mathcal{A}}^{\mathsf{indist\text{-}LPKE\text{-}1}}(\lambda) \leq \mathsf{negl}(\lambda)$, where

$$\mathbf{Adv}_{\mathcal{A}}^{\mathsf{indist\text{-}LPKE\text{-}1}}(\lambda) := |\Pr[(pk, tk) \hookleftarrow D_{\mathrm{inj}} : \mathcal{A}(pk, tk) = 1] - \Pr[(pk, tk) \hookleftarrow D_{\mathrm{loss}} : \mathcal{A}(pk, tk) = 1]| \ .$$

(ii) For any distinct initialization values $K, K' \in \mathcal{K}_\lambda$, the two distributions $\{pk \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, K)\}$ and $\{pk \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, K')\}$ are statistically indistinguishable. We require them to be $2^{-\Omega(\lambda)}$-close in terms of statistical distance.

Lossiness: *For any initialization value $K \in \mathcal{K}_\lambda$ and tag $t \in \mathcal{T}_\lambda$ such that $(K, t) \notin \mathcal{R}$, any $(pk, sk, tk) \leftarrow \mathsf{Keygen}(\varGamma, K)$, and any $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$, the following distributions are statistically close:*

$$\{C \mid C \leftarrow \mathsf{Encrypt}(pk, t, \mathsf{Msg}_0)\} \quad \approx_s \quad \{C \mid C \leftarrow \mathsf{Encrypt}(pk, t, \mathsf{Msg}_1)\}.$$

For any $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, K)$, the above holds for any tag t (and not only those for which $(K, t) \notin \mathcal{R}$).

Efficient opening under lossy tags: *Let D_R denote the distribution, defined over the randomness space R^{LPKE}, from which the random coins used by* $\mathsf{Encrypt}$ *are sampled. For any message $\mathsf{Msg} \in \mathsf{MsgSp}$ and ciphertext C, let $D_{PK,\mathsf{Msg},C,t}$ denote the probability distribution on R^{LPKE} with support*

$$S_{PK,\mathsf{Msg},C,t} = \{\overline{r} \in R^{\mathsf{LPKE}} \mid \mathsf{Encrypt}(pk, t, \mathsf{Msg}, \overline{r}) = C\} \ ,$$

and such that, for each $\overline{r} \in S_{PK,\mathsf{Msg},C,t}$, we have

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

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

Efficient opening under lossy keys: *There exists a PPT sampling algorithm* $\mathsf{LOpener}$ *such that, for any $K \in \mathcal{K}_\lambda$, any keys $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, K)$, any random coins $r \hookleftarrow D_R$, any tag $t \in \mathcal{T}_\lambda$, and any distinct messages $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$, takes as input $C = \mathsf{Encrypt}(pk, t, \mathsf{Msg}_0, r)$, t and sk. It outputs a sample $\overline{r}$ from a distribution statistically close to $D_{PK,\mathsf{Msg}_1,C,t}$.*

In Definition 2.10, some of the first four properties were defined in [23, Definition 4.1]. The last two properties are a natural extension of the definition of efficient opening introduced by Bellare *et al.* [13]. We note that property of decryption under injective tags does not assume that random coins are honestly sampled, but only that they belong to some pre-defined set R^{LPKE}.

For our applications to simulation-sound proofs, it would be sufficient to have algorithms ($\mathsf{Opener}, \mathsf{LOpener}$) that have access to the initial messages Msg_0 *and* the random coins r_0 of the ciphertext to be equivocated (as was the case in the opening algorithms of [13]). In our LWE-based construction, however, the initial messages and random coins are not needed.

3 Direct Construction of Unbounded Simulation-Sound NIZK Arguments

We provide a method that directly compiles any trapdoor Σ-protocol into an unbounded simulation-sound NIZK argument using an $\mathcal{R}$-lossy encryption scheme for the bit-matching relation $\mathcal{R}_{\mathsf{BM}}$ and a correlation intractable hash function.

Definition 3.1. *Let $\mathcal{K} = \{0,1,\perp\}^L$ and $\mathcal{T} = \{0,1\}^\ell$, for some $\ell, L \in \mathsf{poly}(\lambda)$ such that $\ell < L$. Let F_{ADH} the partitioning function defined by $\mathsf{AHF} : \{0,1\}^\ell \to \{0,1\}^L$ in Definition 2.6. The* **bit-matching relation** *$\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ for AHF is the relation where $\mathcal{R}_{\mathsf{BM}}(K,t) = 1$ if and only if $K = K_1 \dots K_L$ and $t = t_1 \dots t_\ell$ satisfy $F_{\mathsf{ADH}}(K,t) = 0$ (namely, $\bigwedge_{i=1}^{L} (K_i = \perp) \vee (K_i = \mathsf{AHF}(t)_i)$).*

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

We describe an $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme below. Our scheme builds on a variant of the primal Regev cryptosystem [90] suggested in [52].

Let $\mathsf{AHF} : \{0,1\}^\ell \to \{0,1\}^L$ an admissible hash function with key space $\mathcal{K} = \{0,1,\perp\}^L$ and let $\mathcal{R}_{\mathsf{BM}} \subset \mathcal{K} \times \{0,1\}^\ell$ the corresponding bit-matching relation. We construct an $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme in the following way.

$\mathsf{Par\text{-}Gen}(1^\lambda)$: Given a security parameter $\lambda \in \mathbb{N}$, let $n_0 = \mathsf{poly}(\lambda)$ the length of messages. Choose a prime modulus $q = \mathsf{poly}(\lambda)$; dimensions $n = n_0 + \Omega(\lambda)$ and $m = 2n\lceil \log q \rceil + O(\lambda)$. Define the tag space as $\mathcal{T} = \{0,1\}^\ell$ where $\ell = \Theta(\lambda)$. Define the initialization value space $\mathcal{K} = \{0,1,\perp\}^L$ and Gaussian parameters $\sigma = O(m) \cdot L$ and $\alpha \in (0,1)$ such that $m\alpha q \cdot (L+1) \cdot \sigma\sqrt{2m} < q/4$. Define public parameters as $\Gamma = (\ell, L, n_0, q, n, m, \alpha, \sigma)$.

$\mathsf{Keygen}(\Gamma, K)$: On input of public parameters Γ and an initialization value $K \in \{0,1,\perp\}^L$, generate a key pair as follows.

1. Sample random matrices $\bar{\mathbf{B}} \hookleftarrow U(\mathbb{Z}_q^{(n-n_0)\times m})$, $\mathbf{S} \hookleftarrow U(\mathbb{Z}_q^{(n-n_0)\times n_0})$ and a small-norm $\mathbf{E} \hookleftarrow \chi^{m \times n_0}$ to compute

$$\mathbf{A} = \left[\frac{\bar{\mathbf{B}}}{\mathbf{S}^\top \cdot \bar{\mathbf{B}} + \mathbf{E}^\top} \right] \in \mathbb{Z}_q^{n \times m}.$$

2. Parse K as $K_1 \dots K_L \in \{0,1,\perp\}^L$. Letting $\mathbf{G} \in \mathbb{Z}_q^{n \times m}$ denote the gadget matrix, for each $i \in [L]$ and $b \in \{0,1\}$, compute matrices $\mathbf{A}_{i,b} \in \mathbb{Z}_q^{n \times m}$ as

$$\mathbf{A}_{i,b} = \begin{cases} \mathbf{A} \cdot \mathbf{R}_{i,b} + \mathbf{G} & \text{if } \ (K_i \neq \perp) \wedge (b = 1 - K_i) \\ \mathbf{A} \cdot \mathbf{R}_{i,b} & \text{if } \ (K_i = \perp) \vee (b = K_i). \end{cases} \tag{1}$$

where $\mathbf{R}_{i,b} \hookleftarrow U(\{-1,1\}^{m \times m})$ for all $i \in [L]$ and $b \in \{0,1\}$.

Define $R^{\mathsf{LPKE}} = \{\mathbf{r} \in \mathbb{Z}^{2m} \mid \|\mathbf{r}\| \leq \sigma\sqrt{2m}\}$ and output $sk = (K, \mathbf{S})$ as well as

$$pk := \Big(\mathbf{A},\ \{\mathbf{A}_{i,b}\}_{(i,b)\in[L]\times\{0,1\}}\Big), \qquad tk = (K, \{\mathbf{R}_{i,b}\}_{(i,b)\in[L]\times\{0,1\}}).$$

LKeygen(Γ, K)**:** This algorithm proceeds identically to Keygen except that steps 1 and 2 are modified in the following way.

1. Run $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathsf{GenTrap}(1^\lambda, 1^n, 1^m, q)$ so as to obtain a statistically uniform matrix $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$ with a trapdoor for the lattice $\Lambda^\perp(\mathbf{A})$.
2. Define matrices $\{\mathbf{A}_{i,b} \in \mathbb{Z}_q^{n \times m}\}_{(i,b)\in[L]\times\{0,1\}}$ as in (1).

Define R^{LPKE} as in Keygen and output

$$pk := \Big(\mathbf{A},\ \{\mathbf{A}_{i,b}\}_{(i,b)\in[L]\times\{0,1\}}\Big), \quad sk = \mathbf{T_A}, \quad tk = (K, \{\mathbf{R}_{i,b}\}_{(i,b)\in[L]\times\{0,1\}}).$$

Encrypt(pk, t, Msg)**:** To encrypt $\mathsf{Msg} \in \{0,1\}^{n_0}$ for the tag $t = t_1 \dots t_\ell \in \{0,1\}^\ell$, conduct the following steps.

1. Encode the tag t as $t' = t'_1 \dots t'_L = \mathsf{AHF}(t) \in \{0,1\}^L$ and compute $\mathbf{A}_{F,t} = \sum_{i=1}^L \mathbf{A}_{i,t'_i} \in \mathbb{Z}_q^{n \times m}$. Note that $\mathbf{A}_{F,t} = \mathbf{A} \cdot \mathbf{R}_{F,t} + d_t \cdot \mathbf{G}$ for some $\mathbf{R}_{F,t} \in \mathbb{Z}^{m \times m}$ of norm $\|\mathbf{R}_{F,t}\|_\infty \leq L$ and where $d_t \in \{0, \dots, L\}$ is the number of non-$\perp$ entries of K for which $K_i \neq t'_i$.
2. Choose $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and output $\perp$ if $\mathbf{r} \notin R^{\mathsf{LPKE}}$. Otherwise, output

$$\mathbf{c} = [\mathbf{A} \mid \mathbf{A}_{F,t}] \cdot \mathbf{r} + \left[\frac{\mathbf{0}^{n-n_0}}{\mathsf{Msg} \cdot \lfloor q/2 \rfloor}\right] \in \mathbb{Z}_q^n \quad . \tag{2}$$

Decrypt$(sk, t, \mathbf{c})$**:** Given $sk = (K, \mathbf{S})$ and the tag $t \in \{0,1\}^\ell$, compute $t' = t'_1 \dots t'_L = \mathsf{AHF}(t) \in \{0,1\}^L$ and return $\perp$ if $R_{\mathsf{BM}}(K, t') = 0$. Otherwise, compute $\mathbf{w} = [-\mathbf{S}^\top \mid \mathbf{I}_{n_0}] \cdot \mathbf{c} \in \mathbb{Z}_q^{n_0}$. For each $i \in [n_0]$, do the following:

1. If neither $\mathbf{w}[i]$ nor $|\mathbf{w}[i] - \lfloor q/2 \rfloor|$ is close to 0, halt and return $\perp$.
2. Otherwise, set $\mathsf{Msg}[i] \in \{0,1\}$ so as to minimize $|\mathbf{w}[i] - \mathsf{Msg}[i] \cdot \lfloor q/2 \rfloor|$.

Return $\mathsf{Msg} = \mathsf{Msg}[1] \dots \mathsf{Msg}[n_0]$.

Opener$(pk, tk, t, \mathbf{c}, \mathsf{Msg}_1)$**:** Given $tk = (K, \{\mathbf{R}_{i,b}\}_{i,b})$ and $t \in \{0,1\}^\ell$, compute $t' = t'_1 \dots t'_L = \mathsf{AHF}(t) \in \{0,1\}^L$ and return $\perp$ if $R_{\mathsf{BM}}(K, t') = 1$. Otherwise,

1. Compute the small-norm matrix $\mathbf{R}_{F,t} = \sum_{i=1}^L \mathbf{R}_{i,t'_i} \in \mathbb{Z}^{m \times m}$ such that $\mathbf{A}_{F,t} = \mathbf{A} \cdot \mathbf{R}_{F,t} + d_t \cdot \mathbf{G}$ and $\|\mathbf{R}_{F,t}\|_\infty \leq L$ with $d_t \in [L]$.

2. Use $\mathbf{R}_{F,t} \in \mathbb{Z}^{m \times m}$ as a trapdoor for the matrix
$$\bar{\mathbf{A}}_{F,t} = [\mathbf{A} \mid \mathbf{A}_{F,t}] = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_{F,t} + d_t \cdot \mathbf{G}] \ \in \mathbb{Z}_q^{n \times 2m}$$
to sample a Gaussian vector $\bar{\mathbf{r}} \in \mathbb{Z}^{2m}$ such that
$$\bar{\mathbf{A}}_{F,t} \cdot \bar{\mathbf{r}} = \mathbf{c} - \left[\frac{\mathbf{0}^{n-n_0}}{\mathsf{Msg}_1 \cdot \lfloor q/2 \rfloor}\right] \ . \tag{3}$$
Namely, defining $\mathbf{c}_{\mathsf{Msg}_1} = \mathbf{c} - [(\mathbf{0}^{n-n_0})^\top \mid \mathsf{Msg}_1^\top \cdot \lfloor q/2 \rfloor]^\top$, sample and output fake random coins $\bar{\mathbf{r}} \hookleftarrow D_{\Lambda_q^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}),\sigma}$.

$\mathsf{LOpener}(sk, t, \mathbf{c}, \mathsf{Msg}_1)$: Given $sk = \mathbf{T}_{\mathbf{A}}$ and $t \in \{0,1\}^\ell$, use $\mathbf{T}_{\mathbf{A}}$ to derive a trapdoor $\mathbf{T}_{\mathbf{A},t}$ for the lattice $\Lambda_q^\perp(\bar{\mathbf{A}}_{F,t})$ and use $\mathbf{T}_{\mathbf{A},t}$ to sample a Gaussian vector $\bar{\mathbf{r}} \hookleftarrow D_{\Lambda_q^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}),\sigma}$ satisfying (3).

The above construction requires $2L = \Theta(\lambda)$ matrices in the public key but allows for a relatively small modulus $q = \Theta(m^{5/2} n^{1/2} L^2)$. A technique suggested by Yamada [94] can be used to reduce the number of public matrices to $O(\log^2 \lambda)$ at the expense of a larger (but still polynomial) modulus. Since our application to Naor-Yung requires a public key containing a large correlation-intractable hashing key anyway, we chose to minimize the modulus size.

Theorem 3.2 states that the construction has the required properties under the LWE assumption.

Theorem 3.2. *The above construction is an $\mathcal{R}_{\mathsf{BM}}$-lossy public-key encryption scheme with efficient opening under the* LWE *assumption.*

Proof. To prove the statement, we prove that the scheme enables correct decryption with overwhelming probability in injective mode. We also prove the indistinguishability properties using the LWE assumption on one occasion.

Decryption under injective tags. For any initialization value $K \in \mathcal{K}$, any tag $t \in \{0,1\}^\ell$ such that $(K, t) \in \mathcal{R}_{\mathsf{BM}}$, any message $\mathsf{Msg} \in \{0,1\}^{n_0}$, and any encryption $\mathbf{c} \in \mathbb{Z}_q^n$ of Msg under the $\mathsf{pk} = (\mathbf{A}, \{\mathbf{A}_{i,b}\}_{i,b})$ and t, we have:
$$[-\mathbf{S}^\top \mid \mathbf{I}_{n_0}] \cdot \mathbf{c} = \mathbf{E}^\top \cdot [\mathbf{I}_m \mid \mathbf{R}_{F,t}] \cdot \mathbf{r} + \mathsf{Msg} \cdot \lfloor q/2 \rfloor \in \mathbb{Z}_q^{n_0}$$
We show that, for any $\mathbf{r} \in \mathbb{Z}^{2m}$ of norm smaller than $\|\mathbf{r}\|_\infty \leq \|\mathbf{r}\| \leq \sigma\sqrt{2m}$, we have $\|\mathbf{E}^\top [\mathbf{I}_m \mid \mathbf{R}_{F,t}] \cdot \mathbf{r}\|_\infty < q/4$ with overwhelming probability over the randomness of Keygen, so that the decryption algorithm recovers the message. To prove this, notice that our definition of the randomness space R^{LPKE} imposes $\|\mathbf{r}\|_\infty \leq \|\mathbf{r}\| \leq \sigma\sqrt{2m}$. Besides, we also have
$$\|[\mathbf{I}_m \mid \mathbf{R}_{F,t}]\|_\infty \leq 1 + \|\mathbf{R}_{F,t}\|_\infty \leq 1 + L$$
with probability 1 and
$$\|\mathbf{E}^\top\|_\infty = \max_{i \in [n_0]} \sum_{j=1}^m |e_{ij}| \leq \sqrt{m} \cdot \max_{i \in [n_0]} \sqrt{\sum_{j=1}^m e_{ij}^2} \leq m \cdot \alpha q$$

with overwhelming probability when $\mathbf{E}^\top \hookleftarrow D_{\mathbb{Z}^{n_0 \times m}, \alpha q}$. Putting the above altogether, our choice of parameters implies that

$$\|\mathbf{E}^\top\|_\infty \cdot \|[\mathbf{I}_m \mid \mathbf{R}_{F,t}]\|_\infty \cdot \|\mathbf{r}\|_\infty \leq m\alpha q \cdot (L+1) \cdot \sigma\sqrt{2m} < q/4 \ .$$

Indistinguishability. The key generation algorithm $\mathsf{LKeygen}$ and Keygen satisfy the following properties:

(i) The LWE assumption implies that, for any $K \in \mathcal{K}_\lambda$, the distributions $D_{\mathrm{loss}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\mathit{\Gamma}, K)\}$ and $D_{\mathrm{inj}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{Keygen}(\mathit{\Gamma}, K)\}$ are computationally indistinguishable. These distributions only differ in the generation of the matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$. The matrix $\mathbf{A}$ produced by the Keygen algorithm is pseudorandom since, under the $\mathsf{LWE}_{q,m,n-n_0,\alpha}$ assumption, we can replace $\mathbf{S}^\top \bar{\mathbf{B}} + \mathbf{E}^\top$ by a uniform matrix $\mathbf{B} \sim U(\mathbb{Z}_q^{n_0 \times m})$ without the adversary noticing. When using $\mathsf{LKeygen}$, the matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ is statistically uniform by the properties of the $\mathsf{TrapGen}$ algorithm (specifically, Lemma 2.2).

(ii) For any distinct initialization values $K, K' \in \mathcal{K}_\lambda$, the two distributions $\{pk \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\mathit{\Gamma}, K)\}$ and $\{pk \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\mathit{\Gamma}, K')\}$ are statistically indistinguishable since the public matrices $(\mathbf{A}, \{\mathbf{A}_i\})$ are statistically uniform and independent regardless of which K is used as input by $\mathsf{LKeygen}$. Recall the matrix $\mathbf{A}$ produced by the $\mathsf{LKeygen}$ algorithm is statistically close to $U(\mathbb{Z}_q^{n \times m})$ by the properties of the $\mathsf{TrapGen}$. As for the matrices $\mathbf{A}_{i,b} = \mathbf{A} \cdot \mathbf{R}_{i,b} + \kappa_{i,b} \cdot \mathbf{G}$, for $\kappa_{i,b} \in \{0,1\}$, the Leftover Hash Lemma implies that the statistical distance between the distributions $\{(\mathbf{A}, \mathbf{A} \cdot \mathbf{R}_{i,b}) \mid \mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m}), \mathbf{R}_{i,b} \hookleftarrow U(\{-1,1\}^{m \times m})\}$ and $\{(\mathbf{A}, \mathbf{A}_{i,b}) \mid \mathbf{A}, \mathbf{A}_{i,b} \hookleftarrow U(\mathbb{Z}_q^{n \times m})\}$ is smaller than $m \cdot \sqrt{q^n/2^m} < 2^{-\lambda}$, where the last inequality is implied by our choice of $m = 2n\lceil \log q \rceil + O(\lambda)$.

Lossiness. It is enough to prove that the distribution of a ciphertext obtained by encrypting under a lossy tag is statistically close to the uniform distribution on $\mathbb{Z}_q^n$.

For any initialization value $K \in \mathcal{K}_\lambda$ and tag $t \in \{0,1\}^\ell$ such that $(K, t) \notin \mathcal{R}_{\mathsf{BM}}$, any pair $\big(pk = (\mathbf{A}, \{\mathbf{A}_i\}_{i=1}^u), sk = (\mathbf{S}, K), tk\big) \leftarrow \mathsf{Keygen}(\mathit{\Gamma}, K)$, and any message $\mathsf{Msg} \in \{0,1\}^{n_0}$, an encryption of Msg is generated as

$$\mathbf{c} = [\mathbf{A} \mid \mathbf{A}_{F,t}] \cdot \mathbf{r} + \left[\begin{array}{c} \mathbf{0}^{n-n_0} \\ \hline \mathsf{Msg} \cdot \lfloor q/2 \rfloor \end{array}\right] \in \mathbb{Z}_q^n \ . \tag{4}$$

where $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^{2m}, \sigma}$ and $\bar{\mathbf{A}}_{F,t} = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_{F,t} + d_t \cdot \mathbf{G}] \in \mathbb{Z}_q^{n \times 2m}$. The matrix $\bar{\mathbf{A}}_{F,t}$ is of this form because t is a lossy tag (i.e., $(K, t) \notin \mathcal{R}_{\mathsf{BM}}$). This implies that the columns of $\bar{\mathbf{A}}_{F,t}$ generate $\mathbb{Z}_q^n$. By [85, Lemma 5.3], we know that $\bar{\mathbf{A}}_{F,t}$ has a trapdoor $\mathbf{T}_{F,t} \in \mathbb{Z}^{2m \times 2m}$ (namely, a short basis of the lattice $\mathit{\Lambda}^\perp(\bar{\mathbf{A}}_{F,t})$) such that $\|\tilde{\mathbf{T}}_{F,t}\| \leq (\|\mathbf{R}_{F,t}\| + 1) \cdot \sqrt{5}$ and thus $\|\tilde{\mathbf{T}}_{F,t}\| \leq \sqrt{5} \cdot (\sqrt{m} \cdot L + 1)$. Again, by [52, Lemma 3.1], we know that $\eta_{2^{-m}}(\mathit{\Lambda}^\perp(\bar{\mathbf{A}}_{F,t})) \leq \|\tilde{\mathbf{T}}_{F,t}\| \cdot O(\sqrt{m})$. By choosing

$\sigma = O(m) \cdot L$, we have $\sigma \geq \eta_{2^{-m}}(\Lambda^{\perp}(\bar{\mathbf{A}}_{F,t}))$. By applying [52, Lemma 5.2], we conclude that $\bar{\mathbf{A}}_{F,t} \cdot \mathbf{r}$ is statistically close to the uniform distribution $U(\mathbb{Z}_q^n)$ when $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$.

Efficient opening under lossy tags. From the previous paragraph, we know that the lattice $\Lambda_q^{\perp}(\bar{\mathbf{A}}_{F,t})$ has a basis satisfying $\|\tilde{\mathbf{T}}_{F,t}\| \leq \sqrt{5} \cdot (\sqrt{m}L + 1)$. By the choice of $\sigma = O(m) \cdot L$, the condition $\sigma \geq \|\tilde{\mathbf{T}}_{F,t}\| \cdot \omega(\sqrt{\log 2m})$ holds. For any $\mathbf{c}_{\mathsf{Msg}_1} \in \mathbb{Z}_q^n$, we can thus apply Lemma 2.3 and sample a Gaussian vector $\bar{\mathbf{r}} \in \mathbb{Z}^{2m}$ from the distribution $D_{\Lambda_q^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}),\sigma}$. Our argument to prove the lossiness under lossy tags implies that encrypting any message $\mathsf{Msg}_0 \in \{0,1\}^{n_0}$ under a lossy tag leads to a statistically uniform ciphertext $\mathbf{c} \sim_s U(\mathbb{Z}_q^n)$. In particular, for any $\mathsf{Msg}_1 \in \{0,1\}^{n_0}$, the distribution

$$\Big\{\big(\bar{\mathbf{A}}_{F,t},\ \mathbf{c}_{\mathsf{Msg}_1} = \bar{\mathbf{A}}_{F,t} \cdot \mathbf{r}_0 + \big[\tfrac{\mathbf{0}^{n-n_0}}{\mathsf{Msg}_0 \cdot \lfloor q/2 \rceil}\big] - \big[\tfrac{\mathbf{0}^{n-n_0}}{\mathsf{Msg}_1 \cdot \lfloor q/2 \rceil}\big],\ \bar{\mathbf{r}}\big) \ \mid \\ \mathbf{r}_0 \hookleftarrow D_{\mathbb{Z}^{2m},\sigma},\ \bar{\mathbf{r}} \hookleftarrow D_{\Lambda^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}),\sigma}\Big\}$$

is statistically close to

$$\Big\{\big(\ \bar{\mathbf{A}}_{F,t},\ \mathbf{c}_{\mathsf{Msg}_1} = \mathbf{c} - \big[\tfrac{\mathbf{0}^{n-n_0}}{\mathsf{Msg}_1 \cdot \lfloor q/2 \rceil}\big],\ \bar{\mathbf{r}}\ \big) \mid \mathbf{c} \hookleftarrow U(\mathbb{Z}_q^n),\ \bar{\mathbf{r}} \hookleftarrow D_{\Lambda^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}),\sigma}\Big\} \ ,$$

which is itself statistically close to $\big\{\big(\bar{\mathbf{A}}_{F,t},\ \mathbf{c}_{\mathsf{Msg}_1} = \bar{\mathbf{A}}_{F,t} \cdot \mathbf{r},\ \mathbf{r}\big) \mid\ \mathbf{r} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}\big\}$.

Efficient opening under lossy keys. By [35, Lemma 3.2], we know that a basis $\mathbf{T}_{\mathbf{A},t} \in \mathbb{Z}^{2m \times 2m}$ for the lattice $\Lambda_q^{\perp}([\mathbf{A}|\mathbf{A}_{F,t}])$ can be efficiently computed given a basis $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ of the lattice $\Lambda_q^{\perp}(\mathbf{A})$. Moreover, this basis satisfies $\|\tilde{\mathbf{T}}_{\mathbf{A}}\| = \|\tilde{\mathbf{T}}_{\mathbf{A},t}\|$. By Lemma 2.2, it follows that $\|\tilde{\mathbf{T}}_{\mathbf{A},t}\| \leq O(\sqrt{n \log q}) = O(\sqrt{m})$. By the choice of parameters, we obtain that $\sigma \geq \|\tilde{\mathbf{T}}_{\mathbf{A},t}\| \cdot \omega(\sqrt{\log 2m})$. Hence, by Lemma 2.3, we can sample $\bar{\mathbf{r}} \in \mathbb{Z}^{2m}$ from a distribution statistically close to $D_{\Lambda_q^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}),\sigma}$. The claim follows from the same arguments as in the case of efficient openings under lossy tags. □

3.2 A Generic Construction from Trapdoor Σ-Protocols and $\mathcal{R}_{\mathsf{BM}}$-lossy PKE

We construct unbounded simulation-sound NIZK proofs by combining trapdoor Σ-protocols and $\mathcal{R}$-lossy public-key encryption schemes. Our proof system is inspired by ideas from [83] and relies on the following ingredients:

- A trapdoor Σ-protocol $\Pi' = (\mathsf{Gen}'_{\mathsf{par}}, \mathsf{Gen}'_{\mathcal{L}}, \mathsf{P}', \mathsf{V}')$ with challenge space $\mathcal{C}$, for the same language $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$ and which satisfies the properties of Definition 2.9. In addition, $\mathsf{BadChallenge}(\tau_{\Sigma}, \mathsf{crs}, x, \mathbf{a})$ should be computable within time $T \in \mathsf{poly}(\lambda)$ for any input $(\tau, \mathsf{crs}, x, \mathbf{a})$.
- A strongly unforgeable one-time signature scheme $\mathsf{OTS} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys of length $\ell \in \mathsf{poly}(\lambda)$.

- An admissible hash function $\mathsf{AHF} : \{0,1\}^\ell \to \{0,1\}^L$, for some $L \in \mathsf{poly}(\lambda)$ with $L > \ell$, which induces the relation $\mathcal{R}_{\mathsf{BM}} : \{0,1,\perp\}^L \times \{0,1\}^\ell \to \{0,1\}$.
- An $\mathcal{R}$-lossy PKE scheme $\mathcal{R}\text{-}\mathsf{LPKE} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{LOpener})$ for the relation $\mathcal{R}_{\mathsf{BM}} : \{0,1,\perp\}^L \times \{0,1\}^\ell \to \{0,1\}$ with public (resp. secret) key space $\mathcal{PK}$ (resp. $\mathcal{SK}$). We assume that $\mathsf{Decrypt}$ is computable within time T. We denote the message (resp. ciphertext) space by MsgSp (resp. CtSp) and the randomness space by R^{LPKE}. Let also D_R^{LPKE} denote the distribution from which the random coins of $\mathsf{Encrypt}$ are sampled.
- A correlation intractable hash family $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ for the class $\mathcal{R}_{\mathsf{CI}}$ of relations that are efficiently searchable within time T.

We also assume that these ingredients are compatible in the sense that P' outputs a first prover message $\mathbf{a}$ that fits in the message space MsgSp of $\mathcal{R}$-LPKE.

Our argument system $\Pi^{\mathrm{uss}} = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ allows P and V to input a label lbl consisting of public data. While this label will be the empty string in our KDM-CCA scheme of Section, it may be useful when several non-interactive arguments have to be bound together. The construction goes as follows.

