Non-Interactive CCA2-Secure Threshold Cryptosystems: Achieving Adaptive Security in the Standard Model Without Pairings

Julien Devevey, Benoît Libert, Khoa Nguyen, Thomas Peters, Moti Yung

Non-Interactive CCA2-Secure Threshold Cryptosystems: Achieving Adaptive Security in the Standard Model Without Pairings

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

[1] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, Inria, UCBL), France
[2] CNRS, Laboratoire LIP, France
[3] Nanyang Technological University, SPMS, Singapore
[4] FNRS and Université catholique de Louvain, Belgium
[5] Google and Columbia University, USA

Abstract. We consider threshold public-key encryption, where the decryption servers distributively hold the private key shares, and we need a threshold of these servers to decrypt the message (while the system remains secure when less than the threshold is corrupt). We investigate the notion of chosen-ciphertext secure threshold systems which has been historically hard to achieve. We further require the systems to be, both, adaptively secure (i.e., secure against a strong adversary making corruption decisions dynamically during the protocol), and non-interactive (i.e., where decryption servers do not interact amongst themselves but rather efficiently contribute, each, a single message). To date, only pairing-based implementations were known to achieve security in the standard security model without relaxation (i.e., without assuming the random oracle idealization) under the above stringent requirements. Here, we investigate how to achieve the above using other assumptions (in order to understand what other algebraic building blocks and mathematical assumptions are needed to extend the domain of encryption methods achieving the above). Specifically, we show realizations under the Decision Composite Residuosity (DCR) and Learning-With-Errors (LWE) assumptions.

Keywords. Threshold cryptography, adaptive security, non-interactive schemes, standard model, chosen-ciphertext security, DCR, LWE.

1 Introduction

Threshold cryptography [41,42,19,39] avoids a single point of failure by splitting the secret key into $\ell > 1$ shares and handing them over to different servers. This is done in such a way that any set of size at least $t \leq \ell$ servers can jointly compute private key operations whereas no subset of up to $t-1$ servers can similarly compute or otherwise compromise the cryptosystem's security.

Chosen-ciphertext (IND-CCA) security [76,81] is recognized as the *de facto* security notion for public-key encryption. Designing threshold IND-CCA2-secure

cryptosystems is non-trivial, and particularly challenging when we aim to combine all desirable properties. In this paper, we are interested in CCA2-secure threshold public-key encryption schemes that are simultaneously: secure under adaptive corruptions (namely, where adversaries can choose whom to corrupt based on the previously obtained information during the protocol), and non-interactive. By "non-interactive" we mean that decryption servers do not communicate with one another in a time consuming protocol, but rather only send a single message to a combiner which gathers these partial decryptions to produce the cleartext. In addition, our goal is to prove security in the standard model (i.e., without the random oracle idealization) and without assuming reliable erasures on behalf of decryption servers. Finally, we also wish to achieve robustness and prevent corrupted servers from hindering the decryption process.

We re-emphasize that we aim at simple non-interactive client/servers protocols where, in order to decrypt a message, a client sends a ciphertext to a decryption server that responds with a decryption share (along with a non-interactive proof of share correctness) without having to talk to other servers. As advocated in [85], such non-interactive protocols are attractive as they require no synchronization among servers, and do not rely on network latency guarantees.

To our knowledge, all solutions that combine all the aforementioned properties [70,67] rely on bilinear maps. In this paper, we consider the problem of schemes realizing the above under other well-established and non-pairing-related standard assumptions.

Non-Interactive Schemes. When we aim to avoid interaction during the decryption process in the design of threshold CCA2 schemes, the common stumbling block is that decryption servers often need to know whether an incoming ciphertext is valid or not before releasing their partial decryption result. The early solutions to this problem involved non-interactive zero-knowledge (NIZK) proofs [84,49] of ciphertext well-formedness in the random oracle model. In the standard model, Canetti and Goldwasser [25] thresholdized the Cramer-Shoup cryptosystem [34] by means of a randomized decryption protocol. Their approach involves shared randomizers in order to prevent partial decryptions on invalid ciphertexts from leaking information on the secret key shares. To remove interaction from the process, shareholders have to store a large number of pre-shared randomizers, which entails a prohibitively large storage cost. Cramer, Damgård and Ishai suggested [31] a non-interactive distributed randomization technique but it only supports a small number of servers. Boneh *et al.* [18] observed that, at least for static adversaries, these limitations can be avoided if shared randomizers are generated using non-interactive distributed pseudorandom functions.

In the static corruption setting, generic or partially generic CCA2-secure threshold constructions were proposed in [44,16,88,17]. Boneh, Boyen and Halevi [16] notably came up with the first fully non-interactive realization in the standard model. Their scheme crucially relies on pairings to publicly check the validity of ciphertexts, which drastically simplifies the problem of proving security in the threshold setting. Bilinear maps also provide robustness essentially for free, by making the validity of decryption shares publicly verifiable. Similar appli-

cations of the Canetti-Halevi-Katz [26] methodology to threshold cryptography were considered in [20,63]. Wee [88] subsequently laid out a framework for the design of non-interactive threshold signatures and CCA2-secure cryptosystems in the random oracle model under static corruptions.

More recently, Boneh *et al.* [17] introduced a tool, called *universal thresholdizer*, that essentially turns any non-interactive cryptographic scheme (such as public-key encryption, digital signatures or pseudorandom functions) into a threshold variant of the same primitive. Their compiler builds on fully homomorphic encryption (FHE) [53] and notably implies CCA2-secure non-interactive threshold encryption schemes in the static corruption setting.

Adaptive Corruptions. Most threshold cryptosystems (e.g., [84,25,44,49,16]) have been analyzed in a static corruption model, where the adversary commits to the set of corrupted servers *before* the protocol execution. Unfortunately, security under static corruptions does not imply security against more realistic adversaries that can adaptively corrupt servers based on previously and dynamically collected information. Canetti *et al.* [24] put forth adaptively secure key generation protocols for the distributed generation of discrete-log-based keys as well as adaptively secure threshold DSA signatures. Frankel, MacKenzie and Yung [51,50] independently showed different methods to achieve adaptive security. Their techniques were extended [7] to obtain proactive [77] RSA signatures.

The constructions of [24,51,50] inherently require interaction as they rely on the so-called "single inconsistent player" (SIP) technique. The latter consists of transforming a t-out-of-ℓ secret sharing into an additive t-out-of-t sharing of the same secret. In the latter case, only one server (which is randomly chosen ahead of time by the simulator among the ℓ severs) has an inconsistent internal state that causes the simulation to fail if it gets corrupted. Since this occurs with probability $\approx 1/2$, the stuck simulator can rewind the adversary and use different random coins with the hope of avoiding a new corruption of the inconsistent player. The threshold schemes of [51,50] thus proceed by switching from a (t, ℓ) polynomial secret sharing to a (t, t) additive secret sharing by first choosing a set of t participants. If a single participant fails to provide a valid contribution, the whole protocol must restart from scratch.[6] Jarecki and Lysyanskaya [59] extended the SIP technique to eliminate the need for erasures and described an adaptively secure variant of the Canetti-Goldwasser scheme [25]. Abe and Fehr [2] removed zero-knowledge proofs from the Jarecki-Lysyanskaya construction and proved it secure in (a variant of) the universal composability framework. However, [59,2] both require interaction during the decryption protocol. As in previous threshold variants of Cramer-Shoup, it requires either a large amount of synchronized interaction or a storage of a large number (i.e., proportional to the total number of decryption queries over the lifetime of the system) of pre-shared secrets. As argued in [85], none of the schemes in [25,1,59,2] is a simple, non-interactive, client/server protocol.

[6] An alternative approach, suggested in [80,7], requires each participant to store back-up shares of other participant's shares in such a way that the missing contributions of faulty servers can be reconstructed. However, it still requires additional interaction.

An adaptively secure extension of the Boneh-Boyen-Halevi construction [16] was proposed by Libert and Yung [69] using bilinear maps in composite order groups. It was subsequently shown [70,67] that pairing-based NIWI/NIZK arguments [56,60] can be used to remove interaction from threshold variants of Cramer-Shoup while proving security under adaptive corruptions in the standard model. A natural question to ask (from an algebraic perspective aiming not to put all one's eggs in the same [pairing] basket) is whether similarly simple non-interactive adaptively secure systems can be realized in the standard model outside the world of pairing-based cryptography.

Our Contribution. In this paper, we provide IND-CCA-secure non-interactive threshold cryptosystems proven secure in the sense of a game-based definition of *adaptive security* under the Decision Composite Residuosity assumption [78] (DCR) and the Learning-With-Errors (LWE) assumption [82].

