Simulation-Sound Proofs 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 proof system for the LWE relation. In turn, this relation makes it possible to express the equality of plaintexts encrypted under different keys in the dual Regev cryptosystem. As an application, we obtain an LWE-based public-key encryption scheme for which we can prove key-dependent message (KDM-CCA2) security under chosen-ciphertext attacks in the standard model.

Keywords. LWE, standard model, Naor-Yung, KDM-CCA security, NIZK proofs, simulation-soundness.

1 Introduction

The Fiat-Shamir transformation [47] is a well-known technique that turns any 3-move honest-verifier zero-knowledge proof system (a.k.a. Σ-protocol [40]) 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 [91,57,73] and chosen-ciphertext-secure public-key encryption schemes [94,48,1,17]. In the standard model, however, counter-examples [60] 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.* [29] and Peikert and Shiehian [89], which imply the existence of Fiat-Shamir-based NIZK proofs for all NP languages under the sole Learning-With-Errors (LWE) assumption [92]. Their results followed a line of research [95,31,70,28] 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 [29,89] 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 [87,93,48].

In order to instantiate the Naor-Yung paradigm of CCA2-secure encryption [87] 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 [46] with the modifications suggested by Canetti *et al.* [29,34]. In addition, a direct application of [29,34,89] to CCA2 security requires to apply the generic compiler of [43] that turns any NIZK proof system into simulation-sound [93] proofs. In this paper, we consider the problem of more efficiently instantiating the Naor-Yung paradigm in the standard model under lattice assumptions. Using correlation intractable hash functions, our goal is to directly construct simulation-sound proofs of plaintext equality *without* going through a reduction to an NP complete problem.

1.1 Our Contributions

We describe the first non-trivial instantiation of the Naor-Yung paradigm under lattice assumptions. As an application, we obtain a direct construction of a public key encryption scheme for which we can prove key-dependent message security under chosen-ciphertext attacks (or KDM-CCA2 security for short) under the standard Learning-With-Errors (LWE) assumption [92]. By "non-trivial" and "direct construction", we mean that our scheme is *not* the result of merely combining generic NIZK techniques [93,43] with the recent results [29,34,89] on NIZK proofs based on correlation intractable hash functions. In particular, we bypass the use of a proof system for the graph Hamiltonicity language [46,29,34].

Instead, as a key building block, we directly construct a simulation-sound NIZK proof system showing that two dual Regev ciphertexts [55] are encryptions of the same plaintext. We show that our proof system provides *unbounded* simulation-soundness [43] (as opposed to one-time simulation-soundness [93,79]), meaning that the adversary remains unable to prove a false statement, even after having seen simulated proofs for polynomially many (possibly false) statements. This makes our proof system suitable to prove KDM-CCA2 security by applying the Naor-Yung technique to variants [4,66] of the dual Regev cryptosystem that

are known to provide key-dependent message security for affine functions.

As a result, we obtain a public-key encryption (PKE) scheme for which we can prove KDM-CCA2 security under the LWE assumption with polynomial approximation factors. 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 [7,4], 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.

1.2 Technical Overview

Our starting point is a trapdoor Σ-protocol [29,34] for the LWE language. Namely, it allows proving that a given vector $\mathbf{y} \in \mathbb{Z}_q^m$ is of the form $\mathbf{y} = \mathbf{B}^\top \cdot \mathbf{s} + \mathbf{e}$, for a public matrix $\mathbf{B} \in \mathbb{Z}_q^{n \times m}$ and secret vectors $\mathbf{s} \in \mathbb{Z}_q^n$, $\mathbf{e} \in [-B, B]^m$ such that $m > n$ and $B \ll q$. Recall that a standard Σ-protocol [40,39] is a 3-move protocol with transcripts of the form $(\mathbf{a}, \mathbf{c}, \mathbf{z})$ where $\mathbf{c}$ 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 Σ-protocol [29,34] has the property that, for any statement x outside the language L and any first message $\mathbf{a}$ sent by the prover, a trapdoor makes it possible to determine the unique challenge $\mathbf{c}$ for which a valid response $\mathbf{z}$ exists. There is an efficiently computable function BadChallenge that takes as input a trapdoor τ, a false statement $x \notin L$, and a first prover message $\mathbf{a}$, and computes the unique $\mathbf{c}$ such that there exists an accepting transcript $(\mathbf{a}, \mathbf{c}, \mathbf{z})$ (that is, there is no accepting transcript of the form $(\mathbf{a}, \mathbf{c}', \mathbf{z})$ for any $\mathbf{c}' \neq \mathbf{c}$).

Our first observation is that the Σ-protocol of Asharov *et al.* [9,8] can be turned into a trapdoor Σ-protocol for the LWE language. Indeed, if we know a short basis for the lattice $\Lambda^\perp(\mathbf{B})$, we can determine the unique binary challenge for which a given first prover message $\mathbf{a}$ admits a valid response $\mathbf{z}$ (assuming that the statement $\mathbf{y}$ is not a vector of the form $\mathbf{y} = \mathbf{B}^\top \cdot \mathbf{s} + \mathbf{e}$, for some small $\mathbf{e} \in \mathbb{Z}^m$). While very simple, the resulting trapdoor Σ-protocol actually requires a super-polynomial modulus q as its honest-verifier zero-knowledge property relies on the noise flooding technique (see, e.g., [9]). In order to work with a polynomial modulus and inverse error rate in the LWE assumption, we can actually use a statistical honest-verifier zero-knowledge protocol due to Micciancio and Vadhan [86], which was previously used to prove similar languages in [61]. Again, we rely on the observation that a BadChallenge function is efficiently computable using a trapdoor for the lattice $\Lambda^\perp(\mathbf{B})$. Using the Micciancio-Vadhan protocol, we thus obtain a trapdoor Σ-protocol for proving that two dual Regev ciphertexts decrypt to the same plaintext since this is equivalent to stating that a ciphertext-dependent vector $\mathbf{y} \in \mathbb{Z}_q^m$ is of the form $\mathbf{y} = \mathbf{B}^\top \cdot \mathbf{s} + \mathbf{e}$, for some small $\mathbf{e} \in \mathbb{Z}^m$.

The main difficulty, however, is to turn the aforementioned trapdoor Σ-protocol into a non-interactive proof system with unbounded simulation-soundness. This problem is non-trivial since the Canetti *et al.* protocol [29,34] is not known to satisfy this security notion[6]. The NIZK simulator of [29,34] generates simulated proofs by "programming" the CI hash function from which the verifier's challenge is derived. In the context of unbounded simulation-soundness [93,43], 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.* [34,29] in the original Feige-Lapidot-Shamir protocol [46]. In [34, Section 5.2], the first prover message $\mathbf{a}$ is computed using a lossy encryption scheme [16,13] 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.* [13,16] that specific lossy encryption schemes [58,88] 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.* [34,29] is instantiated using a lossy encryption scheme with efficient opening, we can use a strategy introduced by Damgård [42] to simulate NIZK proofs without programming the CI hash function. Namely, we can generate the first prover message as a lossy encryption of 0. When receiving the verifier's challenge $\mathbf{c}$, we 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 $\mathbf{a}$.

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 that a ciphertext encrypted for a tag $t \in \mathcal{T}$ is injective if $\mathcal{R}(K, t) = 1$ and lossy otherwise. Boyle *et al.* [23] gave $\mathcal{R}$-lossy encryption schemes that (with noticeable

[6] It can be generically achieved using NIZK for general NP relations [43] 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 [29,34]

probability) are lossy for polynomially many tags and injective on an adversarially chosen tag. 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 provide a construction of $\mathcal{R}$-lossy encryption that satisfies our syntactic/security definitions under the LWE assumption. The scheme can be viewed as a combination of the primal Regev cryptosystem [92] – which is known [90] to be a lossy PKE scheme and is easily seen to support efficient openings as defined in [16,13] – 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 $\mathbf{c} = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_t + (1 - \mathcal{R}(K,t)) \cdot \mathbf{G}] \cdot \mathbf{r} + [\mathbf{0} \mid \boldsymbol{\mu} \cdot (q/2)]^\top$, 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 $\Lambda^\perp(\mathbf{A})$ to equivocate lossy ciphertexts for any arbitrary tag.

Our simulation-sound proof 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 Σ-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 zero-knowledge protocols described by Garay, MacKenzie and Yang [52,82] and Gennaro [54]. Our extension of $\mathcal{R}$-lossy encryption actually resembles their notion of simulation-sound trapdoor commitments. The difference is that, while [82] only requires commitments to be computationally binding for tags that have

never been equivocated, we need adversarially-chosen tags to be statistically binding and even extractable.

1.3 Related Work

Fiat-Shamir in the Standard Model. The Fiat-Shamir methodology was shown [60] not to be sound in the standard model in general as it may fail to preserve the soundness of pathological 3-move arguments, regardless of which hash function is used. Known negative results (see [60,18] 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 [64] 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 [95,31,70,28] focused on the design of correlation intractable hash functions leading to sound instantiation of Fiat–Shamir in the standard model. For a broad class of assumptions, this was first achieved [30,95,70] using indistinguishability obfuscation [53] or non-standard exponential hardness assumptions [31]. Canetti *et al.* [29] 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 [89] recently gave an elegant FHE-based solution relying on the hardness of the LWE problem [92] with polynomial approximation factors. While specific to the Gentry-Sahai-Waters (GSW) FHE [56], their construction does not require any non-standard circular security assumption. Together with the techniques of [34,29], it implies NIZK for all NP languages.

In [34,29], Canetti *et al.* showed that, besides the language of Hamiltonian graphs considered in [46], trapdoor Σ-protocols also exist of other languages like that of quadratic residues modulo a composite integer [59]. Using the CI hash function of [89], they thus obtained a NIZK proof for the Quadratic Residuosity language under the LWE assumption. Choudhuri *et al.* [38] showed that the hash families of [29] make the transformation sound for the sumcheck protocol [81]. Here, we exploit the observation that existing Σ-protocols [86,9] for the LWE relation can easily be turned into trapdoor Σ-protocols.

KDM Security. Key-dependent message security is not implied by standard security notions like IND-CPA security (see, e.g., [2,35]). It was first formalized by Black, Rogaway and Shrimpton [19] and motivated by applications in anonymous credentials [26] or in the context of disk encryption (e.g., in the BitLocker encryption utility [22]), where the encryption key may be stored on the disk being encrypted. The first examples of KDM-secure secret-key encryption were

[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.

given by Black *et al.* [19] in the random oracle model.

In the standard model, the feasibility of KDM security remained open during several years. Hofheinz and Unruh [69] described a secret-key encryption scheme for which they proved KDM-CPA security against adversaries that obtain a bounded number of encryptions. In the public-key setting, Boneh, Halevi, Hamburg, and Ostrovsky [22] constructed a scheme for which they proved 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 cryptosystem [92] is also KDM secure for all affine functions under the LWE assumption. They also described a secret-key construction based on the hardness of the Learning Parity with Noise (LPN) problem and Döttling subsequently gave a public key variant [45]. Under the Quadratic (QR) and Composite Residuosity (DCR) [88] assumptions, Brakerski and Goldwasser [24] gave alternative constructions that additionally provide security under key leakage. In the context of identity-based encryption (IBE) [21], Alperin-Sheriff and Peikert [4] showed that a variant of the IBE scheme of Agrawal *et al.* [3] provides KDM security for a bounded number of challenge ciphertexts. We note that applying the Canetti-Halevi-Katz transform [33] to the IBE scheme of [4] does not immediately give KDM-CCA security as this would require to begin with an IBE system providing KDM security with respect to master secret keys [51].

Haitner and Holenstein [63] gave black-box impossibility results when the adversary makes encryption queries for poly-wise independent functions. Brakerski, Goldwasser, and Kalai [25] and Barak *et al.* [11] independently came up with different techniques that bypass the impossibility results of [63] so as to prove KDM security for richer function families. Malkin *et al.* [83] suggested a much more efficient scheme with ciphertexts of $O(d)$ group elements for function families containing degree d polynomials. Applebaum [5,6] 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 bounded-KDM security [11] (i.e., for any circuit of a priori bounded polynomial size). Bellare *et al.* [12] suggested a more efficient amplification technique but, unlike Applebaum's transformation, it only applies in the KDM-CPA setting and does not preserve CCA security. Kitagawa *et al.* [75] later extended the optimized transformation of [12] to the KDM-CCA case.

KDM-CCA Security. The first standard model realization of PKE scheme with KDM security under chosen-ciphertext attacks appeared in the work of Camenisch, Chandran, and Shoup [27]. 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* [22] and Groth-Sahai proofs [62]. Our scheme is an instantiation of the generic construction of [27] in the lattice setting, where we cannot simply use Groth-Sahai proofs.

Hofheinz [67] 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. The latter scheme builds on different ideas and relies on the Composite Residuosity assumption and bilinear maps.

A first attempt to obtain KDM-CCA security without bilinear maps was made by Lu, Li, and Jia [80]. Han, Liu, and Lyu [65] identified a bug in [80] and gave a patch using the same methodology. They achieved KDM-CCA security for bounded-degree polynomial functions (with ciphertexts of polynomial size in the degree of the functions) under the DDH and DCR assumptions. More recently, Kitigawa and Tanaka [77] described a framework for the design of KDM-CCA secure PKE schemes under a single number theoretic assumption. Their framework extends ideas from [96] and provides instantiations under the DDH, QR, and DCR assumptions. Since the framework of [77] relies on hash proof systems [41], it is not known to provide LWE-based realizations (indeed, hash proof systems do not readily enable chosen-ciphertext security from LWE so far).

In the random oracle model, several well-known constructions happen to remain secure under KDM queries. For example, Backes *et al.* [10] gave evidence that RSA-OAEP [15] provides KDM-CCA2 security. Kitigawa *et al.* [76] demonstrated a similar result for the Fujisaki-Okamoto transformation of [50].

2 Background

Here we define some of the tools involved in our constructions. A few additional standard tools, such as NIZK, are defined in Appendix A.

2.1 Lattices

For any $q \geq 2$, we let $\mathbb{Z}_q$ denote the ring of integers with addition and multiplication modulo q. If $\mathbf{x} \in \mathbb{R}^n$ is a vector, then $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots x_n^2}$ 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, we let $U(S)$ denote the uniform distribution over S. If X and Y are distributions over the same domain, then $\Delta(X, Y)$ 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 $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ 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:*

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

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 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*

$$\mathbf{A} = \left[\bar{\mathbf{A}} \mid \bar{\mathbf{A}}\mathbf{R} + \mathbf{H}\cdot\mathbf{G}\right] \in \mathbb{Z}_q^{n\times m}$$

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 ([55, 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 [29] 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. For simplicity, we abuse notation and often omit unique-output.