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

$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$: Given public parameters par and a language $\mathcal{L} \subset \{0,1\}^N$, let $\mathcal{K} = \{0,1,\perp\}^L$ and $\mathcal{T} = \{0,1\}^\ell$. The CRS is generated as follows.

1. Generate a CRS $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ for the trapdoor Σ-protocol Π'.
2. Generate public parameters $\Gamma \hookleftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$ for the $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme where the relation $\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ is defined by an admissible hash function $\mathsf{AHF} : \{0,1\}^\ell \to \{0,1\}^L$. Choose a random initialization value $K \leftarrow \mathcal{K}$ and generate lossy keys $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\Gamma, K)$.
3. Generate a key $k \leftarrow \mathsf{Gen}(1^\lambda)$ for a correlation intractable hash function with output length $\kappa = \Theta(\lambda)$.

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

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

1. Compute $\big(\mathbf{a}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa), st'\big) \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w)$ via κ invocations of the prover for Π'. Then, for each $i \in [\kappa]$, compute $\mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{a}'_i; \mathbf{r}_i)$ using random coins $\mathbf{r}_i \hookleftarrow D_R^{\mathsf{LPKE}}$. Let $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$ and $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_\kappa)$.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK})) \in \{0,1\}^\kappa$.
3. Compute $\mathbf{z}' = (\mathbf{z}'_1, \ldots, \mathbf{z}'_\kappa) = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, \mathbf{a}', \mathsf{Chall}, st')$ via κ executions of the prover of Π'. Define $\mathbf{z} = (\mathbf{z}', \mathbf{a}', \mathbf{r})$.
4. Generate $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, \mathbf{a}, \mathbf{z}, \mathsf{lbl}))$ and output $\boldsymbol{\pi} = \big(\mathsf{VK}, (\mathbf{a}, \mathbf{z}), sig\big)$.

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

1. Write $\mathbf{z}$ as $\mathbf{z} = \big((\mathbf{z}'_1, \ldots, \mathbf{z}'_\kappa), (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa), (\mathbf{r}_1, \ldots, \mathbf{r}_\kappa)\big)$ and return 0 if it does not parse properly. Return 0 if there exists $i \in [\kappa]$ such that $\mathbf{a}_i \neq \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{a}'_i; \mathbf{r}_i)$ or $\mathbf{r}_i \notin R^{\mathsf{LPKE}}$.
2. Let $\mathsf{Chall} = \mathsf{Hash}\big(k, (x, (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa), \mathsf{VK})\big)$. If $\mathsf{V}'(\mathsf{crs}'_{\mathcal{L}}, x, (\mathbf{a}'_i, \mathsf{Chall}[i], \mathbf{z}'_i)) = 1$ for each $i \in [\kappa]$, return 1. Otherwise, return 0.

Our NIZK simulator uses a technique due to Damgård [43], which uses a trapdoor commitment scheme to achieve a straight-line simulation of 3-move zero-knowledge proofs in the common reference string model.

Theorem 3.3. *The above argument system is multi-theorem zero-knowledge assuming that the trapdoor Σ-protocol Π' is special zero-knowledge.*

Proof (Sketch). We describe a simulator $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ which uses the lossy secret key $\tau_{\mathrm{zk}} = sk$ of $\mathcal{R}$-LPKE to simulate transcripts $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ without using the witnesses. Namely, on input of $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$, Sim_0 generates $\mathsf{crs}_{\mathcal{L}}$ by proceeding identically to $\mathsf{Gen}_{\mathcal{L}}$ while Sim_1 is described hereunder.

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

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

In Appendix C, we show that the simulation is statistically indistinguishable from proofs generated by the real prover. □

If we just target multi-theorem NIZK without simulation-soundness, the construction can be simplified as shown in Appendix B , where we explain how it can provide statistical zero-knowledge in the common random string (instead of the common reference string) model.

Going back to simulation-soundness, our proof builds on techniques used in [43,83]. The interactive proof systems of [83] rely on commitment schemes where the adversary cannot break the computational binding property of the commitment for some tag after having seen equivocations of commitments for different tags. Here, in order to use a correlation-intractable hash function, we need a commitment scheme which is equivocable on some tags but (with noticeable probability) becomes statistically binding on an adversarially-chosen tag. For this purpose, we exploit the observation that an $\mathcal{R}$-lossy PKE scheme can be used as a commitment scheme with these properties. Namely, it can serve as a trapdoor commitment to equivocate lossy encryptions of the first prover message in $\mathsf{\Pi}'$ while forcing the adversary to create a fake proof on a statistically binding (and even extractable) commitment.

At a high level, the proof also bears similarities with [76] in that they also use a commitment scheme that is statistically hiding in adversarial queries but becomes statistically binding in the adversary's output. The difference is that we need to equivocate the statistically-hiding commitment in simulated proofs here.

Theorem 3.4. *The above argument system provides unbounded simulation-soundness if: (i)* OTS *is a strongly unforgeable one-time signature; (ii)* $\mathcal{R}$-LPKE *is an* $\mathcal{R}_{\mathsf{BM}}$*-lossy PKE scheme; (iii) The hash family* $\mathcal{H}$ *is correlation-intractable for all relations that are searchable within time* T*, where* T *denotes the maximal running time of algorithms* $\mathsf{BadChallenge}(\cdot,\cdot,\cdot,\cdot)$ *and* $\mathsf{Decrypt}(\cdot,\cdot,\cdot)$.

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

Game_0: This is the real game of Definition A.3. Namely, the challenger runs $(\mathsf{crs}, \tau_{\mathrm{zk}}) \leftarrow \mathsf{Sim}_0(\mathsf{par}, 1^N)$ and gives $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, \varGamma, pk, \mathsf{AHF}, \varPi^{\mathrm{ots}}, \kappa)$ to the adversary $\mathcal{A}$. At the same time, the challenger generates a trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}_{\mathsf{sound}}$ in such a way that it can efficiently test if $\mathcal{A}$'s output satisfies the winning condition (ii). The adversary is granted oracle access to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$. At each query, $\mathcal{A}$ chooses a statement $x \in \{0,1\}^N$ with a label lbl and the challenger replies by returning a simulated argument $\boldsymbol{\pi} \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, x, \mathsf{lbl})$. When $\mathcal{A}$ halts, it outputs a triple $(x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star)$, where $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, (\mathbf{a}^\star, \mathbf{z}^\star), sig^\star\big)$. The Boolean variable W_0 is thus set to $W_0 = \mathtt{true}$ under the following three conditions: (i) $(x^\star, \mathsf{lbl}^\star, \boldsymbol{\pi}^\star) \notin \mathcal{Q}$, where $\mathcal{Q} = \{(x_i, \mathsf{lbl}_i, \boldsymbol{\pi}_i)\}_{i=1}^{Q}$ denotes the set of queries to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ and the corresponding responses $\boldsymbol{\pi}_i = \big(\mathsf{VK}^{(i)}, (\mathbf{a}_i, \mathbf{z}_i), sig_i\big)$; (ii) $x^\star \notin \mathcal{L}_{\mathsf{sound}}$; and (iii) $V(\mathsf{crs}, x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star) = 1$. We may assume w.l.o.g. that the one-time verification keys $\{\mathsf{VK}^{(i)}\}_{i=1}^{Q}$ are chosen ahead of time at the beginning of the game. By definition we have $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{uss}}(\lambda) = \Pr[W_0]$.

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

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

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

$$F_{\mathsf{ADH}}(K, \mathsf{VK}^{(1)}) = \cdots = F_{\mathsf{ADH}}(K, \mathsf{VK}^{(Q)}) = 1 \ \wedge \ F_{\mathsf{ADH}}(K, \mathsf{VK}^\star) = 0 \qquad (5)$$

are satisfied, where $\mathsf{VK}^\star$ is the one-time verification key in the adversary's output and $\mathsf{VK}^{(1)}, \ldots, \mathsf{VK}^{(Q)}$ are those in adversarial queries. If these conditions do not hold, the challenger aborts and sets $W_3 = \mathtt{false}$. For simplicity, we assume that $\mathcal{B}$ aborts at the very beginning of the game if it detects that there exists $i \in [Q]$ such that $F_{\mathsf{ADH}}(K, \mathsf{VK}^{(i)}) = 0$ (recall that $\{\mathsf{VK}^{(i)}\}_{i=1}^{Q}$ are chosen at the outset of the game by $\mathcal{B}$). If conditions (5) are satisfied, the challenger sets $W_3 = \mathtt{true}$ whenever $W_1 = \mathtt{true}$. Letting Fail denote the event that $\mathcal{B}$ aborts because (5) does not hold, we have $W_3 = W_2 \wedge \neg\mathsf{Fail}$. Since the key K of the admissible hash function is statistically independent of the adversary's view, we can apply Theorem 2.7 to argue that there is a noticeable function $\delta(\lambda)$ such that $\Pr[\neg\mathsf{Fail}] \geq \delta(\lambda)$. This implies

$$\Pr[W_3] = \Pr[W_2 \wedge \neg\mathsf{Fail}] \geq \delta(\lambda) \cdot \Pr[W_2] \ , \qquad (6)$$

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

We remark that, if conditions (5) are satisfied in Game_2, the sequence of one-time verification keys $(\mathsf{VK}^{(1)}, \ldots, \mathsf{VK}^{(Q)}, \mathsf{VK}^\star)$ satisfies $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{VK}^\star) = 1$ and $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{VK}^{(i)}) = 0$ for all $i \in [Q]$.

Game_4**:** In this game, we modify the oracle $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ and by exploiting the efficient opening property of $\mathcal{R}$-LPKE for lossy tags (instead of lossy keys). At the i-th query (x_i, lbl_i) to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$, we must have $F_{\mathsf{ADH}}(K, \mathsf{VK}^{(i)}) = 1$ (meaning that $\mathsf{VK}^{(i)}$ is a lossy tag as $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{VK}^{(i)}) = 0$) if $\mathcal{B}$ did not abort. This allows $\mathcal{B}$ to equivocate $\mathbf{a}$ using the trapdoor key tk instead of the lossy secret key sk of $\mathcal{R}$-LPKE. Namely, at step 4 of Sim_1, the modified $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ oracle computes random coins $\mathbf{r} \leftarrow \mathsf{Opener}(pk, tk, \mathsf{VK}, \mathbf{a}, \mathbf{a}')$ instead of running $\mathsf{LOpener}$ using sk. We define the Boolean variable W_4 exactly as W_3. Since Opener and $\mathsf{LOpener}$ output samples from the same distribution D_R^{LPKE} over R^{LPKE}, this implies that $|\Pr[W_4] - \Pr[W_3]| \leq 2^{-\Omega(\lambda)}$.

Game_5**:** We now modify the distribution of crs. At step 2 of Gen, we generate the keys for $\mathcal{R}$-LPKE as injective keys $(pk, sk, tk) \leftarrow \mathsf{Keygen}(\mathit{\Gamma}, K)$ instead of lossy keys $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\mathit{\Gamma}, K)$. The indistinguishability property (i) of $\mathcal{R}$-LPKE ensures that $\Pr[W_5]$ and $\Pr[W_4]$ are negligibly far apart. Recall that this indistinguishability property ensures that the distributions of pairs (pk, tk) produced by Keygen and $\mathsf{LKeygen}$ are computationally indistinguishable. We can thus easily build a distinguisher $\mathcal{B}$ against $\mathcal{R}$-LPKE that bridges between Game_4 and Game_5 (by using tk to simulate $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ as in Game_4). It comes that $|\Pr[W_5] - \Pr[W_4]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{indist\text{-}LPKE\text{-}1}}(\lambda)$.

Due to the modification introduced in Game_5, if the conditions (5) are all satisfied, we have $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{VK}^\star) = 1$, meaning that the adversary's fake proof $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, \big(\mathbf{a}^\star, \mathbf{z}^\star = (\mathbf{z}'^\star, \mathbf{a}'^\star, \mathbf{r}^\star)\big), sig^\star\big)$ involves an injective tag $\mathsf{VK}^\star$. Since pk is now an injective key, this implies that $\mathbf{a}^\star$ is an injective encryption of $\mathbf{a}'^\star$ under the tag $\mathsf{VK}^\star$ using the randomness $\mathbf{r}^\star$.

Game_6: We change again the distribution of $\mathsf{crs} = \big(\mathsf{par}, (\mathsf{crs}'_{\mathcal{L}}, k), pk, \mathsf{AHF}, \mathsf{OTS}\big)$ by relying on the CRS indistinguishability property of the trapdoor Σ-protocol Π'. Namely, we use the $\mathsf{TrapGen}'$ algorithm of Definition 2.9 to generate $\mathsf{crs}'_{\mathcal{L}}$ as $(\mathsf{crs}'_{\mathcal{L}}, \tau_\Sigma) \leftarrow \mathsf{TrapGen}'(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$ instead of $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$. We immediately have $|\Pr[W_6] - \Pr[W_5]| \leq \mathbf{Adv}^{\text{indist-}\Sigma}_{\mathcal{A}}(\lambda)$.

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

Game_7: In this game, we use the decryption algorithm of $\mathcal{R}$-LPKE. If $\mathcal{B}$ did not fail, we know that $\mathcal{A}$'s output $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, \big(\mathbf{a}^\star, \mathbf{z}^\star = (\mathbf{z}'^\star, \mathbf{a}'^\star, \mathbf{r}^\star)\big), sig^\star\big)$ involves an injective tag $\mathsf{VK}^\star$, so that $\mathbf{a}^\star = (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$ is a statistically binding commitment to $\mathbf{a}'^\star = (\mathbf{a}'^\star_1, \ldots, \mathbf{a}'^\star_\kappa)$. With probability $2^{-\Omega(\lambda)}$, for each $i \in [\kappa]$, there thus exists only one message $\mathbf{a}'^\star_i$ such that $\mathbf{a}^\star_i = \mathsf{Encrypt}(pk, \mathsf{VK}^\star, \mathbf{a}'^\star_i; \mathbf{r}^\star_i)$ for some $\mathbf{r}^\star_i \in R^{\mathsf{LPKE}}$. We thus consider the relation R_{bad} defined by

$$\begin{aligned} ((x, \mathbf{a}, \mathsf{VK}), \mathsf{Chall}) \in R_{\mathsf{bad}} \quad &\Leftrightarrow \quad x \notin \mathcal{L} \quad \wedge \\ &\forall i \in [\kappa] \;:\; \mathsf{Chall}[i] = \mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Decrypt}(sk, \mathsf{VK}, \mathbf{a}_i)). \end{aligned} \tag{7}$$

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

$$\begin{aligned} \mathsf{Hash}(k, (x^\star, \mathbf{a}^\star, \mathsf{VK}^\star)) \neq \big(&\mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x^\star, \mathsf{Decrypt}(sk, \mathsf{VK}^\star, \mathbf{a}^\star_1),\\ &\ldots,\; \mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x^\star, \mathsf{Decrypt}(sk, \mathsf{VK}^\star, \mathbf{a}^\star_\kappa)\big) \end{aligned}$$

and $W_7 = W_6$ otherwise. The decryption property under injective tags implies $|\Pr[W_7] - \Pr[W_6]| \leq 2^{-\Omega(\lambda)}$ since, unless some $\mathbf{a}^\star_i$ does not decrypt to $\mathbf{a}'^\star_i$, $\boldsymbol{\pi}^\star$ cannot correctly verify if the above inequality does not hold.

In Game_7, we have $\Pr[W_7] \leq \mathbf{Adv}^{\mathrm{CI}}_{\mathcal{A}}(\lambda)$ as the equality

$$\begin{aligned} \mathsf{Hash}(k, (x^\star, \mathbf{a}^\star, \mathsf{VK}^\star)) = \big(&\mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x^\star, \mathsf{Decrypt}(sk, \mathsf{VK}^\star, \mathbf{a}^\star_1),\\ &\ldots,\; \mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x^\star, \mathsf{Decrypt}(sk, \mathsf{VK}^\star, \mathbf{a}^\star_\kappa)\big) \end{aligned}$$

would break the correlation-intractability of $\mathcal{H}$ for the relation R_{bad}. Note that the work of [87] implies a correlation intractable hash function for the relation R_{bad} defined above. Their bootstrapping theorem implies the existence of such a hash family under the LWE assumption with polynomial approximation factors.

Putting the above altogether, we obtain

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

which completes the proof. □

We note that the above security proof is not tight as the use of admissible hash functions induces a security loss $1/\delta(\lambda)$ (where $\delta(\lambda)$ is the non-negligible function of Theorem 2.7) in the upper bound on the adversary's advantage. In Section 4, we give a variant of our argument system where the simulation-soundness property tightly relates to the security of a pseudorandom function.

4 Tightly Secure Simulation-Sound Arguments

To achieve tight simulation-soundness, we describe an $\mathcal{R}$-lossy PKE scheme for a relation induced by a pseudorandom function family. In Definition 4.1, we assume that the tag space $\mathcal{T}$ has a special structure. Namely, each tag $t = (t_c, t_a) \in \mathcal{T}$ consists of a core component $t_c \in \{0,1\}^\lambda$ and an auxiliary component $t_a \in \{0,1\}^\ell$.

Definition 4.1. *Let a pseudorandom function* $\mathsf{PRF} : \mathcal{K} \times \{0,1\}^\ell \to \{0,1\}^\lambda$ *with key space* $\mathcal{K} = \{0,1\}^\lambda$ *and input space* $\{0,1\}^\ell$*. Let* $\mathcal{T} = \{0,1\}^\lambda \times \{0,1\}^\ell$*, for some* $\ell \in \mathsf{poly}(\lambda)$*. We define the PRF relation* $\mathcal{R}_{\mathsf{PRF}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ *as* $\mathcal{R}_{\mathsf{PRF}}(K, (t_c, t_a)) = 1$ *if and only if* $t_c \neq \mathsf{PRF}(K, t_a)$*.*

We rely on the idea (previously used in [21,77]) of homomorphically evaluating the circuit of a PRF using the GSW FHE [53]. As observed in [26], when the circuit is in NC1, it is advantageous to convert it into a branching program using Barrington's theorem. This enables the use of a polynomial modulus q.

Lemma 4.2 (Adapted from [53,18]). *Let* $C : \{0,1\}^L \to \{0,1\}$ *be a NAND Boolean circuit of depth* d*. Let* $\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i + k_i \cdot \mathbf{G} \in \mathbb{Z}_q^{n \times m}$ *with* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$*,* $\mathbf{R}_i \in \{-1,1\}^{m \times m}$ *and* $k_i \in \{0,1\}$*, for* $i \leq L$*. There exist deterministic algorithms* $\mathsf{Eval}_{\mathsf{BP}}^{\mathsf{pub}}$ *and* $\mathsf{Eval}_{\mathsf{BP}}^{\mathsf{priv}}$ *with running time* $\mathsf{poly}(4^d, L, m, n, \log q)$ *that satisfy:* $\mathsf{Eval}_{\mathsf{BP}}^{\mathsf{pub}}(C, (\mathbf{A}_i)_i) = \mathbf{A} \cdot \mathsf{Eval}_{\mathsf{BP}}^{\mathsf{priv}}\big(C, ((\mathbf{R}_i, k_i))_i\big) + C(k_1, \ldots, k_L) \cdot \mathbf{G}$*, and* $\|\mathsf{Eval}_{\mathsf{BP}}^{\mathsf{priv}}(C, (\mathbf{R}_i, k_i)_i)\| \leq 4^d \cdot O(m^{3/2})$*.*

4.1 An $\mathcal{R}_{\mathsf{PRF}}$-Lossy PKE Scheme

We describe an $\mathcal{R}$-lossy PKE scheme for the relation $\mathcal{R}_{\mathsf{PRF}}$ of Definition 4.1.

Let $\mathsf{PRF} : \mathcal{K} \times \{0,1\}^\ell \to \{0,1\}^\lambda$ with key space $\mathcal{K} = \{0,1\}^\lambda$ and input space $\{0,1\}^\ell$ and let $\mathcal{R}_{\mathsf{PRF}} \subset \mathcal{K} \times \mathcal{T}$ the corresponding relation. We construct an $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme in the following way.

Par-Gen(1^λ)**:** Given a security parameter $\lambda \in \mathbb{N}$, let $n_0 = \mathsf{poly}(\lambda)$ the length of messages. Choose a prime modulus $q = \mathsf{poly}(\lambda)$; dimensions $n = n_0 + \Omega(\lambda)$ and $m = 2n\lceil \log q \rceil + O(\lambda)$. Define the tag space as $\mathcal{T} = \{0,1\}^\lambda \times \{0,1\}^\ell$ where $\ell = \Theta(\lambda)$. Define the initialization value space $\mathcal{K} = \{0,1\}^\lambda$ and Gaussian parameters $\sigma = 4^d \cdot O\left(m^2\right)$ and $\alpha \in (0,1)$ such that $4^d m^{3.5} \alpha q \cdot \sigma < q$. Define public parameters as $\Gamma = (\ell, L, n_0, q, n, m, u, \alpha, \sigma)$.

Keygen(Γ, K)**:** On input of public parameters Γ and an initialization value $K \in \{0,1\}^\lambda$, generate a key pair as follows.

1. Sample random matrices $\bar{\mathbf{B}} \hookleftarrow U(\mathbb{Z}_q^{(n-n_0)\times m})$, $\mathbf{S} \hookleftarrow U(\mathbb{Z}_q^{(n-n_0)\times n_0})$ and a small-norm $\mathbf{E} \hookleftarrow \chi^{m \times n_0}$ to compute $\mathbf{A} = [\bar{\mathbf{B}}^\top \mid \bar{\mathbf{B}}^\top \mathbf{S} + \mathbf{E}]^\top \in \mathbb{Z}_q^{n \times m}$.
2. Parse K as $k_1 \ldots k_\lambda \in \{0,1\}^\lambda$. For each $i \in [L]$, compute matrices $\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i + k_i \cdot \mathbf{G}$, where $\mathbf{R}_i \hookleftarrow U(\{-1,1\}^{m \times m})$, for all $i \in [\lambda]$.

Define $R^{\mathsf{LPKE}} = \{\mathbf{r} \in \mathbb{Z}^{2m} \mid \|\mathbf{r}\| \leq \sigma\sqrt{2m}\}$ and output $sk = (K, \mathbf{S})$ as well as

$$pk := \Big(\mathbf{A},\ \{\mathbf{A}_i\}_{i\in[\lambda]}\Big), \qquad tk = (K, \{\mathbf{R}_i\}_{i\in[\lambda]}).$$

LKeygen(Γ, K)**:** This algorithm proceeds identically to Keygen except that steps 1 and 2 are modified in the following way.

1. Run $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \mathsf{GenTrap}(1^\lambda, 1^n, 1^m, q)$ to obtain a statistically uniform $\mathbf{A} \sim U(\mathbb{Z}_q^{n\times m})$ with a trapdoor for $\Lambda^\perp(\mathbf{A})$.
2. Define matrices $\{\mathbf{A}_i \in \mathbb{Z}_q^{n\times m}\}_{i\in[\lambda]}$ as in Keygen.

Output $pk := \Big(\mathbf{A},\ \{\mathbf{A}_i\}_{i\in[\lambda]}\Big)$, $sk = \mathbf{T}_\mathbf{A}$, and $tk = (K, \{\mathbf{R}_i\}_{i\in[\lambda]})$.

Encrypt(pk, t, Msg)**:** To encrypt a message $\mathsf{Msg} \in \{0,1\}^{n_0}$ for the structured tag $t = (t_c, t_a) \in \mathcal{T} = \{0,1\}^\lambda \times \{0,1\}^\ell$, conduct the following steps.

1. Let $C_{\mathsf{PRF},t} : \{0,1\}^\lambda \to \{0,1\}$ the circuit, where $t = (t_c, t_a)$ is hard-wired, which inputs a λ-bit key $K = k_1 \ldots k_\lambda \in \{0,1\}^\lambda$ and outputs $C_{\mathsf{PRF},t}(K)$ such that $C_{\mathsf{PRF},t}(K) = 1 \;\Leftrightarrow\; t_c = \mathsf{PRF}_K(t_a) \;\Leftrightarrow\; \mathcal{R}_{\mathsf{PRF}}(K,t) = 0$. Compute $\mathbf{A}_{F,t} \leftarrow \mathsf{Eval}^{\mathsf{pub}}_{\mathsf{BP}}(C_{\mathsf{PRF}}, (\mathbf{A}_i)_i) \in \mathbb{Z}_q^{n\times m}$ such that
$$\mathbf{A}_{F,t} = \mathbf{A} \cdot \mathbf{R}_t + C_{\mathsf{PRF},t}(K) \cdot \mathbf{G},$$
where $\mathbf{R}_t = \mathsf{Eval}^{\mathsf{priv}}_{\mathsf{BP}}(C_{\mathsf{PRF},t}, (\mathbf{R}_i, k_i)_i) \in \mathbb{Z}^{m\times m}$ s.t. $\|\mathbf{R}_t\| \leq 4^d \cdot O(m^{3/2})$.
2. Choose $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^{2m},\sigma}$ and output $\perp$ if $\mathbf{r} \notin R^{\mathsf{LPKE}}$. Otherwise, output
$$\mathbf{c} = [\mathbf{A} \mid \mathbf{A}_{F,t}] \cdot \mathbf{r} + [\mathbf{0}^{n-n_0\top} \mid \mathsf{Msg} \cdot \lfloor q/2 \rfloor]^\top \;\in \mathbb{Z}_q^n \;.$$

Decrypt$(sk, t, \mathbf{c})$**:** Given the secret key $sk = (K, \mathbf{S})$ and the tag $t = (t_c, t_a) \in \mathcal{T}$, compute $C_{\mathsf{PRF},t}(K) \in \{0,1\}$ and return $\perp$ if $C_{\mathsf{PRF},t}(K) = 1$. Otherwise, Compute and return $\mathsf{Msg} = \mathsf{Msg}[1] \ldots \mathsf{Msg}[n_0]$ exactly as in Section 3.1.

Opener$(pk, tk, t, \mathbf{c}, \mathsf{Msg}_1)$**:** Given $tk = (K, \{\mathbf{R}_i\}_i)$ and $t = (t_c, t_a) \in \mathcal{T}$, compute $C_{\mathsf{PRF},t}(K) \in \{0,1\}$ and return $\perp$ if $C_{\mathsf{PRF},t}(K) = 0$. Otherwise,

1. Compute the matrix $\mathbf{R}_t = \mathsf{Eval}^{\mathsf{priv}}_{\mathsf{BP}}(C_{\mathsf{PRF},t}, (\mathbf{R}_i, k_i)_i) \in \mathbb{Z}^{m \times m}$ such that $\mathbf{A}_{F,t} = \mathbf{A} \cdot \mathbf{R}_t + \mathbf{G}$ and $\|\mathbf{R}_t\| \leq 4^d \cdot O(m^{3/2})$.
2. Use $\mathbf{R}_t \in \mathbb{Z}^{m \times m}$ as a trapdoor for $\bar{\mathbf{A}}_{F,t} = [\mathbf{A} \mid \mathbf{A}_{F,t}] = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_t + \mathbf{G}]$ to sample $\bar{\mathbf{r}} \in \mathbb{Z}^{2m}$ such that $\bar{\mathbf{A}}_{F,t} \cdot \bar{\mathbf{r}} = \mathbf{c} - [\mathbf{0}^{n-n_0 \top} \mid \mathsf{Msg}_1 \cdot \lfloor q/2 \rfloor]^\top$. Namely, defining $\mathbf{c}_{\mathsf{Msg}_1} = \mathbf{c} - [(\mathbf{0}^{n-n_0})^\top \mid \mathsf{Msg}_1^\top \cdot \lfloor q/2 \rfloor]^\top$, sample and output fake random coins $\bar{\mathbf{r}} \hookleftarrow D_{\Lambda_q^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}),\sigma}$.

LOpener$(sk, t, \mathbf{c}, \mathsf{Msg}_1)$**:** Given $sk = \mathbf{T}_{\mathbf{A}}$ and $t = (t_c, t_a) \in \mathcal{T}$, use $\mathbf{T}_{\mathbf{A}}$ to derive a trapdoor $\mathbf{T}_{\mathbf{A},t}$ for the lattice $\Lambda_q^{\perp}(\bar{\mathbf{A}}_{F,t})$ and use $\mathbf{T}_{\mathbf{A},t}$ to sample a Gaussian vector $\bar{\mathbf{r}} \hookleftarrow D_{\Lambda_q^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}),\sigma}$ in the same coset of $\Lambda_q^{\perp}(\bar{\mathbf{A}}_{F,t})$ as in Opener.

The proof of Theorem 4.3 is identical to that of Theorem 3.2 and omitted.

Theorem 4.3. *The above construction is an $\mathcal{R}_{\mathsf{PRF}}$-lossy public-key encryption scheme with efficient opening under the* LWE *assumption.*

4.2 Unbounded Simulation-Sound Argument

We construct a tightly secure USS argument from the following ingredients:

- A pseudorandom function family $\mathsf{PRF} : \mathcal{K} \times \{0,1\}^\ell \rightarrow \{0,1\}^\lambda$ with key space $\mathcal{K} = \{0,1\}^\lambda$ and input space $\{0,1\}^\ell$, which induces the relation $\mathcal{R}_{\mathsf{PRF}} : \mathcal{K} \times \mathcal{T} \rightarrow \{0,1\}$ of Definition 4.1.
- An $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme $\mathcal{R}$-LPKE = (Par-Gen, Keygen, LKeygen, Encrypt, Decrypt, Opener, LOpener) for the relation $\mathcal{R}_{\mathsf{PRF}} : \mathcal{K} \times \mathcal{T} \rightarrow \{0,1\}$ with public (resp. secret) key space $\mathcal{PK}$ (resp. $\mathcal{SK}$). We assume that Decrypt is computable within time T. We denote the message (resp. ciphertext) space by MsgSp (resp. CtSp) and the randomness space by R^{LPKE}. Let also D_R^{LPKE} denote the distribution of the random coins of Encrypt.
- A trapdoor Σ-protocol $\Pi' = (\mathsf{Gen}'_{\mathsf{par}}, \mathsf{Gen}'_{\mathcal{L}}, \mathsf{P}', \mathsf{V}')$, a one-time signature scheme $\mathsf{OTS} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ and a correlation intractable hash family $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ that satisfy the same conditions as in Section 3.2.