Our first construction relies on both assumptions and features ciphertexts that are about as short as in standard (i.e., non-threshold) DCR-based CCA2-secure encryption schemes based on the Cramer-Shoup paradigm [35,22]. Indeed, ciphertexts are only roughly 3 times as large as those of [22]. Our scheme offers at least two advantages over the DCR-based system obtained by applying universal thresholdizers [17] to, e.g., the Camenisch-Shoup cryptosystem [22, Section 3.2]. First, in line with our goal, we can prove adaptive security under a polynomial reduction for $t, \ell = \mathsf{poly}(\lambda)$ without relying on complexity leveraging.[7] Indeed, universal thresholdizers are not known to enable adaptive security under our definition. Second, the scheme implies a more efficient *voting-friendly* threshold cryptosystem [14] in the sense that its ciphertexts can be publicly "downgraded" (by discarding the ciphertext components that ensure CCA2 security) into an additively homomorphic encryption scheme which is much more efficient than the voting-friendly scheme derived from [17,22]. The reason is that the shared decryption algorithm of [17] would proceed by homomorphically evaluating the decryption circuit of the standard Paillier-based scheme [22] over FHE-encrypted Paillier secret keys.

Our second construction relies on the sole LWE assumption. To our knowledge, it is the first adaptively secure non-interactive threshold cryptosystem with CCA2 security in the standard model under a quantum-safe assumption. One caveat is that, analogously to previous LWE-based threshold cryptosystems, it relies on an LWE assumption with super-polynomial approximation factor. The reason is that, as in all earlier threshold LWE-based constructions [12,17], decryption servers need to add a noise flooding term (which requires a super-polynomial modulus-to-noise ratio) in order to not leak their secret key shares when computing partial decryptions. It remains an open problem to prove adaptive security under a more common LWE assumption with polynomial approximation factor.

Technical Overview. Our schemes build on hash proof systems [35] and can be seen as pairing-free adaptation of constructions proposed by Libert and Yung

[7] When $t, \ell = O(\log \lambda)$, statically secure schemes can be proven adaptively secure by guessing the set of corrupted servers upfront.

[70]. In [70], they exploit the property that, in the security proofs of encryption schemes built upon hash proof systems, the simulator always knows the secret keys, which makes it easier to answer adaptive corruption queries (a similar observation was made by Dodis and Fazio [43] in the context of trace-and-revoke schemes). In the threshold setting, the reduction knows all secret key shares and can always provide a consistent internal state for adaptively corrupted servers.

To address the difficulty that valid ciphertexts are not publicly recognizable, [70,67] replaced the designated-verifier NIZK proofs of ciphertext validity [34,35] by publicly verifiable pairing-based NIZK arguments. This eliminates the need for randomized decryption – which was the culprit of interaction in [25,59] – since the shared decryption oracle can just reject invalid ciphertexts. This, in turn, preserves the entropy of the centralized secret key (which is used to create the challenge ciphertext in [34,35]) as decryption queries on valid ciphertexts do not decrease the entropy of secret keys conditionally on the adversary's view. In the challenge ciphertext, the reduction must be able to simulate a fake argument of ciphertext validity while making sure that the adversary cannot come up with such a fake argument in decryption queries. For this purpose, the underlying NIZK argument has to provide one-time simulation-soundness [83].

Our first scheme is a threshold version of (a variant of) an Elgamal-Paillier combination proposed in [22]. The public key contains $h = g^{4N \cdot x} \bmod N^2$, where N is a safe-prime product and $x \in \mathbb{Z}$ is the secret key. Messages $\mathsf{Msg} \in \mathbb{Z}_N$ are encrypted as $(C_0, C_1) = (g^{2N \cdot r}, (1+N)^{\mathsf{Msg}} \cdot h^r) \in (\mathbb{Z}^*_{N^2})^2$ and can be decrypted using x. The security proof of [22] involves a hybrid game where C_0 is sampled as a random quadratic residue (instead of a $2N$-th residue) in $\mathbb{Z}^*_{N^2}$ before computing $C_1 = (1+N)^{\mathsf{Msg}} \cdot C_0^{2x} \bmod N^2$. In order to exploit the entropy of $x \bmod N$ in the challenge phase, each ciphertext (C_0, C_1) should come with a simulation-sound NIZK proof/argument that C_0 is an N-th residue in $\mathbb{Z}^*_{N^2}$.

This NIZK component can be realized from recent results [23,27,79] on the standard-model instantiability of the Fiat-Shamir paradigm [48]. In our setting, we can use an argument of composite residuosity described by Libert *et al.* [66], which argues soundness in one shot (i.e., without parallel repetitions). However, the latter construction is somewhat an overkill for our purposes as it provides *unbounded* simulation-soundness (USS) while we only need *one-time* simulation-soundness in the context of threshold CCA2 security. We, thus, construct an optimized version of the NIZK argument of [66], where the common reference string (CRS) only contains $O(1)$ Paillier ciphertexts, instead of $O(\lambda)$ in [66]. This new optimized NIZK argument suffices for all applications that only need one-time simulation soundness.[8]

Like its unbounded counterpart [66], our one-time simulation-sound argument adapts a compiler from [65], which builds USS arguments from trapdoor Σ-protocols. In short, these are Σ-protocols in the CRS model where an efficiently computable function $\mathsf{BadChallenge}$ uses a trapdoor to compute the only

[8] Faust *et al.* [45] showed that Fiat-Shamir provides simulation-soundness "for free" in the ROM. However, their proof crucially relies on the random oracle modeling of hash functions and it is not known to immediately carry over to the standard model.

challenge Chall admitting a valid response z for a given false statement $x \notin \mathcal{L}$ and a given first prover message a. The USS argument of [65] uses a technique due to Damgård [36] that consists of having the prover first send an equivocable commitment to its first Σ-protocol message before opening the commitment in the response z. In [65], the equivocable commitment was replaced by a strengthened version of the $\mathcal{R}$-lossy encryption primitive of Boyle *et al.* [21]. In a nutshell, an $\mathcal{R}$-lossy PKE scheme is a tag-based encryption scheme where ciphertexts are injective for all tags t satisfying some relation $R(K,t)$ (where K is an initialization value chosen at key generation time) and equivocable when $R(K,t)=0$. By equivocating the ciphertext in all simulated proofs while keeping it extractable in the adversary's fake proof, we can use the extraction trapdoor to compute the $\mathsf{BadChallenge}$ function (in order to ensure soundness via the techniques of [27]), even after having simulated proofs by means of ciphertext equivocation. This can be seen as applying the simulation-sound zero-knowledge techniques of Garay *et al.* [52] in the context of the $\mathsf{BadChallenge}$ function methodology [27].

In [65], it was shown that the underlying equivocable $\mathcal{R}$-lossy PKE scheme can be instantiated from the DCR assumption using public keys comprised of $O(\lambda)$ Paillier ciphertexts. In Section 3, we show that, if we only need to simulate *one* argument of a false statement, we can use a more efficient $\mathcal{R}$-lossy PKE scheme for a different relation allowing for constant-size public keys. While [66] uses the bit-matching relation of [21] which incurs long public keys as it must be combined with admissible hash functions [15], we can simply use the inequality relation where $R(K,t)=1$ if and only if $K \neq t$. In our DCR-based instantiation, we can thus encrypt/commit to $\mathsf{Msg} \in \mathbb{Z}_N$ under the tag t by computing $\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}} \cdot r^N \bmod N^2$, which can be decrypted as a standard Paillier ciphertext when N divides the order of $u^t \cdot v$. When $u^t \cdot v$ is an N-th residue, we can equivocate ct by finding $r \in \mathbb{Z}_N^*$ that explains ct as an encryption of an arbitrary plaintext. By suitably programming $u, v \in \mathbb{Z}_{N^2}^*$, we can make sure that $u^t \cdot v$ is an N-th residue for one specific tag $t = K$. Importantly, we need to equivocate without knowing the factorization of N since, in our application to simulation-soundness, we rely on the DCR assumption to switch between settings where either only one tag is equivocable or all tags are equivocable.

We note that the above tools do not quite make valid ciphertexts publicly recognizable because the NIZK argument only guarantees that C_0 is a composite residue without proving that it is also a square in $\mathbb{Z}_{N^2}^*$. However, it does not affect the application to CCA2 security since decryption servers can simply square C_0 themselves to make sure that C_0^2 lives in the subgroup of $2N$-th residues before releasing decryption shares $C_0^{2 \cdot \mathsf{sk}_i}$.