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 PPT adversary* $\mathcal{A}$*, we have*

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

Definition 2.6 ([34,29]). *Given a collection of relation ensemble* $\mathfrak{R}$*, a hash family* $\mathcal{H}$ *is* **somewhere statistically correlation intractable** *w.r.t.* $\mathfrak{R}$ *if there is an efficient algorithm* $\mathsf{StatGen}$ *with the following properties:*

- $\mathsf{StatGen}(1^\lambda, \mathsf{aux})$ *is a fake key generation that takes as input a security parameter* λ *and an auxiliary input* aux*. It outputs a hashing key* k*.*
- *For any relation* $R \in \mathfrak{R}$*, there exists an auxiliary input* aux_R *with the following properties:*
  - **Key indistinguishability:** *The distributions* $\{k \mid k \leftarrow \mathsf{Gen}(1^\lambda)\}$ *and* $\{k \mid k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{aux}_R)\}$ *are computationally indistinguishable. For any PPT distinguisher* $\mathcal{A}$*, the following function should be negligible:*

$$\begin{aligned}\mathbf{Adv}^{\text{indist-CI}}_{\mathcal{A}}(\lambda) := |\Pr[k \leftarrow \mathsf{Gen}(1^\lambda) : 1 \leftarrow \mathcal{A}(k)] \\ - \Pr[k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{aux}_R) : 1 \leftarrow \mathcal{A}(k)]| \ .\end{aligned}$$

  - **Statistical Correlation Intractability:** *With overwhelming probability over the choice of* $k \leftarrow \mathsf{StatGen}(1^\lambda, \mathsf{aux}_R)$*, no pair* $(k, h(k,x))$ *satisfies* R*:*

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

Clearly, a somewhere statistical correlation intractable hash function for a relation class $\mathfrak{R}$ is also an R-correlation intractable for any relation ensemble $R \in \mathfrak{R}$.

Peikert and Shiehian [89] recently described a somewhere 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 LWE assumption with polynomial approximation factors.

2.3 Admissible Hash Functions

Admissible hash functions were introduced by Boneh and Boyen [20] as a combinatorial tool for partitioning-based security proofs for which Freire *et al.* [49] gave a simplified definition.

Definition 2.7 ([20,49]). *Let* $\ell(\lambda), L(\lambda) \in \mathbb{N}$ *be functions of a security parameter* $\lambda \in \mathbb{N}$*. Let* $\mathsf{AHF} : \{0,1\}^\ell \to \{0,1\}^L$ *be an efficiently computable function. For every* $K \in \{0,1,\perp\}^L$*, 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*

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

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

Theorem 2.8 ([71, 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*.*

In [71], Jager actually proved the result of Theorem 2.8 for *balanced* admissible hash functions which provide both a lower bound and a close upper bound for the probability in Definition 2.7. In our setting, we only need the standard definition of admissible hash functions since we use them to prove security in a game where the adversary aims at outputting a hard-to-compute result (rather than breaking an indistinguishability property). However, the result of Theorem 2.8 still applies to standard admissible hash functions.

In the context of LWE, the standard use of admissible hash functions (used in, e.g., [36]) is to encode K using $\Theta(\lambda)$ matrices of the form $\mathbf{A} \cdot \mathbf{R}_i + K_i \cdot \mathbf{G}$. Yamada [97] observed that, since the admissible hash function of [71] has sparse keys K with only $\eta = O(\log\lambda)$ non-$\perp$ entries, it can be more compactly encoded using $O(\log\lambda)$ integers of size $O(\log\lambda)$ each. Yamada defined a modified partitioning function $F_{\mathsf{MAH}}(\cdot)$, which is equivalent to $F_{\mathsf{ADH}}(\cdot)$ and operates over a compact representation T of K. The homomorphic operations of GSW then make it possible to homomorphically compute $\mathbf{A} \cdot \mathbf{R}'_x + F_{\mathsf{MAH}}(\mathsf{T}, X) \cdot \mathbf{G}$ from GSW encryptions of a compact encoding T of K.

Lemma 2.9 ([97]). *Let* $\mathcal{K}_{\mathsf{MAH}} = \{\mathsf{K} \subset [2\ell] \ \mid \ |\mathsf{K}| < \eta\}$ *and* $\mathcal{X} = \{0,1\}^\ell$*. For any key* $K \in \{0,1,\perp\}^L$ *with at most* $\eta = O(\log\lambda)$ *entries in* $\{0,1\}$*, define* $\mathsf{K} = \mathsf{Encode}_{\mathsf{MAH}}(K)$ *as*

$$\mathsf{K} := \{2i - K_i \mid i \in [L], \ K_i \neq \perp\} \in \mathcal{K}_{\mathsf{MAH}} \ .$$

Then, the partitioning function

$$F_{\mathsf{MAH}}(\mathsf{K}, X) := \begin{cases} 0 & \text{if} \quad \mathsf{K} \subseteq \mathsf{S}(X) \quad \text{where} \quad \mathsf{S}(X) = \{2i - \mathsf{AHF}(X)_i \mid i \in [L]\} \\ 1 & \text{otherwise} \end{cases}$$

satisfies $F_{\mathsf{ADH}}(K, X) = 0 \Leftrightarrow F_{\mathsf{MAH}}(\mathsf{K}, X) = 0$*. Moreover,* $\eta' := |\mathsf{K}| = O(\log\lambda)$*, so that* K *costs* $u = O(\log^2\lambda)$ *bits to represent. Finally, there exist deterministic polynomial-time algorithms* $(\mathsf{PubEval}_{\mathsf{MAH}}, \mathsf{TrapEval}_{\mathsf{MAH}})$ *where* $\mathsf{PubEval}_{\mathsf{MAH}}$ *(resp.* $\mathsf{TrapEval}_{\mathsf{MAH}}$*) takes as input* $X \in \{0,1\}^\ell$ *and* $\{\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i + \kappa_i \cdot \mathbf{G}\}_{i\in[u]}$ *(resp.*

X and $\{\mathbf{R}_i, \kappa_i\}_{i\in[u]}$), where $u = O(\log^2 \lambda)$ and $\kappa_i \in \{0,1\}$, $\mathbf{R}_i \in \{-1,0,1\}^{m\times m}$ for all $i \in [u]$. It outputs a matrix $\mathbf{A}_{F,X} = \mathsf{PubEval}_{\mathsf{MAH}}(X, \{\mathbf{A}_i\}_{i\in[u]})$ (resp. $\mathbf{R}_{F,X} = \mathsf{TrapEval}_{\mathsf{MAH}}(X, \{\kappa_i, \mathbf{R}_i\}_{i\in[u]})$) such that

$$\mathbf{A}_{F,X} = \mathbf{A} \cdot \mathbf{R}_{F,X} \cdot F_{\mathsf{MAH}}(\mathsf{K}, X) \cdot \mathbf{G} \quad \in \mathbb{Z}_q^{n\times m}$$

and $\mathbf{R}_{F,X} \in \mathbb{Z}^{m\times m}$ has norm $\|\mathbf{R}_{F,X}\|_\infty \leq m^3 u(L+1)$.

2.4 Trapdoor Σ-protocols

Canetti *et al.* [34] considered a definition of Σ-protocols that slightly differs from the usual formulation [40,39].

Definition 2.10 (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 $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(1^\lambda, \mathcal{L})$, and a statement x and proceed as follows: (i) The prover takes as input $w \in R_{\mathsf{zk}}(x)$, computes $(\mathbf{a}, st) \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$ and sends $\mathbf{a}$ to the verifier; (ii) The verifier then sends back a random challenge $\mathbf{c}$ from the challenge space $\mathcal{C}$; (iii) The prove finally sends a response $\mathbf{z} = \mathsf{P}(\mathsf{crs}, x, w, \mathbf{a}, \mathbf{c}, st)$ to the verifier; (iv) On input of a transcript $(\mathbf{a}, \mathbf{c}, \mathbf{z})$, V outputs 1 or 0.*
- **Completeness:** *If $(x, w) \in R_{\mathsf{zk}}$ and the prover honestly computes $(\mathbf{a}, \mathbf{z})$ for a challenge $\mathbf{c}$ sent by V, $\mathsf{V}(\mathsf{crs}, x, (\mathbf{a}, \mathbf{c}, \mathbf{z}))$ outputs 1 with probability $1 - \mathsf{negl}(\lambda)$.*
- **Special Honest Verifier Zero-Knowledge (SHVZK):** *There is a PPT simulator HVSim that, on input of crs, $x \in \mathcal{L}_{\mathsf{zk}}$ and a random $\mathbf{c} \in \mathcal{C}$, outputs $(\mathbf{a}, \mathbf{z}) \leftarrow \mathsf{HVSim}(\mathsf{crs}, x, \mathbf{c})$ such that $(\mathbf{a}, \mathbf{c}, \mathbf{z})$ is computationally indistinguishable from a real transcript with challenge $\mathbf{c}$ (for $w \in R_{zk}(x)$).*
- **Special soundness:** *For any common reference string $\mathsf{crs} \leftarrow \mathsf{Gen}(1^\lambda)$, any $x \notin \mathcal{L}_{\mathsf{sound}}$, and any first message $\mathbf{a}$ sent by the prover, there is at most one challenge $\mathbf{c} = f(\mathsf{crs}, x, \mathbf{a})$ for which an accepting transcript $(\mathsf{crs}, x, \mathbf{a}, \mathbf{c}, \mathbf{z})$ exists for some third message $\mathbf{z}$. The function f is called the "bad challenge function" associated with Π. That is, if $x \notin \mathcal{L}_{\mathsf{sound}}$ and the challenge differs from the bad challenge, the verifier never accepts.*

Definition 2.10 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 2.11 (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 2.10 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 the public parameters outputs the language-dependent part $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ of $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathsf{par}})$.*
- $\mathsf{TrapGen}(\mathsf{par}, \tau_{\mathcal{L}})$ *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 as inputs a trapdoor τ_{Σ}, a common reference string $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$, an instance x, and a first prover message $\mathbf{a}$. It outputs a challenge $\mathbf{c}$.*

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 $\mathsf{TrapGen}(\mathsf{par}, \tau_{\mathcal{L}})$. Namely, for any aux and any PPT distinguisher $\mathcal{A}$, we have*

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

- **Correctness:** *There exists a language-specific trapdoor $\tau_{\mathcal{L}}$ such that, for any instance $x \notin \mathcal{L}_{\mathsf{sound}}$ and all $(\mathsf{crs}_{\mathcal{L}}, \tau_{\Sigma}) \leftarrow \mathsf{TrapGen}(\mathsf{par}, \tau_{\mathcal{L}})$, 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 $\mathcal{L}_{\mathsf{sound}}$. For this reason, in Section 4.1, we compute the challenge $\mathbf{c}$ by hashing both x and $\mathbf{a}$ using a somewhere correlation intractable hash function.

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 [74] 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 $(K, t) \notin \mathcal{R}$ 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 consider a stronger notion of $\mathcal{R}$-lossy PKE scheme which imposes some 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 $(K, t) \notin \mathcal{R}$). 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 Opener'. The former operates over injective public keys for lossy tags while the latter can equivocate ciphertexts encrypted under lossy keys for any tag.

Definition 2.12. *Let $\mathcal{R} \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ be an efficiently computable binary relation. An $\mathcal{R}$-lossy public-key encryption scheme with efficient opening is a 7-uple of PPT algorithms* $(\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{Opener}')$ *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: *The key generation algorithm $\mathsf{LKeygen}$ and Keygen satisfy the following properties:*

(i) For any $K \in \mathcal{K}_\lambda$, the distributions $D_{\text{inj}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{Keygen}(\Gamma, K)\}$ and $D_{\text{loss}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ are computationally indistinguishable. Namely, for any PPT adversary, we

have $\mathbf{Adv}^{\mathsf{indist\text{-}LPKE\text{-}1}}(\lambda) \leq \mathsf{negl}(\lambda)$*, where*

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

(ii) For any distinct initialization values $K, K' \in \mathcal{K}_\lambda$*, the two distributions* $\{pk \mid (pk,sk,tk) \leftarrow \mathsf{LKeygen}(\Gamma,K)\}$ *and* $\{pk \mid (pk,sk,tk) \leftarrow \mathsf{LKeygen}(\Gamma,K')\}$ *are statistically indistinguishable.*

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

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

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 sampling algorithm Opener *such that, for any* $K \in \mathcal{K}_\lambda$*, any keys* $(pk,sk,tk) \leftarrow \mathsf{Keygen}(\Gamma,K)$ *and* $(pk,sk,tk) \leftarrow \mathsf{LKeygen}(\Gamma,K)$*, 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* $C = \mathsf{Encrypt}(pk,t,\mathsf{Msg}_0,r)$*,* t*, and* tk*. It outputs an independent 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* Opener' *such that, for any* $K \in \mathcal{K}_\lambda$*, any keys* $(pk,sk,tk) \leftarrow \mathsf{LKeygen}(\Gamma,K)$*, 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 2.12, 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 $(\mathsf{Opener}, \mathsf{Opener}')$ 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 [13,68]). In our LWE-based construction, however, the initial messages and random coins are not needed.

As in [23], we consider $\mathcal{R}$-lossy PKE schemes for the bit-matching relation, which evaluates to 1 if t agrees with K in all positions where the latter is not $\perp$.

Definition 2.13. *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 an admissible hash function $\mathsf{AHF} : \{0,1\}^\ell \to \{0,1\}^L$ in Definition 2.7. The* **bit-matching relation** *$\mathcal{R}_{\mathsf{BM}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ for AHF is the relation where we have $\mathcal{R}_{\mathsf{BM}}(K,t) = 1$ 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 An $\mathcal{R}$-Lossy Public-Key Encryption Scheme with Efficient Opening from LWE

We describe an $\mathcal{R}_{\mathsf{BM}}$-lossy PKE scheme for the bit-matching relation. Our scheme builds on a variant of the primal Regev cryptosystem [92] suggested in [55].

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.

Par-Gen(1^λ)**:** Given a security parameter $\lambda \in \mathbb{N}$, let $n_0 = \mathsf{poly}(\lambda)$ denote the bit length of encrypted 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 $\mathcal{K} = \{0,1\}^L$ and Gaussian parameters $\sigma = O(m^4) \cdot u(L+1)$ and $\alpha \in (0,1)$ such that $m\alpha q \cdot (1 + m^3 u(L+1)) \cdot \sigma\sqrt{2m} < q/4$. Set $\Gamma = (\ell, L, n_0, q, n, m, \alpha, \sigma)$.

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

1. Generate a key pair for the primal Regev encryption scheme. Namely, sample a matrix $\bar{\mathbf{B}} \hookleftarrow U(\mathbb{Z}_q^{(n-n_0)\times m})$ and compute
$$\mathbf{A} = \left[\frac{\bar{\mathbf{B}}}{\mathbf{S}^\top \cdot \bar{\mathbf{B}} + \mathbf{E}^\top}\right] \in \mathbb{Z}_q^{n\times m} \ ,$$
where $\mathbf{S} \hookleftarrow U(\mathbb{Z}_q^{(n-n_0)\times n_0})$ and $\mathbf{E} \hookleftarrow \chi^{m\times n_0}$.
2. Let $u = O(\log^2 \lambda)$ as specified by Lemma 2.9. Compute $\mathsf{K} = \mathsf{Encode}_{\mathsf{MAH}}(K)$ and let $\kappa_1 \dots \kappa_u \in \{0,1\}^u$ its u-bit encoding. Define matrices
$$\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i + \kappa_i \cdot \mathbf{G} \ \in \mathbb{Z}_q^{n\times m} \qquad\qquad \forall i \in [u]$$
where $\mathbf{G} \in \mathbb{Z}_q^{n\times m}$ is the gadget matrix and $\mathbf{R}_i \hookleftarrow U(\{-1,1\}^{m\times m})$ for all $i \in [u]$.