Our construction $\Pi^{\mathrm{uss}} = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ goes as follows.

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

Gen$_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$**:** Given public parameters par and a language $\mathcal{L} \subset \{0,1\}^N$, let $\mathcal{K} = \{0,1\}^\lambda$ and $\mathcal{T} = \{0,1\}^\ell$. The CRS is generated as follows.

1. Generate a CRS $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ for the trapdoor Σ-protocol Π'.
2. Generate public parameters $\Gamma \hookleftarrow$ Par-Gen(1^λ) for the $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme where the relation $\mathcal{R}_{\mathsf{PRF}} : \mathcal{K} \times \mathcal{T} \rightarrow \{0,1\}$ is defined by a PRF family $\mathsf{PRF} : \mathcal{K} \times \{0,1\}^\ell \rightarrow \{0,1\}^\lambda$. Generate lossy keys $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\Gamma, \mathbf{0}^\lambda)$, where the initialization value is the all-zeroes string $\mathbf{0}^\lambda$.
3. Generate a key $k \leftarrow \mathsf{Gen}(1^\lambda)$ for a correlation intractable hash function with output length $\kappa = \Theta(\lambda)$.

Output the language-dependent $\mathsf{crs}_{\mathcal{L}} := \big(\mathsf{crs}'_{\mathcal{L}}, k\big)$ and the simulation trapdoor $\tau_{\mathrm{zk}} := sk$. The global CRS consists of $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, pk, \mathsf{PRF}, \mathsf{OTS})$.

$\mathbf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$: To prove x with respect to a label lbl using $w \in R_{\mathsf{zk}}(x)$, generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^\lambda)$. Then, choose a random core tag component $t_c \hookleftarrow U(\{0,1\}^\lambda)$ and do the following.

1. Compute $\big(\mathbf{a}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa), st'\big) \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w)$ via κ invocations of the prover for Π'. For each $i \in [\kappa]$, compute $\mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, (t_c, \mathsf{VK}), \mathbf{a}'_i; \mathbf{r}_i)$ using random coins $\mathbf{r}_i \hookleftarrow D_R^{\mathsf{LPKE}}$. Let $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$ and $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_\kappa)$.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}, t_c, \mathsf{VK})) \in \{0,1\}^\kappa$.
3. Compute $\mathbf{z}' = (\mathbf{z}'_1, \ldots, \mathbf{z}'_\kappa) = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, \mathbf{a}', \mathsf{Chall}, st')$ via κ executions of the prover of Π'. Define $\mathbf{z} = (\mathbf{z}', \mathbf{a}', \mathbf{r})$.
4. Generate a one-time signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, t_c, \mathbf{a}, \mathbf{z}, \mathsf{lbl}))$ and output the proof $\boldsymbol{\pi} = \big((t_c, \mathsf{VK}), (\mathbf{a}, \mathbf{z}), sig\big)$.

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

1. Write $\mathbf{z}$ as $\mathbf{z} = \big((\mathbf{z}'_1, \ldots, \mathbf{z}'_\kappa), (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa), (\mathbf{r}_1, \ldots, \mathbf{r}_\kappa)\big)$. Return 0 if there exists $i \in [\kappa]$ such that $\mathbf{a}_i \neq \mathsf{Encrypt}(pk, (t_c, \mathsf{VK}), \mathbf{a}'_i; \mathbf{r}_i)$ or $\mathbf{r}_i \notin R^{\mathsf{LPKE}}$.
2. Let $\mathsf{Chall} = \mathsf{Hash}\big(k, (x, (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa), t_c, \mathsf{VK})\big)$. If there exists $i \in [\kappa]$ such that $\mathsf{V}'(\mathsf{crs}'_{\mathcal{L}}, x, (\mathbf{a}'_i, \mathsf{Chall}[i], \mathbf{z}'_i)) = 0$, return 0. Otherwise, return 1.

In Appendix D, we show that the unbounded simulation-soundness of the above argument system is tightly related to the security of its underlying building blocks, which are all instantiable (with tight security reductions) from the LWE assumption with polynomial approximation factors.

5 Trapdoor Σ-Protocols for ACPS Ciphertexts

The KDM-CPA system of Applebaum *et al.* [7] uses a modulus $q = p^2$, for some prime p. Its public key $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^m$ contains a random matrix $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$ and a vector $\mathbf{b} = \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}$, for some $\mathbf{s} \sim D_{\mathbb{Z}^n, \alpha q}$, $\mathbf{e} \sim D_{\mathbb{Z}^m, \alpha q}$. Its encryption algorithm proceeds analogously to the primal Regev cryptosystem [90] and computes $\mathbf{c} = (\bar{\mathbf{c}}, c) = (\mathbf{A} \cdot \mathbf{r}, \mathbf{b}^\top \mathbf{r} + \mu \cdot p + \chi) \in \mathbb{Z}_q^{n+1}$, where $\mathbf{r} \sim D_{\mathbb{Z}^m, r}$ is a Gaussian vector and $\chi \sim D_{\mathbb{Z}, r'}$ is sampled from a Gaussian with a slightly larger standard deviation. Decryption proceeds by rounding $c - \mathbf{s}^\top \cdot \bar{\mathbf{c}} \mod q$ to the nearest multiple of p.

In this section, we describe a trapdoor Σ-protocol allowing to prove that two ACPS ciphertexts $\mathbf{c}_0 = (\bar{\mathbf{c}}_0, c_0)$, $\mathbf{c}_1 = (\bar{\mathbf{c}}_1, c_1)$ are both encryptions of the same $\mu \in \mathbb{Z}_p$. This protocol is obtained by extending a simpler protocol (described in Appendix E.1), which argues that a given vector $\mathbf{c} \in \mathbb{Z}_q^{n+1}$ is an ACPS encryption of some plaintext $\mu \in \mathbb{Z}_p$.

We note that Ciampi *et al.* [38] recently gave a construction of trapdoor Σ-protocol with binary challenges from any Σ-protocol. The Σ-protocol described hereunder is natively trapdoor without applying the transformation of [38].

Proving Plaintext Equalities in ACPS Ciphertexts. Let $q = p^2$, for some prime p, and a matrix $\mathbf{A}$ which is used to set up two Regev public keys $(\mathbf{A}, \mathbf{b}_0) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$ and $(\mathbf{A}, \mathbf{b}_1) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$, where $\mathbf{b}_0 = \mathbf{A}^\top \cdot \mathbf{s}_0 + \mathbf{e}_0$ and $\mathbf{b}_1 = \mathbf{A}^\top \cdot \mathbf{s}_1 + \mathbf{e}_1$ for some $\mathbf{s}_0, \mathbf{s}_1 \sim D_{\mathbb{Z}^n,\alpha q}$, $\mathbf{e}_0, \mathbf{e}_1 \sim D_{\mathbb{Z}^m,\alpha q}$. Let also the matrix

$$\mathbf{A}_{\mathrm{eq}} = \left[\begin{array}{c|c|c|c} \mathbf{A} & & & \\ \hline \mathbf{b}_0^\top & 1 & & \\ \hline & & \mathbf{A} & \\ \hline & & \mathbf{b}_1^\top & 1 \end{array}\right] \in \mathbb{Z}_q^{2(n+1)\times 2(m+1)}, \tag{8}$$

We give a trapdoor $\varSigma$-protocol for the language $\mathcal{L}^{\mathrm{eq}} = (\mathcal{L}^{\mathrm{eq}}_{\mathsf{zk}}, \mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}})$, where

$$\begin{aligned} \mathcal{L}^{\mathrm{eq}}_{\mathsf{zk}} := \Big\{ (\mathbf{c}_0, \mathbf{c}_1) \in (\mathbb{Z}_q^{n+1})^2 \ \mid\ & \exists \mathbf{r}_0, \mathbf{r}_1 \in \mathbb{Z}^m,\ \chi_0, \chi_1 \in \mathbb{Z},\ \mu \in \mathbb{Z}_p\ : \\ & \|\mathbf{r}_b\| \le B_r,\ |\chi_b| \le B_\chi \quad \forall b \in \{0,1\} \\ & \wedge \quad \mathbf{c}_b = \bar{\mathbf{A}}_b \cdot [\mathbf{r}_b^\top \mid \chi_b]^\top + \mu \cdot \left[\mathbf{0}^{n\top} \mid p\right]^\top \bmod q \ \Big\}, \end{aligned}$$

$$\begin{aligned} \mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}} := \Big\{ (\mathbf{c}_0, \mathbf{c}_1) \in (\mathbb{Z}_q^{n+1})^2 \ \mid\ & \exists\ \bar{\mathbf{c}}_0, \bar{\mathbf{c}}_1 \in \mathbb{Z}_q^n,\ v_0, v_1 \in [-B^*, B^*],\ \mu \in \mathbb{Z}_p \\ \wedge\ \mathbf{c}_b = & \left[\begin{array}{c} \bar{\mathbf{c}}_b \\ \hline \mathbf{s}_b^\top \cdot \bar{\mathbf{c}}_b + p \cdot \mu + v_b \end{array}\right] \quad \forall b \in \{0,1\} \Big\}, \end{aligned}$$

where

$$\bar{\mathbf{A}}_b = \left[\begin{array}{c|c} \mathbf{A} & \\ \hline \mathbf{b}_b^\top & 1 \end{array}\right] \in \mathbb{Z}_q^{(n+1)\times(m+1)} \qquad\qquad \forall b \in \{0,1\}.$$

We note that $\mathcal{L}^{\mathrm{eq}}_{\mathsf{zk}} \subseteq \mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}}$ when $B_r \alpha q \sqrt{m} + B_\chi < B^* \ll p$. Also, $\mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}}$ is equivalently defined as the language of pairs $(\mathbf{c}_0, \mathbf{c}_1)$ such that such that

$$\left[\begin{array}{c|c|c|c} -\mathbf{s}_0^\top & 1 & & \\ \hline & & -\mathbf{s}_1^\top & 1 \end{array}\right] \cdot \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix} \bmod q \ = \ \begin{bmatrix} v_0 \\ v_1 \end{bmatrix} + \mu \cdot \begin{bmatrix} p \\ p \end{bmatrix}$$

for some $\mu \in \mathbb{Z}_p$, $v_0, v_1 \in [-B^*, B^*]$.

$\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$: On input of a security parameter $\lambda \in \mathbb{N}$, choose moduli q, p with $q = p^2$, dimensions n, m, and error rate $\alpha > 0$ and a Gaussian parameter $\sigma_{\mathrm{eq}} \ge \log(2m+2) \cdot \sqrt{B_r^2 + B_\chi^2}$. Define public parameters $\mathsf{par} = \{\lambda, q, p, n, m, \alpha, \sigma_{\mathrm{eq}}\}$.

$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}^{\mathrm{eq}})$: Takes in global parameters par and the description of a language $\mathcal{L}^{\mathrm{eq}} = (\mathcal{L}^{\mathrm{eq}}_{\mathsf{zk}}, \mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}})$ specifying real numbers $B^*, B_r, B_\chi > 0$ such that $B_r \alpha q \sqrt{m} + B_\chi < B^* \ll p$, and a matrix $\mathbf{A}_{\mathrm{eq}}$ from the distribution (8). It defines the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{\bar{\mathbf{A}}, B^*, B_r, B_\chi\}$. The global CRS is

$$\mathsf{crs} = \big(\{\lambda, q, p, n, m, \alpha, \sigma_{\mathrm{eq}}\}, \{\mathbf{A}_{\mathrm{eq}}, B^*, B_r, B_\chi\}\big).$$

$\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$: Given par and a language description $\mathcal{L}^{\mathrm{eq}}$ that specifies $B^*, B_r, B_\chi > 0$ satisfying the same constraints as in $\mathsf{Gen}_{\mathcal{L}}$, a matrix $\mathbf{A}_{\mathrm{eq}}$

sampled from the distribution (8), as well as a membership-testing trapdoor $\tau_{\mathcal{L}} = (\mathbf{s}_0, \mathbf{s}_1) \sim (D_{\mathbb{Z}^n,\alpha q})^2$ for $\mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}}$, output $\mathsf{crs}_{\mathcal{L}} = \{\bar{\mathbf{A}}, B^*, B_r, B_\chi\}$. The global CRS is $\mathsf{crs} = \big(\{\lambda, q, p, n, m, \alpha, \sigma_{\mathrm{eq}}\}, \{\mathbf{A}_{\mathrm{eq}}, B^*, B_r, B_\chi\}\big)$ and the trapdoor $\tau_\Sigma = (\mathbf{s}_0, \mathbf{s}_1) \in \mathbb{Z}^n \times \mathbb{Z}^n$.

$\mathbf{P}\big(\mathsf{crs}, (\mathbf{c}_0, \mathbf{c}_1), (\mu, \mathbf{w})\big) \leftrightarrow \mathbf{V}(\mathsf{crs}, \mathbf{x})$: Given crs and a statement

$$\begin{bmatrix} \mathbf{c}_0 \\ \hline \mathbf{c}_1 \end{bmatrix} = \mathbf{A}_{\mathrm{eq}} \cdot [\mathbf{r}_0{}^\top \mid \chi_0 \mid \mathbf{r}_1{}^\top \mid \chi_1]^\top + \mu \cdot [\mathbf{0}^{n\top} \mid p \mid \mathbf{0}^{n\top} \mid p]^\top \in \mathbb{Z}_q^{2(n+1)},$$

the prover P (who has $\mathbf{w} = [\mathbf{r}_0^\top \mid \chi_0 \mid \mathbf{r}_1^\top \mid \chi_1]^\top \in \mathbb{Z}^{2(m+1)}$ and $\mu \in \mathbb{Z}_p$) and the verifier V interact as follows.

1. The prover P samples a uniform scalar $r_\mu \hookleftarrow U(\mathbb{Z}_p)$ and Gaussian vector $\mathbf{r}_w \hookleftarrow D_{\mathbb{Z}^{2(m+1)},\sigma_{\mathrm{eq}}}$. It computes the following which is sent to V:

$$\mathbf{a} = \mathbf{A}_{\mathrm{eq}} \cdot \mathbf{r}_w + r_\mu \cdot [\mathbf{0}^{n\top} \mid p \mid \mathbf{0}^{n\top} \mid p]^\top \in \mathbb{Z}_q^{2(n+1)}.$$

2. V sends a random challenge $\mathsf{Chall} \in \{0,1\}$ to P.
3. P computes $\mathbf{z} = \mathbf{r}_w + \mathsf{Chall} \cdot \mathbf{w} \ \in \mathbb{Z}^{2(m+1)}$, $z_\mu = r_\mu + \mathsf{Chall} \cdot \mu \mod p$. It sends $(\mathbf{z}, z_\mu)$ to V with probability $\theta = \min\Big(\frac{D_{\mathbb{Z}^{2(m+1)},\sigma_{\mathrm{eq}}}(\mathbf{z})}{M \cdot D_{\mathbb{Z}^{2(m+1)},\sigma_{\mathrm{eq}},\mathsf{Chall}\cdot\mathbf{w}}(\mathbf{z})}, 1\Big)$, where $M = e^{12/\log(2(m+1)) + 1/(2\log^2(2(m+1)))}$. With probability $1-\theta$, P aborts.
4. Given $(\mathbf{z}, z_\mu) \in \mathbb{Z}^{2(m+1)} \times \mathbb{Z}_p$, V checks if $\|\mathbf{z}\| \leq \sigma_{\mathrm{eq}}\sqrt{2(m+1)}$ and

$$\mathbf{a} + \mathsf{Chall} \cdot \begin{bmatrix} \mathbf{c}_0 \\ \hline \mathbf{c}_1 \end{bmatrix} = \mathbf{A}_{\mathrm{eq}} \cdot \mathbf{z} + z_\mu \cdot [\mathbf{0}^{n\top} \mid p \mid \mathbf{0}^{n\top} \mid p]^\top \bmod q. \tag{9}$$

If these conditions do not both hold, V halts and returns $\perp$.

$\mathsf{\mathbf{BadChallenge}}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, (\mathbf{c}_0, \mathbf{c}_1), \mathbf{a}\big)$: Given $\tau_\Sigma = (\mathbf{s}_0, \mathbf{s}_1) \in \mathbb{Z}^n \times \mathbb{Z}^n$, parse the first prover message as $\mathbf{a} = (\mathbf{a}_0^\top \mid \mathbf{a}_1^\top)^\top \in \mathbb{Z}_q^{2(n+1)}$. If there exists $d \in \{0,1\}$ such that no pair $(\mu'_d, \mathbf{v}_d) \in [-(p-1)/2, (p-1)/2] \times [-B^*/2, B^*/2]^2$ satisfies

$$\left[\begin{array}{c|c|c|c} -\mathbf{s}_0^\top & 1 & & \\ \hline & & -\mathbf{s}_1^\top & 1 \end{array}\right] \cdot (\mathbf{a} + d \cdot \mathbf{c}) \bmod q \ = \ \mathbf{v}_d + \mu'_d \cdot \begin{bmatrix} p \\ \hline p \end{bmatrix} \tag{10}$$

over $\mathbb{Z}$, then return $\mathsf{Chall} = 1 - d$. Otherwise, return $\mathsf{Chall} = \perp$.

The completeness of the protocol crucially uses the fact that p divides q to ensure that the response $z_\mu = r_\mu + \mathsf{Chall} \cdot \mu \bmod p$ satisfies (9).

The intuition of $\mathsf{BadChallenge}$ is that, for a false statement $(\mathbf{c}_0, \mathbf{c}_1) \notin \mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}}$, there exists $d \in \{0,1\}$ such that no pair $(\mu'_d, \mathbf{v}_d)$ satisfies (10) for a small enough $\mathbf{v}_d \in \mathbb{Z}^2$. Moreover, for this challenge $\mathsf{Chall} = d$, no valid response can exist, as shown in the proof of Lemma 5.1. We note that $\mathsf{BadChallenge}$ may output a bit even when there is no bad challenge at all for a given $\mathbf{a}$. These "false positives" are not a problem since, in order to soundly instantiate Fiat-Shamir, we only need the somewhere CI hash function to avoid the bad challenge when it exists.

Lemma 5.1. *The above construction is a trapdoor Σ-protocol for $\mathcal{L}^{\mathrm{eq}}$ if we set* $\sigma_{\mathrm{eq}} \geq \log(2m+2) \cdot \sqrt{B_r^2 + B_\chi^2}$ *and*

$$B^* > \max(2\sigma_{\mathrm{eq}}\sqrt{2m+2} \cdot (\alpha q\sqrt{m}+1), B_r \alpha q \sqrt{m} + B_\chi).$$

(The proof is given in Appendix E.2.)

Parallel Repetitions. To achieve negligible soundness error, the protocol is repeated $\kappa = \Theta(\lambda)$ times in parallel by first computing $(\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$ before obtaining $\mathsf{Chall} = \mathsf{Chall}[1] \ldots \mathsf{Chall}[\kappa]$ and computing the response $\bar{\mathbf{z}} = (\mathbf{z}_1, \ldots, \mathbf{z}_\kappa)$, $(z_{\mu,1}, \ldots, z_{\mu,\kappa})$. We then handle $\bar{\mathbf{z}}$ as an integer vector in $\mathbb{Z}^{\kappa \cdot (m+1)}$ and reject it with probability $\theta = \min\big(1, D_{\mathbb{Z}^{2\kappa \cdot (m+1)}, \sigma_{\mathrm{eq}}}(\mathbf{z})/M \cdot D_{\mathbb{Z}^{2\kappa \cdot (m+1)}, \sigma_{\mathrm{eq}}, \mathsf{Chall} \cdot (\mathbf{1}^\kappa \otimes \mathbf{w})}(\mathbf{z})\big)$, where $M = e^{12/\log(2\kappa \cdot (m+1)) + 1/(2\log^2(2\kappa \cdot (m+1)))}$. Then, we need to slightly increase σ_{eq} and set $\sigma_{\mathrm{eq}} \geq \log(2\kappa(m+1)) \cdot \sqrt{\kappa(B_r^2 + B_\chi^2)}$.

Acknowledgements

Part of this research was funded by the French ANR ALAMBIC project (ANR-16-CE39-0006). This work was also supported in part by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701). 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.

References

1. M. Abdalla, F. Bourse, A. De Caro, and D. Pointcheval. Simple functional encryption schemes for inner products. In *PKC*, 2015.
2. M. Abe and S. Fehr. Perfect NIZK with adaptive soundness. In *TCC*, 2007.
3. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Eurocrypt*, 2010.
4. S. Agrawal, B. Libert, and D. Stehlé. Fully secure functional encryption for inner products from standard assumptions. In *Crypto*, 2016.
5. J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In *PKC*, 2012.
6. B. Applebaum. Key-dependent message security: Generic amplification and completeness theorems. *J. of Cryptology*, 27(3), 2013.
7. B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In *Crypto*, 2009.
8. G. Asharov, A. Jain, A. López-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Eurocrypt*, 2012.
9. G. Asharov, A. Jain, and D. Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. Cryptology ePrint Archive: Report 2011/613, 2012.

10. A. Banerjee, C. Peikert, and A. Rosen. Pseudorandom functions and lattices. In *Eurocrypt*, 2012.
11. B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai. Bounded key-dependent message security. In *Eurocrypt*, 2010.
12. D. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. In *STOC*, 1986.
13. M. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Eurocrypt*, 2009.
14. M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *ACM-CCS*, 1993.
15. N. Bitansky, D. Dachman-Soled, S. Garg, A. Jain, T. Tauman Kalai, A. Lopez-Alt, and D. Wichs. Why "Fiat-Shamir for proofs" lacks a proof. In *TCC*, 2013.
16. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *SAC*, 2002.
17. D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Crypto*, 2004.
18. D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In *Eurocrypt*, 2014.
19. D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from Decision Diffie-Hellman. In *Crypto*, 2008.
20. D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan. Key-homomorphic PRFs and their applications. In *Crypto*, 2013.
21. X. Boyen and Q. Li. Towards tightly secure lattice short signature and ID-based encryption. In *Asiacrypt*, 2016.
22. X. Boyen and Q. Li. Almost tight multi-instance multi-ciphertext identity-based encryption on lattices. In *ACNS*, 2018.
23. E. Boyle, G. Segev, and D. Wichs. Fully leakage-resilient signatures. In *Eurocrypt*, 2011.
24. Z. Brakerski and S. Goldwasser. Circular and leakage resilient public-key encryption under subgroup indistinguishability (or: Quadratic residuosity strikes back). In *Crypto*, 2010.
25. Z. Brakerski, S. Goldwasser, and Y. Tauman Kalai. Black-box circular-secure encryption beyond affine functions. In *TCC*, 2011.
26. Z. Brakerski and V. Vaikuntanathan. Lattice-based FHE as secure as PKE. In *ITCS*, 2014.
27. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Eurocrypt*, 2001.
28. J. Camensich, N. Chandran, and V. Shoup. A public key encryption scheme secure against key dependent chosen plaintext and adaptive chosen ciphertext attacks. In *Eurocrypt*, 2009.
29. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. Rothblum, and R. Rothblum. Fiat-Shamir from simpler assumptions. Cryptology ePrint Archive: Report 2018/1004.
30. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. Rothblum, R. Rothblum, and D. Wichs. Fiat-Shamir: From practice to theory. In *STOC*, 2019.
31. R. Canetti, Y. Chen, L. Reyzin, and R. Rothblum. Fiat-Shamir and correlation intractability from strong KDM-secure encryption. In *Eurocrypt*, 2018.
32. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisted. *J. of the ACM*, 51(4), 2004.

33. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Eurocrypt*, 2004.
34. R. Canetti, A. Lombardi, and D. Wichs. Fiat-Shamir: From Practice to Theory, Part II (NIZK and Correlation Intractability from Circular-Secure FHE). Cryptology ePrint Archive: Report 2018/1248.
35. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. Journal of Cryptology, 25(4), 2010.
36. M. Chase and A. Lysyanskaya. Simulatable VRFs with applications to multi-theorem NIZK. In *Crypto*, 2007.
37. A. Choudhuri, P. Hubacek, K. C., K. Pietrzak, A. Rosen, and G. Rothblum. Finding a Nash equilibrium is no easier than breaking Fiat-Shamir. In *STOC*, 2019.
38. M. Ciampi, R. Parisella, and D. Ventury. On adaptive security of delayed-input sigma protocols and Fiat-Shamir NIZKs. In *SCN*, 2020.
39. G. Couteau and D. Hofheinz. Designated-verifier pseudorandom generators, and their applications. In *Eurocrypt*, 2019.
40. R. Cramer. Modular design of secure, yet practical cryptographic protocols. PhD thesis, University of Amsterdam, 1996.
41. R. Cramer, I. Damgård, and B. Schoenmaekers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Crypto*, 1994.
42. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Eurocrypt*, 2002.
43. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Eurocrypt*, 2000.
44. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero-knowledge. In *Crypto*, 2001.
45. A. De Santis and M. Yung. Cryptograpic applications of the non-interactive metaproof and many-prover systems. In *Crypto*, 1990.
46. N. Döttling. Low-noise LPN: KDM secure public key encryption and sample amplification. In *PKC*, 2015.
47. U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero-knowledge under general assumptions. *SIAM J. of Computing*, 29(1), 1999.
48. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto*, 1986.
49. P.-A. Fouque and D. Pointcheval. Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. In *Asiacrypt*, 2001.
50. E. Freire, D. Hofheinz, K. Paterson, and C. Striecks. Programmable hash functions in the multilinear setting. In *Crypto*, 2013.
51. C. Gentry, J. Groth, Y. Ishai, C. Peikert, A. Sahai, and A. Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *J. of Cryptology*, 28(4), 2015.
52. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
53. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Crypto*, 2013.
54. O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *J. of ACM*, volume 33, 1986.
55. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 1989.
56. S. Goldwasser and Y. Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *FOCS*, 2003.

57. J. Groth, R. Ostrovsky, and A. Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 2012.
58. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt*, 2008.
59. S. Halevi, S. Myers, and C. Rackoff. On seed-incompressible functions. In *TCC*, 2008.
60. S. Han, S. Liu, and L. Lyu. Efficient KDM-CCA secure public-key encryption for polynomial functions. In *Asiacrypt*, 2016.
61. K. Hara, K. F., T. Matsuda, G. Hanaoka, and K. Tanaka. Simulation-based receiver selective opening cca secure pke from standard computational assumptions. In *SCN*, 2018.
62. C. Hazay and M. Venkitasubramaniam:. On the power of secure two-party computation. In *Crypto*, 2016.
63. D. Hofheinz. All-but-many lossy trapdoor functions. In *Eurocrypt*, 2012.
64. D. Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In *Eurocrypt*, 2013.
65. D. Hofheinz and T. Jager. Tightly secure signatures and public-key encryption. In *Crypto*, 2012.
66. J. Holmgren and A. Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In *FOCS*, 2018.
67. Y. Ishai, E. Kushilevitz, R. Ostrovksy, and A. Sahai. Zero-knowledge from secure multiparty computation. In *STOC*, 2007.
68. T. Jager. Verifiable random functions from weaker assumptions. In *TCC*, 2015.
69. C. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *Asiacrypt*, 2013.
70. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, 2006.
71. S. Kim and D.-J. Wu. Multi-theorem preprocessing NIZKs from lattices. In *Crypto*, 2018.
72. F. Kitagawa and T. Matsuda. CPA-to-CCA transformation for KDM security. In *TCC*, 2019.
73. F. Kitagawa, T. Matsuda, and K. Tanaka. Simple and efficient KDM-CCA secure public key encryption. In *Asiacrypt*, 2019.
74. F. Kitagawa and K. Tanaka. A framework for achieving KDM-CCA secure public-key encryption. In *Asiacrypt*, 2018.
75. Q. Lai, F. Liu, and Z. Wang. Almost tight security in lattices with polynomial moduli - PRF, IBE, all-but-many LTF, and more. In *PKC*, 2020.
76. B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption from homomorphic signatures. In *Eurocrypt*, 2014.
77. B. Libert, A. Sakzad, D. Stehlé, and R. Steinfeld. All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In *Crypto*, 2017.
78. A. Lombardi, W. Quach, R. Rothblum, D. Wichs, and D. Wu. New constructions of reusable designated-verifier NIZKs. In *Crypto*, 2019.
79. X. Lu, B. Li, and D. Jia. KDM-CCA security from RKA secure authenticated encryption. In *Eurocrypt*, 2015.
80. V. Lyubashevsky. Lattice-Based Identification Schemes Secure Under Active Attacks. In *PKC*, 2008.
81. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Asiacrypt*, 2009.

82. V. Lyubashevsky. Lattice signatures without trapdoors. In *Eurocrypt*, 2012.
83. P. MacKenzie and K. Yang. On simulation-sound trapdoor commitments. In *Eurocrypt*, 2004.
84. T. Malkin, I. Teranishi, and M. Yung. Efficient circuit-size independent public key encryption with KDM security. In *Eurocrypt*, 2012.
85. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012.
86. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, 1990.
87. C. Peikert and S. Shiehian. Non-interactive zero knowledge for NP from (plain) Learning With Errors. In *Crypto*, 2019.
88. C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Crypto*, 2008.
89. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Jo. of Cryptology*, 13(3), 2000.
90. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.
91. A. Sahai. Non-malleable non-interactive zero-knowledge and adaptive chosen-ciphertext security. In *FOCS*, 1999.
92. C.-P. Schnorr. Efficient identification and signatures for smart cards. In *Crypto*, 1989.
93. Y. Tauman Kalai, G. Rothblum, and R. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In *Crypto*, 2017.
94. S. Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In *Crypto*, 2017.