Our LWE-based construction relies on the dual Regev cryptosystem [54], where public keys contain random matrices $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and $\mathbf{U} = \mathbf{A} \cdot \mathbf{R} \in \mathbb{Z}_q^{n \times L}$, for some $n, m, L \in \mathsf{poly}(\lambda)$ such that $n < m$, and secret keys are small-norm integer matrices $\mathbf{R} \in \mathbb{Z}^{m \times L}$. Since the columns of $\mathbf{R}$ have a lot of entropy conditionally on $(\mathbf{A}, \mathbf{U})$, it is tempting to adapt the approach of our DCR-based system and use LWE in an hash-proof-like fashion (as previously done in, e.g., [8]). However, this requires preventing the adversary from inferring information on $\mathbf{R}$ by

making decryption queries on ill-formed ciphertexts. This cannot be achieved via designated-verifier NIZK proofs [35] since known LWE-based hash proof systems (e.g., [62,89]) do not provide smoothness in the worst-case. Namely, nothing is guaranteed on the unpredictability of $\mathbf{R}^\top \mathbf{c}_0$ when $\mathbf{c}_0$ is neither a vector of LWE samples for $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ nor a uniform vector over $\mathbb{Z}_q^m$, but something in between (e.g., a vector $\mathbf{c}_0 = \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}_0$ where $\mathbf{e}_0 \in \mathbb{Z}^m$ is slightly too large).

To address the problem of showing that $\mathbf{c}_0$ is well-formed (i.e., of the form $\mathbf{c}_0 = \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}_0$ for a small enough $\mathbf{e}_0 \in \mathbb{Z}^m$), we replace the designated-verifier NIZK proof by a Fiat-Shamir-based [48] publicly verifiable NIZK argument, which is known to provide soundness in the standard model under the LWE assumption [79]. To avoid relying on a generic Karp reduction to the Graph Hamiltonicity language used in [27], we rely on the simulation-sound NIZK argument of Libert *et al.* [65, Appendix G] which allows showing that a vector $\mathbf{c}_0$ is indeed of the form $\mathbf{c}_0 = \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}_0$ for a small $\mathbf{e}_0 \in \mathbb{Z}^m$. Since their construction provides publicly verifiable arguments, its soundness property does not rely on the entropy of a verifier's secret key and bypasses the difficulties arising from the use of designated-verifier NIZK proofs. In particular, it keeps the verifier from accepting proofs for vectors $\mathbf{c}_0 = \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}_0$ where $\mathbf{e}_0$ is only slightly too large, which preserves the entropy of the centralized secret key $\mathbf{R} \in \mathbb{Z}^{m \times \ell}$.

In the threshold setting, both schemes share their secret keys using the linear integer secret sharing (LISS) primitive of Damgård and Thorbek [38], which are similar to linear secret sharing schemes except that they work over $\mathbb{Z}$. In our LWE-based construction, we crucially exploit the fact that LISS schemes have small linear reconstruction coefficients that can multiply decryption shares without blowing up the underlying noise terms. We could have alternatively used $\{0,1\}$-linear secret sharing (which can also express monotone Boolean formulas [64]) as in [17]. However, as observed in [68] in the adaptive corruption setting, LISS nicely interact with discrete Gaussian distributions and make it easier to analyze the remaining entropy of shared secret keys after all decryption queries and corruption queries. Indeed, our DCR-based TPKE bears similarities to the inner product functional encryption scheme of Agrawal *et al.* [6] in that it samples secret keys $x \in \mathbb{Z}$ from a Gaussian distribution over the integers. By sharing them with a LISS, we can adapt arguments used in [6,68] in order to assess the entropy of secret keys after all queries.

Related Work. Back in 2001, Fouque and Pointcheval [49] used the Naor-Yung paradigm [76] to construct a CCA2-secure threshold cryptosystem under the DCR assumption in the random oracle model. In Supplementary Material E, we show, as a comment, that the proof of IND-CCA security of [49] is actually incorrect as an adversary can break the soundness of the proof of plaintext equalities between Paillier ciphertexts with different moduli. It can be fixed by having the encryptor prove that the plaintext is a positive integer smaller than both moduli.

The first LWE-based threshold encryption scheme was proposed by Bendlin and Damgård [12] who showed a threshold version of Regev's cryptosystem [82]. Xie *et al.* [90] gave a threshold CCA-secure realization where the size of public

keys and ciphertexts grows at least linearly with the number of servers. Boneh *et al.* gave a compiler [17] that turns any IND-CCA secure into a non-interactive threshold variant thereof using fully homomorphic encryption. Bendlin *et al.* [13] considered lattice-based threshold signatures and IBE schemes. However, the servers can only compute an a priori bounded number of non-interactive private key operations without using interaction. Libert *et al.* [68] described non-interactive threshold pseudorandom functions from LWE. Our LWE-based TPKE and its security proof are actually inspired by their use of LISS schemes.

Organization. In Section 2, we first recall some definitions and tools that will be used in our constructions. Section 3 then presents our one-time simulation-sound NIZK arguments, which builds on our DCR-based $\mathcal{R}$-lossy PKE scheme described in Section 3.1. Our DCR-based threshold cryptosystem is explained in Section 4. Its variant based on the sole LWE assumption is given in Section 5. For simplicity, we first present non-robust variants of both schemes. In Supplementary Material C, we show that standard techniques can be applied to hedge against malicious adversaries.

2 Background and Definitions

2.1 Lattices

For any $q \geq 2$, $\mathbb{Z}_q$ denotes the ring of integers with addition and multiplication modulo q. If $\mathbf{x} \in \mathbb{R}^n$ is a vector, $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2}$ denotes its Euclidean norm and $\|\mathbf{x}\|_\infty = \max_i |x_i|$ its infinity norm. If $\mathbf{M}$ is a matrix over $\mathbb{R}$, then $\|\mathbf{M}\| := \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{M}\mathbf{x}\|}{\|\mathbf{x}\|}$ and $\|\mathbf{M}\|_\infty := \sup_{\mathbf{x}\neq 0} \frac{\|\mathbf{M}\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty}$ denote its induced norms. For a finite set S, $U(S)$ stands for the uniform distribution over S. If X and Y are distributions over the same domain, $\Delta(X, Y)$ denotes their statistical distance.

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

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

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

We now recall the definition of the Learning-With-Errors (LWE) assumption introduced by Regev [82].

Definition 2.1 (LWE assumption). *Let $m \geq n \geq 1$, $q \geq 2$ and $\alpha \in (0, 1)$ be functions of a security parameter λ. The* LWE *problem consists in distinguishing between the distributions $(\mathbf{A}, \mathbf{A}\mathbf{s}+\mathbf{e})$ and $U(\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^m)$, where $\mathbf{A} \sim U(\mathbb{Z}_q^{m\times n})$, $\mathbf{s} \sim U(\mathbb{Z}_q^n)$ and $\mathbf{e} \sim D_{\mathbb{Z}^m,\alpha q}$.*

Lemma 2.2 ([54, 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* $D_{\Lambda,s,\mathbf{c}}$*.*

Lemma 2.3 ([75], Lemma 4.4). *For* $\sigma = \omega(\sqrt{\log n})$*, there exists a negligible function* $\epsilon = \epsilon(n)$ *such that* $\Pr_{\mathbf{x} \hookleftarrow D_{\mathbb{Z}^n,\sigma}}[\|\mathbf{x}\| > \sigma\sqrt{n}] \leq \frac{1+\epsilon}{1-\epsilon} \cdot 2^{-n}$*.*

Lemma 2.4 ([68, Lemma 2.6]). *Let* $\epsilon \in (0,1), c \in \mathbb{R}$ *and* $\sigma > 0$*, such that* $\sigma \geq \sqrt{\ln 2(1+1/\epsilon)/\pi}$*. Then* $H_\infty(D_{\mathbb{Z},\sigma,c}) \geq \log\sigma - \log\left(1 + \frac{2\epsilon}{1-\epsilon}\right)$*. For* $\sigma = \Omega(\sqrt{n})$*, we get* $H_\infty(D_{\mathbb{Z},\sigma,c}) \geq \log(\sigma) - 2^{-n}$*.*

Lemma 2.5 ([47]). *Let* $\beta > 0, q \in \mathbb{Z}$ *and* $y \in \mathbb{Z}$*. Then, the following holds:* $\Delta(D_{\mathbb{Z}_q,\beta\cdot q,0}, D_{\mathbb{Z}_q,\beta\cdot q,y}) \leq \frac{|y|}{\beta q}$*.*

Lemma 2.6 ([74, Theorem 2]). *There exists an efficient randomized algorithm* $\mathsf{TrapGen}(1^n, 1^m, q)$ *that given any integers* $n \geq 1, q \geq 2$ *and sufficiently large* $m = O(n \log q)$ *outputs a matrix* $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ *and a trapdoor* $\mathbf{T_A}$ *such that the distribution of* $\mathbf{A}$ *is statistically close to uniform.*

Lemma 2.7 (Adapted from [54, Cor. 2.8]). *Let* $\Lambda' \subseteq \Lambda \subseteq \mathbb{R}^n$ *be two lattices with the same dimension. Let* $\epsilon \in (0, 1/2)$*. Then, for any* $c \in \mathbb{R}^n$ *and any* $\sigma \geq \eta_\epsilon(\Lambda')$*, the distribution* $D_{\Lambda,\sigma,c} \bmod \Lambda'$ *is within statistical distance* 2ϵ *from the uniform distribution over* Λ/Λ'*.*

2.2 Composite Residuosity Assumption

We now recall Paillier's Composite Residuosity assumption and its variant considered by Damgård and Jurik.