Define $R^{\mathsf{LPKE}} = \{\mathbf{r} \in \mathbb{Z}^{2m} \mid \|\mathbf{r}\| \le \sigma\sqrt{2m}\}$ and output

$$pk := \Big(\mathbf{A},\ \{\mathbf{A}_i\}_{i=1}^u\Big), \qquad sk = (K, \mathbf{S}), \qquad tk = (K, \{\mathbf{R}_i\}_{i=1}^u).$$

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

1. Run $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow \mathsf{GenTrap}(1^\lambda, 1^n, 1^m, q)$ so as to obtain a statistically uniform matrix $\mathbf{A} \sim U(\mathbb{Z}_q^{n \times m})$ with a trapdoor for the lattice $\Lambda^\perp(\mathbf{A})$. Notice $m = 2n\lceil \log q \rceil + O(\lambda)$ is required by Lemma 2.2 in order to run algorithm $\mathsf{GenTrap}$.
2. Define matrices

$$\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i + \kappa_i \cdot \mathbf{G} \ \in \mathbb{Z}_q^{n \times m} \qquad\qquad \forall i \in [u]$$

where $\mathbf{R}_i \hookleftarrow U(\{-1, 1\}^{m \times m})$ for all $i \in [u]$.

Define R^{LPKE} as in Keygen and output

$$pk := \Big(\mathbf{A},\ \{\mathbf{A}_i\}_{i=1}^u\Big), \qquad sk = \mathbf{T}_{\mathbf{A}}, \qquad tk = (K, \{\mathbf{R}_i\}_{i=1}^u).$$

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

1. Compute $\mathbf{A}_{F,t} = \mathsf{PubEval}_{\mathsf{MAH}}(t, \{\mathbf{A}_i\}_{i \in [u]}) \in \mathbb{Z}_q^{n \times m}$ using the $\mathsf{PubEval}_{\mathsf{MAH}}$ algorithm of Lemma 2.9. Note that $\mathbf{A}_{F,t} = \mathbf{A} \cdot \mathbf{R}_{F,t} + F_{\mathsf{MAH}}(\mathsf{K}, t) \cdot \mathbf{G}$ for some $\mathbf{R}_{F,t} \in \mathbb{Z}^{m \times m}$ of norm $\|\mathbf{R}_{F,t}\|_\infty \leq m^3 u(L+1)$.
2. Choose $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^{2m}, \sigma}$ and output $\perp$ if $\mathbf{r} \notin R^{\mathsf{LPKE}}$. Otherwise, compute and output the ciphertext

$$\mathbf{c} = [\mathbf{A} \mid \mathbf{A}_{F,t}] \cdot \mathbf{r} + \left[\frac{\mathbf{0}^{n-n_0}}{\mathsf{Msg} \cdot \lfloor q/2 \rfloor}\right] \in \mathbb{Z}_q^n \ . \tag{1}$$

$\mathsf{Decrypt}(sk, t, \mathbf{c})$: Given the secret key $sk = (K, \mathbf{S})$ and the tag $t \in \{0,1\}^\ell$, compute $\mathsf{K} = \mathsf{Encode}_{\mathsf{MAH}}(K)$ and return $\perp$ if $F_{\mathsf{MAH}}(\mathsf{K}, t) = 1$. Otherwise, compute $\mathbf{w} = [-\mathbf{S}^\top \mid \mathbf{I}_{n_0}] \cdot \mathbf{c} \in \mathbb{Z}_q^{n_0}$. Then, 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 $\mathsf{Msg} = \mathsf{Msg}[1] \ldots \mathsf{Msg}[n_0]$.

$\mathsf{Opener}(pk, tk, \mathbf{c}, \mathsf{Msg}_1)$: Given $tk = (K, \{\mathbf{R}_i\}_{i=1}^u)$ and $t \in \{0,1\}^\ell$, compute $\mathsf{K} = \mathsf{Encode}_{\mathsf{MAH}}(K)$ and return $\perp$ if $F_{\mathsf{MAH}}(\mathsf{K}, t) = 0$. Otherwise,

1. Compute the matrix $\mathbf{R}_{F,t} = \mathsf{TrapEval}_{\mathsf{MAH}}(t, \{\kappa_i, \mathbf{R}_i\}_{i \in [u]}) \in \mathbb{Z}^{m \times m}$ such that $\mathbf{A}_{F,t} = \mathbf{A} \cdot \mathbf{R}_{F,t} + \mathbf{G}$ and $\|\mathbf{R}_{F,t}\|_\infty \leq m^3 u(L+1)$.
2. Use $\mathbf{R}_{F,t} \in \mathbb{Z}^{m \times m}$ as a trapdoor for the matrix

$$\bar{\mathbf{A}}_{F,t} = [\mathbf{A} \mid \mathbf{A}_{F,t}] = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_{F,t} + \mathbf{G}] \ \in \mathbb{Z}_q^{n \times 2m}$$

to sample a Gaussian vector $\bar{\mathbf{r}} \in \mathbb{Z}^{2m}$ such that

$$\bar{\mathbf{A}}_{F,t} \cdot \bar{\mathbf{r}} = \mathbf{c} - \left[\frac{\mathbf{0}^{n-n_0}}{\mathsf{Msg}_1 \cdot \lfloor q/2 \rfloor}\right] \ . \tag{2}$$

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 $\bar{\mathbf{r}} \hookleftarrow D_{\Lambda_q^{\mathbf{c}_{\mathsf{Msg}_1}}(\bar{\mathbf{A}}_{F,t}), \sigma}$.

Opener′$(pk, sk, t, \mathbf{c}, \mathsf{Msg}_1)$**:** 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 $\Lambda^\perp(\bar{\mathbf{A}}_{F,t})$ 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 (2).

We prove the following theorem, stating that the above construction satisfies all the required properties under the LWE assumption, in Appendix B.

Theorem 3.1. *The above construction is an $\mathcal{R}_{\mathsf{BM}}$-lossy public-key encryption scheme with efficient opening under the* LWE *assumption.*

4 Direct Constructions of Unbounded Simulation-Sound NIZK Arguments

In this section, we first provide a method that directly compiles (i.e., without relying on generic NIZK techniques [43]) any trapdoor Σ-protocol into an unbounded simulation-sound NIZK argument system using an $\mathcal{R}$-lossy encryption scheme and a correlation intractable hash function.

In a second step, we show a trapdoor Σ-protocol based on the Micciancio-Vadhan protocol [86] which can be used to prove plaintext equalities in the dual Regev cryptosystem.

4.1 A Generic Construction from Trapdoor Σ-Protocols and $\mathcal{R}$-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 [52,82,54] 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 2.11. 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 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 $L \in \mathsf{poly}(\lambda)$ such that $L > \ell$, which induces the relation $\mathcal{R}_{\mathsf{BM}} : \{0,1,\perp\}^L \times \{0,1\}^\ell \to \{0,1\}$ of Definition 2.13.
- An $\mathcal{R}$-lossy PKE scheme $\mathcal{R}$-$\mathsf{LPKE} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{Opener}')$ 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 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.

- A somewhere correlation intractable hash family $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ for the relation 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 construction $\Pi^{\mathrm{uss}} = (\mathsf{Gen}_{\mathsf{par}}, \mathsf{Gen}_{\mathcal{L}}, \mathsf{P}, \mathsf{V})$ 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 $\mathcal{L} \subset \{0,1\}^N$, let $\mathcal{K} = \{0, 1, \perp\}^L$ and $\mathcal{T} = \{0,1\}^\ell$. The common reference string is generated 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 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 the somewhere correlation intractable hash function.

Output the language-dependent $\mathsf{crs}_{\mathcal{L}} := \big(\mathsf{crs}'_{\mathcal{L}}, pk, k, \mathsf{AHF}, \mathsf{OTS}\big)$ 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 $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$.

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

1. Compute $(\mathbf{a}', st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w)$ as a first prover message for Π'. Then, compute $\mathbf{a} \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{a}'; \mathbf{r})$ using random coins $\mathbf{r} \hookleftarrow D_R^{\mathsf{LPKE}}$ sampled from the distribution D_R^{LPKE} over R^{LPKE}.
2. Compute $\mathbf{c} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK}))$.
3. Compute $\mathbf{z}' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, \mathbf{a}', \mathbf{c}, st')$ and define the prover's response to be $\mathbf{z} = (\mathbf{z}', \mathbf{a}', \mathbf{r})$.
4. Generate a one-time signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, \mathbf{a}, \mathbf{z}))$ and output

$$\boldsymbol{\pi} = \big(\mathsf{VK}, (\mathbf{a}, \mathbf{z}), sig\big). \tag{3}$$

$\mathsf{V}(\mathsf{crs}, x, \boldsymbol{\pi})$: Given a statement and a candidate proof $\boldsymbol{\pi}$, parse $\boldsymbol{\pi}$ as in (3). If $\mathcal{V}(\mathsf{VK}, (x, \mathbf{a}, \mathbf{z}), sig) = 0$, return 0. Otherwise,

1. Write $\mathbf{z}$ as $\mathbf{z} = (\mathbf{z}', \mathbf{a}', \mathbf{r})$ and return 0 if it does not parse properly. Return 0 if $\mathbf{a} \neq \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{a}'; \mathbf{r})$ or $\mathbf{r} \notin R^{\mathsf{LPKE}}$.
2. Compute $\mathbf{c} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK}))$. If $\mathsf{V}'(\mathsf{crs}'_{\mathcal{L}}, x, (\mathbf{a}', \mathbf{c}, \mathbf{z}')) = 1$, return 1. Otherwise, return 0.

Our NIZK simulator uses a technique due to Damgård [42], 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 4.1. *The above argument system is multi-theorem zero-knowledge assuming that the trapdoor Σ-protocol Π' is special honest-verifier 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 $(\mathbf{a}, \mathbf{c}, \mathbf{z})$ 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.

$\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, x, \varepsilon)$: On input a statement $x \in \{0,1\}^N$ 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 the same length as the first prover message of Π'. Compute
$$\mathbf{a} \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{0}^{|\mathbf{a}'|}; \mathbf{r}_0)$$
using random coins $\mathbf{r}_0 \hookleftarrow D_R^{\mathsf{LPKE}}$ sampled from the distribution D_R^{LPKE}.
2. Compute $\mathbf{c} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK}))$.
3. Run the HVZK simulator $(\mathbf{a}', \mathbf{z}') \leftarrow \mathsf{HVSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathbf{c})$ of Π' so as to obtain a simulated transcript $(\mathbf{a}', \mathbf{c}, \mathbf{z}')$ of Π' for the challenge $\mathbf{c}$.
4. Using the lossy secret key sk of $\mathcal{R}$-LPKE, compute random coins $\mathbf{r} \leftarrow \mathsf{Opener}'(pk, sk, \mathsf{VK}, \mathbf{a}, \mathbf{a}')$ 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. Generate a one-time signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, \mathbf{a}, \mathbf{z}))$ 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 honest-verifier zero-knowledge property of Π' implies that its simulator produces $(\mathbf{a}', \mathbf{z}') \leftarrow \mathsf{HVSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathbf{c})$ such that $(\mathbf{a}', \mathbf{c}, \mathbf{z}')$ is computationally indistinguishable from a real transcript with challenge $\mathbf{c}$. This implies that the distribution

$$\begin{aligned}\{(\mathbf{a}, \mathbf{a}', \mathbf{r}, \mathbf{z}') \mid \mathbf{r}_0 \hookleftarrow D_R^{\mathsf{LPKE}},\ \mathbf{a} \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{0}^{|\mathbf{a}'|}; \mathbf{r}_0)\ ,\\ (\mathbf{a}', \mathbf{z}') \leftarrow \mathsf{HVSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathbf{c}),\ \mathbf{r} \leftarrow \mathsf{Opener}'(pk, sk, \mathsf{VK}, \mathbf{a}, \mathbf{a}')\}\ ,\end{aligned} \tag{4}$$

is computationally indistinguishable from

$$\begin{aligned}\{(\mathbf{a}, \mathbf{a}', \mathbf{r}, \mathbf{z}') \mid \mathbf{r}_0 \hookleftarrow D_R^{\mathsf{LPKE}},\ \mathbf{a} \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{0}^{|\mathbf{a}'|}; \mathbf{r}_0)\ ,\\ (\mathbf{a}', st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w),\ \mathbf{z}' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, \mathbf{a}', \mathbf{c}, st')\ ,\\ \mathbf{r} \leftarrow \mathsf{Opener}'(pk, sk, \mathsf{VK}, \mathbf{a}, \mathbf{a}')\}\ .\end{aligned} \tag{5}$$

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

$$\begin{aligned}\{(\mathbf{a}, \mathbf{a}', \mathbf{r}, \mathbf{z}') \mid\ & (\mathbf{a}', st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w),\ \mathbf{r} \hookleftarrow D_R^{\mathsf{LPKE}}\\ & \mathbf{a} \leftarrow \mathsf{Encrypt}(pk, \mathsf{VK}, \mathbf{a}'; \mathbf{r})\ ,\\ & \mathbf{z}' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, \mathbf{a}', \mathbf{c}, st')\}\ .\end{aligned} \tag{6}$$

Clearly, the distribution (4) corresponds to proof generated by the simulator while (6) is the distribution generated by the real prover. This implies that simulated proofs are computationally (resp. statistically) indistinguishable from real proofs if the simulator of Π' is computationally (resp. statistically) HVZK. □

Our proof of unbounded simulation-soundness builds on techniques used in [42,52,82,54]. The interactive proof systems of [82,54] 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 the aforementioned properties. Namely, it can serve as a trapdoor commitment to equivocate lossy encryptions of the first prover message in Π' while forcing the adversary to create a fake proof on a statistically binding commitment.

At a high level, our proof of simulation-soundness also bears similarities with [78] 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 (i.e., the lossy ciphertext) in simulated proofs here.

Theorem 4.2. *The above argument system provides unbounded simulation-soundness assuming that: (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 somewhere 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. To prove the result, we consider a sequence of games. For each i, we define a 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 game of Definition A.2. Namely, the challenger runs $(\mathsf{crs}, \tau_{\mathrm{zk}}) \leftarrow \mathsf{Sim}_0(\mathsf{par}, 1^N)$ and gives $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$ 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 $x \in \{0,1\}^N$ and the challenger replies by returning a simulated proof $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, x, \varepsilon)$. When the adversary $\mathcal{A}$ halts, it outputs $(x^\star, \boldsymbol{\pi}^\star)$, 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 conditions: (i) $(x^\star, \boldsymbol{\pi}^\star) \notin \mathcal{Q}$, where $\mathcal{Q} = \{(x_i, \boldsymbol{\pi}_i)\}_{i=1}^{Q}$ denotes the set of queries to the oracle $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ 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) $V(\mathsf{crs}, x^\star, \boldsymbol{\pi}^\star) = 1$. 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 the definition of the adversary's advantage, 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 $(x^\star, \boldsymbol{\pi}^\star)$, where $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, (\mathbf{a}^\star, \mathbf{z}^\star), sig^\star\big)$ contains a $\mathsf{VK}^\star$ that coincide 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 $|\Pr[W_1] - \Pr[W_0]| \leq \mathbf{Adv}^{\mathrm{ots}}_{\mathcal{B}}(\lambda)$.