A Additional Definitions

A.1 Lossy Encryption With Efficient Opening

We recall the notion of lossy encryption with efficient opening as considered by Bellare *et al.* [13].

Definition A.1. *A lossy PKE scheme with efficient opening is a tuple of PPT algorithms* $(\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener})$ *such that:*

Public parameters: $\mathsf{Par\text{-}Gen}$ *inputs a security parameter* $\lambda \in \mathbb{N}$ *and outputs public parameters* Γ*, which specify a message space* MsgSp*, a ciphertext space* CtSp *and a randomness space* R^{LPKE}*.*

Key generation: *On input of public parameters* Γ*,* Keygen *outputs an injective public key* $pk \in \mathcal{PK}$ *and a secret key* $sk \in \mathcal{SK}$*.*

Lossy Key generation: *On input of public parameters* Γ*,* $\mathsf{LKeygen}$ *outputs a lossy public key* $pk \in \mathcal{PK}$ *and a lossy secret key* $sk \in \mathcal{SK}$*.*

Decryption under injective keys: *For any* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$*, any injective key pair* $(pk, sk) \leftarrow \mathsf{Keygen}(\Gamma)$*, and any message* $\mathsf{Msg} \in \mathsf{MsgSp}$*, we have*

$$\Pr\left[\exists r \in R^{\mathsf{LPKE}} : \mathsf{Decrypt}\big(sk, \mathsf{Encrypt}(pk, \mathsf{Msg}; r)\big) \neq \mathsf{Msg}\right] < \nu(\lambda) \ ,$$

for some negligible function $\nu(\lambda)$*, where* $(pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda)$ *and the probability is taken over the randomness of* Keygen*.*

Indistinguishability: *The distributions* $D_{\mathrm{inj}} = \{pk \mid (pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda)\}$ *and* $D_{\mathrm{loss}} = \{pk \mid (pk, sk) \leftarrow \mathsf{LKeygen}(1^\lambda)\}$ *are indistinguishable. For any PPT adversary, we have* $\mathbf{Adv}^{\mathsf{indist\text{-}LPKE}}(\lambda) \leq \mathsf{negl}(\lambda)$*, where*

$$\mathbf{Adv}^{\mathsf{indist\text{-}LPKE}}(\lambda) := |\Pr[(pk, sk) \hookleftarrow D_{\mathrm{inj}} : \mathcal{A}(pk) = 1] \\ - \Pr[(pk, sk) \hookleftarrow D_{\mathrm{loss}} : \mathcal{A}(pk) = 1]| \ .$$

Lossiness under lossy keys: *For any* $(pk, sk) \leftarrow \mathsf{LKeygen}(1^\lambda)$ *and any messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, the distributions* $\{C \mid C \leftarrow \mathsf{Encrypt}(pk, \mathsf{Msg}_0)\}$ *and* $\{C \mid C \leftarrow \mathsf{Encrypt}(pk, \mathsf{Msg}_1)\}$ *are statistically close.*

Efficient opening under lossy keys: *Let* D_R *the distribution over* R^{LPKE} *from which the random coins of* $\mathsf{Encrypt}$ *are sampled. For any* $\mathsf{Msg} \in \mathsf{MsgSp}$ *and ciphertext* C*, let* $D_{PK,\mathsf{Msg},C}$ *denote the probability distribution on* R^{LPKE} *with support* $S_{PK,\mathsf{Msg},C} = \{\overline{r} \in R^{\mathsf{LPKE}} \mid \mathsf{Encrypt}(pk, \mathsf{Msg}, \overline{r}) = C\}$*, and such that, for each* $\overline{r} \in S_{PK,\mathsf{Msg},C}$*, we have*

$$D_{PK,\mathsf{Msg},C}(\overline{r}) = \Pr_{r' \hookleftarrow D_R}[r' = \overline{r} \mid \mathsf{Encrypt}(pk, \mathsf{Msg}, r') = C] \ .$$

There is a PPT sampling algorithm Opener *such that, for any keys* $(pk, sk) \leftarrow \mathsf{LKeygen}(1^\lambda)$*, any randomness* $r \hookleftarrow D_R$*, and any* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, takes as inputs* $C = \mathsf{Encrypt}(pk, \mathsf{Msg}_0, r)$ *and* Msg_1 *and outputs an independent sample* $\overline{r}$ *from a distribution statistically close to* $D_{PK,\mathsf{Msg}_1,C}$*.*

A.2 Non-Interactive Zero-Knowledge and Simulation-Sound Proofs

We recall the definitions of NIZK proofs. Since it is sufficient for our applications, we allow the common reference string to be generated as a function of the language $\mathcal{L}$ (analogously to quasi-adaptive NIZK proofs [69]). We actually give a slightly different definition than the standard ones, defining NIZK for gap languages. That is, a language is defined by a pair of language $\mathcal{L}_{\mathsf{zk}} \subseteq \mathcal{L}_{\mathsf{sound}}$, and completeness is guaranteed for statements in $\mathcal{L}_{\mathsf{zk}}$ while soundness is guaranteed for statement outside $\mathcal{L}_{\mathsf{sound}}$. This is sufficient for our purpose.

In addition, we consider NIZK argument systems where each argument comes with a label lbl taken as input by both the prover and the verifier. Labels will only be useful when we consider simulation-soundness, which is necessary in our CCA-anonymous group signature (the CPA-variant only uses non-labeled argument systems).

Definition A.2. *A non-interactive zero-knowledge (NIZK) argument system* Π *for a language* $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$ *associated to two NP relations* $(R_{\mathsf{zk}}, R_{\mathsf{sound}})$ *consists of four PPT algorithms* $(\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ *with the following syntax:*

- $\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$ *takes as input a security parameter* λ *and outputs public parameters* par.
- $\mathsf{Gen}_{\mathcal{L}}(1^\lambda, \mathcal{L}, \tau_{\mathcal{L}})$ *takes as input a security parameter* λ*, the description of* $\mathcal{L}$ *which specifies a statement length* N*, and a membership testing trapdoor* $\tau_{\mathcal{L}}$ *for* $\mathcal{L}$*. It outputs the language-dependent part* $\mathsf{crs}_{\mathcal{L}}$ *of the common reference string* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$*.*

- $\mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$ *is a proving algorithm taking as input the common reference string* crs*, a statement* $x \in \{0,1\}^N$*, a witness* w *such that* $(x, w) \in R_{\mathsf{zk}}$ *and a label* lbl*. It outputs a proof* π*.*
- $\mathsf{V}(\mathsf{crs}, x, \pi, \mathsf{lbl})$ *is a verification algorithm taking as input a common reference string* crs*, a statement* $x \in \{0,1\}^N$*, and a proof* π*. It outputs* 1 *or* 0*.*

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

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

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

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

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

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

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

  Here, $\mathsf{P}(\mathsf{crs}, \cdot, \cdot)$ *is an oracle that outputs* $\perp$ *on input* (x, w, lbl) *such that* $(x, w) \notin R_{\mathsf{zk}}$*. Otherwise, it outputs* $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$*.* $\mathcal{O}(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ *is an oracle that outputs* $\perp$ *on input of* (x, w, lbl) *such that* $(x, w) \notin R_{\mathsf{zk}}$ *and outputs a simulated argument* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, x, \mathsf{lbl})$ *on input of* (x, w, lbl) *such that* $(x, w) \in R_{\mathsf{zk}}$*. Note that this simulated proof* π *is generated independently of the witness* w *provided as input.*[8]

The notion of soundness captured by Definition A.2 is *non-adaptive* in that the statement is given as input to the dishonest prover and chosen independently of the common reference string. The stronger notion of *adaptive soundness* allows the target statement to be chosen by the adversary after having received the common reference string. It is known (see, e.g., [2]) that perfect or statistical NIZK arguments cannot provide adaptive soundness under falsifiable assumptions. The reason lies in the impossibility of recognizing when the adversary wins and outputs a proof for a false statement. One way to bypass the impossibility results is to consider *trapdoor languages*, where a trapdoor can be used to recognize false statements. In the context of CCA security, we will consider a notion of adaptive soundness for trapdoor languages.

Definition A.2 captures a notion of multi-theorem zero-knowledge, which allows the adversary to obtain proofs for multiple statements. Feige *et al.* [47] gave a generic transformation of a multi-theorem NIZK argument system from a

[8] In particular, Sim_1 can be run on any statement x, even $x \notin \mathcal{L}_{\mathsf{sound}}$. This is central in the definition of unbounded simulation soundness (Definition A.3).

single-theorem one (where the adversary can only invoke the oracle once).

Simulation-Soundness. We now recall the definition of simulation-soundness introduced in [91], which informally captures the adversary's inability to create a new proof for a false statement $x^\star$ even after having seen simulated proofs for possibly false statements $\{x_i\}_i$ of its choice.

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

Definition A.3 ([91,44]). *Let a language $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$. A NIZK argument system for $\mathcal{L}$ provides* **unbounded simulation soundness** *if no PPT adversary has noticeable advantage in this game.*

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

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

B Simpler Construction of Multi-Theorem NIZK Arguments

In order to compile trapdoor Σ-protocols into multi-theorem NIZK proof systems for the same languages. To this end, we use the following building blocks.

- A trapdoor Σ-protocol $\Pi' = (\mathsf{Gen}'_{\mathsf{par}}, \mathsf{Gen}'_{\mathcal{L}}, \mathsf{P}', \mathsf{V}')$ with challenge space $\mathcal{C}$, for a language $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$.
- A correlation intractable hash family $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ with output length $\kappa \in \mathsf{poly}(\lambda)$ for the class $\mathcal{R}_{\mathsf{CI}}$ of relations that are efficiently searchable within time T.
- A lossy PKE scheme $\Pi^{\mathsf{LPKE}} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener})$ with public (resp. secret) key space $\mathcal{PK}$ (resp. $\mathcal{SK}$), as defined in Appendix A.1. We assume that the decryption algorithm $\mathsf{Decrypt}$ is computable

within time T. We denote the message (resp. ciphertext) space by MsgSp (resp. CtSp) and the randomness space by R^{LPKE}. Let also D_R^{LPKE} denote the distribution from which the random coins of $\mathsf{Encrypt}$ are sampled.

Our construction $\Pi = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ goes as follows.

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

$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$: Given public parameters par and a language $\mathcal{L}$, generate the common reference string as follows.

1. Generate a common reference string $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ for the trapdoor Σ-protocol Π'.
2. Generate a key $k \leftarrow \mathsf{Gen}(1^\lambda)$ for the correlation intractable hash function.
3. Generate public parameters $\Gamma \hookleftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$ for the lossy PKE scheme Π^{LPKE}. Then, generate lossy keys $(pk, sk) \leftarrow \mathsf{LKeygen}(\Gamma)$.

Output the language-dependent $\mathsf{crs}_{\mathcal{L}} := \big(\mathsf{crs}'_{\mathcal{L}}, k\big)$. The global common reference string consists of

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

$\mathsf{P}(\mathsf{crs}, \mathbf{x}, \mathbf{w})$: To prove a statement $\mathbf{x}$ using a witness $\mathbf{w} \in R_{\mathsf{zk}}(x)$,

1. Compute $\big(\mathbf{a}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa), st\big) \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathbf{w})$ via κ invocations of the prover algorithm of Π'. For each $i \in [\kappa]$, compute $\mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathbf{a}'_i; \mathbf{r}_i)$ using randomness $\mathbf{r}_i \hookleftarrow D_R^{\mathsf{LPKE}}$ sampled from the distribution D_R^{LPKE} over R^{LPKE}. Define $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$ and $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_\kappa)$.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (\mathbf{x}, \mathbf{a})) \in \{0,1\}^\kappa$.
3. Compute $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_\kappa) = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathbf{w}, \mathbf{a}, \mathsf{Chall}, st)$ via κ parallel executions of the prover of Π'.
4. Output the proof $\boldsymbol{\pi} = \big(\mathbf{a}', \mathbf{z}, \mathbf{r}\big)$.

$\mathsf{V}(\mathsf{crs}, \mathbf{x}, \boldsymbol{\pi})$: Given a statement $\mathbf{x}$ and a candidate proof $\boldsymbol{\pi} = \big(\mathbf{a}', \mathbf{z}, \mathbf{r}\big)$, do the following. For each $i \in [\kappa]$, compute $\mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathbf{a}'_i; \mathbf{r}_i)$. Compute the challenge $\mathsf{Chall} = \mathsf{Hash}\big(k, (\mathbf{x}, (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa))\big)$. If $\mathsf{V}'(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, (\mathbf{a}'_i, \mathsf{Chall}[i], \mathbf{z}_i)) = 1$ for each $i \in [\kappa]$, return 1. Otherwise, return 0.

The construction can be applied to any trapdoor Σ-protocol. In the modified FLS protocol of [34] for the graph Hamiltonicity language, we note that the first prover message is already a lossy encryption so that we do not need to super-encrypt it using a second layer of lossy encryption. Instead, we can use the simulator of Theorem 3.3 and simulate proofs by equivocating lossy encryptions.

As a candidate lossy encryption scheme with efficient opening, we can use the variant of Regev's cryptosystem suggested in [52, Section 8.1] where encryption coins are sampled from a Gaussian distribution: This construction has random lossy public keys and supports efficient opening by embedding a lattice trapdoor in (statistically uniform) lossy public keys. Since the CI hash function of [87] can support uniformly random hashing keys, by applying [87, Theorem 5.3], we then obtain a multi-theorem NIZK argument for all NP in the common random string model (assuming that the CRS of the trapdoor Σ-protocol is itself random), which provides statistical zero-knowledge and non-adaptive soundness under the LWE assumption. In contrast, applying the FLS transformation [47] to [87] requires a common reference string with a special structure since one of the CRS components has to be in the image of a pseudorandom generator in order to achieve statistical zero-knowledge.

Theorem B.1. *The above argument system is multi-theorem zero-knowledge assuming that the trapdoor Σ-protocol Π' is special zero-knowledge.*

Proof. We describe a zero-knowledge simulator $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ that uses the lossy secret key $\tau_{\mathrm{zk}} = sk$ of Π^{LPKE} to generate proofs $\boldsymbol{\pi} = (\mathbf{a}', \mathbf{z}, \mathbf{r})$ without using the witnesses. Namely, on input of $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$, Sim_0 generates $\mathsf{crs}_{\mathcal{L}}$ by proceeding identically to $\mathsf{Gen}_{\mathcal{L}}$ while Sim_1 is described hereunder.

$\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \mathbf{x})$: Given a statement $\mathbf{x}$ and the simulation trapdoor $\tau_{\mathrm{zk}} = sk$, algorithm Sim_1 proceeds as follows.

1. Let $\mathbf{0}^{|\mathbf{a}'_i|}$ the all-zeroes string of the same length as the first prover message of Π'. For each $i \in [\kappa]$, compute
$$\mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathbf{0}^{|\mathbf{a}'_i|}; \mathbf{r}_{i,0})$$
using random coins $\mathbf{r}_{i,0} \hookleftarrow D_R^{\mathsf{LPKE}}$ sampled from the distribution D_R^{LPKE}.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (\mathbf{x}, \mathbf{a}))$, where $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$.
3. For each $i \in [\kappa]$, run the ZK simulator $(\mathbf{a}'_i, \mathbf{z}'_i) \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathsf{Chall}[i])$ of Π' so as to obtain a simulated transcript $(\mathbf{a}'_i, \mathsf{Chall}[i], \mathbf{z}'_i)$ of Π' for the challenge $\mathsf{Chall}[i] \in \{0, 1\}$.
4. Using the lossy secret key sk of Π^{LPKE}, compute random coins $\mathbf{r}_i \leftarrow \mathsf{Opener}(pk, sk, \mathbf{a}_i, \mathbf{a}'_i)$ that explain $\mathbf{a}_i$ as an encryption of $\mathbf{a}'_i$. Then, output the proof $\boldsymbol{\pi} = (\mathbf{a}', \mathbf{z}, \mathbf{r}) = \big((\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa), (\mathbf{z}_1, \ldots, \mathbf{z}_\kappa), (\mathbf{r}_1, \ldots, \mathbf{r}_\kappa)\big)$.

We now prove that the simulation is indistinguishable from proofs generated by the real prover. The special ZK property of Π' implies that its simulator produces $(\mathbf{a}'_i, \mathbf{z}'_i) \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathsf{Chall}[i])$ such that $(\mathbf{a}'_i, \mathsf{Chall}[i], \mathbf{z}'_i)$ is indistinguishable from a real transcript with challenge $\mathsf{Chall}[i]$. This implies that the distribution

$$\begin{aligned}\{(\mathbf{a}_i, \mathbf{a}'_i, \mathbf{r}_i, \mathbf{z}'_i) \mid \mathbf{r}_{i,0} \hookleftarrow D_R^{\mathsf{LPKE}},\ \mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathbf{0}^{|\mathbf{a}'_i|}; \mathbf{r}_{i,0}),\\ (\mathbf{a}'_i, \mathbf{z}'_i) \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathsf{Chall}[i]),\ \mathbf{r}_i \leftarrow \mathsf{Opener}(pk, sk, \mathbf{a}_i, \mathbf{a}'_i)\},\end{aligned} \tag{11}$$

is indistinguishable from

$$\begin{aligned}\{(\mathbf{a}_i, \mathbf{a}'_i, \mathbf{r}_i, \mathbf{z}'_i) \mid\ & \mathbf{r}_{i,0} \hookleftarrow D_R^{\mathsf{LPKE}},\ \mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathbf{0}^{|\mathbf{a}'_i|}; \mathbf{r}_{i,0}),\\ & (\mathbf{a}'_i, st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathbf{w}),\ \mathbf{z}'_i = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathbf{w}, \mathbf{a}'_i, \mathsf{Chall}[i], st'),\\ & \mathbf{r}_i \leftarrow \mathsf{Opener}(pk, sk, \mathbf{a}_i, \mathbf{a}'_i)\}.\end{aligned}$$

By the property of efficient opening under lossy keys, we know that the above is statistically indistinguishable from

$$\begin{aligned}\{(\mathbf{a}_i, \mathbf{a}'_i, \mathbf{r}_i, \mathbf{z}'_i) \mid (\mathbf{a}'_i, st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathbf{w}),\ \mathbf{r}_i \hookleftarrow D_R^{\mathsf{LPKE}}\\ \mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathbf{a}'_i; \mathbf{r}_i),\ \mathbf{z}'_i = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathbf{w}, \mathbf{a}'_i, \mathsf{Chall}[i], st')\}.\end{aligned} \tag{12}$$

The distribution (11) corresponds to proofs generated by Sim_1 the simulator while (12) is identical to the distribution generated by the real prover. Simulated proofs are thus statistically indistinguishable from real proofs if the simulator of Π' is statistically ZK. □

Theorem B.2. *The above argument system provides non-adaptive soundness assuming that: (i) Π^{LPKE} is a lossy encryption scheme; (ii) The hash family $\mathcal{H}$ is somewhere correlation-intractable for all relations that are searchable within time T, where T denotes the maximal running time of algorithm* $\mathsf{BadChallenge}(\cdot,\cdot,\cdot,\cdot)$.

Proof. To prove the result, we consider a sequence of games. For each i, we define a Boolean variable $W_i \in \{\mathtt{true}, \mathtt{false}\}$ where $W_0 = \mathtt{true}$ if and only if the adversary wins in Game_0.

Game_0: This is the real soundness experiment. Namely, the challenger gives $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, \Gamma, pk)$ to the adversary $\mathcal{A}$ and a statement $\mathbf{x}^\star \notin \mathcal{L}_{\mathsf{sound}}$. The variable W_0 is thus set to $W_0 = \mathtt{true}$ if the latter event occurs. By the definition of $\mathcal{A}$'s advantage, we have $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{soundness}}(\lambda) = \Pr[W_0]$.

Game_1: We change the distribution of $\mathsf{crs} = \big(\mathsf{crs}', k\big)$ by leveraging the CRS indistinguishability property of the trapdoor Σ-protocol Π'. Namely, we use the $\mathsf{TrapGen}'$ algorithm of Definition 2.9 to generate $\mathsf{crs}'_{\mathcal{L}}$ as $(\mathsf{crs}'_{\mathcal{L}}, \tau_\Sigma) \leftarrow \mathsf{TrapGen}'(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$ instead of $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$. We immediately have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}_{\mathcal{A}}^{\mathrm{indist}\text{-}\Sigma}(\lambda)$.

We note that the trapdoor τ_Σ produced by $\mathsf{TrapGen}'$ in Game_1 can be used in subsequent games to compute the $\mathsf{BadChallenge}$ function of the trapdoor Σ-protocol Π'.

Game_2: We modify the distribution of crs. Namely, at step 3 of $\mathsf{Gen}_{\mathcal{L}}$, we generate the keys for Π^{LPKE} as injective keys $(pk, sk) \leftarrow \mathsf{Keygen}(\Gamma)$ instead of lossy keys $(pk, sk) \leftarrow \mathsf{LKeygen}(\Gamma)$. The indistinguishability property of Π^{LPKE} guarantees $\Pr[W_2]$ is within negligible distance from $\Pr[W_1]$. We can easily build a distinguisher $\mathcal{B}$ against Π^{LPKE} such that

$$|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{indist\text{-}LPKE}}(\lambda).$$

Game_3: We now use the decryption algorithm of Π^{LPKE}. We consider the relation R_{bad} defined by

$$\begin{aligned}\big((\mathbf{x}, (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)), \mathsf{Chall}\big) \in R_{\mathsf{bad}} \quad &\Leftrightarrow \quad \mathbf{x} \notin \mathcal{L} \quad \wedge \\ \mathsf{Chall}[i] = \mathsf{BadChallenge}\big(\tau_\Sigma, &\mathsf{crs}'_{\mathcal{L}}, \mathbf{x}, \mathsf{Decrypt}(sk, \mathbf{a}_i)\big) \quad \forall i \in [\kappa].\end{aligned} \tag{13}$$

We now set $W_3 = \mathsf{false}$ if

$$\begin{aligned}\mathsf{Hash}(k, (\mathbf{x}^\star, \mathbf{a}^\star)) \neq \big(&\mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, \mathbf{x}^\star, \mathsf{Decrypt}(sk, \mathbf{a}_1^\star)\big), \\ &\ldots, \ \mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, \mathbf{x}^\star, \mathsf{Decrypt}(sk, \mathbf{a}_\kappa^\star))\big)\end{aligned} \tag{14}$$

and $W_3 = W_2$ otherwise. By the statistical decryption correctness of the lossy PKE scheme under injective keys, we have $|\Pr[W_3] - \Pr[W_2]| \leq 2^{-\Omega(\lambda)}$ since $\boldsymbol{\pi}^\star$ cannot properly verify if $\mathbf{x}^\star \notin \mathcal{L}$ and (14) holds.

In Game_3, we claim that $\Pr[W_3] \leq \mathbf{Adv}_{\mathcal{A}}^{\mathrm{CI}}(\lambda)$. Indeed, if $\mathbf{x}^\star \notin \mathcal{L}$ and

$$\mathsf{Hash}(k, (\mathbf{x}^\star, \mathbf{a}^\star)) = \big(\mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, \mathbf{x}^\star, \mathsf{Decrypt}(sk, \mathbf{a}_1^\star)\big), \\ \ldots, \ \mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, \mathbf{x}^\star, \mathsf{Decrypt}(sk, \mathbf{a}_\kappa^\star))\big)$$

the correlation intractability of $\mathcal{H}$ is broken for the relation R_{bad} defined in (13).

Putting the above altogether, the advantage of a PPT adversary is thus smaller than

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{soundness}}(\lambda) \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{indist\text{-}LPKE}}(\lambda) + \mathbf{Adv}_{\mathcal{A}}^{\mathrm{indist}\text{-}\Sigma}(\lambda) + \mathbf{Adv}_{\mathcal{A}}^{\mathrm{CI}}(\lambda) + 2^{-\Omega(\lambda)},$$

which proves the claim. □

C Proof of Theorem 3.3

Proof. To prove the result, we describe a simulator $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ which uses the lossy secret key $\tau_{\mathrm{zk}} = sk$ of $\mathcal{R}$-LPKE to simulate transcripts $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ without using the witnesses. Namely, on input of $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$, Sim_0 generates $\mathsf{crs}_{\mathcal{L}}$ by proceeding identically to $\mathsf{Gen}_{\mathcal{L}}$ while Sim_1 is described hereunder.

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

1. Generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^\lambda)$. Let $\mathbf{0}^{|\mathbf{a}'_i|}$ the all-zeroes string of the same length as the first prover message of Π'. Compute $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$ as
$$\mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{0}^{|\mathbf{a}'_i|}; \mathbf{r}_{i,0}) \qquad \forall i \in [\kappa]$$
using random coins $\mathbf{r}_{i,0} \hookleftarrow D_R^{\mathsf{LPKE}}$ independently sampled from the distribution D_R^{LPKE}.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK})) \in \{0,1\}^\kappa$.
3. For each $i \in [\kappa]$, run the ZK simulator $(\mathbf{a}'_i, \mathbf{z}'_i) \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Chall})$ of Π' to obtain a simulated transcript $(\mathbf{a}'_i, \mathsf{Chall}[i], \mathbf{z}'_i)$ of Π' for the challenge $\mathsf{Chall}[i] \in \{0,1\}$.
4. For each $i \in [\kappa]$, using the lossy secret key sk of $\mathcal{R}$-LPKE, compute random coins $\mathbf{r}_i \leftarrow \mathsf{LOpener}(pk, sk, \mathsf{VK}, \mathbf{a}_i, \mathbf{a}'_i)$ which explain $\mathbf{a}_i$ as an encryption of $\mathbf{a}'_i$ under the tag VK. Then, define
$$\mathbf{z} = \big(\mathbf{z}' = (\mathbf{z}'_1, \ldots, \mathbf{z}'_\kappa), \mathbf{a}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa), \mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_\kappa)\big).$$
5. Generate a one-time signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, \mathbf{a}, \mathbf{z}, \mathsf{lbl}))$ and output the proof $\boldsymbol{\pi} = \big(\mathsf{VK}, (\mathbf{a}, \mathbf{z}), sig\big)$.

We now prove that the simulation is statistically indistinguishable from proofs generated by the real prover. The special zero-knowledge property of Π' implies that its simulator produces $(\mathbf{a}', \mathbf{z}') \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Chall})$ such that $(\mathbf{a}', \mathsf{Chall}, \mathbf{z}')$

is indistinguishable from a real transcript with challenge Chall. For each $i \in [\kappa]$, this implies that the distribution

$$\{(\mathbf{a}_i, \mathbf{a}'_i, \mathbf{r}_i, \mathbf{z}'_i) \mid \mathbf{r}_{i,0} \hookleftarrow D_R^{\mathsf{LPKE}},\ \mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{0}^{|\mathbf{a}'_i|}; \mathbf{r}_{i,0}) \ , \\ (\mathbf{a}'_i, \mathbf{z}'_i) \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Chall}[i]),\ \mathbf{r}_i \leftarrow \mathsf{LOpener}(pk, sk, \mathsf{VK}, \mathbf{a}_i, \mathbf{a}'_i)\} \ , \quad (15)$$

is indistinguishable from

$$\{(\mathbf{a}_i, \mathbf{a}'_i, \mathbf{r}_i, \mathbf{z}'_i) \mid \mathbf{r}_{i,0} \hookleftarrow D_R^{\mathsf{LPKE}},\ \mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{0}^{|\mathbf{a}'_i|}; \mathbf{r}_{i,0}) \ , \\ (\mathbf{a}'_i, st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w),\ \mathbf{z}' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, \mathbf{a}'_i, \mathsf{Chall}[i], st') \ , \\ \mathbf{r}_i \leftarrow \mathsf{LOpener}(pk, sk, \mathsf{VK}, \mathbf{a}_i, \mathbf{a}'_i)\} \ .$$

By the property of efficient opening under lossy keys, we know that the above is statistically indistinguishable from

$$\{(\mathbf{a}_i, \mathbf{a}'_i, \mathbf{r}_i, \mathbf{z}'_i) \mid \ (\mathbf{a}'_i, st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w),\ \mathbf{r}_i \hookleftarrow D_R^{\mathsf{LPKE}} \\ \mathbf{a}_i \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{a}'_i; \mathbf{r}_i) \ , \quad (16) \\ \mathbf{z}'_i = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, \mathbf{a}'_i, \mathsf{Chall}[i], st')\} \ .$$

Clearly, the distribution (15) corresponds to proof generated by the simulator while (16) is the distribution generated by the real prover. This implies that simulated proofs are statistically indistinguishable from real proofs if the simulator of Π' is statistically special ZK. □

D Security Proofs for the USS Arguments of Section 4.2

Theorem D.1. *The argument system of Section 4.2 is multi-theorem zero-knowledge assuming that the trapdoor Σ-protocol Π' is special zero-knowledge.*

Proof. To prove the result, we describe a simulator $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ which uses the lossy secret key $\tau_{\mathrm{zk}} = sk$ of $\mathcal{R}$-LPKE to simulate transcripts $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ without using the witnesses. Namely, on input of $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^\lambda)$, Sim_0 generates $\mathsf{crs}_{\mathcal{L}}$ by proceeding identically to $\mathsf{Gen}_{\mathcal{L}}$ while Sim_1 is described hereunder.

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

1. Generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^\lambda)$. Then, choose a random $t_c \hookleftarrow U(\{0,1\}^\lambda)$.
2. Let $\mathbf{0}^{|\mathbf{a}'_i|}$ the all-zeroes string of the same length as the first prover message of Π'. Compute $\mathbf{a} = (\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$ as

$$\mathbf{a}_i \leftarrow \mathsf{Encrypt}\big(pk, (t_c, \mathsf{VK}), \mathbf{0}^{|\mathbf{a}'_i|}; \mathbf{r}_{i,0}\big) \qquad \forall i \in [\kappa]$$

using random coins $\mathbf{r}_{i,0} \hookleftarrow D_R^{\mathsf{LPKE}}$ independently sampled from the distribution D_R^{LPKE}.
3. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}, t_c, \mathsf{VK})) \in \{0,1\}^\kappa$.