Definition 2.8 ([78,37]). *Let integers* $N = pq$ *and* $s > 1$ *for primes* p, q*. The* ***s*-Decision Composite Residuosity** (s-DCR) *assumption states that the distributions* $\{x = w^{N^s} \bmod N^{s+1} \mid w \leftarrow U(\mathbb{Z}_N^\star)\}$ *and* $\{x \mid x \leftarrow U(\mathbb{Z}_{N^{s+1}}^\star)\}$ *are computationally indistinguishable.*

It is known [37] that the s-DCR assumption is equivalent to the standard 1-DCR of [78] for any $s > 1$.

2.3 Linear Integer Secret Sharing

This section recalls the concept of linear integer secret sharing (LISS), as defined by Damgård and Thorbek [38]. Definitions below are taken from [86] where the secret to be shared lives in an interval $[-2^l, 2^l]$ centered in 0, for some $l \in \mathbb{N}$.

Definition 2.9. *A* ***monotone*** *access structure on* $[\ell]$ *is a non-empty collection* $\mathbb{A}$ *of sets* $A \subseteq [\ell]$ *such that* $\emptyset \notin \mathbb{A}$ *and, for all* $A \in \mathbb{A}$ *and all sets* B *such that* $A \subseteq B \subseteq [\ell]$*, we have* $B \in \mathbb{A}$*. For an integer* $t \in [\ell]$*, the* ***threshold*-***t* *access structure* $T_{t,\ell}$ *is the collection of sets* $A \subseteq [\ell]$ *such that* $|A| \geq t$*. Sets* $A \in \mathbb{A}$ *are called qualified and sets* $B \notin \mathbb{A}$ *are called forbidden.*

Let $P = [\ell]$ be a set of shareholders. In a LISS scheme, a dealer D wants to share a secret s in a publicly known interval $[-2^l, 2^l]$. To this end, D uses a share generating matrix $\mathbf{M} \in \mathbb{Z}^{d \times e}$ and a random vector $\boldsymbol{\rho} = (s, \rho_2, \ldots, \rho_e)^\top$, where s is the secret to be shared and $\{\rho_i\}_{i=2}^e$ are randomly sampled in $[-2^{l_0+\lambda}, 2^{l_0+\lambda}]^{e-1}$, for some $l_0 \geq l \in \ell$. Usually, the distribution of the ρ_i is uniform but, in the following, we will set $l_0 = l$ and $\rho_i \hookleftarrow D_{\mathbb{Z},\sigma}$. The dealer D computes a vector $\boldsymbol{s} = (s_1, \ldots, s_d)^\top$ of share units as $\boldsymbol{s} = (s_1, \ldots, s_d)^\top = \mathbf{M} \cdot \boldsymbol{\rho} \in \mathbb{Z}^d$. Each party in $P = \{1, \ldots, \ell\}$ is assigned a set of share units. Letting $\psi : \{1, \ldots, d\} \to P$ be a surjective function, the i-th share unit s_i is assigned to the shareholder $\psi(i) \in P$, in which case player $\psi(i)$ is said to own the i-th row of $\mathbf{M}$. If $A \subseteq P$ is a set of shareholders, $\mathbf{M}_A \in \mathbb{Z}^{d_A \times e}$ denotes the set of rows jointly owned by A. Likewise, $\boldsymbol{s}_A \in \mathbb{Z}^{d_A}$ denotes the restriction of $\boldsymbol{s} \in \mathbb{Z}^d$ to the coordinates jointly owned by the parties in A. The j-th shareholder's share consists of $\boldsymbol{s}_{\psi^{-1}(j)} \in \mathbb{Z}^{d_j}$, so that it receives $d_j = |\psi^{-1}(j)|$ out of the $d = \sum_{j=1}^{\ell} d_j$ share units. The *expansion rate* $\mu = d/\ell$ is defined to be the average number of share units per player.

There exist security notions for LISS schemes but, since we do not explicitly rely on them, we omit their exposition for conciseness.

To construct LISS schemes, Damgård and Thorbek [38] used integer span programs [33].

Definition 2.10 ([33]). *An integer span program (ISP) is a tuple formed by three elements $\mathcal{M} = (\mathbf{M}, \psi, \boldsymbol{\varepsilon})$, where $\mathbf{M} \in \mathbb{Z}^{d \times e}$ is an integer matrix whose rows are labeled by a surjective function $\psi : \{1, \ldots, d\} \to \{1, \ldots, \ell\}$ and $\boldsymbol{\varepsilon} = (1, 0, \ldots, 0)$ is called target vector. The size of $\mathcal{M}$ is the number of rows d in $\mathbf{M}$.*

Definition 2.11. *Let Γ be a monotone access structure and let $\mathcal{M} = (\mathbf{M}, \psi, \boldsymbol{\varepsilon})$ an integer span program. Then, $\mathcal{M}$ is an ISP for Γ if it computes Γ: namely, for all $A \subseteq \{1, \ldots, \ell\}$, the following conditions hold:*

1. *If $A \in \Gamma$, there is a reconstruction vector $\boldsymbol{\lambda} \in \mathbb{Z}^{d_A}$ such that $\boldsymbol{\lambda}^\top \cdot \mathbf{M}_A = \boldsymbol{\varepsilon}^\top$.*
2. *If $A \notin \Gamma$, there exists $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_e)^\top \in \mathbb{Z}^e$ such that $\mathbf{M}_A \cdot \boldsymbol{\kappa} = \mathbf{0} \in \mathbb{Z}^d$ and $\boldsymbol{\kappa}^\top \cdot \boldsymbol{\varepsilon} = 1$. In this case, $\boldsymbol{\kappa}$ is called a sweeping vector for A.*

We also define $\kappa_{\max} = \max\{|a| \mid a \text{ is an entry in some sweeping vector}\}$.

Damgård and Thorbek [38] observed that a LISS can be built by setting the share generating matrix to be the matrix $\mathbf{M}$ of an ISP $\mathcal{M} = (\mathbf{M}, \psi, \boldsymbol{\varepsilon})$ that computes the access structure Γ. We may then specify a LISS scheme $\mathcal{L} = (\mathcal{M} = (\mathbf{M}, \psi, \boldsymbol{\varepsilon}), \Gamma, \mathcal{R}, \mathcal{K})$ by an ISP for the access structure Γ, a space $\mathcal{R}$ of reconstruction vectors satisfying Condition 1 of Definition 2.11, and a space $\mathcal{K}$ of sweeping vectors satisfying Condition 2.

The last step is building an ISP for any access structure with small reconstruction vectors and small sweeping vectors. Damgård and Thorbek showed in [38] that LISS schemes can be obtained from [11,33]. While the Benaloh-Leichter (BL) secret sharing [11] was designed to work over finite groups, it was generalized in [38] to share integers using access structures consisting of monotone Boolean formulas. In turn, this implies a LISS scheme for any threshold access structure by applying a result of Valiant [87,55]. Their LISS scheme

built upon Benaloh-Leichter [11] satisfies what we want: as can be observed from [38, Lemma 4], every coefficient of any reconstruction vector $\boldsymbol{\lambda}$ lives in $\{-1, 0, 1\}$ and [38, Lemma 5] shows that $\kappa_{\max} = 1$. Let a monotone Boolean formula f, then the BL-based technique allows us to build binary share distribution matrices $\mathbf{M} \in \{0,1\}^{d \times e}$ such that $d, e = O(\text{size}(f))$. Moreover they have at most $\text{depth}(f) + 1$ non-zero entries, so that each share unit s_i has magnitude $O(2^{l_0+\lambda} \cdot \text{depth}(f))$.

Finally, Valiant's result [87] implies the existence of a monotone Boolean formula of the threshold-t function $T_{t,\ell}$, which has size $d = O(\ell^{5.3})$ and depth $O(\log \ell)$. Recall that each player will receive about d/ℓ rows of $\mathbf{M}$ on average, then the average share size is $O(\ell^{4.3} \cdot (l_0 + \lambda + \log\log \ell))$ bits. Valiant's construction was improved by Hoory *et al.* [58] who gave a monotone formula of size $O(\ell^{1+\sqrt{2}})$ and depth $O(\log \ell)$ for the majority function.[9] This in turn reduces the average share size to $O(\ell^{\sqrt{2}} \cdot (l_0 + \lambda + \log\log \ell))$ bits.

2.4 Threshold PKE

In this section, we slightly adapt the TPKE syntax defined by Boneh *et al.* [17].