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. When the adversary halts and outputs $x^\star$, the challenger checks if the conditions

$$F_{\mathsf{ADH}}(K, \mathsf{VK}^{(1)}) = \cdots = F_{\mathsf{ADH}}(K, \mathsf{VK}^{(Q)}) = 1 \ \wedge \ F_{\mathsf{ADH}}(K, \mathsf{VK}^\star) = 0 \qquad (7)$$

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 $W_2 = \mathtt{false}$. 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 (7) are satisfied, the challenger sets $W_2 = \mathtt{true}$ whenever $W_1 = \mathtt{true}$. Letting Fail denote the event that $\mathcal{B}$ aborts because (7) does not hold, we have $W_2 = W_1 \wedge \neg\mathsf{Fail}$. Since the key K of the admissible hash function is statistically independent of the adversary's view, we can apply Theorem 2.8 to argue that there is a noticeable function $\delta(\lambda)$ such that $\Pr[\neg\mathsf{Fail}] \geq \delta(\lambda)$. This implies

$$\Pr[W_2] = \Pr[W_1 \wedge \neg\mathsf{Fail}] \geq \delta(\lambda) \cdot \Pr[W_1] \ , \qquad (8)$$

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 (7) 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_3: 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 $x_i \in \{0,1\}^N$, 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

Opener' using sk. We define the Boolean variable W_3 exactly as W_2. Since Opener and Opener' output samples from the same distribution D_R^{LPKE} over R^{LPKE}, this implies that $|\Pr[W_3] - \Pr[W_2]| \leq 2^{-\lambda}$.

Game_4: We now modify the distribution of crs. Namely, at step 2 of Gen, we generate the keys for $\mathcal{R}$-LPKE as injective keys $(pk, sk, tk) \leftarrow \mathsf{Keygen}(\Gamma, K)$ instead of lossy keys $(pk, sk, tk) \leftarrow \mathsf{LKeygen}(\Gamma, K)$. The indistinguishability property (i) of $\mathcal{R}$-LPKE guarantees $\Pr[W_4]$ is within negligible distance from $\Pr[W_3]$. 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_3 and Game_4 (by using tk to simulate $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot, \cdot)$ as in Game_3). It comes that

$$|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}_{\mathcal{B}}^{\text{indist-LPKE-1}}(\lambda) \ .$$

We note that the modification introduced in Game_4 implies that, if the conditions (7) are 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_5: We change again the distribution of $\mathsf{crs} = \big(\mathsf{crs}', pk, k, \mathsf{AHF}, \mathsf{OTS}\big)$ by leveraging the CRS indistinguishability property of the trapdoor Σ-protocol Π'. Namely, we use the $\mathsf{TrapGen}'$ algorithm of Definition 2.11 to generate $\mathsf{crs}'_{\mathcal{L}}$ as $(\mathsf{crs}'_{\mathcal{L}}, \tau_\Sigma) \leftarrow \mathsf{TrapGen}'(\mathsf{par}, \tau_{\mathcal{L}})$ instead of $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$. We immediately have $|\Pr[W_5] - \Pr[W_4]| \leq \mathbf{Adv}_{\mathcal{A}}^{\text{indist-}\Sigma}(\lambda)$.

We note that the trapdoor τ_Σ produced by $\mathsf{TrapGen}'$ in Game_5 can henceforth be used 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 the ciphertext $\mathbf{a}^\star$ contained in $\boldsymbol{\pi}^\star$ when $\mathcal{R}_{\mathsf{BM}}(K, \mathsf{VK}^\star) = 1$.

Game_6: We introduce another change in the distribution of $\mathsf{crs}_{\mathcal{L}}$. We consider the relation R_{bad} defined by

$$((x, \mathbf{a}, \mathsf{VK}), \mathbf{c}) \in R_{\mathsf{bad}} \quad \Leftrightarrow \quad \mathbf{c} = \mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Decrypt}(sk, \mathsf{VK}, \mathbf{a})).$$

We now generate the key of the correlation-intractable hash function as $k \leftarrow \mathsf{StatGen}\big(1^\lambda, \mathsf{aux}_{R_{\mathsf{bad}}}\big)$ instead of $k \leftarrow \mathsf{Gen}(1^\lambda)$. By the key indistinguishability property of $\mathcal{H}$, we have $|\Pr[W_6] - \Pr[W_5]| \leq \mathbf{Adv}_{\mathcal{A}}^{\text{indist-CI}}(\lambda)$.

In Game_6, we claim that $\Pr[W_6] \leq 2^{-\Omega(\lambda)}$. Indeed, if $\mathcal{B}$ did not fail, we know that the adversary'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 $\mathbf{a}^\star$ is a statistically binding commitment to $\mathbf{a}'^\star$. With probability $2^{-\Omega(\lambda)}$, there thus exists only one message $\mathbf{a}'^\star$ such that $\mathbf{a}^\star = \mathsf{Encrypt}(pk, \mathsf{VK}^\star, \mathbf{a}'^\star; \mathbf{r}^\star)$ for some $\mathbf{r}^\star \in R^{\mathsf{LPKE}}$. Said otherwise, there

exists only one $\mathbf{a}'^{\star}$ for which a pair $(\mathbf{a}'^{\star}, \mathbf{r}^{\star})$ satisfies step 1 of the verification algorithm. Moreover, since $\mathbf{a}^{\star}$ uniquely determines $\mathbf{a}'^{\star}$, the statistical correlation intractability property of $\mathcal{H}$ implies that we can only have

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

with exponentially small probability. The probability to have $W_6 = \mathtt{true}$ is thus smaller than $2^{-\Omega(\lambda)}$ as claimed.

Putting the above altogether, we obtain

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

which completes the proof. □

The work of Peikert and Shiehian [89] implies a somewhere statistically correlation intractable hash function for the relation R_{bad} defined in the proof of Theorem 4.2 (in Game_6). The bootstrapping theorem of [89] actually implies the existence of such a hash family under the LWE assumption with polynomial approximation factors.

4.2 A Trapdoor Σ-Protocol based on the Micciancio-Vadhan SZK Proof System

In Appendix C, we show that the Gap Σ-protocol of Asharov *et al.* [9,8] provides a very simple trapdoor Σ-protocol for the LWE relation. Its disadvantage is that the gap between its languages $\mathcal{L}_{\mathsf{zk}}$ and $\mathcal{L}_{\mathsf{sound}}$ is very large due to the use of the noise flooding technique, which is necessary for the honest-verifier zero-knowledge property. In this section, we describe a trapdoor Σ-protocol based on the Micciancio-Vadhan protocol [86] which yields a polynomial gap between $\mathcal{L}_{\mathsf{zk}}$ and $\mathcal{L}_{\mathsf{sound}}$. In turn, this will make it possible to work with a polynomial-size modulus q in Section 5.

Let integers $m > n$, a modulus q and rational numbers $\gamma, d > 0$. Given $\mathbf{B} \in \mathbb{Z}_q^{m \times n}$, consider the language $\mathcal{L}_{\gamma,d} = \{\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}}\}$, where

$$\mathcal{L}_{\mathsf{zk}} := \big\{\mathbf{y} \in \mathbb{Z}_q^m \mid \exists \mathbf{s} \in \mathbb{Z}^n : \|\mathbf{y} - \mathbf{B} \cdot \mathbf{s}\| \leq d\big\} \ ,$$
$$\mathcal{L}_{\mathsf{sound}} := \big\{\mathbf{y} \in \mathbb{Z}_q^m \mid \exists \mathbf{s} \in \mathbb{Z}^n : \|\mathbf{y} - \mathbf{B} \cdot \mathbf{s}\| \leq \gamma \cdot d\big\} \ .$$

For $\gamma = m^{0.5+\Omega(1)}$, Micciancio and Vadhan [86] gave a 3-move interactive statistical zero-knowledge proof for $\mathcal{L}_{\gamma,d} = \{\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}}\}$, where the length of a proof is $O(\xi \cdot m \cdot \log q)$ bits, for $\xi = \omega(1)$. We show that it implies a trapdoor Σ-protocol for the language $\mathcal{L}_{\gamma,d}$. The protocol of [86] allows a prover P in possession of witnesses $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}^m$ such that $\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e}$ and $\|\mathbf{e}\| \leq d$ to convince a verifier V that $\mathbf{y} \in \mathcal{L}_{\mathsf{sound}}$.

The trapdoor Σ-protocol is parameterized by an integer $\xi = \omega(1)$. We assume that the language $\mathcal{L}_{\gamma,d}$ specifies a distribution $D_{\mathbf{B}}$ of matrices over $\mathbb{Z}_q^{m\times n}$ for which there exists an efficient algorithm $\mathsf{TrapSamp}_{\mathbf{B}}(1^\lambda, 1^n, 1^m, q)$ which outputs a matrix $\mathbf{B} \in \mathbb{Z}_q^{m\times n}$ whose distribution is statistically close to $D_{\mathbf{B}}$ together with a small-norm full-rank integer matrix $\mathbf{T_B} \in \mathbb{Z}^{m\times m}$ such that $\mathbf{T_B}\cdot\mathbf{B} = \mathbf{0}^{m\times n} \bmod q$.

$\mathbf{Gen}_{\mathsf{par}}(1^\lambda)$: On input of a security parameter λ, choose a modulus q, dimensions n, m, and error rate $\alpha > 0$. Define $\mathsf{par} = \{\lambda, q, n, m, \alpha\}$.

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

$$\mathsf{crs} = \big(\{\lambda, q, n, m, \alpha\}, \{\mathbf{B}, \gamma, d, \xi\}\big) \ .$$

$\mathbf{TrapGen}(\mathsf{par}, \tau_{\mathcal{L}_{\gamma,d}})$: On input of public parameters par and a membership-testing trapdoor $\tau_{\mathcal{L}_{\gamma,d}}$ for $\mathcal{L}_{\gamma,d}$ consisting of a matrix $\mathcal{L}_{\gamma,d} = \mathbf{T_B}$ obtained as $(\mathbf{B}, \mathbf{T_B}) \leftarrow \mathsf{TrapSamp}_{\mathbf{B}}(1^\lambda, 1^n, 1^m, q)$, output $\mathsf{crs}_{\mathcal{L}} = \{\mathbf{B}, \gamma, d, \xi\}$, which defines $\mathsf{crs} = \big(\{\lambda, q, n, m, \alpha\}, \{\mathbf{B}, \gamma, d, \xi\}\big)$, as well as $\tau_\Sigma = \mathbf{T_B}$.

$\mathbf{P}\big(\mathsf{crs}, \mathbf{y}, (\mathbf{s}, \mathbf{e})\big) \leftrightarrow \mathbf{V}(\mathsf{crs}, \mathbf{y})$: Given crs, a statement $\mathbf{y} \in \mathbb{Z}_q^m$ and P (who has the witness $\mathbf{e} \in \mathbb{Z}^m$ such that $\|\mathbf{e}\| \le d$) and V interact in the following way.

1. The prover P chooses an arbitrary basis $\mathbf{L_B} \in \mathbb{Z}^{m\times m}$ for the q-ary lattice $\Lambda(\mathbf{B}) = \mathbf{B}\cdot\mathbb{Z}^n + q\cdot\mathbb{Z}^m$ and does the following.
   a. Choose random bits $c_1 \dots c_\xi \hookleftarrow U(\{0,1\})$ as well as ξ random vectors $\mathbf{r}_1, \dots, \mathbf{r}_\xi \in \mathbb{Z}^m$ of norm $\|\mathbf{r}_i\| \le \gamma\cdot d/2$ which are uniformly sampled in a ball centered in $\mathbf{0}^m$. Check if there exists $i^\star \in [\xi]$ such that $\|\mathbf{r}_{i^\star} + (2c_{i^\star} - 1)\cdot\mathbf{e}\| \le \gamma\cdot d/2$. If not, repeat the process of choosing $\mathbf{r}_1, \dots, \mathbf{r}_\xi \in \mathbb{Z}^m$ until such an $i^\star$ exists.
   b. For each $i \in [\xi]$, compute $\mathbf{m}_i \in \mathbb{Z}_q^m$ as

   $$\mathbf{m}_i = (c_i\cdot\mathbf{y} + \mathbf{r}_i) - \mathbf{L_B}\cdot\big\lfloor \mathbf{L_B}^{-1}\cdot(c_i\cdot\mathbf{y} + \mathbf{r}_i)\big\rceil \bmod q \ ,$$

   which can be seen as reducing $c_i\cdot\mathbf{y} + \mathbf{r}_i$ modulo the lattice basis $\mathbf{L_B}$ and taking the result modulo q.

   Then, P sends $\mathbf{a} := \{\mathbf{m}_i\}_{i\in[\xi]}$ to V.
2. V sends a random challenge $\mathsf{Chall} \in \{0,1\}$ to P.
3. If $\mathsf{Chall} = \oplus_{i=1}^{\xi} c_i$, set $\bar{c}_i = c_i$ and $\bar{\mathbf{r}}_i = \mathbf{r}_i$ for all $i \in [\xi]$. If $\mathsf{Chall} \neq \oplus_{i=1}^{\xi} c_i$, set

   $$\begin{aligned} &\bar{c}_{i^\star} = 1 - c_{i^\star}, && \bar{\mathbf{r}}_{i^\star} = \mathbf{r}_{i^\star} + (2c_{i^\star} - 1)\cdot\mathbf{e} \ . \\ &\bar{c}_i = c_i, && \bar{\mathbf{r}}_i = \mathbf{r}_i && \forall i \in [\xi]\setminus\{i^\star\} \ . \end{aligned}$$

   Compute $\mathbf{z}_i = \mathbf{m}_i - (\bar{c}_i\cdot\mathbf{y} + \bar{\mathbf{r}}_i) \bmod q$ for each $i \in [\xi]$. Then, send the response $\mathbf{z} = \{\mathbf{z}_i, \bar{c}_i\}_{i\in[\xi]}$ to V.

4. Upon receiving $\{\mathbf{z}_i, \bar{c}_i\}_{i\in[\xi]}$, V checks whether the following two conditions are satisfied: (i) $\mathsf{Chall} = \oplus_{i=1}^{\xi} \bar{c}_i$; (ii) For each $i \in [\xi]$, $\mathbf{z}_i$ is in the lattice $\Lambda(\mathbf{B})$ and $\|\mathbf{m}_i - (\mathbf{z}_i + \bar{c}_i \cdot \mathbf{y})\| \leq \gamma \cdot d/2$. If these conditions do not both hold, V halts and returns $\perp$.