4. For each $i \in [\kappa]$, run the ZK simulator $(\mathbf{a}'_i, \mathbf{z}'_i) \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Chall})$ of Π' to obtain a simulated transcript $(\mathbf{a}'_i, \mathsf{Chall}[i], \mathbf{z}'_i)$ of Π' for the challenge $\mathsf{Chall}[i] \in \{0,1\}$.
5. For each $i \in [\kappa]$, using the lossy secret key sk of $\mathcal{R}$-LPKE, compute random coins $\mathbf{r}_i \leftarrow \mathsf{LOpener}\big(pk, sk, (t_c, \mathsf{VK}), \mathbf{a}_i, \mathbf{a}'_i\big)$ which explain $\mathbf{a}_i$ as an encryption of $\mathbf{a}'_i$ under the tag VK. Then, define
$$\mathbf{z} = \big(\mathbf{z}' = (\mathbf{z}'_1, \ldots, \mathbf{z}'_\kappa), \mathbf{a}' = (\mathbf{a}'_1, \ldots, \mathbf{a}'_\kappa), \mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_\kappa)\big).$$
6. Generate a one-time signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, t_c, \mathbf{a}, \mathbf{z}, \mathsf{lbl}))$ and output the proof $\boldsymbol{\pi} = \big((t_c, \mathsf{VK}), (\mathbf{a}, \mathbf{z}), sig\big)$.

Using the same arguments as in the proof of Theorem 3.3, we can show that simulated proofs are indistinguishable from real proofs. □

To obtain tight simulation-soundness, we need a one-time signature providing tight security in the multi-user setting in the sense of Definition D.2. Such a candidate was given under the SIS assumption in [77, Appendix C.3].

Definition D.2. *A one-time signature $(\mathcal{G}, \mathsf{S}, \mathcal{V})$ provides one-time strong unforgeability in the multi-user setting if no* ppt *adversary has non-negligible advantage in the following game.*

1. *The challenger generates N key pairs $(\mathsf{VK}_i, \mathsf{SK}_i) \leftarrow \mathcal{G}(\Gamma, 1^\lambda)$ and gives $(\mathsf{VK}_i)_{i \leq N}$ to the adversary $\mathcal{A}$.*
2. *The adversary adaptively makes up to N queries of the form (i, M_i). At each query, it obtains $sig_i \leftarrow \mathcal{S}(\mathsf{SK}_i, M_i)$. Note that a single query is allowed for each index $i \leq N$.*
3. *$\mathcal{A}$ outputs $(i^\star, M^\star, sig^\star)$ and wins if $\mathcal{V}(\mathsf{VK}_{i^\star}, M^\star, sig^\star) = 1$ and $(M^\star, sig^\star) \neq (M_{i^\star}, sig_{i^\star})$.*

The adversary's advantage $\mathbf{Adv}_{\mathcal{A}}^{N,\text{suf-}\mathsf{OTS}}(\lambda)$ its probability of success taken over all random coins.

In order to make the security loss of the reduction independent of the adversarial number Q of simulation queries, we need a PRF family with a tight security proof. The LWE-based candidates of [10,20] have (almost) tight security proofs in the sense that the reduction induces a multiplicative gap[9] that only depends on the input length, no matter how many evaluation queries the adversary is allowed to make.

In order to keep the modulus of our $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme polynomial, we can apply the result of Brakerski and Vaikuntanathan [26], which require to homomorphically evaluate the PRF via an NC1 circuit. By turning the latter into a branching program of polynomial length using Barrington's theorem [12] and exploiting the asymmetric noise growth of GSW [53], the techniques of [26] prevent the noise from growing too large and enable the use of a polynomial

[9] The proof of [20] loses a factor $\Theta(Q)$ but this gap can be avoided as shown in [77].

modulus. More specifically, we need a PRF that makes it possible to evaluate an input-dependent circuit over an FHE-encrypted key using a log-depth circuit. The key-homomorphic PRF of [20] fulfills these requirements, but requires an LWE assumption with a very large modulus.

Fortunately, a recent result of Lai *et al.* [75] makes it possible to bypass the use of Barrington's theorem by combining any tightly secure LWE-based PRF with an FHE scheme that has a decryption circuit in NC1. For this purpose, a GGM-based construction [54] can be used by exploiting the tight security result of [75, Section 3]. As a result, we can obtain tight security under the same LWE assumption as in [75, Section 3].

Theorem D.3. *The unbounded simulation-soundness of the argument system in Section 4.2 is tightly related to: (i) The strong unforgeability of* OTS*; (ii) The indistinguishability properties of the* $\mathcal{R}_{\mathsf{PRF}}$*-lossy PKE scheme; (iii) The pseudorandomness of* PRF*; (iv) The security of the correlation-intractable hash function and the indistinguishability of the trapdoor* Σ*-protocol. Concretely, the advantage of any PPT simulation-soundness adversary* $\mathcal{A}$ *as*

$$\begin{aligned}\mathbf{Adv}^{\mathrm{uss}}_{\mathcal{A}}(\lambda) \leq\ & \mathbf{Adv}^{Q,\mathrm{suf}\text{-}\mathsf{OTS}}_{\mathcal{B}_1}(\lambda) + \mathbf{Adv}^{\mathsf{indist\text{-}LPKE\text{-}1}}_{\mathcal{B}_2}(\lambda) \\ & + \mathbf{Adv}^{\mathrm{indist}\text{-}\Sigma}_{\mathcal{B}_3}(\lambda) + \mathbf{Adv}^{\mathrm{CI}}_{\mathcal{B}_4}(\lambda) \\ & + 2 \cdot \mathbf{Adv}^{\mathsf{PRF}}_{\mathcal{B}_5}(\lambda) \ + 2^{-\Omega(\lambda)},\end{aligned} \tag{17}$$

for PPT algorithms $\mathcal{B}_1, \ldots, \mathcal{B}_5$ *running in about the same time as* $\mathcal{A}$*.*

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

Game_0**:** This is the real game of Definition A.3. Namely, the challenger runs $(\mathsf{crs}, \tau_{\mathrm{zk}}) \leftarrow \mathsf{Sim}_0(\mathsf{par}, 1^N)$ and gives $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, pk, \mathsf{PRF}, \mathsf{OTS})$ to the adversary $\mathcal{A}$. At the same time, the challenger generates a trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}_{\mathsf{sound}}$ in such a way that it can efficiently test if $\mathcal{A}$'s output satisfies the winning condition (ii). The adversary is given $\tau_{\mathcal{L}}$ and is granted oracle access to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$. At each query, $\mathcal{A}$ chooses a statement $x \in \{0,1\}^N$ and a label lbl. The challenger replies by returning a simulated proof $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, x, \mathsf{lbl})$. When the adversary $\mathcal{A}$ halts, it outputs $(x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star)$, where $\boldsymbol{\pi}^\star = \big((t_c^\star, \mathsf{VK}^\star), (\mathbf{a}^\star, \mathbf{z}^\star), sig^\star\big)$. The Boolean variable W_0 is thus set to $W_0 = \mathtt{true}$ under the following three conditions: (i) $(x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star) \notin \mathcal{Q}$, where $\mathcal{Q} = \{(x_i, \boldsymbol{\pi}_i, \mathsf{lbl}_i)\}_{i=1}^Q$ denotes the set of queries to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ and the corresponding responses $\boldsymbol{\pi}_i = \big((t_c^{(i)}, \mathsf{VK}^{(i)}), (\mathbf{a}_i, \mathbf{z}_i), sig_i\big)$; (ii) $x^\star \notin \mathcal{L}_{\mathsf{sound}}$; and (iii) $V(\mathsf{crs}, x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star) = 1$. We may assume w.l.o.g. that the one-time verification keys $\{\mathsf{VK}^{(i)}\}_{i=1}^Q$ are chosen ahead of time at the beginning of the game. By definition, we have $\mathbf{Adv}^{\mathrm{uss}}_{\mathcal{A}}(\lambda) = \Pr[W_0]$.

Game_1**:** This game like Game_0 except that the challenger $\mathcal{B}$ sets $W_1 = \mathtt{false}$ if $\mathcal{A}$ outputs a fake proof $(x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star)$, where $\boldsymbol{\pi}^\star = \big((t_c^\star, \mathsf{VK}^\star), (\mathbf{a}^\star, \mathbf{z}^\star), sig^\star\big)$ contains a verification key $\mathsf{VK}^\star$ that coincides with the verification key $\mathsf{VK}^{(i)}$

contained in an output $\boldsymbol{\pi}_i = \big((t_c^{(i)}, \mathsf{VK}^{(i)}), (\mathbf{a}_i, \mathbf{z}_i), sig_i\big)$ of the simulation oracle $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$. The strong unforgeability of OTS in the Q-user setting implies that $|\Pr[W_1] - \Pr[W_0]| \leq \mathbf{Adv}_{\mathcal{B}}^{Q,\text{suf-}\mathsf{OTS}}(\lambda)$ is negligible.

Game_2: This game is like Game_1 with the difference that, at the outset of the game, the challenger chooses a random PRF secret key $K \leftarrow U(\{0,1\}^\lambda)$. At each query to the simulation oracle $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$, the challenger now computes $t_c^{(i)} = \mathsf{PRF}(K, \mathsf{VK}^{(i)}) \in \{0,1\}^\lambda$ instead of sampling $t_c^{(i)} \hookleftarrow U(\{0,1\}^\lambda)$ as before. Since the output of $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, \mathbf{0}^\lambda)$ is independent of the PRF secret key K, we have $|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{PRF}}(\lambda)$.

Game_3: This game is like Game_2 except that, when the adversary halts and outputs $x^\star$, the challenger checks if the condition

$$t_c^\star \neq \mathsf{PRF}(K, \mathsf{VK}^\star) \tag{18}$$

is satisfied, where $(t_c^\star, \mathsf{VK}^\star)$ are contained in the adversary's fake proof $\boldsymbol{\pi}^\star = \big((t_c^\star, \mathsf{VK}^\star), (\mathbf{a}^\star, \mathbf{z}^\star), sig^\star\big)$. If this condition does not hold, the challenger aborts and sets $W_3 = \mathtt{false}$. If (18) is satisfied, the challenger sets $W_3 = \mathtt{true}$ whenever $W_2 = \mathtt{true}$. Letting Fail denote the event that $\mathcal{B}$ aborts because (18) does not hold, we have $|\Pr[W_3] - \Pr[W_2]| \leq \Pr[\mathsf{Fail}]$. We rely again on the pseudorandomness of PRF to argue that there exists a negligible function $\mathbf{Adv}_{\mathcal{B}}^{\mathsf{PRF}}(\lambda)$ such that

$$|\Pr[W_3] - \Pr[W_2]| \leq \Pr[\mathsf{Fail}] \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{PRF}}(\lambda) + 2^{-\lambda}. \tag{19}$$

Game_4: This game is like Game_3 with the following changes. At step 2 of $\mathsf{Gen}_{\mathcal{L}}$, the challenger uses the PRF secret key $K \leftarrow U(\{0,1\}^\lambda)$ chosen at the outset of the game to compute $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, K)$ instead of $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, \mathbf{0}^\lambda)$. The second (statistical) indistinguishability property of the $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme ensures that changing the initialization value does not significantly affect $\mathcal{A}$' view. As the two distributions of pk are $2^{-\varOmega(\lambda)}$-close in terms of statistical distance, we have $|\Pr[W_4] - \Pr[W_3]| \leq 2^{-\varOmega(\lambda)}$.

Game_5: We modify the oracle $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ and by exploiting the efficient opening property of $\mathcal{R}$-LPKE for lossy tags (instead of lossy keys). At the i-th query $x_i \in \{0,1\}^N$, we have $\mathcal{R}_{\mathsf{PRF}}(K, (t_c^{(i)}, \mathsf{VK}^{(i)})) = 0$ (meaning that $(t_c^{(i)}, \mathsf{VK}^{(i)})$ is a lossy tag) since $t_c^{(i)} = \mathsf{PRF}(K, \mathsf{VK}^{(i)})$ by construction. This allows $\mathcal{B}$ to equivocate $\mathbf{a}$ using the trapdoor key tk instead of the lossy secret key sk of $\mathcal{R}$-LPKE. Specifically, at step 5 of Sim_1, the modified $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ oracle samples random coins as $\mathbf{r} \leftarrow \mathsf{Opener}\big(pk, tk, (t_c^{(i)}, \mathsf{VK}^{(i)}), \mathbf{a}, \mathbf{a}'\big)$ instead of running $\mathsf{LOpener}$ using sk. We define the Boolean variable W_5 like W_4. Since algorithms Opener and $\mathsf{LOpener}$ output samples from the same distribution D_R^{LPKE} over R^{LPKE}, this implies $|\Pr[W_5] - \Pr[W_4]| \leq 2^{-\varOmega(\lambda)}$.

Game_6: We modify the distribution of crs. At step 2 of Gen, we generate the keys for $\mathcal{R}$-LPKE as injective keys $(pk, sk, tk) \leftarrow \mathsf{Keygen}(\varGamma, K)$ instead of lossy keys $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\varGamma, K)$. The indistinguishability property (i) of

$\mathcal{R}$-LPKE ensures that the distributions of pairs (pk, tk) produced by Keygen and $\mathsf{LKeygen}$ are computationally indistinguishable. We can thus build a distinguisher $\mathcal{B}$ against $\mathcal{R}$-LPKE that bridges between Game_5 and Game_6, so that $|\Pr[W_6] - \Pr[W_5]| \leq \mathbf{Adv}_{\mathcal{B}}^{\text{indist-LPKE-1}}(\lambda)$.

The latter modification implies that, if the condition (18) is satisfied, the adversary's fake proof $\boldsymbol{\pi}^\star = \big((t_c^\star, \mathsf{VK}^\star), \big(\mathbf{a}^\star, \mathbf{z}^\star = (\mathbf{z}'^\star, \mathbf{a}'^\star, \mathbf{r}^\star)\big), sig^\star\big)$ involves an injective tag $(t_c^\star, \mathsf{VK}^\star)$ for which $C_{\mathsf{PRF},K}((t_c^\star, \mathsf{VK}^\star)) = 0$. Since pk is now an injective key, $\mathbf{a}^\star$ is thus an injective encryption of $\mathbf{a}'^\star$ under the injective tag $(t_c^\star, \mathsf{VK}^\star)$ using the randomness $\mathbf{r}^\star$.

Game_7: We change again the distribution of $\mathsf{crs} = \big(\mathsf{par}, (\mathsf{crs}'_{\mathcal{L}}, k), pk, \mathsf{PRF}, \mathsf{OTS}\big)$ by relying on the CRS indistinguishability property of the trapdoor Σ-protocol Π'. Namely, we use the $\mathsf{TrapGen}'$ algorithm of Definition 2.9 to generate $\mathsf{crs}'_{\mathcal{L}}$ as $(\mathsf{crs}'_{\mathcal{L}}, \tau_\Sigma) \leftarrow \mathsf{TrapGen}'(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$ instead of $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$. We immediately have $|\Pr[W_7] - \Pr[W_6]| \leq \mathbf{Adv}_{\mathcal{A}}^{\text{indist-}\Sigma}(\lambda)$.

We note that the trapdoor τ_Σ produced by $\mathsf{TrapGen}'$ in Game_7 can be used in later games to compute the $\mathsf{BadChallenge}$ function of the trapdoor Σ-protocol Π'. In order to evaluate $\mathsf{BadChallenge}$, we also use the secret key sk produced by $(pk, sk, tk) \leftarrow \mathsf{Keygen}(\Gamma, K)$ which allows decrypting $\mathbf{a}^\star$ as long as $\mathcal{R}_{\mathsf{PRF}}(K, (t_c^\star, \mathsf{VK}^\star)) = 1$.

Game_8: We now use the decryption algorithm of the $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme. If $\mathcal{B}$ did not fail, $\mathcal{A}$'s output $\boldsymbol{\pi}^\star = \big((t_c^\star, \mathsf{VK}^\star), \big(\mathbf{a}^\star, \mathbf{z}^\star = (\mathbf{z}'^\star, \mathbf{a}'^\star, \mathbf{r}^\star)\big), sig^\star\big)$ involves an injective tag $(t_c^\star, \mathsf{VK}^\star)$, so that $\mathbf{a}^\star = (\mathbf{a}_1^\star, \ldots, \mathbf{a}_\kappa^\star)$ is a statistically binding commitment to $\mathbf{a}'^\star = (\mathbf{a}_1'^\star, \ldots, \mathbf{a}_\kappa'^\star)$. With probability $2^{-\Omega(\lambda)}$, there thus exists only one $\mathbf{a}'^\star = (\mathbf{a}_1'^\star, \ldots, \mathbf{a}_\kappa'^\star)$ for which a pair $\big(\mathbf{a}'^\star, (\mathbf{r}_1^\star, \ldots, \mathbf{r}_\kappa^\star)\big)$ satisfies step 1 of the verification algorithm. We then consider the relation R_{bad} defined by

$$\begin{aligned}&\big((x, \mathbf{a}, t_c, \mathsf{VK}), \mathsf{Chall}\big) \in R_{\mathsf{bad}} \quad \Leftrightarrow \quad x \notin \mathcal{L} \ \wedge \\ &\quad \mathsf{Chall}[i] = \mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Decrypt}(sk, (t_c, \mathsf{VK}), \mathbf{a}_i)) \qquad \forall i \in [\kappa].\end{aligned} \tag{20}$$

We now set $W_8 = \mathsf{false}$ if

$$\begin{aligned}&\mathsf{Hash}(k, (x^\star, \mathbf{a}^\star, t_c^\star, \mathsf{VK}^\star)) \\ &\qquad \neq \big(\mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x^\star, \mathsf{Decrypt}(sk, (t_c^\star, \mathsf{VK}^\star), \mathbf{a}_1^\star\big), \\ &\qquad\quad \ldots,\ \mathsf{BadChallenge}\big(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x^\star, \mathsf{Decrypt}(sk, (t_c^\star, \mathsf{VK}^\star), \mathbf{a}_\kappa^\star)\big)\end{aligned} \tag{21}$$

Unless one of the $\mathbf{a}_i^\star$'s fails to decrypt to the $\mathbf{a}_i'^\star$ contained in $\boldsymbol{\pi}^\star$, the latter cannot properly verify if inequality (21) holds. The property of correct decryption under injective tags thus implies $|\Pr[W_8] - \Pr[W_7]| \leq 2^{-\Omega}$

In Game_8, we have $\Pr[W_8] \leq \mathbf{Adv}_{\mathcal{A}}^{\mathrm{CI}}(\lambda)$. Indeed, the correlation intractability property of $\mathcal{H}$ is broken for the relation R_{bad} of (20) if we have $x^\star \not\in \mathcal{L}$ and

$$\begin{aligned}
&\mathsf{Hash}(k,(x^\star,\mathbf{a}^\star,t_c^\star,\mathsf{VK}^\star)) \\
&\qquad\qquad = \big(\mathsf{BadChallenge}\big(\tau_\Sigma,\mathsf{crs}'_{\mathcal{L}},x^\star,\mathsf{Decrypt}(sk,(t_c^\star,\mathsf{VK}^\star),\mathbf{a}_1^\star),\\
&\qquad\qquad\qquad \ldots,\ \mathsf{BadChallenge}\big(\tau_\Sigma,\mathsf{crs}'_{\mathcal{L}},x^\star,\mathsf{Decrypt}(sk,(t_c^\star,\mathsf{VK}^\star),\mathbf{a}_\kappa^\star)\big).
\end{aligned}$$

When counting probabilities, we obtain the stated upper bound on the adversary's advantage. □

E Deferred Proofs for the Trapdoor Σ-Protocols of Section 5

To prove the statistical special zero-knowledge property of the involved Σ-protocols, we use the following lemma proved by Lyubashevsky [81,82].

Lemma E.1 ([82, Th. 4.6]). *Let V be a subset of $\mathbb{Z}^m$ in which all elements have norms less than T, let σ be a real number such that $\sigma = \omega(T\sqrt{\log m})$, and $h : V \to \mathbb{R}$ be a probability distribution. Then, there exists a real number M such that the distribution of the following algorithm $\mathcal{A}$:*

```
1: $\mathbf{v} \hookleftarrow h$
2: $\mathbf{z} \hookleftarrow D_{\mathbb{Z}^m,\sigma,\mathbf{v}}$
3: output $(\mathbf{z},\mathbf{v})$ with probability $\min\left(\frac{D_{\mathbb{Z}^m,\sigma}(\mathbf{z})}{M D_{\mathbb{Z}^m,\sigma,\mathbf{v}}(\mathbf{z})}, 1\right)$
```

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

```
1: $\mathbf{v} \hookleftarrow h$
2: $\mathbf{z} \hookleftarrow D_{\mathbb{Z}^m,\sigma}$
3: output $(\mathbf{z},\mathbf{v})$ with probability $1/M$.
```

Moreover, the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-\omega(\log m)}}{M}$. More concretely, if $\sigma = \alpha T$ for any positive α, then $M = e^{12/\alpha + 1/(2\alpha^2)}$, the output of $\mathcal{A}$ is within statistical distance $2^{-100}/M$ of the output of $\mathcal{F}$, and the probability that $\mathcal{A}$ outputs something is at least $\frac{1-2^{-100}}{M}$.

E.1 Trapdoor Σ-Protocol Showing that a Ciphertext is Valid

We give a simpler variant of the Σ-protocol in Section 5 which allows proving that an ACPS ciphertext is a valid encryption of some $\mu \in \mathbb{Z}_p$. It combines techniques from Σ-protocols allowing to prove an inhomogeneous SIS relation (which rely on rejection sampling as in [81]) with standard Schnorr proofs. This combination exploits the property that the order p of the message space divides the modulus $q = p^2$. The constraint $p|q$ can be removed if we just want to prove simpler statements like a ciphertext encrypting 0 (or encrypting 0 or 1 via [41]).

For an ACPS public key $PK = (\mathbf{A}, \mathbf{b})$, we define the matrix

$$\bar{\mathbf{A}} = \left[\begin{array}{c|c} \mathbf{A} & \\ \hline \mathbf{b}^\top & 1 \end{array}\right] \in \mathbb{Z}_q^{(n+1)\times(m+1)}. \tag{22}$$

We give a trapdoor Σ-protocol for the language $\mathcal{L}^\mu = (\mathcal{L}^\mu_{\mathsf{zk}}, \mathcal{L}^\mu_{\mathsf{sound}})$, where

$$\begin{aligned} \mathcal{L}^\mu_{\mathsf{zk}} := \Big\{ \mathbf{c} \in \mathbb{Z}_q^{n+1} \ \mid \exists \mathbf{r} \in \mathbb{Z}^m,\ \chi \in \mathbb{Z},\ \mu \in \mathbb{Z}_p \ :\ \|\mathbf{r}\| \leq B_r,\ |\chi| \leq B_\chi \\ \wedge \quad \mathbf{c} = \bar{\mathbf{A}} \cdot [\mathbf{r}^\top \mid \chi]^\top + \mu \cdot \big[\mathbf{0}^{n\top} \mid p\big]^\top \bmod q \ \Big\}, \end{aligned}$$

$$\mathcal{L}^\mu_{\mathsf{sound}} := \Big\{ \mathbf{c} \in \mathbb{Z}_q^{n+1} \mid \exists\ \bar{\mathbf{c}} \in \mathbb{Z}_q^n,\ v \in [-B^*, B^*],\ \mu \in \mathbb{Z}_p : \mathbf{c} = \left[\begin{array}{c} \bar{\mathbf{c}} \\ \hline \mathbf{s}^\top \cdot \bar{\mathbf{c}} + p \cdot \mu + v \end{array}\right] \Big\},$$

We note that $\mathcal{L}^\mu_{\mathsf{zk}} \subseteq \mathcal{L}^\mu_{\mathsf{sound}}$ when $B_r \alpha q \sqrt{m} + B_\chi < B^* \ll p$.

While the Σ-protocol can be turned into a trapdoor Σ-protocol using the transformation of [38], we provide a $\mathsf{BadChallenge}$ function showing that the version below is already a trapdoor Σ-protocol.

$\mathbf{Gen}_{\mathsf{par}}(1^\lambda)$: On input of a security parameter $\lambda \in \mathbb{N}$, choose moduli q, p with $q = p^2$, dimensions n, m, and error rate $\alpha > 0$ and a Gaussian parameter $\sigma_0 \geq \log(m+1) \cdot \sqrt{B_r^2 + B_\chi^2}$. Define public parameters $\mathsf{par} = \{\lambda, q, p, n, m, \alpha, \sigma_0\}$.

$\mathbf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}^\mu)$: Takes as inputs global parameters par and the description of a language $\mathcal{L}^\mu = (\mathcal{L}^\mu_{\mathsf{zk}}, \mathcal{L}^\mu_{\mathsf{sound}})$ which specifies reals $B^*, B_r, B_\chi > 0$ such that such that $B_r \alpha q \sqrt{m} + B_\chi < B^* \ll p$, as well as a matrix $\bar{\mathbf{A}}$ from the distribution (22). It defines the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{\bar{\mathbf{A}}, B^*, B_r, B_\chi\}$. The global common reference string is

$$\mathsf{crs} = \big(\{\lambda, q, p, n, m, \alpha, \sigma_0\}, \{\bar{\mathbf{A}}, B^*, B_r, B_\chi\}\big).$$

$\mathbf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$: Given public parameters par and the description of a language $\mathcal{L}^\mu$ that specifies real numbers $B^*, B_r, B_\chi > 0$ satisfying the same constraints as in $\mathsf{Gen}_{\mathcal{L}}$, a matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{(n+1)\times(m+2)}$ sampled from the distribution (22), and a membership-testing trapdoor $\tau_{\mathcal{L}} = \mathbf{s} \sim D_{\mathbb{Z}^n, \alpha q}$ for $\mathcal{L}^\mu_{\mathsf{sound}}$, output $\mathsf{crs}_{\mathcal{L}} = \{\bar{\mathbf{A}}, B^*, B_r, B_\chi\}$. The global common reference string is

$$\mathsf{crs} = \big(\{\lambda, q, p, n, m, \alpha, \sigma_0\}, \{\bar{\mathbf{A}}, B^*, B_r, B_\chi\}\big)$$

and the trapdoor $\tau_\Sigma = \mathbf{s} \in \mathbb{Z}^n$.

$\mathbf{P}\big(\mathsf{crs}, \mathbf{c}, (\mu, \mathbf{w})\big) \leftrightarrow \mathbf{V}(\mathsf{crs}, \mathbf{x})$: Given $\mathbf{c} = \bar{\mathbf{A}} \cdot [\mathbf{r}^\top \mid \chi]^\top + \mu \cdot \big[\mathbf{0}^{n\top} \mid p\big]^\top \in \mathbb{Z}_q^{n+1}$, the prover P (who has $\mathbf{w} = [\mathbf{r}^\top \mid \chi]^\top \in \mathbb{Z}^{m+1}$ and $\mu \in \mathbb{Z}_p$) and the verifier V interact as follows.

1. P samples a uniform $r_\mu \hookleftarrow U(\mathbb{Z}_p)$ and a Gaussian $\mathbf{r}_w \hookleftarrow D_{\mathbb{Z}^{m+1},\sigma_0}$. It sends V the message $\mathbf{a} = \bar{\mathbf{A}} \cdot \mathbf{r}_w + r_\mu \cdot [\mathbf{0}^{n\top} \mid p]^\top \in \mathbb{Z}_q^{n+1}$.
2. V sends a random challenge $\mathsf{Chall} \in \{0,1\}$ to P.
3. P computes $\mathbf{z} = \mathbf{r}_w + \mathsf{Chall} \cdot \mathbf{w} \ \in \mathbb{Z}^{m+1}$ and $z_\mu = r_\mu + \mathsf{Chall} \cdot \mu \mod p$. It sends $(\mathbf{z}, z_\mu)$ to V with probability $\theta = \min\left(\frac{D_{\mathbb{Z}^{m+1},\sigma_0}(\mathbf{z})}{M \cdot D_{\mathbb{Z}^{m+1},\sigma_0,\mathsf{Chall}\cdot\mathbf{w}}(\mathbf{z})}, 1\right)$, where $M = e^{12/\log(m+1)+1/(2\log^2(m+1))}$. With probability $1-\theta$, P aborts.
4. Upon receiving $(\mathbf{z}, z_\mu) \in \mathbb{Z}^{m+1} \times \mathbb{Z}_p$, V checks if $\|\mathbf{z}\| \leq \sigma_0\sqrt{m+1}$ and
$$\mathbf{a} + \mathsf{Chall} \cdot \mathbf{c} = \bar{\mathbf{A}} \cdot \mathbf{z} + z_\mu \cdot \left[\begin{array}{c}\mathbf{0}^n \\ \hline p\end{array}\right] \bmod q. \tag{23}$$

   If these conditions do not both hold, V halts and returns $\perp$.

$\mathsf{\textbf{BadChallenge}}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, \mathbf{c}, \mathbf{a}\big)$: Given $\tau_\Sigma = \mathbf{s} \in \mathbb{Z}^n$, parse the first prover message as $\mathbf{a} \in \mathbb{Z}_q^{n+1}$ and do the following.

1. Compute $a_0' = [\ -\mathbf{s}^\top \mid 1\ ] \cdot \mathbf{a} \bmod q$ and interpret it as an element of $[-(q-1)/2, (q-1)/2]$. If a_0' cannot be written as $a_0' = p \cdot \mu_0 + v_0$ for some $\mu_0 \in [-(p-1)/2, (p-1)/2]$ and $v_0 \in [-B^*/2, B^*/2]$, return $\mathsf{Chall} = 1$.
2. Compute $a_1' = [\ -\mathbf{s}^\top \mid 1\ ] \cdot (\mathbf{a} + \mathbf{c}) \bmod q$ and interpret it as an element of $[-(q-1)/2, (q-1)/2]$. If a_1' cannot be written as $a_1' = p \cdot \mu_1 + v_1$ for some $\mu_1 \in [-(p-1)/2, (p-1)/2]$ and $v_1 \in [-B^*/2, B^*/2]$, return $\mathsf{Chall} = 0$.