Definition 2.12 (Threshold PKE). *For any $\ell \in \mathbb{N}$, let $\mathbb{S}(\ell)$ be a class of efficient monotone access structure on $[\ell]$. A Threshold PKE scheme (TPKE) for the family $\mathbb{S}(\cdot)$ is a tuple of efficient* PPT *algorithms* $(\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{PartDec}, \mathsf{PartVerify}, \mathsf{Combine})$ *with the following specifications:*

- $\mathsf{Keygen}(1^\lambda, \mathbb{A}) \to (\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$*: On input a security parameter λ, $\mathbb{A} \in \mathbb{S}(\ell)$ an access structure on $[\ell]$, for some ℓ (corresponding to the biggest element(s) of $\mathbb{A}$ for the inclusion), the algorithm outputs a set of public parameters* pp *(which are implicit in the inputs of all other algorithms and) that defines the message space $\mathcal{M}$, a public key* pk *and a set of secret key shares* $\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell$*.*
- $\mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg}) \to \mathsf{ct}$*: On input the public parameters* pp*, the encryption key* pk *and a message* $\mathsf{Msg} \in \mathcal{M}$*, the algorithm outputs a ciphertext* ct*.*
- $\mathsf{PartDec}(\mathsf{pk}, \mathsf{ct}, \mathsf{sk}_i) \to \mu_i$*: Given public parameters* pp*, a ciphertext* ct *and a secret key share* sk_i*, this algorithm outputs a partial decryption μ_i.*
- $\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}, \mu_i) \to \mathsf{b} \in \{0,1\}$*: On input of public parameters* pp*, a ciphertext* ct *and a partial decryption μ_i, this algorithm outputs a bit* b*.*
- $\mathsf{Combine}(\mathsf{pk}, B = (\mathcal{S}, \{\phi(\mu_i)\}_{i \in \mathcal{S}}), \mathsf{ct}) \to \mathsf{Msg}'$*: Given public parameters and a set of images of ϕ of partial decryptions, the algorithm outputs a message $\mathsf{Msg}' \in \mathcal{M}$. The function ϕ is public and deterministic.*[10]

We aim to construct a TPKE scheme that satisfies the following compactness, correctness and requirements. Our definition of compactness captures that the

[9] Note that a threshold-t function can be obtained from the majority function by fixing the desired number of input bits, so that we need a majority function of size $\leq 2\ell$ to construct a threshold function $T_{t,\ell}$.

[10] It helps defining robustness. For non-robust TPKE, ϕ is the identity function.

size of ciphertexts and public keys is not impacted by the number of servers. Unlike [17], the next definition relies on the family $\mathbb{S} = \cup_{\ell \in \mathsf{poly}(\lambda)} \mathbb{S}(\ell)$.

Definition 2.13 (Compactness [17]). *A TPKE scheme satisfies compactness if there exist polynomials P and Q such that $\forall \lambda, \forall \mathbb{A} \in \mathbb{S}, |\mathsf{pk}| \leq P(\lambda) \wedge |\mathsf{ct}| \leq Q(\lambda)$, where $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$ and the ciphertext is generated as $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg})$ for any $\mathsf{Msg} \in \mathcal{M}$.*

Definition 2.14 (Decryption Correctness). *A TPKE provides decryption correctness if the following holds. For any $\lambda \in \mathbb{N}$, any access structure $\mathbb{A} \in \mathbb{S}$, any set $\mathcal{S} \in \mathbb{A}$ and any message $\mathsf{Msg} \in \mathcal{M}$, if we run $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$, $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg})$ and then $\mu_i \leftarrow \mathsf{PartDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}), \forall i \in \mathcal{S}$, we have $\Pr[\mathsf{Combine}(\mathsf{pk}, (\mathcal{S}, \{\phi(\mu_i)\}_{i \in \mathcal{S}}), \mathsf{ct}) = \mathsf{Msg}] = 1 - \mathrm{negl}(\lambda)$.*

Definition 2.15 (Partial Verification Correctness). *A TPKE provides partial verification correctness if the following holds. For any $\lambda \in \mathbb{N}$, any $\mathbb{A} \in \mathbb{S}$, any $\mathcal{S} \in \mathbb{A}$ and any message $\mathsf{Msg} \in \mathcal{M}$, if we run $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$, $\mathsf{ct} \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg})$ and $\mu_i \leftarrow \mathsf{PartDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}), \forall i \in \mathcal{S}$, then $\Pr[\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}, \mu_i) = 1] = 1 - \mathrm{negl}(\lambda)$.*

We can now define chosen-ciphertext security in a model allowing the adversary to adaptively corrupt decryption servers.

Definition 2.16 (Adaptive-CCA security for TPKE). *A TPKE scheme provides chosen-ciphertext security under adaptive corruptions if no PPT adversary $\mathcal{A}$ has non-negligible advantage in the following game.*

1. *On input the the security parameter λ, $\mathcal{A}$ chooses an access structure $\mathbb{A}$.*
2. *The challenger generates $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$. It sends $(\mathsf{pp}, \mathsf{pk})$ to $\mathcal{A}$ and initializes an empty set $\mathcal{C} = \emptyset$.*
3. *$\mathcal{A}$ can adaptively interleave the following queries:*
   - ***Corruption**: $\mathcal{A}$ sends the challenger an index $i \in [\ell]$. The challenger replies by returning the share sk_i and updating the set $\mathcal{C} = \mathcal{C} \cup \{i\}$.*
   - ***Partial Decryption**: $\mathcal{A}$ chooses an index $i \in [\ell]$ and a ciphertext ct and the challenger returns a partial decryption $\mu_i \leftarrow \mathsf{PartDec}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct})$.*
4. *$\mathcal{A}$ chooses $\mathsf{Msg}_0^\star, \mathsf{Msg}_1^\star \in \mathcal{M}$. The challenger replies with a challenge ciphertext $\mathsf{ct}^* \leftarrow \mathsf{Encrypt}(\mathsf{pk}, \mathsf{Msg}_b^\star)$, where $b \hookleftarrow U(\{0,1\})$ is a random bit.*
5. *$\mathcal{A}$ makes more corruption and partial decryption queries subject to the following condition which must be satisfied at any time. Let $\mathcal{C} \subset [\ell]$ the set of corrupted servers and let $\mathcal{C}^*$ the subset of indexes $j \in [\ell]$ such that $\mathcal{A}$ made a decryption query of the form (j, ct^*). Then, it is required that $\mathcal{C} \cup \mathcal{C}^* \notin \mathbb{A}$.*
6. *The experiment ends with $\mathcal{A}$ outputting a bit $b' \in \{0,1\}$.*

The advantage of $\mathcal{A}$ is defined as $\mathbf{Adv}^{\text{ind-cca}}(\mathcal{A}) := \left| \Pr[b' = b] - \frac{1}{2} \right|$.

We now recall the notion of *robustness*, which informally captures that no malicious adversary can prevent a honest majority from decrypting a valid ciphertext.

Definition 2.17 ([17]). *A TPKE scheme satisfies* **robustness** *if no PPT adversary $\mathcal{A}$ can cause the following experiment $\mathsf{Expt}^{\mathrm{robust}}_{\mathcal{A},\mathsf{TPKE}}(1^\lambda)$ to output 1 with non-negligible probability.*

1. *On input the security parameter λ, $\mathcal{A}$ chooses an access structure $\mathbb{A}$.*
2. *The challenger samples $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$ and provides $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$ to $\mathcal{A}$.*
3. *$\mathcal{A}$ outputs a partial decryption forgery $\left(\mathsf{ct}^\star, \mu_i^\star, i\right)$, where $i \in [\ell]$.*
4. *The experiment outputs 1 if we have $\phi(\hat{\mu}_i^\star) \neq \phi(\mathsf{PartDec}(\mathsf{pk}, \mathsf{sk}_i, \mathsf{ct}^\star))$ while $\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}^\star, \mu_i^\star) = 1$.*

We note that the function ϕ allows considering as robust a TPKE such that $\mu_i^\star = (\hat{\mu}_i^\star, \pi_i^\star)$ and where $\mathsf{Combine}$ only runs on $\hat{\mu}_i^\star$ and not on $\pi_i^\star$. While, given $(\hat{\mu}_i^\star, \pi_i^\star)$, $\mathsf{Combine}$ could have simply striped $\pi_i^\star$, such a formalization would prevent showing as robust a TPKE where $\hat{\mu}_i^\star$ is an element of an admissible language and $\pi_i^\star$ is a probabilistic membership argument whose validity does not necessarily guarantee that $\pi_i^\star$ is in the range of honestly computed arguments. Thanks to ϕ, such case will not be artificially discarded.

A weaker robustness notion, a.k.a. consistency [16], captures the robustness of schemes where the generation of $\hat{\mu}_i^\star$ is probabilistic and its validity tolerates a gap with respect to honestly computed statements. We recall it in Supplementary Material A.2. Here, we will focus on the (stronger) notion of robustness.