$\mathsf{BadChallenge}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, \mathbf{y}, \mathbf{a}\big)$: Given $\tau_\Sigma = \mathbf{T}_\mathbf{B}$, parse the first prover message $\mathbf{a}$ as $\mathbf{a} = \{\mathbf{m}_i\}_{i\in[\xi]}$ where $\mathbf{m}_i \in \mathbb{Z}_q^m$ for each $i \in [\xi]$. For $i = 1$ to ξ, conduct the following steps for each $b \in \{0, 1\}$.

1. Compute $\boldsymbol{\ell}_{i,b} = \mathbf{T}_\mathbf{B} \cdot (\mathbf{m}_i - b \cdot \mathbf{y}) \bmod q$.
2. Solve $\mathbf{T}_\mathbf{B} \cdot \mathbf{r}_{i,b} = \boldsymbol{\ell}_{i,b}$ over $\mathbb{Q}$ by computing $\mathbf{T}_\mathbf{B}^{-1} \cdot \boldsymbol{\ell}_{i,b}$. If the solution is an integer vector $\mathbf{r}_{i,b} \in \mathbb{Z}^m$ such that $\|\mathbf{r}_{i,b}\| \leq \gamma \cdot d/2$, set $c_i = b$.

If there exists an $i \in [\xi]$ such that c_i was not defined, set $\mathsf{Chall} = \perp$. Otherwise, output the bad challenge $\mathsf{Chall} = \oplus_{i=1}^{\xi} c_i \in \{0, 1\}$.

Lemma 4.3. *The above construction is a trapdoor Σ-protocol for the language $\mathcal{L}_{\gamma,d} = \{\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}}\}$.*

Proof. We prove that the given construction satisfies the requirements Definition 2.10 and Definition 2.11.

Completeness. We show that, if $\mathbf{y} \in \mathcal{L}_{\mathsf{zk}}$, then an honest prover P runs in polynomial time and produces $(\mathbf{a}, \mathbf{z})$ that always gets accepted by the verifier V.

First, given $\mathbf{B}$, one can efficiently compute an arbitrary basis $\mathbf{L}_\mathbf{B}$ of the lattice $\Lambda(\mathbf{B})$. Next, in Step (1.a.), the probability that P – in possession of the witness $\mathbf{e} \in \mathbb{Z}^m$ such that $\|\mathbf{e}\| \leq d$ – has to repeat the sampling of $c_1, \ldots, c_\xi, \mathbf{r}_1, \ldots, \mathbf{r}_\xi$ is negligible. In fact, as analyzed by Micciancio and Vadhan [86], the probability that there does not exist $i^\star \in [\xi]$ such that $\|\mathbf{r}_{i^\star} + (2c_{i^\star} - 1) \cdot \mathbf{e}\| \leq \gamma \cdot d/2$ is at most $2\big(1 - \beta(2/\gamma)\big)^\xi$, where $\beta(\epsilon)$ is the relative volume of the intersection of two m-dimensional unit spheres whose centers are at distance ϵ. Since $\beta(\epsilon)$ satisfies $\beta(\epsilon) \geq \max(3e^{-\epsilon^2 m/2}, 1 - \epsilon\sqrt{m})$, such probability is negligible in m for $\gamma = m^{0.5+\Omega(1)}$ and $\xi = \omega(1)$. Then, at the end of Step (1.b.), the prover can obtain and send $\mathbf{a} = \{\mathbf{m}_i\}_{i\in[\xi]}$ where, for each $i \in [\xi]$, the vector $\mathbf{m}_i - (c_i \cdot \mathbf{y} + \mathbf{r}_i) \bmod q$ is in the lattice $\Lambda(\mathbf{B})$ for some $c_i \in \{0, 1\}$ and $\|\mathbf{r}_i\| \leq \gamma \cdot d/2$.

Upon receiving Chall from V, in the event that $\mathsf{Chall} \neq \oplus_{i=1}^{\xi} c_i$, P flips the bit $c_{i^\star}$ and modifies $\mathbf{r}_{i^\star}$ accordingly, ensuring that the response $\mathbf{z}$ contains $\{\mathbf{z}_i, \bar{c}_i\}_{i=1}^{\xi}$ satisfying $\mathsf{Chall} = \oplus_{i=1}^{\xi} \bar{c}_i$ and $\mathbf{z}_i \in \Lambda(\mathbf{B})$ for all $i \in [\xi]$. Furthermore, we have $\|\mathbf{m}_i - (\bar{c}_i \cdot \mathbf{y} + \mathbf{z}_i)\| \leq \gamma \cdot d/2$ for all $i \in [\xi]$. In particular, for $i = i^\star$ and in the case where $c_{i^\star}$ is flipped, we have

$$\begin{aligned}\mathbf{z}_{i^\star} &= \mathbf{m}_{i^\star} - (\bar{c}_{i^\star} \cdot \mathbf{y} + \bar{\mathbf{r}}_{i^\star}) = \mathbf{m}_{i^\star} - (1 - c_{i^\star}) \cdot \mathbf{y} - \mathbf{r}_{i^\star} - (2c_{i^\star} - 1) \cdot \mathbf{e} \\ &= \big(\mathbf{m}_{i^\star} - (c_{i^\star}\mathbf{y} + \mathbf{r}_{i^\star})\big) + (2c_{i^\star} - 1)(\mathbf{y} - \mathbf{e}) \ ,\end{aligned}$$

which belongs to $\Lambda(\mathbf{B})$ since both $\mathbf{m}_{i^\star} - (c_{i^\star}\mathbf{y} + \mathbf{r}_{i^\star})$ and $\mathbf{y} - \mathbf{e}$ do, and which satisfies

$$\|\mathbf{m}_{i^\star} - (\bar{c}_{i^\star} \cdot \mathbf{y} + \mathbf{z}_{i^\star})\| = \|\bar{\mathbf{r}}_{i^\star}\| = \|\mathbf{r}_{i^\star} + (2c_{i^\star} - 1) \cdot \mathbf{e}\| \leq \gamma \cdot d/2 \ ,$$

as established in Step (1.a.). As a result, V always outputs 1 and the protocol has perfect completeness.

Statistical honest verifier zero-knowledge. As in [86], the simulator exploits the fact that the distribution of $\{\mathbf{m}_i\}_{i=1}^{\xi}$ can be efficiently sampled without knowing the witness $\mathbf{e}$ (which determines a lattice point $\mathbf{B}\cdot\mathbf{s}$ close to $\mathbf{y}$). The simulator also uses the standard techniques for OR-proofs to simulate $\bar{c}_1,\ldots,\bar{c}_\xi$ such that $\oplus_{i=1}^{\xi}\bar{c}_i = \mathsf{Chall}$. On input crs, statement $\mathbf{y}\in\mathcal{L}_{\mathsf{zk}}$ and a random challenge $\mathsf{Chall}\in\{0,1\}$, the simulator chooses an arbitrary basis $\mathbf{L_B}$ of $\Lambda(\mathbf{B})$ and then proceeds as follows.

1. Pick $\xi-1$ bits $\bar{c}_1,\ldots,\bar{c}_{\xi-1}\hookleftarrow U(\{0,1\})$ and let $\bar{c}_\xi = \oplus_{i=1}^{\xi}\bar{c}_i \ \oplus\ \mathsf{Chall}$.
2. For all $i\in[\xi]$, pick $\mathbf{r}_i\in\mathbb{Z}^m$ uniformly at random in the ball that has radius $\gamma\cdot d/2$ and center at $\mathbf{0}^m$. Then, compute
$$\begin{aligned}\mathbf{m}_i &= (\bar{c}_i\cdot\mathbf{y}+\mathbf{r}_i)-\mathbf{L_B}\cdot\left\lfloor\mathbf{L_B}^{-1}\cdot(\bar{c}_i\cdot\mathbf{y}+\mathbf{r}_i)\right\rfloor \bmod q;\\ \mathbf{z}_i &= \mathbf{m}_i-(\bar{c}_i\cdot\mathbf{y}+\mathbf{r}_i)\bmod q.\end{aligned}$$
3. Output $\left(\{\mathbf{m}_i\}_{i=1}^{\xi},\mathsf{Chall},\{\mathbf{z}_i,\bar{c}_i\}_{i=1}^{\xi}\right)$.

It can be checked that the transcript output by the simulator is accepted by the verifier and its distribution is statistically indistinguishable from a real transcript with challenge Chall.

Special soundness. Towards a contradiction, assume that $\mathbf{y}\notin\mathcal{L}_{\mathsf{sound}}$ and there exist valid transcripts $\left(\{\mathbf{m}_i\}_{i=1}^{\xi},0,\{\mathbf{z}_{i,0},\bar{c}_{i,0}\}_{i=1}^{\xi}\right)$ and $\left(\{\mathbf{m}_i\}_{i=1}^{\xi},1,\{\mathbf{z}_{i,1},\bar{c}_{i,1}\}_{i=1}^{\xi}\right)$ with the same first message $\mathbf{a}=\{\mathbf{m}_i\}_{i=1}^{\xi}$ and distinct challenges $\mathsf{Chall}_0=0$, $\mathsf{Chall}_1=1$. Since $\oplus_{i=1}^{\xi}\bar{c}_{i,0}\neq\oplus_{i=1}^{\xi}\bar{c}_{i,1}$, there must exist an index $i'\in[\xi]$ such that $\bar{c}_{i',0}\neq\bar{c}_{i',1}$. Note that $\bar{c}_{i',1}-\bar{c}_{i',0}\in\{-1,1\}$.

Since the two transcripts are valid, we have $\|\mathbf{m}_{i'}-(\mathbf{z}_{i',1}+\bar{c}_{i',1}\cdot\mathbf{y})\|\leq\gamma\cdot d/2$ and $\|\mathbf{m}_{i'}-(\mathbf{z}_{i',0}+\bar{c}_{i',0}\cdot\mathbf{y})\|\leq\gamma\cdot d/2$. The triangle inequality thus implies $\|(\bar{c}_{i',1}-\bar{c}_{i',0})\cdot\mathbf{y}-(\mathbf{z}_{i',0}-\mathbf{z}_{i',1})\|\leq\gamma\cdot d$. Since $\mathbf{z}_{i',0},\mathbf{z}_{i',1}\in\Lambda(\mathbf{B})$ and $\bar{c}_{i',1}-\bar{c}_{i',0}\in\{-1,1\}$, the vector $\mathbf{v}':=(\mathbf{z}_{i',0}-\mathbf{z}_{i',1})/(\bar{c}_{i',1}-\bar{c}_{i',0})$ also belongs to the lattice $\Lambda(\mathbf{B})$. This implies that there exists $\mathbf{s}'\in\mathbb{Z}^n$ such that $\|\mathbf{y}-\mathbf{B}\cdot\mathbf{s}'\|\leq\gamma\cdot d$. However, this contradicts the hypothesis that $\mathbf{y}\notin\mathcal{L}_{\mathsf{sound}}$.

CRS indistinguishability. This property follows directly from the fact that the distribution of matrix $\mathbf{B}$ obtained from $\mathsf{TrapSamp}_{\mathbf{B}}(1^\lambda,1^n,1^m,q)$ is statistically close to $D_{\mathbf{B}}$.

Correctness of BadChallenge. In the $\mathsf{BadChallenge}$ algorithm, we observe that, if $\mathbf{y}\notin\mathcal{L}_{\mathsf{sound}}$, for each $i\in[\xi]$, at most one $b\in\{0,1\}$ can lead to an $\boldsymbol{\ell}_{i,b}$ such that the corresponding $\mathbf{r}_{i,b}$ satisfies $\|\mathbf{r}_{i,b}\|\leq\gamma\cdot d/2$. Indeed, if there exits an index $i\in[\xi]$ such that the inequality is satisfied for both $c_i\in\{0,1\}$, we would have
$$\mathbf{T_B}\cdot(\mathbf{r}_{i,0}-\mathbf{r}_{i,1})=\boldsymbol{\ell}_{i,0}-\boldsymbol{\ell}_{i,1}=\mathbf{T_B}\cdot\mathbf{y}\bmod q,$$
which would contradict that $\mathbf{y}\notin\mathcal{L}_{\mathsf{sound}}$.

We also remark that a bad challenge can only exist if

$$\|\mathbf{T}_{\mathbf{B}}^{-1} \cdot (\mathbf{T}_{\mathbf{B}} \cdot (\mathbf{m}_i - b_i \cdot \mathbf{y}) \bmod q)\| \leq \gamma \cdot d/2$$

for some $b_i \in \{0,1\}$. Indeed, the verifier only accepts responses when $\mathbf{z}_i \in \Lambda(\mathbf{B})$ for all $i \in [\xi]$. In the verification equation, the condition $\|\mathbf{m}_i - (\mathbf{z}_i + \bar{c}_i \cdot \mathbf{y})\| \leq \gamma \cdot d/2$ implies that $\mathbf{m}_i = \bar{c}_i \cdot \mathbf{y} + \mathbf{r}_i + \mathbf{L}_{\mathbf{B}} \cdot \mathbf{w}_i$ for some $\mathbf{w}_i, \mathbf{r}_i \in \mathbb{Z}^m$ such that $\|\mathbf{r}_i\| \leq \gamma \cdot d/2$. In this case, the $\mathsf{BadChallenge}$ algorithm can always correctly decode $\bar{c}_i \in \{0,1\}$. Hence, if $\mathsf{BadChallenge}$ outputs $\mathsf{Chall} = \perp$, there exists no valid response $\{(\bar{c}_i, \mathbf{z}_i)\}$ regardless of the value of Chall. □

Combining the result of [89] with theorems 4.1, 4.2, 4.3, and 3.1, we obtain the following corollary.

Corollary 4.4. *For the language $\mathcal{L}_{\gamma,d} = \{\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}}\}$ with $\gamma = m^{0.5+\Omega(1)}$, there exists a NIZK argument system that is statistically NIZK and provides unbounded simulation-soundness under the* LWE *assumption.*

5 Public-Key Encryption with KDM-CCA2 Security from LWE

In this section, we describe a PKE scheme with KDM-CCA2 security under the LWE assumption by applying a technique suggested by Chandran *et al.* [27]. In [27], it was shown that applying the Naor-Yung paradigm to two schemes providing KDM-CPA security and standard IND-CPA security, respectively, can give KDM-CCA2 security as long as the underlying NIZK proof system is unbounded simulation-sound. Our scheme is obtained by applying this idea to a variant of the LWE-based system for which Alperin-Sheriff and Peikert [4] gave a proof of KDM-CPA security.