In any other case, return $\mathsf{Chall} =\perp$.

We note that the completeness of the protocol crucially uses the fact that p divides q to ensure that the response $z_\mu = r_\mu + \mathsf{Chall} \cdot \mu \bmod p$ satisfies (23).

Lemma E.2. *The above construction is a trapdoor Σ-protocol for* $\mathcal{L}^\mu$ *if* $\sigma_0 \geq \log(m+1) \cdot \sqrt{B_r^2 + B_\chi^2}$ *and* $B^* > \max(2\sigma_0\sqrt{m+1} \cdot (\alpha q\sqrt{m}+1), B_r\alpha q\sqrt{m} + B_\chi)$.

Proof. We first consider the special ZK property. Given a statement $\mathbf{c} \in \mathcal{L}^\mu_{\mathsf{zk}}$ and a challenge $\mathsf{Chall}^* \in \{0,1\}$, the simulator first samples $\mathbf{z}^* \hookleftarrow D_{\mathbb{Z}^{m+1},\sigma_0}$ and $z_\mu^* \hookleftarrow U(\mathbb{Z}_p)$. Then, it computes $\mathbf{a}^* = \bar{\mathbf{A}} \cdot \mathbf{z}^* + z_\mu^* \cdot \left[\begin{array}{c}\mathbf{0}^n \\ \hline p\end{array}\right] - \mathsf{Chall}^* \cdot \mathbf{c} \bmod q$. It outputs $(\mathbf{a}^*, \mathsf{Chall}^*, (\mathbf{z}^*, z_\mu^*))$ with probability $1/M$. By construction, the triple $(\mathbf{a}^*, \mathsf{Chall}^*, (\mathbf{z}^*, z_\mu^*))$ satisfies the verification conditions with high probability. We show that it is statistically indistinguishable from a real transcript. If $\mathbf{c} \in \mathcal{L}^\mu_{\mathsf{zk}}$, there exist $\mathbf{r} \in \mathbb{Z}^m$, $\chi \in \mathbb{Z}$ and $\mu \in \mathbb{Z}_p$ such that $\|\mathbf{r}\| \leq B_r$, $|\chi| \leq B_\chi$ and $\mathbf{c} = \bar{\mathbf{A}} \cdot [\mathbf{r}^\top \mid \chi]^\top + \mu \cdot \left[\mathbf{0}^{n\top} \mid p\right]^\top$. The distribution of $\mathbf{z} \in \mathbb{Z}^{m+1}$ in a real transcript is thus $D_{\mathbb{Z}^{m+1},\sigma_0,\mathsf{Chall}\cdot\mathbf{w}}$, where $\mathbf{w} = [\mathbf{r}^\top \mid \chi]^\top$. By Lemma E.1 and our choice of $\sigma_0 \geq \log(m+1) \cdot \sqrt{B_r^2 + B_\chi^2}$, the distribution of the simulated $\mathbf{z}^*$ is within statistical distance $2^{-100}/M$ from that of a real non-aborting transcript. Moreover, the component z_μ in the real protocol is uniformly random over $\mathbb{Z}_p$, so is the respective component z_μ^* in the simulation. Finally, in both the real protocol and the simulation, the statement $\mathbf{c}$, the challenge Chall and the response $(\mathbf{z}, z_\mu)$

uniquely determine $\mathbf{a}$.

Soundness can be shown using the usual arguments. Let us assume that, for a given $\mathbf{a} \in \mathbb{Z}_q^{n+1}$, there exist two valid responses $(\mathbf{z}_b, z_{\mu,b}) \in \mathbb{Z}^{m+1} \times \mathbb{Z}_p$ with $\|\mathbf{z}_b\| \leq \sigma_0 \cdot \sqrt{m+1}$ for each $b \in \{0,1\}$ and such that

$$\mathbf{a} + b \cdot \mathbf{c} = \bar{\mathbf{A}} \cdot \mathbf{z}_b + z_{\mu,b} \cdot \left[\frac{\mathbf{0}^n}{p}\right] \mod q.$$

By subtracting them, we obtain $\mathbf{c} = \bar{\mathbf{A}} \cdot \mathbf{z} + (z_{\mu,1} - z_{\mu,0} \bmod p) \cdot \left[\mathbf{0}^{n\top} \mid p\right]^\top \bmod q$, where $\mathbf{z} = \mathbf{z}_1 - \mathbf{z}_0 \in \mathbb{Z}^{m+1}$ has norm $\|\mathbf{z}\| \leq 2\sigma_0\sqrt{m+1}$. Then, we also have

$$[\ -\mathbf{s}^\top \mid 1] \cdot \mathbf{c} = [\ \mathbf{e}^\top \mid 1] \cdot \mathbf{z} + (z_{\mu,1} - z_{\mu,0} \bmod p) \cdot p,$$

which implies that $\mathbf{c} \in \mathcal{L}^{\mu}_{\mathsf{sound}}$ since $|[\ \mathbf{e}^\top \mid 1]^\top \cdot \mathbf{z}| < 2\sigma_0\sqrt{m+1} \cdot (\alpha q\sqrt{m}+1) < B^*$.

We now show that $\mathsf{BadChallenge}$ provides the correct result. First, assuming that $\mathbf{c} \notin \mathcal{L}^{\mu}_{\mathsf{sound}}$, for a given $\mathbf{a} \in \mathbb{Z}_q^{n+1}$, there cannot exist two distinct pairs $(\mu_0, v_0), (\mu_1, v_1) \in [-(p-1)/2, (p-1)/2] \times [-B^*/2, B^*/2]$ such that

$$[\ -\mathbf{s}^\top \ \mid 1\ ] \cdot (\mathbf{a} + b \cdot \mathbf{c}) \bmod q \ = \ p \cdot \mu_b + v_b \qquad \forall b \in \{0,1\}, \tag{24}$$

where the above equality holds over $\mathbb{Z}$.

Let us first assume that there exists no $(\mu_0, v_0) \in \mathbb{Z}_p \times [-B^*/2, B^*/2]$ satisfying (24) for $b = 0$. Then, there can be no valid response for $\mathsf{Chall} = 0$ since the verifier would only accept a response $(\mathbf{z}_0, z_{\mu,0}) \in \mathbb{Z}^{m+1} \times \mathbb{Z}_p$ satisfying

$$\mathbf{a} = \bar{\mathbf{A}} \cdot \mathbf{z}_0 + z_{\mu,0} \cdot \left[\frac{\mathbf{0}^n}{p}\right] \mod q$$

and $\|\mathbf{z}_0\| \leq \sigma_0\sqrt{m+1}$, which would imply

$$[\ -\mathbf{s}^\top \ \mid 1\ ] \cdot \mathbf{a} \bmod q \ = \ p \cdot z_{\mu,0} + [\mathbf{e}^\top \mid 1] \cdot \mathbf{z}_0$$

with $|[\mathbf{e}^\top \mid 1] \cdot \mathbf{z}_0| < \sigma_0\sqrt{m+1} \cdot (\alpha q\sqrt{m}+1) < B^*/2$.

Now, assuming that there exists no $(\mu_1, v_1) \in \mathbb{Z}_p \times [-B^*/2, B^*/2]$ satisfying (24) for $b = 1$, we obtain in a similar way that no valid response can exist for $\mathsf{Chall} = 1$. Since there exists at least one $b \in \{0,1\}$ such that no pair $(\mu_b, v_b) \in [-(p-1)/2, (p-1)/2] \times [-B^*/2, B^*/2]$ satisfies (24), we conclude that $\mathsf{BadChallenge}$ always rules out a $\mathsf{Chall} \in \{0,1\}$ for which no valid response exists for a given $\mathbf{a}$. □

E.2 Proof of Lemma 5.1

Proof. The special ZK property can be shown exactly as in the proof of Lemma E.2. Given a statement $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}^{\mathsf{eq}}_{\mathsf{zk}}$ and a challenge $\mathsf{Chall}^* \in \{0,1\}$, the simulator first samples $\mathbf{z}^* \hookleftarrow D_{\mathbb{Z}^{2(m+1)},\sigma_{\mathrm{eq}}}$ and $z^*_\mu \hookleftarrow U(\mathbb{Z}_p)$. Then, it computes

$$\mathbf{a}^* = \mathbf{A}_{\mathrm{eq}} \cdot \mathbf{z}^* + z^*_\mu \cdot [\mathbf{0}^{n\top} \mid p \mid \mathbf{0}^{n\top} \mid p]^\top - \mathsf{Chall} \cdot \begin{bmatrix} \mathbf{c}_0 \\ \mathbf{c}_1 \end{bmatrix} \mod q.$$

It outputs $(\mathbf{a}^*, \mathsf{Chall}^*, (\mathbf{z}^*, z_\mu^*))$ with probability $1/M$. By construction, the triple $(\mathbf{a}^*, \mathsf{Chall}^*, (\mathbf{z}^*, z_\mu^*))$ satisfies the verification conditions with high probability. We show that it is statistically indistinguishable from a real transcript. If we have $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{zk}}^{\mathrm{eq}}$, there exist $\mathbf{r}_0, \mathbf{r}_1 \in \mathbb{Z}^m$, $\chi_0, \chi_1 \in \mathbb{Z}$ and $\mu \in \mathbb{Z}_p$ such that $\|\mathbf{r}_b\| \leq B_r$, $|\chi_b| \leq B_\chi$ and $\mathbf{c}_b = \bar{\mathbf{A}}_b \cdot [\mathbf{r}_b^\top \mid \chi_b]^\top + \mu \cdot \left[\mathbf{0}^{n\top} \mid p\right]^\top \bmod q$ for $b \in \{0,1\}$. The distribution of $\mathbf{z} \in \mathbb{Z}^{2(m+1)}$ in a real transcript is thus $D_{\mathbb{Z}^{2(m+1)}, \sigma_{\mathrm{eq}}, \mathsf{Chall}\cdot\mathbf{w}}$, where $\mathbf{w} = [\mathbf{r}_0^\top \mid \chi_0 \mid \mathbf{r}_1^\top \mid \chi_1]^\top$. By Lemma E.1 and our choice of $\sigma_{\mathrm{eq}} \geq \log(2m+2) \cdot \sqrt{B_r^2 + B_\chi^2}$, the distribution of the simulated $\mathbf{z}^*$ is within statistical distance $2^{-100}/M$ from that of a real non-aborting transcript. Moreover, the component z_μ in the real protocol is uniformly random over $\mathbb{Z}_p$, so is the respective component z_μ^* in the simulation. Finally, in both the real protocol and the simulation, the statement $(\mathbf{c}_0, \mathbf{c}_1)$, the challenge Chall and the response $(\mathbf{z}, z_\mu)$ uniquely determine $\mathbf{a}$.

Soundness can be shown using the same arguments as well. Let us assume that, for a given $\mathbf{a} \in \mathbb{Z}_q^{2(n+1)}$, there exist two valid responses $(\mathbf{z}_b, z_{\mu,b}) \in \mathbb{Z}^{2m+2} \times \mathbb{Z}_p$ with $\|\mathbf{z}_b\| \leq \sigma_{\mathrm{eq}} \cdot \sqrt{2m+2}$ for each $b \in \{0,1\}$ and such that

$$\mathbf{a} + b \cdot \left[\begin{array}{c}\mathbf{c}_0 \\ \hline \mathbf{c}_1\end{array}\right] = \mathbf{A}_{\mathrm{eq}} \cdot \mathbf{z}_b + z_{\mu,b} \cdot \left[\begin{array}{c}\mathbf{0}^n \\ \hline p \\ \hline \mathbf{0}^n \\ \hline p\end{array}\right] \bmod q.$$

Subtracting them yields

$$\left[\begin{array}{c}\mathbf{c}_0 \\ \hline \mathbf{c}_1\end{array}\right] = \mathbf{A}_{\mathrm{eq}} \cdot (\mathbf{z}_1 - \mathbf{z}_0) + (z_{\mu,1} - z_{\mu,0} \bmod p) \cdot \left[\begin{array}{c}\mathbf{0}^n \\ \hline p \\ \hline \mathbf{0}^n \\ \hline p\end{array}\right] \bmod q,$$

where $\|\mathbf{z}_1 - \mathbf{z}_0\| \leq 2\sigma_{\mathrm{eq}} \cdot \sqrt{2m+2}$. Then, we also have

$$\begin{aligned}\left[\begin{array}{c|c|c|c} -\mathbf{s}_0^\top & 1 & & \\ \hline & & -\mathbf{s}_1^\top & 1\end{array}\right] \cdot \left[\begin{array}{c}\mathbf{c}_0 \\ \hline \mathbf{c}_1\end{array}\right] & \\ = \left[\begin{array}{c|c|c|c} \mathbf{e}_0^\top & 1 & & \\ \hline & & \mathbf{e}_1^\top & 1\end{array}\right] & \cdot (\mathbf{z}_1 - \mathbf{z}_0) + (z_{\mu,1} - z_{\mu,0} \bmod p) \cdot \left[\begin{array}{c}p \\ \hline p\end{array}\right],\end{aligned}$$

which implies that $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}^{\mathrm{eq}}$ since $|[\ \mathbf{e}_0^\top \mid 1 \mid \mathbf{0}^n \mid 0]^\top \cdot (\mathbf{z}_1 - \mathbf{z}_0)| < 2\sigma_{\mathrm{eq}}\sqrt{2m+2} \cdot (\alpha q\sqrt{m} + 1) < B^*$ and $|[\mathbf{0}^n \mid 0 \mid \ \mathbf{e}_1^\top \mid 1]^\top \cdot (\mathbf{z}_1 - \mathbf{z}_0)| < B^*$.

We now show that $\mathsf{BadChallenge}$ provides the correct output. First, assuming that $(\mathbf{c}_0, \mathbf{c}_1) \notin \mathcal{L}_{\mathsf{sound}}^{\mathrm{eq}}$, for a given $\mathbf{a} \in \mathbb{Z}_q^{2n+2}$, there cannot exist two distinct pairs $(\mu_0, \mathbf{v}_0), (\mu_1, \mathbf{v}_1) \in [-(p-1)/2, (p-1)/2] \times [-B^*/2, B^*/2]^2$ such that the following equality holds over $\mathbb{Z}$ for each $b \in \{0,1\}$:

$$\left[\begin{array}{c|c|c|c} -\mathbf{s}_0^\top & 1 & & \\ \hline & & -\mathbf{s}_1^\top & 1\end{array}\right] \cdot \left(\mathbf{a} + b \cdot \left[\begin{array}{c}\mathbf{c}_0 \\ \hline \mathbf{c}_1\end{array}\right]\right) \bmod q \;=\; \mathbf{v}_b + \mu_b \cdot \left[\begin{array}{c}p \\ \hline p\end{array}\right]. \qquad (25)$$

Let us first assume that there exists no $(\mu_0, \mathbf{v}_0) \in \mathbb{Z}_p \times [-B^*/2, B^*/2]^2$ satisfying (25) for $b = 0$. Then, there can be no valid response for $\mathsf{Chall} = 0$. Indeed, the verifier would only accept a response $(\mathbf{z}_0, z_{\mu,0}) \in \mathbb{Z}^{2m+2} \times \mathbb{Z}_p$ satisfying

$$\mathbf{a} = \mathbf{A}_{\mathrm{eq}} \cdot \mathbf{z}_0 + z_{\mu,0} \cdot \left[\begin{array}{c} \mathbf{0}^n \\ \hline p \\ \hline \mathbf{0}^n \\ \hline p \end{array}\right] \mod q$$

and $\|\mathbf{z}_0\| \leq \sigma_{\mathrm{eq}}\sqrt{2m+2}$, which would imply

$$\left[\begin{array}{c|c|c|c} -\mathbf{s}_0^\top & 1 & & \\ \hline & & -\mathbf{s}_1^\top & 1 \end{array}\right] \cdot \mathbf{a} \bmod q \; = \; \left[\begin{array}{c|c|c|c} \mathbf{e}_0^\top & 1 & & \\ \hline & & \mathbf{e}_1^\top & 1 \end{array}\right] \cdot \mathbf{z}_0 + z_{\mu,0} \cdot \left[\begin{array}{c} p \\ \hline p \end{array}\right]$$

with the inequalities $|[\mathbf{e}_0^\top \mid 1 \mid \mathbf{0}^n \mid 0] \cdot \mathbf{z}_0| < \sigma_{\mathrm{eq}}\sqrt{2m+2} \cdot (\alpha q\sqrt{m} + 1) < B^*/2$ and $|[\mathbf{0}^n \mid 0 \mid \mathbf{e}_1^\top \mid 1] \cdot \mathbf{z}_0| < \sigma_{\mathrm{eq}}\sqrt{2m+2} \cdot (\alpha q\sqrt{m} + 1) < B^*/2$.

Similarly, assuming that there exists no pair $(\mu_1, \mathbf{v}_1) \in \mathbb{Z}_p \times [-B^*/2, B^*/2]^2$ satisfying (25) for $b = 1$, we obtain that no valid response can exist for $\mathsf{Chall} = 1$. Since there exists $b \in \{0, 1\}$ such that no pair $(\mu_b, \mathbf{v}_b) \in [-(p-1)/2, (p-1)/2] \times [-B^*/2, B^*/2]^2$ satisfies (25), we conclude that $\mathsf{BadChallenge}$ always eliminates a $\mathsf{Chall} \in \{0, 1\}$ for which no valid response exists for a given $\mathbf{a}$. □

F (Tight) KDM-CCA2 Security from LWE

We now describe a PKE scheme with KDM-CCA2 security under the LWE assumption by applying a technique due to Chandran *et al.* [28]. It was shown [28] that applying the Naor-Yung paradigm to two schemes providing KDM-CPA security and standard IND-CPA security, respectively, gives KDM-CCA2 security as long as the underlying NIZK proof system is unbounded simulation-sound. Our scheme applies this idea to a variant of the KDM-CPA system of Applebaum *et al.* [7]. In order to apply Naor-Yung, we exploit the special shape of the modulus $q = p^2$ to prove that two ACPS ciphertexts encrypt the same plaintext $\mu \in \mathbb{Z}_p$.

By applying the trapdoor Σ-protocol described in Appendix G.2, it is also possible to prove plaintext equalities in the KDM-CPA scheme of Alperin-Sheriff and Peikert [5]. We chose to use the KDM-CPA candidate of Applebaum *et al.* [7] since its proof of KDM-CPA security is tight in the multi-chalenge setting. It thus provides tight KDM-CCA2 security by applying our tightly secure USS argument of Section 4.

F.1 Definition

A public-key encryption scheme consists of a tuple of efficient algorithms ($\mathsf{Par\text{-}Gen}$, Keygen, $\mathsf{Encrypt}$, $\mathsf{Decrypt}$), where $\mathsf{Par\text{-}Gen}$ takes as input a security parameter 1^λ and generates common public parameters Γ, Keygen inputs Γ and outputs a key pair (SK, PK), while $\mathsf{Encrypt}$ and $\mathsf{Decrypt}$ proceed in the usual way.

We recall the definition of KDM-CCA2 security given by Chandran *et al.* [28], which extends the definition of Boneh *et al.* [19] to the chosen-ciphertext setting. As in [19,28,7], the adversary is restricted to encryption queries for functions from a certain family $\mathcal{F} \subset \{f \mid f : \mathcal{SK}^N \to \mathcal{M}\}$, for a polynomial $N \in \mathsf{poly}(\lambda)$, where $\mathcal{SK}$ and $\mathcal{M}$ are the keyspace and the message space, respectively.

Definition F.1 ([28]). *A PKE scheme for a function family $\mathcal{F}$ provides KDM-CCA2 security if no PPT adversary has noticeable advantage in the next game.*

Initialization. *The challenger generates public parameters $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda)$ and N key pairs $(PK_i, SK_i) \leftarrow \mathsf{Keygen}(\Gamma)$. The adversary $\mathcal{A}$ is given Γ and $\{PK_i\}_{i \in [N]}$. The challenger also flips a fair coin $d \hookleftarrow U(\{0,1\})$.*

Queries. *$\mathcal{A}$ adaptively interleaves encryption and decryption queries.*

- **Encryption queries***: The adversary chooses a pair (j, f), where $j \in [N]$ and $f \in \mathcal{F}$. If $d = 0$, the challenger computes $\mu = f(SK_1, \ldots, SK_N) \in \mathcal{M}$ and $C \leftarrow \mathsf{Encrypt}(PK_j, \mu)$. If $d = 1$, the challenger computes $C \leftarrow \mathsf{Encrypt}(PK_j, \mathbf{0}^{|\mu|})$. In either case, the ciphertext C is returned to $\mathcal{A}$.*
- **Decryption queries***: The adversary chooses a ciphertext-index pair (j, C). The challenger returns $\perp$ if C was produced in response to an encryption query $(j, *)$. Otherwise, the challenger computes and returns $\mu \leftarrow \mathsf{Decrypt}(SK_j, C)$ (which may be $\perp$ if C is an invalid ciphertext).*

Guess. *After polynomially many queries, $\mathcal{A}$ halts and outputs $d' \in \{0,1\}$. The adversary is declared successful if $d' = d$ and its advantage is defined to be*

$$\begin{aligned}\mathbf{Adv}^{\text{kdm-cca}}(\mathcal{A}) = |\Pr[\mathcal{A}^{\mathcal{O}_{\text{Enc}}, \mathcal{O}_{\text{Dec}}}(\{PK_i\}_{i=1}^N) = 1 \mid d = 0] \\ - \Pr[\mathcal{A}^{\mathcal{O}_{\text{Enc}}, \mathcal{O}_{\text{Dec}}}(\{PK_i\}_{i=1}^N) = 1 \mid d = 1]|\end{aligned}$$

F.2 Construction

The scheme proceeds by having the sender verifiably encrypt the same $\mu \in \mathbb{Z}_p$ under two independent keys by computing ciphertexts $(\mathbf{c}_{0,0}, c_{0,1})$, $(\mathbf{c}_{1,0}, c_{1,1})$. We note that the non-interactive argument $\boldsymbol{\pi}$ crucially has to prove that these ciphertexts are *both* of the form $(\mathbf{u}_b, \mathbf{u}_b^\top \cdot \mathbf{s}_b + \mathsf{noise}_b + p \cdot \mu)$ for some $\mathbf{u}_0, \mathbf{u}_1 \in \mathbb{Z}_q^n$. In particular, it is not sufficient to argue that the difference $(\mathbf{c}_{0,0}, c_{0,1}) - (\mathbf{c}_{1,0}, c_{1,1})$ decrypts to 0 since the adversary would be able to recover the secret key by making decryption queries where $(\mathbf{c}_{0,0}, c_{0,1})$ and $(\mathbf{c}_{1,0}, c_{1,1})$ are of the form $(\mathbf{u}_b, \mathbf{u}_b^\top \cdot \mathbf{s}_b + \delta)$, for some $\delta \in \mathbb{Z}_q$ that is far from a multiple of p.

$\mathsf{Par\text{-}Gen}(1^\lambda)$: Given λ, select dimensions n, m, moduli q, p where p is prime and $q = p^2$, and Gaussian parameters $r, r', \alpha > 0$ (to be specified in Section F.4) and output $\Gamma = \{\lambda, n, m, q, p, r, r', \alpha\}$.

$\mathsf{Keygen}(\Gamma)$: On input of public parameters Γ, generate a key pair as follows.

1. Choose a random matrix $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m})$.

2. Sample $\mathbf{s}_0, \mathbf{s}_1 \hookleftarrow D_{\mathbb{Z}^n,\alpha q}$, $\mathbf{e}_0, \mathbf{e}_1 \hookleftarrow D_{\mathbb{Z}^m,\alpha q}$ and compute

$$\mathbf{b}_0 = \mathbf{A}^\top \cdot \mathbf{s}_0 + \mathbf{e}_0 \ \in \mathbb{Z}_q^m, \qquad\qquad \mathbf{b}_1 = \mathbf{A}^\top \cdot \mathbf{s}_1 + \mathbf{e}_1 \ \in \mathbb{Z}_q^m. \ .$$

Define the matrix $\mathbf{A}_{\mathrm{eq}}$ as

$$\mathbf{A}_{\mathrm{eq}} = \left[\begin{array}{c|c|c|c} \mathbf{A} & & & \\ \hline \mathbf{b}_0^\top & 1 & & \\ \hline & & \mathbf{A} & \\ \hline & & \mathbf{b}_1^\top & 1 \end{array}\right] \in \mathbb{Z}_q^{2(n+1)\times 2(m+1)}, \qquad (26)$$

and define the language $\mathcal{L}^{\mathrm{eq}} = (\mathcal{L}^{\mathrm{eq}}_{\mathsf{zk}}, \mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}})$, where

$$\begin{aligned} \mathcal{L}^{\mathrm{eq}}_{\mathsf{zk}} := \Big\{ (\mathbf{c}_0, \mathbf{c}_1) \in (\mathbb{Z}_q^{n+1})^2 \ \mid \exists \mathbf{r}_0, \mathbf{r}_1 \in \mathbb{Z}^m,\ \chi_0, \chi_1 \in \mathbb{Z},\ \mu \in \mathbb{Z}_p \ : \\ \|\mathbf{r}_b\| \le B_r,\ |\chi_b| \le B_\chi \ \ \forall b \in \{0,1\} \\ \wedge \quad \mathbf{c}_b = \bar{\mathbf{A}}_b \cdot [\mathbf{r}_b^\top \mid \chi_b]^\top + \mu \cdot \big[\mathbf{0}^{n\top} \mid p\big]^\top \bmod q \ \Big\}, \end{aligned}$$

$$\begin{aligned} \mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}} := \Big\{ (\mathbf{c}_0, \mathbf{c}_1) \in (\mathbb{Z}_q^{n+1})^2 \ \mid \ \exists\ \bar{\mathbf{c}}_0, \bar{\mathbf{c}}_1 \in \mathbb{Z}_q^n,\ v_0, v_1 \in [-B^*, B^*],\ \mu \in \mathbb{Z}_p \\ \wedge \ \ \mathbf{c}_b = \left[\begin{array}{c} \bar{\mathbf{c}}_b \\ \hline \mathbf{s}_b^\top \cdot \bar{\mathbf{c}}_b + p \cdot \mu + v_b \end{array}\right] \ \ \forall b \in \{0,1\} \Big\}, \end{aligned}$$

where bounds B_r, B_χ, B^* are specified in Section F.4.

3. Generate a common reference string $\mathsf{crs} = \big(\mathsf{par}, (\mathsf{crs}'_{\mathcal{L}}, pk_{\mathsf{LPKE}}, k, \mathsf{AHF}, \mathsf{OTS})\big)$ for a simulation-sound argument system Π^{uss} with its simulation trapdoor $\tau_{\mathsf{zk}} := sk_{\mathsf{LPKE}}$ for the language $\mathcal{L}^{\mathrm{eq}}$.

Output (PK, SK), where $PK := \big(\mathbf{A}, \mathbf{b}_0, \mathbf{b}_1, \mathsf{crs}\big)$ and $SK := \mathbf{s}_0 \in \mathbb{Z}^n$.

Encrypt(PK, μ) : To encrypt $\mu \in \mathbb{Z}_p$, conduct the following steps.

1. Choose $\mathbf{r}_0, \mathbf{r}_1 \hookleftarrow D_{\mathbb{Z}^m, r}$, $\chi_0, \chi_1 \hookleftarrow D_{\mathbb{Z}, r'}$ and compute two ciphertexts $(\mathbf{c}_{0,0}, c_{0,1}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, $(\mathbf{c}_{1,0}, c_{1,1}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where

$$\begin{aligned} \mathbf{c}_{0,0} &= \mathbf{A} \cdot \mathbf{r}_0, & c_{0,1} &= \mathbf{b}_0^\top \cdot \mathbf{r}_0 + \chi_0 + p \cdot \mu \\ \mathbf{c}_{1,0} &= \mathbf{A} \cdot \mathbf{r}_1, & c_{1,1} &= \mathbf{b}_1^\top \cdot \mathbf{r}_1 + \chi_1 + p \cdot \mu. \end{aligned}$$

2. Using $\mathbf{w} = [\mathbf{r}_0^\top \mid \chi_0 \mid \mathbf{r}_1^\top \mid \chi_1]^\top$ and $\mu \in \mathbb{Z}_p$, generate a simulation-sound NIZK argument that ciphertexts $\mathbf{c}_0 = (\mathbf{c}_{0,0}, c_{0,1})$ and $\mathbf{c}_1 = (\mathbf{c}_{1,0}, c_{1,1})$ satisfy $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}^{\mathrm{eq}}_{\mathsf{zk}}$. This argument $\boldsymbol{\pi} = \big(\mathsf{VK}, (\mathbf{a}, \mathbf{z}), sig\big)$ is obtained by computing $\mathsf{Chall} = \mathsf{Hash}(k, (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}), \mathbf{a}, \mathsf{VK})$ and a one-time signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, ((\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}), \mathbf{a}, \mathbf{z}))$.

Output the ciphertext $\mathbf{C} = \big(\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}, \boldsymbol{\pi}\big)$.

Decrypt$(SK, \mathbf{C})$: Given $\mathbf{C} = \big(\mathbf{c}_{0,0}, c_{0,}, \mathbf{c}_{1,0}, c_{1,1}, \boldsymbol{\pi}\big)$ and $SK = \mathbf{s}_0 \in \mathbb{Z}^n$, return $\perp$ if $\boldsymbol{\pi}$ does not properly verify. Otherwise, compute $\mu' = c_{0,1} - \mathbf{s}_0^\top \cdot \mathbf{c}_{0,0} \bmod q$ and output $\mu \in \mathbb{Z}_p$ which minimizes $|\mu' - p \cdot \mu \bmod q|$.