2.5 Correlation Intractable Hash Functions

We consider unique-output efficiently searchable relations [23].

Definition 2.18. *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 $f(x) = y$.*

Let $\lambda \in \mathbb{N}$ 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 functions induced by efficient algorithms $(\mathsf{Gen}, \mathsf{Hash})$, where $\mathsf{Gen}(1^\lambda)$ outputs a key $k \in \{0,1\}^{s(\lambda)}$ and $\mathsf{Hash}(k, x)$ computes $h_\lambda(k, x) \in \{0,1\}^{m(\lambda)}$.

Definition 2.19. *For a relation ensemble $\{R_\lambda \subseteq \{0,1\}^{n(\lambda)} \times \{0,1\}^{m(\lambda)}\}$, a hash function family $\mathcal{H} = \{h_\lambda : \{0,1\}^{s(\lambda)} \times \{0,1\}^{n(\lambda)} \to \{0,1\}^{m(\lambda)}\}$ is* **R-correlation intractable** *if, for any probabilistic polynomial time (PPT) adversary $\mathbb{A}$, we have $\Pr\left[k \leftarrow \mathsf{Gen}(1^\lambda)),\ x \leftarrow \mathcal{A}(k) : (x, h_\lambda(k, x)) \in R\right] = \mathsf{negl}(\lambda)$.*

Peikert and Shiehian [79] described a correlation-intractable hash family for any searchable relation (in the sense of Definition 2.18) defined by functions f of bounded depth. When f is computable by a branching program, their construction relies on the standard SIS assumption with polynomial approximation factors. Under the LWE assumption with polynomial approximation factors, their bootstrapping theorem allows handling arbitrary bounded-depth functions.

2.6 Trapdoor Σ-protocols

Canetti *et al.* [27] considered a definition of Σ-protocols that slightly differs from the usual formulation [32,30].

Definition 2.20 (Adapted from [27,9]). *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:** P *and* V *both take as input* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$*, with* $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^{\lambda})$ *and* $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$*, and a statement x and proceed as follows: (i)* P *takes in $w \in R_{\mathsf{zk}}(x)$, computes $(\mathbf{a}, st) \leftarrow \mathsf{P}(\mathsf{crs}, x, w)$ and sends $\mathbf{a}$ to the verifier; (ii)* V *sends back a random challenge* Chall *from the challenge space $\mathcal{C}$; (iii)* P *finally sends a response $\mathbf{z} = \mathsf{P}(\mathsf{crs}, x, w, \mathbf{a}, \mathsf{Chall}, st)$ to* V*; (iv) On input of $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$,* V *outputs 1 or 0.*
- **Completeness:** *If $(x, w) \in R_{\mathsf{zk}}$ and* P *honestly computes $(\mathbf{a}, \mathbf{z})$ for a challenge* Chall*,* $\mathsf{V}(\mathsf{crs}, x, (\mathbf{a}, \mathsf{Chall}, \mathbf{z}))$ *outputs 1 with probability $1 - \mathsf{negl}(\lambda)$.*
- **Special zero-knowledge:** *There is a PPT simulator* ZKSim *that inputs* crs*, $x \in \mathcal{L}_{\mathsf{zk}}$ and a challenge $\mathsf{Chall} \in \mathcal{C}$. It outputs $(\mathbf{a}, \mathbf{z}) \leftarrow \mathsf{ZKSim}(\mathsf{crs}, x, \mathsf{Chall})$ such that $(\mathbf{a}, \mathsf{Chall}, \mathbf{z})$ is computationally indistinguishable from a real transcript with challenge* Chall *(for $w \in R_{zk}(x)$).*
- **Special soundness:** *For any CRS* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$ *obtained as* $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^{\lambda})$*,* $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$*, any $x \notin \mathcal{L}_{\mathsf{sound}}$, and any first message $\mathbf{a}$ sent by* P*, there is at most one challenge* $\mathsf{Chall} = f(\mathsf{crs}, x, \mathbf{a})$ *for which an accepting transcript* $(\mathsf{crs}, x, \mathbf{a}, \mathsf{Chall}, \mathbf{z})$ *exists for some third message $\mathbf{z}$. The function f is called the "bad challenge function" of Π. That is, if $x \notin \mathcal{L}_{\mathsf{sound}}$ and the challenge differs from the bad challenge, the verifier never accepts.*

Definition 2.20 is taken from [27] 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.* [27] define *trapdoor* Σ-protocols as Σ-protocols where the bad challenge function is efficiently computable using a trapdoor. Here, we use a definition where the CRS and the trapdoor may depend on the language.

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 2.21 (Adapted from [27]). *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 2.20 and there exist PPT algorithms* $(\mathsf{TrapGen}, \mathsf{BadChallenge})$ *with the following properties.*

- $\mathsf{Gen}_{\mathsf{par}}$ *inputs $\lambda \in \mathbb{N}$ and outputs public parameters* $\mathsf{par} \leftarrow \mathsf{Gen}_{\mathsf{par}}(1^{\lambda})$.
- $\mathsf{Gen}_{\mathcal{L}}$ *is a randomized algorithm that, on input of public parameters* par*, outputs the language-dependent part* $\mathsf{crs}_{\mathcal{L}} \leftarrow \mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ *of* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$.

- $\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$ *takes as input public parameters* par *and a membership-testing trapdoor* $\tau_{\mathcal{L}}$ *for the language* $\mathcal{L}_{\mathsf{sound}}$*. It outputs a common reference string* $\mathsf{crs}_{\mathcal{L}}$ *and a trapdoor* $\tau_{\Sigma} \in \{0,1\}^{\ell_\tau}$*, for some* $\ell_\tau(\lambda)$*.*
- $\mathsf{BadChallenge}(\tau_{\Sigma}, \mathsf{crs}, x, \mathbf{a})$ *takes in a trapdoor* τ_{Σ}*, a CRS* $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}})$*, an instance* x*, and a first prover message* $\mathbf{a}$*. It outputs a challenge* Chall*.*

In addition, the following properties are required.

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

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

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

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

2.7 $\mathcal{R}$-Lossy Public-Key Encryption With Efficient Opening

In [65], Libert *et al.* formalized a generalization of the notion of $\mathcal{R}$-lossy encryption introduced by Boyle *et al.* [21]. The primitive is a tag-based encryption scheme [63] where the tag space $\mathcal{T}$ is partitioned into *injective* tags and *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 [21] require the initialization value K to be computationally hidden by the public key. The definition of [65] requires 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 $(K, t) \notin \mathcal{R}$). In addition, [65] also asks that the secret key allows equivocating lossy ciphertexts (a property called *efficient opening* by Bellare *et al.* [10]) using an algorithm called Opener. For the purpose of constructing simulation-sound arguments, [65] uses two distinct opening algorithms Opener and $\mathsf{LOpener}$. The former operates over injective public keys for lossy tags while the latter can equivocate ciphertexts encrypted under lossy keys for any tag.

Definition 2.22. *Let $\mathcal{R} \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ be an efficiently computable binary relation. An $\mathcal{R}$-lossy PKE scheme with efficient opening is a 7-uple of PPT algorithms* $(\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{LOpener})$ *such that:*

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

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

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

Decryption under injective tags: *For any $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$, 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*

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

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

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

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

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

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

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

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

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

Equivocation under lossy tags: *For any* $\Gamma \leftarrow \mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$*, any* $K \in \mathcal{K}$*, any keys* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ *let* D_R *denote the distribution, defined over the randomness space* R^{LPKE}*, from which the random coins used by* Encrypt *are sampled. For any message* $\mathsf{Msg} \in \mathsf{MsgSp}$ *and ciphertext* C*, let* $D_{\mathsf{pk},\mathsf{Msg},C,t}$ *denote the probability distribution on* R^{LPKE} *with support*

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

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

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

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

Equivocation under lossy keys: *For any initialization value* $K \in \mathcal{K}_\lambda$*, any keys* $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$*, any random coins* $r \hookleftarrow D_R$*, any tag* $t \in \mathcal{T}_\lambda$*, and any distinct messages* $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathsf{MsgSp}$*, algorithm* LOpener *takes as input* $C = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_0, r)$*,* r*,* t *and* sk*. It outputs* $\overline{r} \in R^{\mathsf{LPKE}}$ *such that* $C = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}_1, \overline{r})$*. We require that, for any* $t \in \mathcal{T}_\lambda$ *such that* $(K, t) \notin \mathcal{R}$*, the distributions*

$$\{\overline{r} \leftarrow \mathsf{LOpener}(\mathsf{pk}, \mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r) \mid r \hookleftarrow D_R\}$$

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

The above definition is slightly weaker than the one of [65] in the property of equivocation under lossy keys. Here, we do not require that the outputs of Opener and LOpener be statistically close to $D_{\mathsf{pk},\mathsf{Msg}_1,C,t}$ as defined in (1): We only require that, on lossy keys and lossy tags, Opener and LOpener sample random coins from statistically close distributions. In fact, the first indistinguishability property implies (since the distinguisher is given tk) that the outputs of both algorithms will be *computationally* indistinguishable from $D_{\mathsf{pk},\mathsf{Msg}_1,C,t}$. Our definition turns out to be sufficient for the purpose of simulation-sound arguments and will allow us to obtain a construction from the DCR assumption.