5.1 Definition

A public-key encryption scheme consists of a tuple of efficient algorithms ($\mathsf{Par\text{-}Gen}$, $\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt}$), 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.* [27], which extends the definition of Boneh *et al.* [22] to the chosen-ciphertext setting. As in [22,27,7], 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 5.1 ([27]). *A public-key encryption scheme for a function family $\mathcal{F}$ provides KDM-CCA2 security if no PPT adversary has noticeable advantage in the following 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. *On polynomially many occasions,* $\mathcal{A}$ *adaptively makes 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*

$$\mathbf{Adv}(\mathcal{A}) = |\Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{Enc}},\mathcal{O}_{\mathrm{Dec}}}(\{PK_i\}_{i=1}^N) \mid d = 0] - \Pr[1 \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{Enc}},\mathcal{O}_{\mathrm{Dec}}}(\{PK_i\}_{i=1}^N) \mid d = 1]|$$

5.2 Construction

In order to apply the Naor-Yung paradigm to the scheme of Alperin-Sheriff and Peikert [4], we have to take into account that the encryption algorithm of [4] relies on the HNF form of LWE [7] and requires the LWE secret $\mathbf{s} \in \mathbb{Z}^n$ to be sampled from the noise distribution. The reason is that their scheme uses public keys of the form $\mathbf{u} = -\mathbf{A} \cdot \mathbf{z}_0 + \mathbf{z}_1$, where $\mathbf{A} \sim U(\mathbb{Z}_q^{n\times m})$ and $\mathbf{z}_0$, $\mathbf{z}_1$ are short vectors with Gaussian entries. Their public key can be seen as dual Regev public keys where a perturbation term $\mathbf{z}_1 \in \mathbb{Z}^n$ has been introduced to prove security in the KDM setting. Since this term $\mathbf{z}_1$ gets multiplied by $\mathbf{s}$ in the encryption algorithm, $\mathbf{s}$ has to be small with respect to q in order not to hinder decryption.

In our setting, one difficulty is that the Micciancio-Vadhan proof system does not easily make it possible to prove the smallness of the random vector $\mathbf{s}$. For this reason, we rely on a modification of the Alperin-Sheriff-Peikert construction [4], which was proven KDM-CPA secure by He *et al.* [66]. In [66], it was proven that the perturbation term $\mathbf{z}_1$ can be removed from the public key of Alperin-Sheriff and Peikert [4] as long as the dimension n is large enough to give a few LWE samples without noise [44]. In turn, this modification allows the encryption algorithm to use a random vector $\mathbf{s}$ sampled from the uniform distribution $U(\mathbb{Z}_q^n)$.

The construction goes as follows.

Par-Gen(1^λ)**:** Given λ, select dimensions n, m, moduli q, p where p is prime and $q = p^2$, and Gaussian parameter r (to be specified below) and output $\Gamma = \{\lambda, n, m, q, p, r\}$.

Keygen(Γ)**:** 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 Gaussian vectors $\mathbf{z}_0, \mathbf{z}_1 \hookleftarrow D_{\mathbb{Z}^m,r}$ and compute

$$\mathbf{u}_0 = -\mathbf{A}\cdot\mathbf{z}_0 \ \in \mathbb{Z}_q^n, \qquad\qquad \mathbf{u}_1 = -\mathbf{A}\cdot\mathbf{z}_1 \ \in \mathbb{Z}_q^n \ .$$

Define the matrix $\mathbf{B}$ as

$$\mathbf{B} = \left[\begin{array}{c|c|c} \mathbf{A}^\top & \mathbf{0}^{m\times n} & \mathbf{0}^m \\ \mathbf{u}_0^\top & \mathbf{0}^{1\times n} & p \\ \hline \mathbf{0}^{m\times n} & \mathbf{A}^\top & \mathbf{0}^m \\ \mathbf{0}^{1\times n} & \mathbf{u}_1^\top & p \end{array}\right] \in \mathbb{Z}_q^{(2m+2)\times(2n+1)} \tag{9}$$

and for parameters $\gamma = m^{0.5+\Omega(1)}$ and $d = r\cdot\sqrt{2m+2}$, define the language $\mathcal{L}^{\mathrm{NY}} = \{\mathcal{L}^{\mathrm{NY}}_{\mathsf{zk}}, \mathcal{L}^{\mathrm{NY}}_{\mathsf{sound}}\}$, where

$$\mathcal{L}^{\mathrm{NY}}_{\mathsf{zk}} = \Big\{ (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}) \in \mathbb{Z}_q^{2m+2} \ \Big|\ \exists\, (\mathbf{s},\mathbf{e}) \ \in \ \mathbb{Z}_q^{2n+1}\times\mathbb{Z}^{2m+2} : \\ \|\mathbf{e}\| \le d \quad \wedge \quad [\mathbf{c}_{0,0}^\top \mid c_{0,1} \mid \mathbf{c}_{1,0}^\top \mid c_{1,1}]^\top = \mathbf{B}\cdot\mathbf{s}+\mathbf{e} \bmod q \Big\},$$

$$\mathcal{L}^{\mathrm{NY}}_{\mathsf{sound}} = \Big\{ (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}) \in \mathbb{Z}_q^{2m+2} \ \Big|\ \exists\, (\mathbf{s},\mathbf{e}) \ \in \ \mathbb{Z}_q^{2n+1}\times\mathbb{Z}^{2m+2} : \\ \|\mathbf{e}\| \le \gamma\cdot d \quad \wedge \quad [\mathbf{c}_{0,0}^\top \mid c_{0,1} \mid \mathbf{c}_{1,0}^\top \mid c_{1,1}]^\top = \mathbf{B}\cdot\mathbf{s}+\mathbf{e} \bmod q \Big\}. \tag{10}$$

3. Generate a common reference string $\mathsf{crs} := \big(\mathsf{crs}', pk_{\mathsf{LPKE}}, k, \mathsf{AHF}, \mathsf{OTS}\big)$ for the simulation-sound proof system Π^{uss} of Section 4.1 with its simulation trapdoor $\tau_{\mathsf{zk}} := sk_{\mathsf{LPKE}}$ for the language $\mathcal{L}^{\mathrm{NY}}$.

Output (PK, SK), where $PK := \big(\mathbf{A}, \mathbf{u}_0, \mathbf{u}_1, \mathsf{crs}\big)$ and $SK := \mathbf{z}_0 \in \mathbb{Z}^m$. The vector $\mathbf{z}_1$ is not used to decrypt and can be discarded.

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

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

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

2. Using witnesses $\mathbf{s} = [\mathbf{s}_0^\top \mid \mathbf{s}_1^\top \mid \mu]^\top \in \mathbb{Z}_q^{2n+1}$ and $\mathbf{e} = [\mathbf{e}_0^\top \mid \mathbf{e}_1^\top \mid \chi_0 \mid \chi_1]^\top$ (note that $\|\mathbf{e}\| \le r\cdot\sqrt{2m+2}$ with overwhelming probability), generate a simulation-sound NIZK proof that $(\mathbf{c}_{0,0}, c_{0,1})$ and $(\mathbf{c}_{1,0}, c_{1,1})$ encrypt the same $\mu \in \mathbb{Z}_p$. Namely, generate a proof $\boldsymbol{\pi}$ that

$$\mathbf{y} := \left[\begin{array}{c} \mathbf{c}_{0,0} \\ \hline c_{0,1} \\ \hline \mathbf{c}_{1,0} \\ \hline c_{1,1} \end{array}\right] = \mathbf{B}\cdot\mathbf{s}+\mathbf{e} \ \in \mathbb{Z}_q^{2m+2} \tag{11}$$

corresponds to an element $(\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1})$ of the language $\mathcal{L}^{\mathrm{NY}}$ defined by (10). This proof $\boldsymbol{\pi} = \big(\mathsf{VK}, (\mathbf{a}, \mathbf{z}), sig\big)$ is obtained by computing $\mathbf{c} = \mathsf{Hash}(k, (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}), \mathbf{a}, \mathsf{VK})$ 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 $SK = \mathbf{z}_0 \in \mathbb{Z}^m$, return $\perp$ if $\boldsymbol{\pi}$ does not properly verify. Otherwise, compute $\mu' = c_{0,1} + \mathbf{z}_0^\top \cdot \mathbf{c}_{0,0} \bmod q$ and output $\mu \in \mathbb{Z}_p$ which minimizes $|\mu' - p \cdot \mu \bmod q|$.

In the ciphertext, we note that it is sufficient to prove the statement (11) for a witness $\mu \in \mathbb{Z}_q$ (i.e., we do not have to prove that $\mu \in \mathbb{Z}_p$) since any $\mu \in \mathbb{Z}_q$ can be written $\mu = \mu_1 \cdot p + \mu_0$ with $\mu_0, \mu_1 \in \mathbb{Z}_p$. Hence, if the statement is true for some $\mu \in \mathbb{Z}_q$, it is also true for some $\mu_0 \in \mathbb{Z}_p$.

In the security proof, we use a trapdoor $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ for the lattice $\Lambda^\perp(\mathbf{A})$ as a membership testing trapdoor for $\mathcal{L}^{\mathrm{NY}}$. We observe that $\mathbf{T_A}$ can be used to compute a trapdoor $\mathbf{T_B} \in \mathbb{Z}^{(2m+2)\times(2m+2)}$ for $\Lambda^\perp(\mathbf{B})$. Indeed, $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ can be used to sample short $\mathbf{e}_0, \mathbf{e}_1 \in \mathbb{Z}^m$ such that $\mathbf{A} \cdot \mathbf{e}_0 = -\mathbf{u}_0 \bmod q$ and $\mathbf{A} \cdot \mathbf{e}_1 = \mathbf{u}_1 \bmod q$, which yield a vector $[\mathbf{e}_0^\top \mid 1 \mid \mathbf{e}_1^\top \mid -1]^\top$ of $\Lambda^\perp(\mathbf{B})$. Then, we can sample m independent vectors $\{\mathbf{e}_{i,0}\}_{i=1}^m$ for which $\mathbf{A} \cdot \mathbf{e}_{i,0} = \mathbf{0} \bmod q$, which yield $2m$ vectors of the form $[\mathbf{e}_{i,0}^\top \mid 0 \mid \mathbf{0}^m \mid 0]$ and $[\mathbf{0}^m \mid 0 \mid \mathbf{e}_{i,0}^\top \mid 0]$. We then obtain a $(2m+2)$-th vector as $[\mathbf{0}^m \mid 0 \mid p \cdot \mathbf{e}_1^\top \mid -p]^\top$. By gathering all these vectors, we obtain a full-rank set of short vectors in $\Lambda^\perp(\mathbf{B})$. We can then apply [84, Lemma 7.1] to turn it into a short basis $\mathbf{T_B}$ of $\Lambda^\perp(\mathbf{B})$. In turn, a trapdoor $\mathbf{T_B}$ for $\Lambda^\perp(\mathbf{B})$ can be used to test whether $\mathbf{y} \in \mathbb{Z}_q^{2m+2}$ belongs to $\mathcal{L}^{\mathrm{NY}}$ because, whenever it does, $\mathbf{T_B}$ allows computing both $\mathbf{s} \in \mathbb{Z}_q^{2n+1}$ and $\mathbf{e} \in \mathbb{Z}^{2m+2}$. We also note that a full-rank set of short vectors is sufficient to invert the LWE function, even without being a basis of $\Lambda^\perp(\mathbf{B})$.

The proof of the following theorem is based on standard techniques and the details are given in Appendix D.

Theorem 5.2. *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.*

Setting the parameters. We now specify a choice of parameters that is compatible with the constructions in Section 3 and Section 4.

Given security parameter λ, we set $n = \Omega(\lambda)$ and $r = \Omega(\sqrt{m})$. Since $r > \eta_\epsilon(\Lambda^\perp(\mathbf{A}))$ w.h.p. by [55, Lemma 5.3], this allows sampling vectors of $\Lambda^\perp(\mathbf{A})$ of norm $\leq r\sqrt{m}$ using a trapdoor (which is necessary for the security proof). The security of the KDM-CPA secure scheme from [66] requires prime p to be $p = \Omega(r^2\sqrt{n}\log n)$. In the security proof, we need to invert the LWE function for the matrix $\mathbf{B}$ using the full-rank set of $2m+2$ vectors (or a basis $\mathbf{T_B}$) obtained above. To this end, we have to ensure that the infinity norm of $\mathbf{T_B} \cdot \mathbf{e} \in \mathbb{Z}^{2m+2}$ is much smaller than modulus $q = p^2$, so that we can compute $\mathbf{T_B} \cdot \mathbf{e}$ over $\mathbb{Z}$ via $\mathbf{T_B} \cdot \mathbf{y} \bmod q$. To bound the entries of $\mathbf{T_B}$, we first note that, by using the

(fresh) trapdoor $\mathbf{T_A} \in \mathbb{Z}^{m \times m}$ for the lattice $\Lambda^{\perp}(\mathbf{A})$ and a Gaussian parameter $s = \Omega(\sqrt{m})$ in the preimage samplings, we can ensure that entries of vectors $\mathbf{e}_0, \mathbf{e}_1, \{\mathbf{e}_{i,0}\}_{i=1}^{m}$ are smaller than $\sqrt{m}r$ with overwhelming probability. Thus, entries of $\mathbf{T_B}$ (even when we apply [84, Lemma 7.1] that incurs a loss of at most $\sqrt{m}$ in norms) are bounded by pmr. As a result, the entries of $\mathbf{T_B} \cdot \mathbf{e}$ are smaller than $pm^{2.5}r^2$ with overwhelming probability. We thus require that $q \gg pm^{2.5}r^2$, i.e., $p \gg m^{2.5}r^2$. Furthermore, to make the scheme compatible with the trapdoor Σ-protocol of Section 4.2, modulus q must be set sufficiently larger than the parameter $\gamma \cdot d$, where $\gamma = m^{0.5+\Omega(1)}$ and $d = r \cdot \sqrt{2m+2}$. To meet all these conditions, we can choose parameters m, p, q such that $m = 2n\lceil \log q \rceil + O(\lambda)$, $p = \Omega(r^2 m^4)$ and $q = p^2$.

The proof system Π^{uss} of Section 4.1 then requires to encrypt the binary decomposition of the first message $\mathbf{a} = \{\mathbf{m}_i\}_{i=1}^{\xi}$ of the prover in the trapdoor Σ-protocol, via the lossy PKE scheme of Section 3. Since the bit-size of $\mathbf{a}$ is $\xi \cdot (2m+1)\lceil \log q \rceil$, where $\xi = \omega(1)$, we can set parameters (n'_0, n', m', q') and σ, α of the lossy PKE scheme as follows:

$$n'_0 = \xi \cdot (2m+1)\lceil \log q \rceil, \quad n' = n'_0 + \Omega(\lambda), \quad m' = 2n'\lceil \log q' \rceil + O(\lambda)$$
$$\sigma = O(m^4 uL), \quad q' = \Omega(m'^6 \sigma u L), \quad \alpha \cdot q = \Omega(\sqrt{n'}),$$

where $u = O(\log^2 \lambda)$ and $L = \mathsf{poly}(\lambda)$.