When the above construction is instantiated with the simulation-sound argument of Section 4, the encryptor additionally chooses a random core tag component t_c for the $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE system, which is included in $\boldsymbol{\pi}$, and the challenge of the Σ-protocol is computed as $\mathsf{Chall} = \mathsf{Hash}(k, (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}), \mathbf{a}, t_c, \mathsf{VK})$.

Comparison with the generic approach. In order to prove plaintext equalities in $\boldsymbol{\pi}$, each iteration of the underlying protocol costs $O(m \cdot \log q)$ bits of communication. If we were to generically apply the results of [34,87] in order to prove that a vector $\mathbf{y} \in \mathbb{Z}_q^m$ is of the form $\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e}$, we estimate that each proof iteration would take $\Omega(m^6 \cdot \log^2 q)$ bits. To our knowledge, the best solution based on a Karp reduction would require to encode the statement as a CNF-SAT instance with $\Omega(mn \cdot \log^2 q)$ clauses and $\Omega(n \cdot \log q)$ variables. Then, a reduction to the Graph Hamiltonicity language would create a graph with $\Omega(m^3 \cdot \log q)$ vertices, of which the adjacency matrix should be encrypted in the first move of the trapdoor Σ-protocol from [34].

The step of a Karp reduction can be avoided using zero-knowledge techniques based on multiparty [67] or two-party computation [62] (in particular, the Σ-protocols of [62] can be turned into trapdoor Σ-protocols). Still, the communication complexity remains linear in the size of the circuit computing the relation. For a language where witnesses are multiplied by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ as in our setting, this circuit size would grow as $\Omega(mn \cdot \log q)$.

While the dependency on the circuit size can be eliminated using homomorphic encryption techniques [51], these require to begin with NIZK proofs for a significantly more complex language – which expresses honest key generation, encryption and decryption in the underlying FHE – than the one we need here. To our knowledge, the most efficient way to prove the honest-FHE-key-generation and honest-encryption parts of the statement in [51, Figure 1] is to use the trapdoor Σ-protocols in Appendices G and E.1, respectively. More precisely, the former handles LWE-type relations (like an FHE public key being correctly generated) and the latter can be adapted to prove that an FHE ciphertext encrypts 0 or 1. As for the proof-of-correct-decryption step, the best solution seems to be the technique of Kim and Wu [71].

The scheme described in this section is considerably simpler as it only uses the trapdoor Σ-protocol of Section 5. In particular, the non-tight version does not rely on FHE outside the CI hash function component.

F.3 Security

The proof of Theorem F.2 follows the same strategy as all proofs based on the Naor-Yung paradigm. In order to rely on Theorem 3.4 in a modular way, the proof crucially relies on the ability to use secret keys $(\mathbf{s}_0, \mathbf{s}_1)$ as a membership testing trapdoor for $\mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}}$ and as a trapdoor to compute the $\mathsf{BadChallenge}$ function of the underlying trapdoor Σ-protocol.

If the simulation-sound argument system is instantiated with our construction of Section 4.2, the tightness of the reduction is not affected by the number Q of encryption queries, but only by the number N of users. We will explain later on how we can also obtain tight security with respect to the number N of users.

Theorem F.2. *The scheme provides KDM-CCA2 security for affine functions assuming that: (i) The* LWE *assumption holds; (ii) The proof system* Π^{uss} *provides unbounded simulation-soundness. Concretely, for any PPT adversary* $\mathcal{A}$*, there exist PPT algorithms* $\mathcal{B}_1$*,* $\mathcal{B}_2$ *and* $\mathcal{B}_3$ *with comparable running time to* $\mathcal{A}$*'s and such that*

$$\mathbf{Adv}^{\text{kdm-cca}}(\mathcal{A}) \leq 2 \cdot \mathbf{Adv}^{\text{lwe}}_{\mathcal{B}_1}(\lambda) + N \cdot \big(\mathbf{Adv}^{\text{sound}}_{\mathcal{B}_2}(\lambda) + \mathbf{Adv}^{\text{uss}}_{\mathcal{B}_3}(\lambda)\big) + 2^{-\Omega(\lambda)}.$$

Proof. The proof uses of a sequence of games starting with a game where the challenger's hidden bit is $d = 0$ and ending with a game where $d = 1$. For each i, S_i is the event that $\mathcal{A}$ wins in Game_i.

Game_1: This game is the real KDM-CCA experiment where the challenger's bit is $d = 0$. In details, the challenger generates a sequence of N public keys $\{PK_i\}_{i=1}^N$, where $PK_i := \big(\mathbf{A}_i, \mathbf{b}_{i,0}, \mathbf{b}_{i,1}, \mathsf{crs}_i\big)$ for each $i \in [N]$. It gives $\{PK_i\}_{i=1}^N$ to the adversary $\mathcal{A}$ and keeps the private keys $\{SK_i = \mathbf{s}_{i,0}\}_{i=1}^N$ to itself. At each decryption query, $\mathcal{B}$ faithfully runs the real decryption algorithm using the private keys $\{SK_i = \mathbf{s}_{i,0}\}_{i=1}^N$. At each encryption query, the adversary $\mathcal{A}$ chooses an index $j \in [N]$ and an affine function $f_{\mathbf{t},w}(\mathbf{s}) = \langle \mathbf{t}, \mathbf{s} \rangle + w \in \mathbb{Z}_p$ specified by a vector $\mathbf{t} \in \mathbb{Z}_p^n$ and a scalar $w \in \mathbb{Z}_p$. The challenger replies by generating a ciphertext $\mathbf{C}^\star = \big(\mathbf{c}^\star_{0,0}, c^\star_{0,1}, \mathbf{c}^\star_{1,0}, c^\star_{1,1}, \boldsymbol{\pi}^\star\big)$ which is an encryption under PK_j of $f_{\mathbf{t},w}(\mathbf{s}_i) = \langle \mathbf{t}, \mathbf{s}_{i,0} \rangle + w \bmod p$. Decryption queries are disallowed for ciphertexts $\mathbf{C}^\star$ returned by the encryption oracle. Eventually, $\mathcal{A}$ halts and outputs a bit $d' \in \{0,1\}$. We denote by S_1 the event that $d' = 0$.

Game_2: We change the decryption oracle. Instead of using the private keys $\{SK_i = \mathbf{z}_{i,0}\}_{i=1}^N$ at each valid decryption query $(j, \mathbf{C})$, where $j \in [N]$ and $\mathbf{C} = \big(\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}, \boldsymbol{\pi}\big)$, $\mathcal{B}$ recalls the short vectors $\{\mathbf{s}_{i,1}\}_{i=1}^N$ for which $\mathbf{b}_{i,1} = \mathbf{A}^\top \cdot \mathbf{s}_{i,1} + \mathbf{e}_{i,1} \in \mathbb{Z}_q^m$ and decrypts $(\mathbf{c}_{1,0}, c_{1,1})$ by computing $c_{1,1} - \mathbf{s}_{j,1}^\top \mathbf{c}_{1,0} \bmod q$. Clearly, $\mathcal{A}$'s view is not affected by this change unless it is able to invoke the decryption oracle on a valid-looking ciphertext although $(\mathbf{c}_{0,0}, c_{0,1})$ and $(\mathbf{c}_{1,0}, c_{1,1})$ are not both valid encryptions of some message $\mu \in \mathbb{Z}_p$ for the public key PK_j. Note that this can only happen for a ciphertext such that $\big((\mathbf{c}_{0,0}, c_{0,1}), (\mathbf{c}_{1,0}, c_{1,1})\big)$ is outside the language $\mathcal{L}_{\mathrm{eq}}$ defined by PK_j. If we call E_2 the event that such a decryption query occurs, we have the inequality $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[E_2]$. Moreover, event E_2 would imply an algorithm $\mathcal{B}$ that breaks the soundness of the proof system when a membership testing trapdoor $\tau_{\mathcal{L}}$ is available. Concretely, Lemma F.3 shows that $\Pr[E_2] \leq N \cdot \mathbf{Adv}^{\text{sound}}_{\mathcal{B}}(\lambda)$.

Game_3: This game is like Game_2 except that, at each encryption query $(j, f_{\mathbf{V},w})$, the returned ciphertext $\mathbf{C}^\star = \big(\mathbf{c}^\star_{0,0}, c^\star_{0,1}, \mathbf{c}^\star_{1,0}, c^\star_{1,1}, \boldsymbol{\pi}^\star\big)$ is obtained by computing $\boldsymbol{\pi}^\star$ as a simulated proof using the simulation trapdoor associated with the language $\mathcal{L}^{\mathrm{eq}}$ defined by PK_j. The statistical zero-knowledge property of the proof system guarantees that $\mathcal{A}$'s view is not affected by this change. We have $|\Pr[S_3] - \Pr[S_2]| \leq 2^{-\Omega(\lambda)}$.

Game_4: We modify the treatment of encryption queries $(j, f_{\mathbf{V},w})$. When $\mathcal{B}$ computes a challenge ciphertext $\mathbf{C}^\star = \big(\mathbf{c}^\star_{0,0}, c^\star_{0,1}, \mathbf{c}^\star_{1,0}, c^\star_{1,1}, \boldsymbol{\pi}^\star\big)$, it computes a hybrid ciphertext where $(\mathbf{c}^\star_{0,0}, c^\star_{0,1})$ is an encryption of $0 \in \mathbb{Z}_p$ and $\mathbf{c}^\star_{1,0}, c^\star_{1,1}$ is an encryption of the queried function $f_{\mathbf{t},w}(\cdot)$. It is easy to prove that any PPT adversary $\mathcal{A}$ that can distinguish between Game_3 and Game_4 would imply an adversary against the KDM-CPA security of the scheme in [7], which would contradict the LWE assumption. The result of [7, Theorem 2] implies that $|\Pr[S_4] - \Pr[S_3]| \leq \mathbf{Adv}^{\mathrm{lwe}}(\lambda)$. Here, the latter inequality uses the fact the reduction of [7, Theorem 2] is tight and does not multiplicatively decline with the number Q of encryption queries made by the adversary $\mathcal{A}$ (nor the number N of users).

Game_5: We modify again the decryption oracle. This time, instead of using the backdoor keys $\{\mathbf{s}_{i,1}\}_{i=1}^N$ to recover the plaintext μ from $(\mathbf{c}_{1,0}, c_{1,1})$ at each valid decryption query $(j, \mathbf{C})$, where $j \in [N]$ and $\mathbf{C} = \big(\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}, \boldsymbol{\pi}\big)$, the challenger $\mathcal{B}$ reverts to using the actual secret keys $\{SK_i = \mathbf{s}_{i,0}\}_{i=1}^N$ to compute $\mu' = c_{0,1} - \mathbf{s}_{j,0}^\top \cdot \mathbf{c}_{0,0} \bmod q$ from $(\mathbf{c}_{0,0}, c_{0,1})$. It is easy to see that the adversary's view remains as in Game_4 until it manages to query the decryption oracle on a valid-looking ciphertext $\mathbf{C}$ for PK_j although $(\mathbf{c}_{0,0}, c_{0,1})$ and $(\mathbf{c}_{1,0}, c_{1,1})$ are not both valid encryptions of a given message μ. If we denote by E_5 the latter event, Lemma F.4 shows that it contradicts the unbounded simulation-soundness of the underlying proof system.

Game_6: We bring yet another modification to the generation of challenge ciphertexts $\mathbf{C}^\star = \big(\mathbf{c}^\star_{0,0}, c^\star_{0,1}, \mathbf{c}^\star_{1,0}, c^\star_{1,1}, \boldsymbol{\pi}^\star\big)$. Namely, in all encryption queries $(j, f_{\mathbf{t},w})$, instead of generating $(\mathbf{c}^\star_{0,0}, c^\star_{0,1})$ and $(\mathbf{c}^\star_{1,0}, c^\star_{1,1})$ as encryptions of 0 and $f_{\mathbf{t},w}(\cdot)$, respectively, $(\mathbf{c}^\star_{0,0}, c^\star_{0,1})$, and $(\mathbf{c}^\star_{1,0}, c^\star_{1,1})$ are now obtained by encrypting $0 \in \mathbb{Z}_p$ twice. Any noticeable change in $\mathcal{A}$'s output distribution would imply an IND-CPA adversary in the multi-user setting against the scheme of [7]. Since KDM-CPA security implies IND-CPA security, the result of [7, Theorem 2] thus implies $|\Pr[S_6] - \Pr[S_5]| \leq \mathbf{Adv}^{\mathrm{lwe}}(\lambda)$.

Game_7: We bring one last change to the generation of the challenge ciphertexts $\mathbf{C}^\star = \big(\mathbf{c}^\star_{0,0}, c^\star_{0,1}, \mathbf{c}^\star_{1,0}, c^\star_{1,1}, \boldsymbol{\pi}^\star\big)$. Instead of computing $\boldsymbol{\pi}^\star$ using the simulation trapdoor of $\mathsf{\Pi}$, we compute it using the witnesses $(\mathbf{r}_0, \chi_0, \mathbf{r}_1, \chi_1, 0)$. This change does not significantly affect $\mathcal{A}$'s view since the obtained proofs are statistically close to those of Game_6. We have $|\Pr[S_7] - \Pr[S_6]| \leq 2^{-\Omega(\lambda)}$.

We observe that Game_7 corresponds to the actual KDM-CCA experiment where the challenger's bit is $d = 1$. If we combine the above, we obtain the stated upper bound for $\mathbf{Adv}^{\text{kdm-cca}}(\mathcal{A}) := |\Pr[S_1] - \Pr[S_7]|$. □

Lemma F.3. *Assuming that an adversary $\mathcal{A}$ can distinguish between Game_1 and Game_2, there exists an algorithm $\mathcal{B}$ with comparable running time that breaks the soundness of the proof system $\mathsf{\Pi}^{\mathrm{uss}}$ with advantage $\mathbf{Adv}^{\mathrm{sound}}(\lambda) \geq \Pr[E_2]/N$.*

Proof. Algorithm $\mathcal{B}$ is given a common reference string crs and the description of a language consisting of a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ together with vectors $\mathbf{b}_0, \mathbf{b}_1 \in \mathbb{Z}_q^m$ as well as a membership testing trapdoor $\tau_{\mathcal{L}}$ consisting of vectors $(\mathbf{s}_0, \mathbf{s}_1) \in \mathbb{Z}_n$ such that $\mathbf{b}_b = \mathbf{A}^\top \mathbf{s}_b + \mathbf{e}_b$ for each $b \in \{0,1\}$. Using $\tau_{\mathcal{L}} = (\mathbf{s}_0, \mathbf{s}_1)$, $\mathcal{B}$ can set up a public key for some user. It thus chooses $i^\star \hookleftarrow U([N])$ as a guess that event E_2 occurs for the first time in a decryption query involving the secret key $SK_{i^\star}$. Next, $\mathcal{B}$ faithfully generates $\{PK_i\}_{i \in [N] \setminus \{i^\star\}}$ and $\{SK_i\}_{i \in [N] \setminus \{i^\star\}}$ by running the real key generation algorithm. In the process of generating $\{PK_i\}_{i \in [N] \setminus \{i^\star\}}$, it also chooses the matrices $\mathbf{A}_i \sim U(\mathbb{Z}_q^{n \times m})$ and vectors $(\mathbf{b}_{i,0}, \mathbf{b}_{i,1})$ together with secret keys $(\mathbf{s}_{i,0}, \mathbf{s}_{i,1})$ for each $i \in [N] \setminus \{i^\star\}$. Then, $\mathcal{B}$ defines the $i^\star$-th public key as $PK_{i^\star} := \big(\mathbf{A}, \mathbf{b}_0 = \mathbf{A}^\top \mathbf{s}_0 + \mathbf{e}_{i^\star,0}, \mathbf{b}_1 = \mathbf{A}^\top \mathbf{s}_1 + \mathbf{e}_{i^\star,1}, \mathsf{crs}\big)$, with $\mathbf{e}_{i^\star,0}, \mathbf{e}_{i^\star,1} \hookleftarrow D_{\mathbb{Z}^m, \alpha q}$, and runs the adversary on input of $\{PK_i\}_{i \in [N]}$.

Since $\mathcal{B}$ knows $\{SK_i\}_{i \in [N]}$, it can properly answer all queries exactly as in the real game. In addition, it can detect any occurrence of event E_2 since it knows $\{\mathbf{s}_{i,0}, \mathbf{s}_{i,1}\}_{i=1}^N$ for all public keys $\{\mathbf{A}_i, \mathbf{b}_{i,0}, \mathbf{b}_{i,1}\}_{i=1}^N$. Recall that an occurrence of event E_2 consists of a valid ciphertext $\mathbf{C} = \big(\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}, \boldsymbol{\pi}\big)$ for which $\big((\mathbf{c}_{0,0}, c_{0,1}), (\mathbf{c}_{1,0}, c_{1,1})\big)$ is outside $\mathcal{L}^{\mathrm{eq}}$. At the first such occurrence, $\mathcal{B}$ aborts if the involved public key is not $PK_{i^\star}$. Otherwise, it halts and outputs the statement $\big((\mathbf{c}_{0,0}, c_{0,1}), (\mathbf{c}_{1,0}, c_{1,1})\big)$ and the proof $\boldsymbol{\pi}$ extracted from $\mathbf{C}$. Clearly, if $\mathcal{B}$ successfully guesses the index $i^\star$ of the public key involved in the first occurrence of E_2, it manages to break the soundness of Π^{uss}. Since $i^\star \hookleftarrow U([N])$ is chosen independently of $\mathcal{A}$'s view, we have $\Pr[E_2] \leq N \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathrm{sound}}(\lambda)$, as claimed. □

Lemma F.4. Game_5 *is computationally indistinguishable from* Game_4. *Assuming that $\mathcal{A}$ can distinguish between these games, there exists a PPT algorithm $\mathcal{B}$ that breaks the unbounded simulation-soundness of the proof system* Π^{uss} *with advantage* $\mathbf{Adv}^{\mathrm{uss}}(\lambda) \geq \Pr[E_5]/N$.

Proof. Let us assume that there exists an adversary $\mathcal{A}$ that can distinguish between the two games. We use $\mathcal{A}$ to build an adversary $\mathcal{B}$ against the unbounded simulation-soundness of the proof system. Algorithm $\mathcal{B}$ is given a common reference string crs and the description of a language $\mathcal{L}^{\mathrm{eq}}$ specified by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and vectors $\mathbf{b}_0, \mathbf{b}_1 \in \mathbb{Z}_q^m$. It also receives a membership testing trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}^{\mathrm{eq}}_{\mathsf{sound}}$, which consists of a pair of short vectors $(\mathbf{s}_0, \mathbf{s}_1) \in (\mathbb{Z}^n)^2$ such that $\mathbf{b}_0 = \mathbf{A}^\top \cdot \mathbf{s}_0 + \mathbf{e}_0 \in \mathbb{Z}_q^m$ and $\mathbf{b}_1 = \mathbf{A}^\top \cdot \mathbf{s}_1 + \mathbf{e}_1 \in \mathbb{Z}_q^m$, where $\mathbf{e}_0, \mathbf{e}_1 \sim D_{\mathbb{Z}^m, \alpha q}$. Using its input, $\mathcal{B}$ can set up a public key for one user for which $\mathcal{B}$ expects $\mathcal{A}$ to break the simulation-soundness of the argument system. It thus draws $i^\star \hookleftarrow U([N])$ as a guess that E_5 will occur for the first time in a decryption query involving the secret key $SK_{i^\star}$, which will be implicitly defined as $SK_{i^\star} = \mathbf{s}_0$. It thus sets $\mathbf{s}_{i^\star,0} := \mathbf{s}_0$ and $\mathbf{s}_{i^\star,1} := \mathbf{s}_1$. Next, $\mathcal{B}$ generates the remaining key pairs $\{(PK_i, SK_i)\}_{i \in [N] \setminus \{i^\star\}}$ as in the real encryption scheme. As part of this process, $\mathcal{B}$ needs to generate public matrices $\mathbf{A}_i \sim U(\mathbb{Z}_q^{n \times m})$ together with vectors $\mathbf{b}_{i,0} = \mathbf{A}_i^\top \cdot \mathbf{s}_{i,0} + \mathbf{e}_{i,0}$ and $\mathbf{b}_{i,1} = \mathbf{A}_i^\top \cdot \mathbf{s}_{i,1} + \mathbf{e}_{i,1}$ for each $i \in [N] \setminus \{i^\star\}$. It also defines $PK_{i^\star} := (\mathbf{A}, \mathbf{b}_0, \mathbf{b}_1, \mathsf{crs})$ and feeds $\mathcal{A}$ with the input $\{PK_i\}_{i \in [N]}$. In the following, we denote by $\mathcal{L}^{\mathrm{eq},(i)}_{\mathsf{sound}}$ the language associated with the j-th public key PK_j (so that $\mathcal{L}_{\mathsf{sound}}^{\mathrm{eq},(i^\star)}$ is $\mathcal{B}$'s challenge language $\mathcal{L}_{\mathsf{sound}}^{\mathrm{eq}}$).

To answer an encryption query $(j, f_{\mathbf{t},\mathbf{w}})$, $\mathcal{B}$ uses PK_j to compute the left ciphertext $(\mathbf{c}_{0,0}, c_{0,1})$ as an encryption of $0 \in \mathbb{Z}_p$ and the right ciphertext $(\mathbf{c}_{1,0}, c_{1,1})$ as an encryption of $f_{\mathbf{t},\mathbf{w}}(\cdot)$. It then invokes its challenger and asks for a simulated proof $\boldsymbol{\pi} \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{zk}, (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}))$. Using the latter, it returns the ciphertext $((\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}), \boldsymbol{\pi})$ to the adversary. To answer a decryption query $(j, \mathbf{C} = (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}, \boldsymbol{\pi}))$ for which $((\mathbf{c}_{0,0}, c_{0,1}), (\mathbf{c}_{1,0}, c_{1,1})) \in \mathcal{L}_{\mathsf{sound}}^{\mathrm{eq},(j)}$ (note that $\mathcal{B}$ can perform this check using the trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}_{\mathsf{sound}}^{\mathrm{eq},(j)}$) and $\mathbf{C}$ was never the result of an encryption query under PK_j, the reduction computes $\mu' = c_{0,0} - \mathbf{s}_{j,0}^\top \cdot \mathbf{c}_{0,0}$ as in Game_5 and returns $\mu \in \mathbb{Z}_p$ such that $|\mu' - p \cdot \mu|$ is minimized.

At the first decryption query $(i, \mathbf{C} = (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}, \boldsymbol{\pi}))$ involving a ciphertext such that $\big((\mathbf{c}_{0,0}, c_{0,1}), (\mathbf{c}_{1,0}, c_{1,1})\big) \not\in \mathcal{L}_{\mathsf{sound}}^{\mathrm{eq},(i)}$, $\mathcal{B}$ halts (again, $\mathcal{B}$ can always detect an occurrence of E_5 using the trapdoor $\tau_{\mathcal{L}}$ for $\mathcal{L}_{\mathsf{sound}}^{\mathrm{eq},(i)}$). If $i \neq i^\star$, $\mathcal{B}$ aborts and reports failure. Otherwise, it relays $x = \big((\mathbf{c}_{0,0}, c_{0,1}), (\mathbf{c}_{1,0}, c_{1,1})\big)$ and $\boldsymbol{\pi}$ to its unbounded simulation-soundness challenger. Since $i^\star \in [N]$ was sampled independently of $\mathcal{A}$'s view, the same arguments as in the proof of Lemma F.3 show that $\mathbf{Adv}_{\mathcal{B}}^{\mathrm{uss}}(\lambda) \geq \frac{1}{N} \cdot \Pr[E_5]$. □

On Achieving Tightness in the Number of Users. The proof of Theorem F.2 only provides tight security with respect to the number of encryption queries. It is possible to also obtain tightness with respect to the number N of users if we have an argument system that provides tight simulation-soundness in the multi-instance setting. This means that we need tight security in a game where the adversary is given N common reference strings $(\mathsf{crs}^{(1)}, \ldots, \mathsf{crs}^{(N)})$ and a simulation oracle $\mathsf{Sim}_1(\mathsf{crs}^{(i)}, \tau_{\mathrm{zk}}^{(i)}, \cdot)$ for each of these. The adversary eventually aims at outputting a non-trivial verifying argument $\boldsymbol{\pi}^\star$ of a false statement $x^\star$ for a common reference string $\mathsf{crs}^{(i^\star)}$ of its choice, with $i^\star \in [N]$.

By inspecting the proof of Theorem D.3, we observe that the argument system of Section 4 can be proved tightly simulation-sound in the N-instance setting with the following changes.

1. The underlying PRF family has to provide tight security in the multi-instance setting. This means pseudorandomness in an experiment where the adversary is granted access to N oracles $\mathsf{PRF}(k^{(i)}, \cdot)$, for $i \in [N]$, and should be unable to distinguish them from N independent truly random functions, even after polynomially many queries to each oracle.

2. The indistinguishability properties of the $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme have to hold in the N-instance setting, with a tight reduction when they rely on the LWE assumption.

   Let the distributions $D_{\mathrm{inj}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{Keygen}(\Gamma, K)\}$ and $D_{\mathrm{loss}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$. For any $K_1, \ldots, K_N \in \mathcal{K}_\lambda$ and any PPT adversary $\mathcal{A}$, the advantage function $\mathbf{Adv}_{\mathcal{A}}^{N\text{-}\mathsf{indist\text{-}LPKE\text{-}1}}(\lambda)$

defined as

$$\begin{aligned}&|\Pr[(pk_1, tk_1), \ldots, (pk_N, tk_N) \hookleftarrow D_{\text{inj}} : \mathcal{A}\big(\{(pk_i, tk_i\}_{i=1}^{N}\big) = 1] \\ &\qquad - \Pr[(pk_1, tk_1), \ldots (pk_N, tk_N) \hookleftarrow D_{\text{loss}} : \mathcal{A}\big(\{(pk_i, tk_i\}_{i=1}^{N}\big) = 1]|\end{aligned}$$

should be negligible.

We also need the second indistinguishability property to hold in the N-instance setting. However, in our $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme of Section 4.1, this property holds in the statistical sense.

3. The CI hash function family has to provide tight correlation intractability in the N-instance setting. Formally:

   For any relations $R_1, \ldots, R_N \in \mathfrak{R}$, the distributions $\{(k_1, \ldots, k_N) \mid k_i \leftarrow \mathsf{Gen}(1^\lambda) \ \ \forall i \in [N]\}$ and $\{(k_1, \ldots, k_N) \mid k_i \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{aux}_{R_i}) \ \ \forall i \in [N]\}$ should be computationally indistinguishable. For any PPT distinguisher $\mathcal{A}$, the following advantage function $\mathbf{Adv}_{\mathcal{A}}^{N\text{-indist-CI}}(\lambda)$

$$\begin{aligned}&|\Pr[k_i \leftarrow \mathsf{Gen}(1^\lambda) \ \ \forall i \in [N] : \mathcal{A}\big(\{(k_i, \mathsf{aux}_{R_i})\}_{i=1}^{N}]\big) = 1] \\ &\quad - \Pr[k_i \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{aux}_{R_i}) \ \ \forall i \in [N] : \mathcal{A}\big(\{(k_i, \mathsf{aux}_{R_i})\}_{i=1}^{N}\big) = 1]|\end{aligned}$$

   has to be negligible.

The first requirement can be met by exploiting an observation of Boyen and Li [22] who showed how to build tightly secure PRFs in the multi-instance setting using tightly secure key-homomorphic PRFs.

As for the second requirement, it can be satisfied by our $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme of Section 4.1 if we modify it by sampling the columns of the secret key $\mathbf{S} \in \mathbb{Z}^{(n-n_0)\times n_0}$ from the noise distribution χ. By doing so, we can prove tight multi-instance indistinguishability by relying on the reduction of [7, Lemma 2] which constructs N independent instances of $\mathsf{HNF\text{-}LWE}$ out of a single LWE instance with one uniform secret. This allows us to bound the advantage of an adversary in the N-instance setting as $\mathbf{Adv}_{\mathcal{A}}^{N\text{-indist-LPKE-1}}(\lambda) \leq n_0 \cdot \mathbf{Adv}_{q,m,n-n_0,\alpha}^{\mathsf{LWE}}(\lambda)$, where n_0 is the message length and $\mathbf{Adv}_{q,m,n-n_0,\alpha}^{\mathsf{LWE}}(\lambda)$ is defined as in Definition 2.1.

Similarly, the CI hash family of Peikert and Shiehian can satisfy the third requirement with one slight change. We just have to modify [87, Construction 3.1] by having their $\mathsf{StatGen}$ algorithm choose the LWE secret from the noise distribution instead of choosing it uniformly. Again, the reduction of [7, Lemma 2] can be used to construct N independent instances of $\mathsf{HNF\text{-}LWE}$ out of a single LWE instance with uniform secret. In turn, these N $\mathsf{HNF\text{-}LWE}$ instances can be used to set up N CI hashing keys $\{k^{(i)}\}_{i=1}^{N}$, each of which contains an $\mathsf{HNF\text{-}LWE}$ instance

$$\left[\begin{array}{c}\bar{\mathbf{A}}^{(i)} \\ \hline {\mathbf{s}^{(i)}}^\top \bar{\mathbf{A}}^{(i)} + {\mathbf{e}^{(i)}}^\top\end{array}\right] \in \mathbb{Z}_q^{n\times m}, \qquad \left[\begin{array}{c}\bar{\mathbf{a}}^{(i)} \\ \hline {\mathbf{s}^{(i)}}^\top \bar{\mathbf{a}}^{(i)} + {e^{(i)}}^\top - \lfloor q/2 \rfloor\end{array}\right] \in \mathbb{Z}_q^{n}.$$

It can be checked that the proof of [87, Theorem 3.5] extends to the N instance setting when each $\mathbf{s}^{(i)}$ is sampled from the noise distribution.