We note that the property of decryption under injective tags does not assume that random coins are honestly sampled, but only that they belong to some predefined set R^{LPKE}.

2.8 Trapdoor Σ-Protocol Showing Composite Residuosity

We recall the trapdoor Σ-protocol of [66], which allows proving that an element of $\mathbb{Z}^*_{N^2}$ is a composite residue (i.e., a Paillier encryption of 0).

Namely, let $N = pq$ be an RSA modulus and let an integer $\zeta > 1$. We describe a trapdoor Σ-protocol for the language

$$\mathcal{L}^{\mathsf{DCR}} := \{x \in \mathbb{Z}^*_{N^{\zeta+1}} \mid \exists w \in \mathbb{Z}^\star_N : x = w^{N^\zeta} \bmod N^{\zeta+1}\}.$$

We assume that the challenge space is $\{0, \ldots, 2^\lambda - 1\}$ and that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > \lambda$ for any sufficiently large $\lambda \in \mathbb{N}$. The condition $p, q > 2^\lambda$ will ensure that the difference between any two challenges be co-prime with N.

In order to obtain a $\mathsf{BadChallenge}$ function that identifies bad challenges for elements $x \notin \mathcal{L}^{\mathsf{DCR}}$, one difficulty is the case of elements $x \in \mathbb{Z}^*_{N^{\zeta+1}}$ that are encryptions of an element $\alpha_x \in \mathbb{Z}_N$ such that $1 < \gcd(\alpha_x, N^\zeta) < N^\zeta$. Indeed, we cannot immediately identify a unique bad challenge by inverting α_x in $\mathbb{Z}_{N^\zeta}$. However, a closer analysis shows that, even when $\zeta > 1$ and $\gcd(\alpha_x, N^\zeta) > 1$, at most one bad challenge can exist in the set $\{0, 1, \ldots, 2^\lambda - 1\}$.

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

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

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

$\mathbf{TrapGen}(\mathsf{par}, \mathcal{L}^{\mathsf{DCR}}, \tau_{\mathcal{L}})$: Given par, the description of a language $\mathcal{L}^{\mathsf{DCR}}$ that specifies an RSA modulus N and a membership-testing trapdoor $\tau_{\mathcal{L}} = (p, q)$ consisting of the factorization of $N = pq$, output the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{N\}$ which defines $\mathsf{crs} = (\{\lambda\}, \mathsf{crs}_{\mathcal{L}})$ and the trapdoor $\tau_\Sigma = (p, q)$.

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

1. P chooses a random $r \hookleftarrow U(\mathbb{Z}^*_N)$ and sends $a = r^{N^\zeta} \bmod N^{\zeta+1}$ to V.
2. V sends a random challenge $\mathsf{Chall} \hookleftarrow U(\{0, \ldots, 2^\lambda - 1\})$ to P.
3. P computes the response $z = r \cdot w^{\mathsf{Chall}} \bmod N$ and sends it to V.
4. V checks if $a \cdot x^{\mathsf{Chall}} \equiv z^{N^\zeta} \pmod{N^{\zeta+1}}$ and returns 0 if this condition is not satisfied.

$\mathbf{BadChallenge}\big(\mathsf{par}, \tau_\Sigma, \mathsf{crs}, x, a\big)$: Given $\tau_\Sigma = (p, q)$, decrypt x and a to obtain $\alpha_x = \mathcal{D}_{\tau_\Sigma}(x) \in \mathbb{Z}_{N^\zeta}$, $\alpha_a = \mathcal{D}_{\tau_\Sigma}(a) \in \mathbb{Z}_{N^\zeta}$.

1. If $\alpha_a = 0$, return $\mathsf{Chall} = 0$.
2. If $\alpha_a \neq 0$, let $d_x = \gcd(\alpha_x, N^\zeta)$, which lives in the set

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

   Then, do the following:

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

In [66], it is shown that the above construction is a trapdoor Σ-protocol with large challenge space. By applying [79], this implies compact NIZK arguments (i.e., without using parallel repetitions to achieve negligible soundness error) for the language $\mathcal{L}^{\mathsf{DCR}}$ assuming that the LWE assumption holds.

Lemma 2.23 ([66]). *The above protocol is a trapdoor Σ-protocol for the language $\mathcal{L}^{\mathsf{DCR}}$.*

3 NIZK Arguments With One-Time Simulation-Soundness

Libert *et al.* [65] gave a method that directly compiles (i.e., without relying on generic NIZK techniques [40]) any trapdoor Σ-protocol for a trapdoor language into an unbounded simulation-sound NIZK argument for the *same* language. As a building block, their construction uses an LWE-based equivocable $\mathcal{R}$-lossy PKE scheme for the bit-matching relation. Under the DCR assumption, a more efficient $\mathcal{R}$-lossy PKE scheme was described in [66]. In this section, we show that, in applications that only require *one-time* simulation-soundness, we can use an $\mathcal{R}$-lossy PKE scheme with a constant-size public key. In contrast, the $\mathcal{R}$-lossy PKE system of [66] has a large public key comprised of $\Theta(\lambda)$ Paillier ciphertexts.

In our *one-time* simulation-sound arguments, we use an $\mathcal{R}$-lossy PKE scheme for the inequality relation.

Definition 3.1. *Let $\mathcal{K} = \{0,1\}^\ell$ and $\mathcal{T} = \{0,1\}^\ell$, for some $\ell \in \mathsf{poly}(\lambda)$. The* **inequality relation** *$\mathcal{R}_{\mathsf{NEQ}} : \mathcal{K} \times \mathcal{T} \to \{0,1\}$ is the relation where $\mathcal{R}_{\mathsf{NEQ}}(K,t) = 1$ if and only if $K \neq t$.*

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

Our DCR-based $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme goes as follows.

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

Keygen(Γ, K)**:** On input of public parameters Γ and $K \in \mathcal{K}$, generate a key pair as follows.

1. Choose an RSA modulus $N = pq$ such that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > L(\lambda)$ for any sufficiently large λ, and an integer $\zeta \in \mathsf{poly}(\lambda)$ such that $N^\zeta > 2^B$.
2. Pick $u \hookleftarrow U(\mathbb{Z}^*_{N^{\zeta+1}})$, $\bar{v} \hookleftarrow U(\mathbb{Z}^*_N)$ and compute $v = u^{-K} \cdot \bar{v}^{N^\zeta} \bmod N^{\zeta+1}$, where K is interpreted as an element of $\mathbb{Z}_{N^\zeta}$.

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

$$\mathsf{pk} := \Big(N, \zeta, u, v\Big), \qquad \mathsf{tk} = (\bar{v}, K).$$

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

1. Choose an RSA modulus $N = pq$ such that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > L(\lambda)$ for any sufficiently large λ, and an integer $\zeta \in \mathsf{poly}(\lambda)$ such that $N^{\zeta} > 2^{B}$.
2. Choose $\bar{u}, \bar{v} \hookleftarrow U(\mathbb{Z}_N^*)$ uniformly. Compute $u = \bar{u}^{N^{\zeta}} \bmod N^{\zeta+1}$ and $v = u^{-K} \cdot \bar{v}^{N^{\zeta}} \bmod N^{\zeta+1}$, where K is interpreted as an element of $\mathbb{Z}_{N^{\zeta}}$.

Define $R^{\mathsf{LPKE}} = \mathbb{Z}_N^*$ and output $\mathsf{sk} = (\bar{u}, \bar{v}, K)$ as well as $\mathsf{pk} := \big(N, \zeta, u, v\big)$ and $\mathsf{tk} = (\bar{v}, K)$.