References

1. M. Abe. Securing "encryption + proof of knowledge" in the random oracle model. In *CT-RSA*, 2002.
2. T. Acar, M. Belenkiy, M. Bellare, and D. Cash. Cryptographic agility and its relation to circular encryption. In *Eurocrypt*, 2010.
3. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Eurocrypt*, 2010.
4. J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In *PKC*, 2012.
5. B. Applebaum. Key-dependent message security: Generic amplification and completeness theorems. In *Eurocrypt*, 2011.
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*, pages 483–501, 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. M. Backes, M. Dürmuth, and D. Unruh. OAEP is secure under key-dependent messages. In *Asiacrypt*, 2008.
11. B. Barak, I. Haitner, D. Hofheinz, and Y. Ishai. Bounded key-dependent message security. In *Eurocrypt*, 2010.

12. M. Bellare, V.-T. Hoang, and P. Rogaway. Foundations of garbled circuits. In *ACM-CCS*, 2012.
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. M. Bellare and P. Rogaway. Optimal asymmetric encryption. In *Eurocrypt*, 1994.
16. M. Bellare and S. Yilek. Encryption schemes secure under selective opening attack. Cryptology ePrint Archive: Report 2009/101, 2009.
17. S. Biagioni, D. Masny, and D. Venturi. Naor-Yung paradigm with shared randomness and applications. In *SCN*, 2016.
18. 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.
19. J. Black, P. Rogaway, and T. Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *SAC*, 2002.
20. D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Crypto*, 2004.
21. D. Boneh and M. K. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.
22. D. Boneh, S. Halevi, M. Hamburg, and R. Ostrovsky. Circular-secure encryption from Decision Diffie-Hellman. In *Crypto*, 2008.
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. J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Eurocrypt*, 2001.
27. 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.
28. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. Rothblum, and R. Rothblum. Fiat-Shamir from simpler assumptions. Cryptology ePrint Archive: Report 2018/1004.
29. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. Rothblum, R. Rothblum, and D. Wichs. Fiat-Shamir: From practice to theory. In *STOC*, 2019.
30. R. Canetti, Y. Chen, and L. Reyzin. On the correlation intractability of obfuscated pseudorandom functions. In *TCC 2016-A*, 2016.
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, M. Green, and S. Hohenberger. New definitions and separations for circular security. In *PKC*, 2012.

36. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. In *Eurocrypt*, 2010.
37. D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert. Bonsai trees, or how to delegate a lattice basis. Journal of Cryptology, 25(4):601-639, 2010.
38. 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.
39. R. Cramer. Modular design of secure, yet practical cryptographic protocols. PhD thesis, University of Amsterdam, 1996.
40. R. Cramer, I. Damgård, and B. Schoenmaeker. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Crypto*, 1994.
41. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *Eurocrypt*, 2002.
42. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Eurocrypt 2000*, 2000.
43. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero-knowledge. In *Crypto*, 2001.
44. Y. Dodis, S. Goldwasser, Y. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*, 2010.
45. N. Döttling. Low noise LPN: KDM secure public key encryption and sample amplification. In *PKC*, 2015.
46. U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero-knowledge under general assumptions. *SIAM J. of Computing*, 29(1), 1999.
47. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto*, 1986.
48. P.-A. Fouque and D. Pointcheval. Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. In *Asiacrypt*, 2001.
49. E. Freire, D. Hofheinz, K. Paterson, and C. Striecks. Programmable hash functions in the multilinear setting. In *Crypto*, 2013.
50. E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. *J. of Cryptology*, 26(1), 2013.
51. D. Galindo, J. Herranz, and J. Villar. Identity-based encryption with master key-dependent message security and leakage-resilience. In *ESORICS*, 2012.
52. J. Garay, P. MacKenzie, and K. Yang. Strengthening zero-knowledge protocols using signatures. In *Eurocrypt*, 2003.
53. S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, 2013.
54. R. Gennaro. Multi-trapdoor commitments and their applications to non-malleable protocols. In *Crypto*, 2004.
55. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
56. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Crypto*, 2013.
57. E.-J. Goh and S. Jarecki. A signature scheme as secure as the Diffie-Hellman problem. In *Eurocrypt*, 2003.
58. S. Goldwasser and S. Micali. Probabilistic encryption. *J. of Computer and System Sciences*, 28, 1984.
59. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 1989.

60. S. Goldwasser and Y. Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *FOCS*, 2003.
61. S. D. Gordon, J. Katz, and V. Vaikuntanathan. A group signature scheme from lattice assumptions. In *Asiacrypt*, 2010.
62. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt*, 2008.
63. I. Haitner and T. Holenstein. On the (im)possibility of key dependent encryption. In *TCC*, 2009.
64. S. Halevi, S. Myers, and C. Rackoff. On seed-incompressible functions. In *TCC*, 2008.
65. S. Han, S. Liu, and L. Lyu. Efficient KDM-CCA secure public-key encryption for polynomial functions. In *Asiacrypt*, 2016.
66. J. He, B. Li, X. Lu, D. Jia, and W. Jing. KDM and selective opening secure IBE based on the LWE problem. In *APKC@AsiaCCS 2017*, 2017.
67. D. Hofheinz. Circular chosen-ciphertext security with compact ciphertexts. In *Eurocrypt*, 2013.
68. D. Hofheinz, T. Jager, and A. Rupp. Public-key encryption with simulation-based selective-opening security and compact ciphertexts. In *TCC-B*, 2016.
69. D. Hofheinz and D. Unruh. Towards key-dependent message security in the standard model. In *Eurocrypt*, 2008.
70. J. Holmgren and A. Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In *FOCS*, 2018.
71. T. Jager. Verifiable random functions from weaker assumptions. In *TCC*, 2015.
72. C. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *Asiacrypt*, 2013.
73. J. Katz and N. Wang. Efficiency improvements for signatures schemes with tight security reductions. In *ACM-CCS*, 2003.
74. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, 2006.
75. F. Kitagawa, T. Matsuda, G. Hanaoka, and K. Tanaka. Efficient key dependent message security amplification against chosen ciphertext attacks. In *ICISC*, 2014.
76. F. Kitagawa, T. Matsuda, G. Hanaoka, and K. Tanaka. On the key dependent message security of the Fujisaki-Okamoto constructions. In *PKC*, 2016.
77. F. Kitagawa and K. Tanaka. A framework for achieving KDM-CCA secure public-key encryption. In *Asiacrypt*, 2018.
78. 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.
79. Y. Lindell. A simpler construction of CCA2-secure public-key encryption under general assumptions. *J. of Cryptology*, 19(3), 2006.
80. X. Lu, B. Li, and D. Jia. KDM-CCA security from RKA secure authenticated encryption. In *Eurocrypt*, 2015.
81. C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. of the ACM*, 1992.
82. P. MacKenzie and K. Yang. On simulation-sound trapdoor commitments. In *Eurocrypt*, 2004.
83. T. Malkin, I. Teranishi, and M. Yung. Efficient circuit-size independent public key encryption with KDM security. In *Eurocrypt*, 2012.
84. D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671. Kluwer, 2002.
85. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012.

86. D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *Crypto*, 2003.
87. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, 1990.
88. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Eurocrypt*, 1999.
89. C. Peikert and S. Shiehian. Non-interactive zero knowledge for NP from (plain) Learning With Errors. In *Crypto*, 2019.
90. C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *Crypto*, 2008.
91. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Jo. of Cryptology*, 13(3), 2000.
92. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.
93. A. Sahai. Non-malleable non-interactive zero knowledgeand adaptive chosen-ciphertext security. In *FOCS*, 1999.
94. V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In *Eurocrypt*, 1998.
95. Y. Tauman Kalai, G. Rothblum, and R. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In *Crypto*, 2017.
96. H. Wee. KDM-security via homomorphic smooth projective hashing. In *PKC*, 2016.
97. S. Yamada. Asymptotically compact adaptively secure lattice IBEs and verifiable random functions via generalized partitioning techniques. In *Crypto*, 2017.

A Additional Definitions

A.1 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 [72]). 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.

Definition A.1. *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*.*
- $\mathsf{Gen}_{\mathcal{L}}(1^\lambda, \mathcal{L})$ *takes as input a security parameter* λ *and the description of* $\mathcal{L}$ *which specifies a statement length* N*. It outputs the language-dependent part* $\mathsf{crs}_{\mathcal{L}}$ *of the common reference string* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$*.*
- $\mathsf{P}(\mathsf{crs}, x, w)$ *is a proving algorithm taking as input the common reference string* crs*, a statement* $x \in \{0,1\}^N$ *and a witness* w *such that* $(x, w) \in R_{\mathsf{zk}}$*. It outputs a proof* π*.*

- $\mathsf{V}(\mathsf{crs}, x, \pi)$ *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* $(x, w) \in R_{\mathsf{zk}}$*, we have*

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

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

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

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

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

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

Definition A.1 captures a notion of multi-theorem zero-knowledge, which allows the adversary to obtain proofs for multiple statements. Feige *et al.* [46] gave a generic transformation of a multi-theorem NIZK argument system from a single-theorem one (where the adversary can only invoke the oracle once).

We now recall the definition of simulation-soundness introduced in [93], 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.2 ([93,43]). *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.*

[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.2).

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. $\mathcal{A}$ *is given oracle access to* $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, \cdot)$. *At each query,* $\mathcal{A}$ *chooses a statement* $x \in \{0,1\}^N$ *and obtains* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathrm{zk}}, x)$.
3. $\mathcal{A}$ *outputs* $(x^\star, \pi^\star)$.

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

B Proof of Theorem 3.1

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 $\mathsf{pk} = (\mathbf{A}, \{\mathbf{A}_i\}_{i=1}^u)$ 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 $\|\mathbf{E}^\top [\mathbf{I}_m \mid \mathbf{R}_{F,t}] \cdot \mathbf{r}\|_\infty < q/4$ with all but negligible probability, so that the decryption algorithm recovers the initial message with probability exponentially close to 1. To prove this, notice that our definition of the randomness space R^{LPKE} imposes the inequality $\|\mathbf{r}\|_\infty \leq \|\mathbf{r}\| \leq \sigma\sqrt{2m}$. Besides, we also have $\|[\mathbf{I}_m \mid \mathbf{R}_{F,t}]\|_\infty \leq 1 + \|\mathbf{R}_{F,t}\|_\infty \leq 1 + m^3 u(L+1)$. Moreover, we have

$$\|\mathbf{E}^\top\|_\infty = \max_{i \in [n_0]} \sum_{j=1}^m |e_{ij}| \leq \sqrt{m} \cdot \max_{i \in [n_0]} \sqrt{\sum_{j=1}^m e_{ij}^2} \leq m \cdot \alpha q$$

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

$$\|\mathbf{E}^\top\|_\infty \cdot \|[\mathbf{I}_m \mid \mathbf{R}_{F,t}]\|_\infty \cdot \|\mathbf{r}\|_\infty \leq m\alpha q \cdot (1 + m^3 u(L+1)) \cdot \sigma\sqrt{2m} < q/4 \ .$$

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 $D_{\mathrm{loss}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ and $D_{\mathrm{inj}} = \{(pk, tk) \mid (pk, sk, tk) \leftarrow \mathsf{Keygen}(\Gamma, K)\}$ 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 $\{pk \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\Gamma, K)\}$ and $\{pk \mid (pk, sk, tk) \leftarrow \mathsf{LKeygen}(\Gamma, K')\}$ 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 $\mathbf{A}_i = \mathbf{A} \cdot \mathbf{R}_i + \kappa_i \cdot \mathbf{G}$, the Leftover Hash Lemma implies that the statistical distance between the distributions $\{(\mathbf{A}, \mathbf{A} \cdot \mathbf{R}_i) \mid \mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n \times m}), \mathbf{R}_i \hookleftarrow U(\{-1,1\}^{m \times m})\}$ and $\{(\mathbf{A}, \mathbf{A}_i) \mid \mathbf{A}, \mathbf{A}_i \hookleftarrow U(\mathbb{Z}_q^{n \times m})\}$ 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 under lossy tags. 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 $\big(pk = (\mathbf{A}, \{\mathbf{A}_i\}_{i=1}^u), sk = (\mathbf{S}, K), tk\big) \leftarrow \mathsf{Keygen}(\Gamma, K)$, and any message $\mathsf{Msg} \in \{0,1\}^{n_0}$, an encryption of Msg is generated as

$$\mathbf{c} = [\mathbf{A} \mid \mathbf{A}_{F,t}] \cdot \mathbf{r} + \begin{bmatrix} \mathbf{0}^{n-n_0} \\ \hline \mathsf{Msg} \cdot \lfloor q/2 \rfloor \end{bmatrix} \in \mathbb{Z}_q^n \ . \tag{12}$$

where $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^{2m}, \sigma}$ and $\bar{\mathbf{A}}_{F,t} = [\mathbf{A} \mid \mathbf{A} \cdot \mathbf{R}_{F,t} + \mathbf{G}] \in \mathbb{Z}_q^{n \times 2m}$. The matrix $\bar{\mathbf{A}}_{F,t}$ is of this form because t is a lossy tag (i.e., $(K, t) \notin \mathcal{R}_{\mathsf{BM}}$), which is equivalent to $F_{\mathsf{MAH}}(K, t) = 1$. 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 $\Lambda^\perp(\bar{\mathbf{A}}_{F,t})$) such that $\|\tilde{\mathbf{T}}_{F,t}\| \leq (\|\mathbf{R}_{F,t}\| + 1) \cdot \sqrt{5}$ and thus $\|\tilde{\mathbf{T}}_{F,t}\| \leq \sqrt{5} \cdot (m^{3.5} u \cdot (L+1) + 1)$. Again, by [55, Lemma 3.1], we know that $\eta_{2^{-m}}(\Lambda^\perp(\bar{\mathbf{A}}_{F,t})) \leq \|\tilde{\mathbf{T}}_{F,t}\| \cdot O(\sqrt{m})$. By the choice of the parameter $\sigma = O(m^4) \cdot u(L+1)$, we can conclude that $\sigma \geq \eta_{2^{-m}}(\Lambda^\perp(\bar{\mathbf{A}}_{F,t}))$. By applying [55, 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 $\|\tilde{\mathbf{T}}_{F,t}\| \leq \sqrt{5} \cdot (m^{3.5} u(L+1) + 1)$. By the choice of $\sigma = O(m^4) \cdot u(L+1)$, 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

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

is statistically close to

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

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 [37, 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. □

C A Simple Trapdoor Σ-Protocol for LWE

In this section, we describe a very simple 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

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

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

The construction is simpler than the Micciancio-Vadhan protocol, but its honest-verifier zero-knowledge property requires a super-polynomial modulus q.

As in the protocol of Section 4.2, the $\mathsf{TrapGen}$ algorithm inputs a membership-testing trapdoor $\tau_{\mathcal{L}_{B,B^*}}$ that consists of a small-norm full-rank integer matrix $\mathbf{T}_{\mathbf{B}} \in \mathbb{Z}^{m \times m}$ such that $\mathbf{T}_{\mathbf{B}} \cdot \mathbf{B} = \mathbf{0}^{m \times n} \bmod q$.

$\mathsf{Gen}_{\mathsf{par}}(1^\lambda)$: On input of a security parameter λ, choose integers B, B^* such that $B/B^* \in \mathsf{negl}(\lambda)$, a modulus q, dimensions n, m, and error rate $\alpha > 0$. Define $\mathsf{par} = \{\lambda, q, n, m, \alpha\}$.