F.4 Setting the Parameters

We now specify choices of parameters that are compatible with the provided constructions.

Let λ be the security parameter. For the correctness and KDM-CPA security of the ACPS system [7], we can set parameters $n, m, r, r', \alpha, p, q$ as follows.

- Dimensions $n = \Omega(\lambda)$, $m = 2(n+1)\lceil \log q \rceil$, and Gaussian parameters $r = \omega(\sqrt{\log m})$, $r' = \Omega(r\sqrt{m})$.
- Prime p, modulus $q = p^2$ and parameter α such that $\alpha q = \Omega(\sqrt{n})$ and $r\alpha q m + r' \ll p$.

For the trapdoor Σ-protocol in Section 5, if $\kappa = \Theta(\lambda)$ is the number of parallel repetitions, then we can work with parameters $B_r = r\sqrt{m}$, $B_\chi = r'\omega(\log m)$, $\sigma_{\mathrm{eq}} \geq \log(2\kappa(m+1)) \cdot \sqrt{\kappa(B_r^2 + B_\chi^2)}$, and $B^* = 3\sigma_{\mathrm{eq}}\sqrt{2m+2} \cdot \alpha q \sqrt{m}$ such that $B^* \ll p$. To meet all these conditions, we can choose prime $p = \widetilde{O}(n^{3.5})$ and modulus $q = p^2 = \widetilde{O}(n^7)$.

In the USS argument of Section 3, we need to encrypt the binary decomposition of the first message $\mathbf{a}$ of the prover in the trapdoor Σ-protocol of Section 5, via the $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme of Section 3.1. Since the bit-size of $\mathbf{a}$ is $2(n+1)\lceil \log q \rceil$, we can set parameters (n_0', n', m', q') and σ', α' of the $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme as follows:

$$n_0' = 2(n+1)\lceil \log q \rceil, \quad n' = n_0' + \Omega(\lambda), \quad m' = 2n'\lceil \log q' \rceil + O(\lambda),$$
$$\sigma' = O(m'L), \quad q' = \Omega({m'}^{5/2}{n'}^{1/2}L^2), \quad \alpha' \cdot q' = \Omega(\sqrt{n'}),$$

where $L = \Theta(\lambda)$.

As for USS argument of Section 4, we need to encrypt $2(n+1)\lceil \log q \rceil$ bits of the prover's message via the $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme of Section 4.1. To this end, we can set parameters (n_0'', n'', m'', q'') and σ'', α'' of the $\mathcal{R}_{\mathsf{PRF}}$-lossy PKE scheme as follows:

$$n_0'' = 2(n+1)\lceil \log q \rceil, \quad n'' = n_0'' + \Omega(\lambda), \quad m'' = 2n''\lceil \log q'' \rceil + O(\lambda),$$
$$\sigma'' = 4^d O({m''}^2), \quad q'' = \Omega(4^d {m''}^{4.5}\sigma''), \quad \alpha'' \cdot q'' = \Omega(\sqrt{n''}),$$

where d is the depth of the underlying PRF circuit.

G Trapdoor Σ-Protocols for LWE Languages

G.1 The Case of LWE with Uniform Secrets

We describe a natural trapdoor Σ-protocol inspired by the Gap Σ-protocol of Asharov *et al.* [9,8] for the language $\mathcal{L}_{B,B^*} = \{\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}}\}$, where

$$\mathcal{L}_{\mathsf{zk}} := \left\{(\mathbf{B}, \mathbf{y}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \mid \exists \mathbf{s} \in \mathbb{Z}^n : \|\mathbf{y} - \mathbf{B} \cdot \mathbf{s}\| \leq B\right\},$$
$$\mathcal{L}_{\mathsf{sound}} := \left\{(\mathbf{B}, \mathbf{y}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \mid \exists \mathbf{s} \in \mathbb{Z}^n : \|\mathbf{y} - \mathbf{B} \cdot \mathbf{s}\| \leq B^*\right\},$$

where $\mathcal{L}_{\mathsf{zk}} \subseteq \mathcal{L}_{\mathsf{sound}}$ when $B \leq B^* \sqrt{m}$.

Here, we use rejection sampling to reduce the multiplicative gap between B and B^* so as to have $\frac{B^*}{B} = \mathsf{poly}(\lambda)$ (whereas [9] requires $B/B^* = \mathsf{negl}(\lambda)$ for the statistical ZK property), which allows using a polynomial modulus q.

$\mathsf{\textbf{Gen}}_{\mathsf{par}}(1^\lambda)$: On input of a security parameter λ, choose integers B, B^* such that $B < B^* \in \mathsf{poly}(\lambda)$, a modulus q, dimensions n, m, an error rate $\alpha > 0$ and a standard deviation $\sigma_{\mathrm{lwe}} > B \cdot \log m$. Define $\mathsf{par} = \{\lambda, q, n, m, \alpha, \sigma_{\mathrm{lwe}}\}$.

$\mathsf{\textbf{Gen}}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}_{B,B^*})$: Given public parameters par and a description of a language $\mathcal{L}_{B,B^*}$ which specifies real numbers $B, B^* > 0$ and a matrix distribution $D_{\mathbf{B}}$, sample a matrix $\mathbf{B} \hookleftarrow D_{\mathbf{B}}$ and define $\mathsf{crs}_{\mathcal{L}} = \{\mathbf{B}, B, B^*\}$. The global common reference string consists of $\mathsf{crs} = \big(\{\lambda, q, n, m, \alpha, \sigma_{\mathrm{lwe}}\}, \{\mathbf{B}, B, B^*\}\big)$.

$\mathsf{\textbf{TrapGen}}(\mathsf{par}, \tau_{\mathcal{L}_{B,B^*}})$: On input of public parameters par and a trapdoor $\tau_{\mathcal{L}_{B,B^*}}$ for the language $\mathcal{L}_{B,B^*}$, which consists of a matrix $\tau_{\mathcal{L}_{B,B^*}} = \mathbf{T_B}$ produced as $(\mathbf{B}, \mathbf{T_B}) \leftarrow \mathsf{TrapSamp}_{\mathbf{B}}(1^\lambda, 1^n, 1^m, q)$, it outputs $\mathsf{crs}_{\mathcal{L}} = \{\mathbf{B}, B, B^*\}$, which defines $\mathsf{crs} = \big(\{\lambda, q, n, m, \alpha, \sigma_{\mathrm{lwe}}\}, \{\mathbf{B}, B, B^*\}\big)$, as well as $\tau_\Sigma = \mathbf{T_B}$.

$\mathsf{\textbf{P}}\big(\mathsf{crs}, \mathbf{y}, (\mathbf{s}, \mathbf{e})\big) \leftrightarrow \mathsf{\textbf{V}}(\mathsf{crs}, \mathbf{y})$: Given crs and a statement $\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$, P (who has $\mathbf{w} = (\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ such that $\|\mathbf{e}\| \leq B$) and V interact as follows.

1. P samples $\mathbf{s}' \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{e}' \hookleftarrow D_{\mathbb{Z}^m, \sigma_{\mathrm{lwe}}}$ and computes its first message as $\mathbf{a} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \in \mathbb{Z}_q^m$, which is sent to V.
2. V sends a random challenge $\mathsf{Chall} \in \{0, 1\}$ to P.
3. P defines $\mathbf{z} = \mathbf{s}' + \mathsf{Chall} \cdot \mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{z}_e = \mathbf{e}' + \mathsf{Chall} \cdot \mathbf{e} \in \mathbb{Z}^m$. It sends $(\mathbf{z}, \mathbf{z}_e) \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ to V with probability $\theta = \min\left(\frac{D_{\mathbb{Z}^m, \sigma_{\mathrm{lwe}}}(\mathbf{z}_e)}{M \cdot D_{\mathbb{Z}^m, \sigma_{\mathrm{lwe}}, \mathsf{Chall} \cdot \mathbf{e}}(\mathbf{z}_e)}, 1\right)$, where $M = e^{12/\log(m) + 1/(2\log^2(m))}$. With probability $1 - \theta$, P aborts. .
4. Upon receiving $(\mathbf{z}, \mathbf{z}_e) \in \mathbb{Z}_q^n \times \mathbb{Z}^m$, V checks if $\|\mathbf{z}_e\| \leq \sigma_{\mathrm{lwe}} \cdot \sqrt{m}$ and
$$\mathbf{a} + \mathsf{Chall} \cdot \mathbf{y} = \mathbf{B} \cdot \mathbf{z} + \mathbf{z}_e \mod q.$$

$\mathsf{\textbf{BadChallenge}}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, \mathbf{y}, \mathbf{a}\big)$: Given $\tau_\Sigma = \mathbf{T_B}$, parse the first prover message as $\mathbf{a} \in \mathbb{Z}_q^m$. Using the trapdoor $\mathbf{T_B}$, determine if there exist vectors $\mathbf{s}' \in \mathbb{Z}_q^n$ and $\mathbf{e}' \in \mathbb{Z}^m$ such that $\mathbf{a} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$ and $\|\mathbf{e}'\| < B^*/2$. If no such pair $(\mathbf{s}', \mathbf{e}')$ exists, set $\mathsf{Chall} = 1$. Otherwise, set $\mathsf{Chall} = 0$ if there exists no pair $(\mathbf{s}', \mathbf{e}') \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ such that $\mathbf{a} + \mathbf{y} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$ and $\|\mathbf{e}'\| < B^*/2$. In any other case, set $\mathsf{Chall} = \perp$.

The intuition of the $\mathsf{BadChallenge}$ function is that, if $\mathbf{y} \notin \mathcal{L}_{\mathsf{sound}}$, there exists at least one $b \in \{0, 1\}$ such that $\mathbf{a} + b \cdot \mathbf{y}$ cannot be written as $\mathbf{B} \cdot \mathbf{s}' + \mathbf{e}'$ for some $\mathbf{e}' \in \mathbb{Z}^m$ of norm $\|\mathbf{e}'\| < B^*/2$. In this case, the bad challenge can only be $1 - b$ if it exists.

Lemma G.1. *The above construction is a trapdoor Σ-protocol for $\mathcal{L}_{B,B^*}$ when $B^* > 2\sigma_{\mathrm{lwe}}\sqrt{m}$.*

Proof. We first prove the special ZK property. Given a statement $\mathbf{y} \in \mathcal{L}_{\mathsf{zk}}$ and a challenge $\mathsf{Chall}^* \in \{0, 1\}$, the simulator first samples vectors $\mathbf{z}^* \hookleftarrow U(\mathbb{Z}_q^n)$,

$\mathbf{z}_e^* \hookleftarrow D_{\mathbb{Z}^m,\sigma_{\mathrm{lwe}}}$ and computes $\mathbf{a}^* = \mathbf{B} \cdot \mathbf{z}^* + \mathbf{z}_e^* - \mathsf{Chall}^* \cdot \mathbf{y} \bmod q$. It outputs $\big(\mathbf{a}^*, \mathsf{Chall}^*, (\mathbf{z}^*, \mathbf{z}_e^*)\big)$ with probability $1/M$. Note that $\big(\mathbf{a}^*, \mathsf{Chall}^*, (\mathbf{z}^*, \mathbf{z}_e^*)\big)$ is an accepting transcript. We now show that it is statistically indistinguishable from a real transcript. If $\mathbf{y} \in \mathcal{L}_{\mathsf{zk}}$, there exists $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ with $\|\mathbf{e}\| \leq B$ such that $\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e}$. The distribution of $\mathbf{z}_e$ in a real transcript is thus $D_{\mathbb{Z}^m,\sigma_{\mathrm{lwe}},\mathsf{Chall}\cdot\mathbf{e}}$. By Lemma E.1, the distribution of the simulated $\mathbf{z}_e^*$ is within statistical distance $2^{-100}/M$ from that of a real non-aborting transcript. Moreover, in both the real protocol and the simulation, the first prover's message $\mathbf{a}^*$ is uniquely determined by the statement $\mathbf{y}$, the challenge Chall and the response $(\mathbf{z}, \mathbf{z}_e)$.

Soundness can be shown as in [9, Theorem F.1]. By subtracting the verification equations for a given $\mathbf{a} \in \mathbb{Z}_q^m$ and two distinct $\mathsf{Chall}_0, \mathsf{Chall}_1 \in \{0,1\}$, we obtain that there exist $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}^m$ of norm $\|\mathbf{e}\| \leq 2\sigma_{\mathrm{lwe}}\sqrt{m}$ such that $\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e}$, which implies $\mathbf{y} \in \mathcal{L}_{\mathsf{sound}}$ since $B^* > 2\sigma_{\mathrm{lwe}}\sqrt{m}$.

We are left with showing that $\mathsf{BadChallenge}$ provides the correct result. First, we note that there cannot exist $\mathbf{z}_0, \mathbf{z}_1 \in \mathbb{Z}_q^n$ and $\mathbf{z}_{e,0}, \mathbf{z}_{e,1} \in \mathbb{Z}^m$ such that

$$\mathbf{a} + b \cdot \mathbf{y} = \mathbf{B} \cdot \mathbf{z}_b + \mathbf{z}_{e,b} \mod q \tag{27}$$

and $\|\mathbf{z}_{e,b}\| \leq B^*/2$ for each $b \in \{0,1\}$ unless $\mathbf{y} \in \mathcal{L}_{\mathsf{sound}}$. Consequently, if $\mathbf{y} \notin \mathcal{L}_{\mathsf{sound}}$, there exists $b \in \{0,1\}$ such that $\mathbf{a} + b \cdot \mathbf{y}$ cannot be written as $\mathbf{a} + b \cdot \mathbf{y} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}'$ for any $(\mathbf{s}, \mathbf{e}') \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ such that $\|\mathbf{e}'\| \leq B^*/2$. For a given message $\mathbf{a} \in \mathbb{Z}_q^m$ sent by the prover, let us assume that there exists no pair $(\mathbf{s}', \mathbf{e}') \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ such that $\mathbf{a} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$ and $\|\mathbf{e}'\| < B^*/2$ (note that $\mathsf{BadChallenge}$ can detect this using the trapdoor $\mathbf{T}_{\mathbf{B}}$). In this case, no valid response can exist for the challenge $\mathsf{Chall} = 0$ since $\sigma_{\mathrm{lwe}}\sqrt{m} < B^*/2$. Now, let us assume that there exists no $(\mathbf{s}', \mathbf{e}') \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ such that $\mathbf{a} + \mathbf{y} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$ and $\|\mathbf{e}'\| < B^*/2$. In this case, we know that no valid response exists for $\mathsf{Chall} = 1$. Hence, if $\mathbf{y} \notin \mathcal{L}_{\mathsf{sound}}$, $\mathsf{BadChallenge}$ always outputs the only $\mathsf{Chall} \in \{0,1\}$ for which a valid response can possibly exist for a given $\mathbf{a} \in \mathbb{Z}_q^m$. □

On Parallel Repetitions. To achieve negligible soundness error, the protocol is repeated $\kappa = \Theta(\lambda)$ times in parallel by first computing $(\mathbf{a}_1, \ldots, \mathbf{a}_\kappa)$ before obtaining the challenge $\mathsf{Chall} = \mathsf{Chall}[1] \ldots \mathsf{Chall}[\kappa]$ and computing the response $\mathbf{z} = (\mathbf{z}[1], \ldots, \mathbf{z}[\kappa])$. We then handle $\mathbf{z}$ as a vector of $\mathbb{Z}^{\kappa \cdot m}$ and reject it with probability $\theta = \min\left(\frac{D_{\mathbb{Z}^{\kappa \cdot m},\sigma_{\mathrm{lwe}}}(\mathbf{z}_e)}{M \cdot D_{\mathbb{Z}^{\kappa \cdot m},\sigma_{\mathrm{lwe}},\mathsf{Chall}\cdot(\mathbf{1}^\kappa \otimes \mathbf{e})}(\mathbf{z}_e)}, 1\right)$, where $\mathbf{z}_e = (\mathbf{z}_e[1], \ldots, \mathbf{z}_e[\kappa])$ and $M = e^{12/\log(\kappa \cdot m) + 1/(2\log^2(\kappa \cdot m))}$. Here, we need to set $\sigma_{\mathrm{lwe}} > B\sqrt{\kappa} \cdot \log(\kappa \cdot m)$.

G.2 A Trapdoor Σ-Protocol for HNF-LWE

We now adapt the trapdoor Σ-protocol of Appendix G.1 to the case of $\mathsf{HNF\text{-}LWE}$ [7], where the secret $\mathbf{s} \in \mathbb{Z}^n$ is sampled from the noise distribution. We describe a trapdoor Σ-protocol for the language $\mathcal{L} = \{\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}}\}$, where

$$\begin{aligned}
\mathcal{L}_{\mathsf{zk}} &:= \left\{(\mathbf{B}, \mathbf{y}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \mid \exists \mathbf{s} \in \mathbb{Z}^n : \|\mathbf{y} - \mathbf{B} \cdot \mathbf{s}\| \leq B_e \ \wedge \ \|\mathbf{s}\| \leq B_s\right\}, \\
\mathcal{L}_{\mathsf{sound}} &:= \left\{(\mathbf{B}, \mathbf{y}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^m \mid \exists \mathbf{s} \in \mathbb{Z}^n : \|\mathbf{y} - \mathbf{B} \cdot \mathbf{s}\| \leq B_e^* \ \wedge \ \|\mathbf{s}\| \leq B_s^*\right\},
\end{aligned}$$

where $\mathcal{L}_{\mathsf{zk}} \subseteq \mathcal{L}_{\mathsf{sound}}$ when $B_s \leq B_s^* \sqrt{n}$ and $B_e \leq B_e^* \sqrt{m}$.

As before, the $\mathsf{TrapGen}$ algorithm uses a membership-testing trapdoor $\tau_{\mathcal{L}}$ that consists of a small-norm full-rank integer matrix $\mathbf{T_B} \in \mathbb{Z}^{m \times m}$ such that $\mathbf{T_B} \cdot \mathbf{B} = \mathbf{0}^{m \times n} \bmod q$.

$\mathsf{\textbf{Gen}}_{\mathsf{par}}(1^\lambda)$: On input of a security parameter λ, choose integers B, B^* such that $B < B^* \in \mathsf{poly}(\lambda)$, a modulus q, dimensions n, m, an error rate $\alpha > 0$ and standard deviation $\sigma_{\text{hlwe}} > \sqrt{B_s^2 + B_e^2} \cdot \log(m+n)$. Define public parameters $\mathsf{par} = \{\lambda, q, n, m, \alpha, \sigma_{\text{hlwe}}\}$.

$\mathsf{\textbf{Gen}}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$: Given public parameters par and a description of a language $\mathcal{L}$ which specifies real numbers $B_s, B_s^*, B_e, B_e^* > 0$ and a matrix distribution $D_{\mathbf{B}}$, sample a matrix $\mathbf{B} \hookleftarrow D_{\mathbf{B}}$ and define $\mathsf{crs}_{\mathcal{L}} = \{\mathbf{B}, B_s, B_s^*, B_e, B_e^*\}$. The global common reference string consists of

$$\mathsf{crs} = \big(\{\lambda, q, n, m, \alpha, \sigma_{\text{hlwe}}\}, \{\mathbf{B}, B_s, B_s^*, B_e, B_e^*\}\big).$$

$\mathsf{\textbf{TrapGen}}(\mathsf{par}, \tau_{\mathcal{L}})$: On input of public parameters par and a trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}$, which consists of a matrix $\tau_{\mathcal{L}} = \mathbf{T_B}$ produced as $(\mathbf{B}, \mathbf{T_B}) \leftarrow \mathsf{TrapSamp}_{\mathbf{B}}(1^\lambda, 1^n, 1^m, q)$, it outputs $\mathsf{crs}_{\mathcal{L}} = \{\mathbf{B}, B_s, B_e, B_s^*, B_e^*\}$, which defines $\mathsf{crs} = \big(\{\lambda, q, n, m, \alpha, \sigma_{\text{hlwe}}\}, \{\mathbf{B}, B_s, B_s^*, B_e, B_e^*\}\big)$, as well as $\tau_{\Sigma} = \mathbf{T_B}$.

$\mathsf{\textbf{P}}\big(\mathsf{crs}, \mathbf{y}, (\mathbf{s}, \mathbf{e})\big) \leftrightarrow \mathsf{\textbf{V}}(\mathsf{crs}, \mathbf{y})$: Given crs, a statement $\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e} \in \mathbb{Z}_q^m$ and P (who has the witness $\mathbf{w} = [\mathbf{s}^\top \mid \mathbf{e}^\top]^\top \in \mathbb{Z}^{n+m}$ such that $\|\mathbf{s}\| \leq B_s$, $\|\mathbf{e}\| \leq B_e$) and V interact in the following way.

1. P samples $\mathbf{s}' \hookleftarrow D_{\mathbb{Z}^n, \sigma_{\text{hlwe}}}$, $\mathbf{e}' \hookleftarrow D_{\mathbb{Z}^m, \sigma_{\text{hlwe}}}$, defines $\mathbf{r}_w = \begin{bmatrix} \mathbf{s}' \\ \mathbf{e}' \end{bmatrix}$ and computes
$$\mathbf{a} = [\mathbf{B} \mid \mathbf{I}_m] \cdot \mathbf{r}_w \in \mathbb{Z}_q^m,$$
which is sent to V.
2. V sends a random challenge $\mathsf{Chall} \in \{0, 1\}$ to P.
3. P computes $\mathbf{z} = [\mathbf{z}_s{}^\top \mid \mathbf{z}_e{}^\top]^\top = \mathbf{r}_w + \mathsf{Chall} \cdot \mathbf{w} \in \mathbb{Z}^{n+m}$. and sends it to V with probability $\theta = \min\left(\frac{D_{\mathbb{Z}^{n+m}, \sigma_{\text{hlwe}}}(\mathbf{z})}{M \cdot D_{\mathbb{Z}^{n+m}, \sigma_{\text{hlwe}}, \mathsf{Chall} \cdot \mathbf{w}}(\mathbf{z})}, 1\right)$, where $M = e^{12/\log(m+n) + 1/(2\log^2(m+n))}$. With probability $1 - \theta$, P aborts. .
4. Upon receiving $\mathbf{z} = [\mathbf{z}_s{}^\top \mid \mathbf{z}_e{}^\top]^\top \in \mathbb{Z}^{n+m}$, V checks if $\|\mathbf{z}_s\| \leq \sigma_{\text{hlwe}} \cdot \sqrt{n}$, $\|\mathbf{z}_e\| \leq \sigma_{\text{hlwe}} \cdot \sqrt{m}$, and and
$$\mathbf{a} + \mathsf{Chall} \cdot \mathbf{y} = [\mathbf{B} \mid \mathbf{I}_m] \cdot \mathbf{z}.$$
If these conditions are not met, V halts and returns $\perp$.

$\mathsf{\textbf{BadChallenge}}\big(\mathsf{par}, \tau_{\Sigma}, \mathsf{crs}, \mathbf{y}, \mathbf{a}\big)$: Given $\tau_{\Sigma} = \mathbf{T_B}$, parse the first prover message as $\mathbf{a} \in \mathbb{Z}_q^m$ and do the following.

1. Using $\mathbf{T_B}$, determine if there exist vectors $\mathbf{s}' \in [-B_s^*/2, B_s^*/2]^n$ and $\mathbf{e}' \in [-B_e^*/2, B_e^*/2]^m$ such that $\mathbf{a} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$. If not, return $\mathsf{Chall} = 1$.

2. Return $\mathsf{Chall} = 0$ if there exist no vectors $\mathbf{s}' \in [-B_s^*/2, B_s^*/2]^n$ and $\mathbf{e}' \in [-B_e^*/2, B_e^*/2]^m$ such that $\mathbf{a} + \mathbf{y} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$.

In any other case, return $\mathsf{Chall} = \perp$.

Lemma G.2. *If $\sigma_{\mathrm{hlwe}} > B \cdot \log(m+n)$, $B_s^* > 2\sigma_{\mathrm{hlwe}}\sqrt{n}$ and $B_e^* > 2\sigma_{\mathrm{hlwe}}\sqrt{m}$, the above construction is a trapdoor Σ-protocol for $\mathcal{L}$.*

Proof. We first consider the special ZK property. Given a statement $\mathbf{y} \in \mathcal{L}_{\mathsf{zk}}$ and a challenge $\mathsf{Chall}^* \in \{0,1\}$, the simulator first samples $\mathbf{z}_s^* \hookleftarrow D_{\mathbb{Z}^n, \sigma_{\mathrm{hlwe}}}$, $\mathbf{z}_e^* \hookleftarrow D_{\mathbb{Z}^m, \sigma_{\mathrm{hlwe}}}$ and computes $\mathbf{a}^* = \mathbf{B} \cdot \mathbf{z}_s^* + \mathbf{z}_e^* - \mathsf{Chall}^* \cdot \mathbf{y} \bmod q$. It outputs $\big(\mathbf{a}^*, \mathsf{Chall}^*, (\mathbf{z}_s^*, \mathbf{z}_e^*)\big)$ with probability $1/M$. By construction, $\big(\mathbf{a}^*, \mathsf{Chall}^*, (\mathbf{z}_s^*, \mathbf{z}_e^*)\big)$ is an accepting transcript and we show that it is statistically indistinguishable from a real transcript. If $\mathbf{y} \in \mathcal{L}_{\mathsf{zk}}$, there exists $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}^n \times \mathbb{Z}^m$ with $\|\mathbf{s}\| \leq B_s$ and $\|\mathbf{e}\| \leq B_e$ such that $\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e}$. The distribution of $(\mathbf{z}_s, \mathbf{z}_e)$ in a real transcript is thus $D_{\mathbb{Z}^n, \sigma_{\mathrm{hlwe}}, \mathsf{Chall} \cdot \mathbf{s}} \times D_{\mathbb{Z}^m, \sigma_{\mathrm{hlwe}}, \mathsf{Chall} \cdot \mathbf{e}}$. By Lemma E.1 and our choice of $\sigma_{\mathrm{hlwe}} > \sqrt{B_s^2 + B_e^2} \cdot \log(m+n)$, the distribution of the simulated $(\mathbf{z}_s^*, \mathbf{z}_e^*)$ is within statistical distance $2^{-100}/M$ from that of a real non-aborting transcript. Moreover, in both the real protocol and the simulation, the statement $\mathbf{y}$, the challenge Chall and the response $(\mathbf{z}_s, \mathbf{z}_e)$ uniquely determine the first prover's message $\mathbf{a}^*$.

Soundness can be shown using the usual arguments. Let us assume that, for a given $\mathbf{a} \in \mathbb{Z}_q^m$, there exist two valid responses $\mathbf{z}_b = [\mathbf{z}_{s,b}^\top \mid \mathbf{z}_{e,b}^\top]^\top \in \mathbb{Z}^{n+m}$ with $\|\mathbf{z}_{s,b}\| \leq \sigma_{\mathrm{hlwe}} \cdot \sqrt{n}$, $\|\mathbf{z}_{e,b}\| \leq \sigma_{\mathrm{hlwe}} \cdot \sqrt{m}$ for each $b \in \{0,1\}$ and such that

$$\mathbf{a} + b \cdot \mathbf{y} = [\mathbf{B} \mid \mathbf{I}_m] \cdot \mathbf{z}_b \mod q.$$

By subtracting them, we have $\mathbf{y} = [\mathbf{B} \mid \mathbf{I}_m] \cdot \mathbf{z} \bmod q$, where

$$\mathbf{z} = \mathbf{z}_1 - \mathbf{z}_0 = \big[(\mathbf{z}_{s,1} - \mathbf{z}_{s,0})^\top \mid (\mathbf{z}_{e,1} - \mathbf{z}_{e,0})^\top\big]^\top \in \mathbb{Z}^{n+m},$$

where $\|\mathbf{z}_{s,1} - \mathbf{z}_{s,0}\| \leq 2\sigma_{\mathrm{hlwe}}\sqrt{n} < B_s^*$, $\|\mathbf{z}_{e,1} - \mathbf{z}_{e,0}\| \leq 2\sigma_{\mathrm{hlwe}}\sqrt{m} < B_e^*$.

We now show that $\mathsf{BadChallenge}$ provides the correct result. First, assuming that $\mathbf{y} \notin \mathcal{L}_{\mathsf{sound}}$, for a given $\mathbf{a} \in \mathbb{Z}_q^m$, there cannot exist two distinct pairs $(\mathbf{s}_0', \mathbf{e}_0'), (\mathbf{s}_1', \mathbf{e}_1') \in [-B_s^*/2, B_s^*/2]^n \times [-B_e^*/2, B_e^*/2]^m$ such that

$$\mathbf{a} + b \cdot \mathbf{y} = [\mathbf{B} \mid \mathbf{I}_m] \cdot \begin{bmatrix} \mathbf{s}_b' \\ \mathbf{e}_b' \end{bmatrix} \mod q \qquad \forall b \in \{0,1\}.$$

Let us assume that there exist no $\mathbf{s}' \in [-B_s^*/2, B_s^*/2]^n$ and $\mathbf{e}' \in [-B_e^*/2, B_e^*/2]^m$ such that $\mathbf{a} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$ (note that $\mathsf{BadChallenge}$ can detect this using $\mathbf{T_B}$). Via the same argument as in the proof of Lemma G.1, this rules out the existence of a valid response for $\mathsf{Chall} = 0$. Now, assuming that there exist no $\mathbf{s}' \in [-B_s^*/2, B_s^*/2]^n$ and $\mathbf{e}' \in [-B_e^*/2, B_e^*/2]^m$ such that $\mathbf{a} + \mathbf{y} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$, we are guaranteed that no valid response can exist for $\mathsf{Chall} = 1$. Since $\mathbf{y} \notin \mathcal{L}_{\mathsf{sound}}$ implies that at least one of these two events occurs, we conclude that $\mathsf{BadChallenge}$ always outputs the only $\mathsf{Chall} \in \{0,1\}$ that admits a valid response if such a response exists at all for the given $\mathbf{a}$. □