$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}_{B,B^*})$: 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

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

$\mathsf{TrapGen}(\mathsf{par}, \tau_{\mathcal{L}_{B,B^*}})$: 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 $\mathsf{crs}_{\mathcal{L}} = \{\mathbf{B}, \gamma, B\}$, which defines $\mathsf{crs} = \big(\{\lambda, q, n, m, \alpha\}, \{\mathbf{B}, B, B^*\}\big)$, as well as the trapdoor $\tau_\Sigma = \mathbf{T_B}$.

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

1. P chooses $\mathbf{s}' \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{e}' \hookleftarrow U([-(\frac{B^*}{2} - B), (\frac{B^*}{2} - B)]^m)$ and computes $\mathbf{a} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \in \mathbb{Z}_q^m$, which is sent to V.
2. V sends a random challenge $\mathsf{Chall} \in \{0, 1\}$ to P.
3. P computes $\mathbf{z} = \mathbf{s}' + \mathsf{Chall} \cdot \mathbf{s} \in \mathbb{Z}_q^n$ and sends it to V.
4. Upon receiving $\mathbf{z} \in \mathbb{Z}_q^n$, V checks if

$$\mathbf{a} + \mathsf{Chall} \cdot \mathbf{y} - \mathbf{B} \cdot \mathbf{z} \ \in \left[-\frac{B^*}{2}, \frac{B^*}{2}\right]^m .$$

If this condition does not both hold, V halts and returns $\perp$.

$\mathsf{BadChallenge}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, \mathbf{y}, \mathbf{a}\big)$: Given $\tau_\Sigma = \mathbf{T_B}$, parse the first prover message as $\mathbf{a} \in \mathbb{Z}_q^m$. It uses the trapdoor $\mathbf{T_B}$ to determine if there exist vectors $\mathbf{s}' \in \mathbb{Z}_q^n$ and $\mathbf{e}' \in [-B^*/2, B^*/2]^m$ such that $\mathbf{a} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$. If so, it sets $\mathsf{Chall} = 0$. Otherwise, it sets $\mathsf{Chall} = 1$ if there exist $\mathbf{s}' \in \mathbb{Z}_q^n$ and $\mathbf{e}' \in [-B^*/2, B^*/2]^m$ such that $\mathbf{a} + \mathbf{y} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$. In any other case, it sets $\mathsf{Chall} = \perp$.

Lemma C.1. *The above construction is a trapdoor Σ-protocol for $\mathcal{L}_{B,B^\star}$.*

Proof. We first prove the HVZK property exactly as in [9, Theorem F.1]. Given a statement $\mathbf{y} \in \mathcal{L}_{\mathsf{zk}}$ and a challenge $\mathsf{Chall}^* \sim U(\{0, 1\})$, the simulator first samples vectors $\mathbf{z}^* \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e}^* \hookleftarrow U([-(\frac{B^*}{2} - B), (\frac{B^*}{2} - B)]^m)$ and computes $\mathbf{a}^* = \mathbf{B} \cdot \mathbf{z}^* + \mathbf{e}^* - \mathsf{Chall}^* \cdot \mathbf{y}$. Note that $(\mathbf{a}^*, \mathsf{Chall}^*, \mathbf{z}^*)$ 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 $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^n \times [-B, B]^m$ such that $\mathbf{y} = \mathbf{B} \cdot \mathbf{s} + \mathbf{e}$, so that the simulated $\mathbf{a}^*$ can be written $\mathbf{a}^* = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}'$ with $\mathbf{s}' = \mathbf{z}^* - \mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e}' = \mathbf{e}^* - \mathsf{Chall}^* \cdot \mathbf{e} \in \mathbb{Z}^m$. We assume that $\mathsf{Chall}^* = 1$ since the two distributions are exactly identical otherwise. In this case, we have $\mathbf{e}' \in [-B^*/2, B^*/2]^m$. Since $\frac{B}{B^*/2 - B} \in \mathsf{negl}(\lambda)$, it follows that the two distributions are statistically indistinguishable by [9, Lemma 2.1].

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 are left with showing that $\mathsf{BadChallenge}$ provides the correct result. For a given message $\mathbf{a} \in \mathbb{Z}_q^m$ sent by the prover, let us assume that there exist $\mathbf{s}' \in \mathbb{Z}_q^n$ and $\mathbf{e}' \in [-B^*/2, B^*/2]^m$ such that $\mathbf{a} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$ (which $\mathsf{BadChallenge}$ can detect using the trapdoor $\mathbf{T_B}$). In this case, $\mathbf{z} = \mathbf{s}'$ is a valid response for $\mathsf{Chall} = 0$. Moreover, no valid response can exist for $\mathsf{Chall} = 1$ as it would

contradict the assumption that $\mathbf{y} \notin \mathcal{L}_{\mathsf{sound}}$ by the soundness property. Now, let us assume that there exist $\mathbf{s}' \in \mathbb{Z}_q^n$ and $\mathbf{e}' \in [-B^*/2, B^*/2]^m$ such that $\mathbf{a} + \mathbf{y} = \mathbf{B} \cdot \mathbf{s}' + \mathbf{e}' \bmod q$. In this case, we know that the corresponding $\mathbf{z} = \mathbf{s}'$ is a valid response to $\mathsf{Chall} = 1$. By applying the same argument as before, we know that no valid response can exist for $\mathsf{Chall} = 0$ as it would contradict $\mathbf{y} \notin \mathcal{L}_{\mathsf{sound}}$. Hence, we find that $\mathsf{BadChallenge}$ always outputs the correct $\mathsf{Chall} \in \{0, 1\}$ that admits a valid response. □

D Proof of Theorem 5.2

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 $PK_i := \big(\mathbf{A}_i, \mathbf{u}_{i,0}, \mathbf{u}_{i,1}, \mathsf{crs}_i\big)$ for each $i \in [N]$. It gives $\{PK_i\}_{i=1}^N$ to the adversary $\mathcal{A}$ and keeps the private keys $\{SK_i = \mathbf{z}_{i,0}\}_{i=1}^N$ to itself. At each decryption query, $\mathcal{B}$ faithfully runs the real decryption algorithm using the private keys $\{SK_i = \mathbf{z}_{i,0}\}_{i=1}^N$. At each encryption query, the adversary $\mathcal{A}$ chooses an index $j \in [N]$ and an affine function $f_{\mathbf{V},w}$ specified by a matrix $\mathbf{V} = [\mathbf{v}_1|\ldots|\mathbf{v}_N] \in \mathbb{Z}_p^{m \times N}$ and a scalar $w \in \mathbb{Z}_p$. The challenger replies by generating a challenge ciphertext $\mathbf{C}^\star = \big(\mathbf{c}_{0,0}^\star, c_{0,1}^\star, \mathbf{c}_{1,0}^\star, c_{1,1}^\star, \boldsymbol{\pi}^\star\big)$ which is an encryption under PK_j of the function

$$f_{\mathbf{V},w}(\mathbf{Z}) = \sum_{i=1}^{N} \langle \mathbf{v}_i, \mathbf{z}_{i,0} \rangle + w \ ,$$

where $\mathbf{Z} = [\mathbf{z}_{1,0} \mid \ldots \mid \mathbf{z}_{N,0}] \in \mathbb{Z}^{m \times N}$. 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 $\{\mathbf{z}_{i,1}\}_{i=1}^N$ for which $\mathbf{u}_{i,1} = -\mathbf{A} \cdot \mathbf{z}_{i,1}$ and decrypts $(\mathbf{c}_{1,0}, c_{1,1})$ by computing $c_{1,1} + \mathbf{z}_{j,1}^\top \mathbf{c}_{1,0} \bmod q$. 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 $\big(\mathbf{c}_{0,0}^\top \mid c_{0,1} \mid \mathbf{c}_{1,0}^\top \mid c_{1,1}\big)^\top$ is outside the language $\mathcal{L}_{\mathrm{NY}}$ 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 D.1 shows that $\Pr[E_2] \leq N \cdot \mathbf{Adv}_{\mathcal{B}}^{\mathrm{sound}}(\lambda)$.

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 $\mathcal{L}_{\mathrm{NY}}$ 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 $f_{\mathbf{V},w}(\mathbf{Z})$. 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 [66], which would contradict the LWE assumption. The result of [66, Theorem 2] implies that $|\Pr[S_4] - \Pr[S_3]| \leq N \cdot Q \cdot \mathbf{Adv}^{\mathrm{lwe}}(\lambda)$, where Q is the number of encryption queries made by the adversary $\mathcal{A}$.

Game_5: We modify again the decryption oracle. This time, instead of using the backdoor keys $\{\mathbf{z}_{i,1}\}_{i=1}^N$ 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 $\{SK_i = \mathbf{z}_{i,0}\}_{i=1}^N$ to compute $\mu' = c_{0,1} + \mathbf{z}_{j,0}^\top \cdot \mathbf{c}_{0,0} \bmod q$ 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 D.2 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 $(j, f_{\mathbf{V},w})$, 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 $f_{\mathbf{V},w}(\mathbf{Z})$, 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 [66]. Since KDM-CPA security implies IND-CPA security, the result of [66, Theorem 2] thus implies $|\Pr[S_6] - \Pr[S_5]| \leq N \cdot Q \cdot \mathbf{Adv}^{\mathrm{lwe}}(\lambda)$.

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 Π, we compute it using the witnesses $(\mathbf{s}_1, \mathbf{s}_2, \mathbf{e}_0, \mathbf{e}_1, \chi_0, \chi_1)$. 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 that $|\Pr[S_1] - \Pr[S_7]| \leq \mathsf{negl}(\lambda)$ assuming that the LWE assumption holds and that Π provides unbounded simulation-soundness. □

Lemma D.1. *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 Π^{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 $\mathbf{u}_0, \mathbf{u}_1 \in \mathbb{Z}_q^n$ as well as a membership testing trapdoor $\tau_{\mathcal{L}}$ consisting of a trapdoor $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ for $\Lambda^{\perp}(\mathbf{A})$. Using $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$, $\mathcal{B}$ can generate short Gaussian vectors $\mathbf{z}_0, \mathbf{z}_1 \in \mathbb{Z}^m$ with standard deviation r such that $\mathbf{u}_0 = -\mathbf{A} \cdot \mathbf{z}_0$ and $\mathbf{u}_1 = -\mathbf{A} \cdot \mathbf{z}_1$. It 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 $\{SK_i\}_{i \in [N] \setminus \{i^\star\}}$. In the process of generating $\{PK_i\}_{i \in [N] \setminus \{i^\star\}}$, it also generates the matrices $\mathbf{A}_i \sim U(\mathbb{Z}_q^{n \times m})$ together with a trapdoor $\mathbf{T}_{\mathbf{A}_i}$ for $\Lambda^{\perp}(\mathbf{A}_i)$ for each $i \in [N] \setminus \{i^\star\}$. Then, $\mathcal{B}$ defines the $i^\star$-th public key as $PK_{i^\star} := \big(\mathbf{A}, \mathbf{u}_0, \mathbf{u}_1, \mathsf{crs}\big)$ 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 trapdoors $\mathbf{T}_{\mathbf{A}_i}$ for all matrices $\{\mathbf{A}_i\}_{i=1}^N$. 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 $\big(\mathbf{c}_{0,0}^\top \mid c_{0,1} \mid \mathbf{c}_{1,0}^\top \mid c_{1,1}\big)^\top$ is outside $\mathcal{L}_{\mathrm{NY}}$. 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 $\big(\mathbf{c}_{0,0}^\top \mid c_{0,1} \mid \mathbf{c}_{1,0}^\top \mid c_{1,1}\big)^\top$ 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. $\square$

Lemma D.2. 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 $\mathcal{L}_{\mathrm{NY}}$ specified by a matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and vectors $\mathbf{u}_0, \mathbf{u}_1 \in \mathbb{Z}_q^n$. It also receives a membership testing trapdoor $\tau_{\mathcal{L}}$ consisting of a trapdoor $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ for the lattice $\Lambda^{\perp}(\mathbf{A})$. Using $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$, $\mathcal{B}$ can generate short Gaussian vectors $\mathbf{z}_0, \mathbf{z}_1 \in \mathbb{Z}^m$ with standard deviation r such that $\mathbf{u}_0 = -\mathbf{A} \cdot \mathbf{z}_0$ and $\mathbf{u}_1 = -\mathbf{A} \cdot \mathbf{z}_1$. It also 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 $SK_{i^\star}$. It also sets $\mathbf{z}_{i^\star,0} := \mathbf{z}_0$ and $\mathbf{z}_{i^\star,1} := \mathbf{z}_1$. 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 their corresponding trapdoors $\mathbf{T}_{\mathbf{A}_i}$ for $\Lambda^{\perp}(\mathbf{A}_i)$ for each $i \in [N] \setminus \{i^\star\}$. It also defines $PK_{i^\star} := (\mathbf{A}, \mathbf{u}_0, \mathbf{u}_1, \mathsf{crs})$ and feeds $\mathcal{A}$ with the input $\{PK_i\}_{i \in [N]}$. In

the following, we denote by $\mathcal{L}_{\mathsf{sound}}^{\mathrm{NY},(i)}$ the language associated with the i-th public key PK_j (so that $\mathcal{L}_{\mathsf{sound}}^{\mathrm{NY},(i^\star)}$ is $\mathcal{B}$'s challenge language $\mathcal{L}_{\mathsf{sound}}^{\mathrm{NY}}$).

To answer an encryption query $(j, f_{\mathbf{V},\mathbf{w}})$, $\mathcal{B}$ 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 $f_{\mathbf{V},\mathbf{w}}(\mathbf{Z})$. 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 $(\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}) \in \mathcal{L}_{\mathsf{sound}}^{\mathrm{NY},(j)}$ (note that $\mathcal{B}$ can perform this check using the trapdoor $\tau_{\mathcal{L}}$ for the language $\mathcal{L}_{\mathsf{sound}}^{\mathrm{NY},(j)}$ $\mathbf{C}$ was never the result of an encryption query under PK_j, the reduction computes $\mu' = c_{0,0} + \mathbf{z}_j^\top \cdot \mathbf{c}_{0,0}$ as in Game_5 and returns μ 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 $(\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}) \notin \mathcal{L}_{\mathsf{sound}}^{\mathrm{NY},(i)}$, $\mathcal{B}$ halts (recall that $\mathcal{B}$ can always detect an occurrence of E_5 using the trapdoor $\tau_{\mathcal{L}}$ for $\mathcal{L}_{\mathsf{sound}}^{\mathrm{NY},(i)}$). If $i \neq i^\star$, $\mathcal{B}$ aborts and reports failure. Otherwise, it relays $(x = (\mathbf{c}_{0,0}, c_{0,1}, \mathbf{c}_{1,0}, c_{1,1}), \boldsymbol{\pi})$ 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 D.1 show that $\mathbf{Adv}_{\mathcal{B}}^{\mathrm{uss}}(\lambda) \geq \frac{1}{N} \cdot \Pr[E_5]$. $\square$