Encrypt$(\mathsf{pk}, t, \mathsf{Msg})$**:** To encrypt $\mathsf{Msg} \in \mathbb{Z}_{N^{\zeta}}$ for the tag $t \in \{0,1\}^L$, interpret t as an element of $\mathbb{Z}_{N^{\zeta}}$. Pick $r \hookleftarrow U(\mathbb{Z}_N^*)$ and compute

$$\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}} \cdot r^{N^{\zeta}} \bmod N^{\zeta+1}.$$

Decrypt$(\mathsf{sk}, t, \mathsf{ct})$**:** Given $sk = (p, q, t^{\star})$ and the tag $t \in \{0,1\}^L$, interpret t as an element of $\mathbb{Z}_{N^{\zeta}}$. Then, do the following:

1. Letting $\lambda(N) = \mathrm{lcm}(p-1, q-1)$, compute $h_t = (u^t \cdot v)^{\lambda(N)} \bmod N^{\zeta+1}$, which can be written $h_t = 1 + g_t N \bmod N^{\zeta+1}$, for some $g_t \in \mathbb{Z}_{N^{\zeta}}$, since its order is at most N^{ζ}. Return $\perp$ if $g_t = 0$ or $\gcd(g_t, N^{\zeta}) > 1$.
2. Otherwise, compute $\mathsf{Msg} = \frac{(\mathsf{ct}^{\lambda(N)} \bmod N^{\zeta+1})-1}{N} \cdot g_t^{-1} \bmod N^{\zeta}$, where the division is computed over $\mathbb{Z}$, and output $\mathsf{Msg} \in \mathbb{Z}_{N^{\zeta}}$.

Opener$(\mathsf{pk}, \mathsf{tk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r)$**:** Given $\mathsf{tk} = (\bar{v}, K)$ and $t \in \{0,1\}^L$, return $\perp$ if $t \neq K$ when they are interpreted as elements of $\mathbb{Z}_{N^{\zeta}}$. Otherwise, given $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathbb{Z}_{N^{\zeta}}$ and $r \in \mathbb{Z}_N^*$ such that

$$\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}_0} \cdot r^{N^{\zeta}} = (\bar{v}^N)^{\mathsf{Msg}_0} \cdot r^{N^{\zeta}} \bmod N^{\zeta+1}, \tag{2}$$

output $\bar{r} = r \cdot \bar{v}^{\mathsf{Msg}_0 - \mathsf{Msg}_1} \bmod N$, so that $\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}_1} \cdot \bar{r}^{N^{\zeta}} \bmod N^{\zeta+1}$.

LOpener$(\mathsf{sk}, t, \mathsf{ct}, \mathsf{Msg}_0, \mathsf{Msg}_1, r)$**:** Given $\mathsf{sk} = (\bar{u}, \bar{v}, K)$ and $t \in \{0,1\}^L$, interpret t as an element of $\mathbb{Z}_{N^{\zeta}}$. Given $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathbb{Z}_{N^{\zeta}}$ and $r \in \mathbb{Z}_N^*$ such that

$$\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}_0} \cdot r^{N^{\zeta}} = (\bar{u}^{t-K} \cdot \bar{v})^{N \cdot \mathsf{Msg}_0} \cdot r^{N^{\zeta}} \bmod N^{\zeta+1}, \tag{3}$$

output $\bar{r} = r \cdot (\bar{u}^{t-K} \cdot \bar{v})^{\mathsf{Msg}_0 - \mathsf{Msg}_1} \bmod N$, which satisfies

$$\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}_1} \cdot \bar{r}^{N^{\zeta}} \bmod N^{\zeta+1}.$$

The scheme enables decryption under injective tags because, with high probability over the randomness of Keygen, the order of u is a multiple of N^{ζ} since u is sampled uniformly in $\mathbb{Z}_{N^{\zeta+1}}^*$ at step 2. Since $t, K \in \{0,1\}^L$ and $p, q > 2^L$, we have $\gcd(t-K, N^{\zeta}) = 1$, so that N^{ζ} divides the order of $u^{t-K} \cdot \bar{v}^{N^{\zeta}}$ when $t \neq K$. This ensures that h_t has order N^{ζ} and $\gcd(g_t, N^{\zeta}) = 1$ at step 1 of Decrypt.

We now prove that the scheme satisfies all the properties of Definition 2.22. The first indistinguishability property crucially imposes that lossy and injective keys be indistinguishable even when the equivocation trapdoor tk of Opener is given. This is important for our proof of one-time simulation-soundness, which requires that Opener be able to equivocate lossy ciphertexts given only tk and without knowing the factorization of N (otherwise, we could not meaningfully rely on the DCR assumption to switch from lossy to injective keys).

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

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

Decryption under injective tags. Let distinct $t, K \in \{0,1\}^L$ and a message $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$. Let $(\mathsf{pk} = (N, \zeta, u, v), \mathsf{sk} = (p, q, K), \mathsf{tk} = (\bar{v}, K)) \leftarrow \mathsf{KeyGen}(\Gamma, K)$. The encryption algorithm outputs $\mathsf{ct} = (u^t \cdot v)^{\mathsf{Msg}} \cdot r^{N^\zeta} \bmod N^{\zeta+1}$, for some $r \in \mathbb{Z}_N^*$. The decryption algorithm first computes

$$h_t = (u^{t-K} \cdot \bar{v}^{N^\zeta})^{\lambda(N)} \bmod N^{\zeta+1} = u^{(t-K)\lambda(N)} \bmod N^{\zeta+1}. \tag{4}$$

Since u was chosen uniformly in $\mathbb{Z}^*_{N^{\zeta+1}}$, its order is a multiple of N^ζ with overwhelming probability $\varphi(N)/N$. In this case, (4) implies that h_t has order N^ζ since $\gcd\left((t-K)\lambda(N), N^\zeta\right) = 1$ (recall that $0 < |t-K| < 2^L < \min(p,q)$, so that $\gcd(t-K, N^\zeta) = 1$). It can thus be parsed as $h_t = 1 + g_t N$ for some $g_t \in \mathbb{Z}^*_{N^\zeta}$. This implies

$$\begin{aligned}\mathsf{ct}^{\lambda(N)} \bmod N^{\zeta+1} &\equiv h_t^{\mathsf{Msg}} \cdot r^{N^\zeta \cdot \lambda(N)} \\ &\equiv (1 + g_t N)^{\mathsf{Msg}} \equiv 1 + (\mathsf{Msg} \cdot g_t \bmod N^\zeta) \cdot N \pmod{N^{\zeta+1}}.\end{aligned}$$

The output of $\mathsf{Decrypt}$ is thus correct as long as $\gcd(g_t, N^\zeta) = 1$, which is the case with overwhelming probability over the random choice of u in Keygen.

Indistinguishability. We have two properties to verify.

(i) When we consider the distributions of pairs $\big(\mathsf{pk} = (N, \zeta, u, v), \mathsf{tk} = (\bar{v}, K)\big)$ produced by Keygen and $\mathsf{LKeygen}$, the only difference is the way to sample u. In the former case, it is sampled from $U(\mathbb{Z}^*_{N^{\zeta+1}})$ while, in the latter case, it is sampled as $u = \bar{u}^{N^\zeta} \bmod N^{\zeta+1}$, with $\bar{u} \leftarrow\!\!\shortmid U(\mathbb{Z}_N^*)$. The DCR assumption states that these two distributions are computationally indistinguishable and the same holds for the two distributions of pairs $(\mathsf{pk}, \mathsf{tk})$.

(ii) We claim that the distributions $\{\mathsf{pk} \mid (\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K_b)\}_{b=0,1}$ are perfectly indistinguishable for any $K_0, K_1 \in \{0,1\}^L$ since u is an N^ζ-th residue. Indeed,

$$\left\{(N, u = \bar{u}^{N^\zeta} \bmod N^{\zeta+1}, (\bar{u}^{-K_0} \cdot \bar{v})^{N^\zeta} \bmod N^{\zeta+1}) \mid \bar{u}, \bar{v} \leftarrow\!\!\shortmid U(\mathbb{Z}_N^*)\right\}$$

and $\{(N, u = \bar{u}^{N^\zeta} \bmod N^{\zeta+1}, (\bar{u}^{-K_1} \cdot \bar{v})^{N^\zeta} \bmod N^{\zeta+1}) \mid \bar{u}, \bar{v} \leftarrow\!\!\shortmid U(\mathbb{Z}_N^*)\}$ are identical distributions as their third component is uniform in the subgroup of N^ζ-th residues either way.

LOSSINESS UNDER LOSSY TAGS. Let Msg_0 and Msg_1 be two messages and an arbitrary $K \in \mathcal{K}$. Let an injective key

$$\big(\mathsf{pk} = (N, \zeta, u, v), \mathsf{sk} = (p, q, K), \mathsf{tk} = (\bar{v}, K)\big) \leftarrow \mathsf{Keygen}(\Gamma, K),$$

where $v = u^{-K} \cdot \bar{v}^{N^\zeta} \bmod N^{\zeta+1}$. Under the tag $t = K$, encrypting Msg_0 or Msg_1 leads to identical ciphertext distributions since $(u^K \cdot v)^{\mathsf{Msg}_b} = \bar{v}^{\mathsf{Msg}_b \cdot N^\zeta} \bmod N^{\zeta+1}$ for each $b \in \{0,1\}$. Indeed, the distributions $\{\bar{v}^{\mathsf{Msg}_0 \cdot N^\zeta} \cdot r^{N^\zeta} \mid r \hookleftarrow U(\mathbb{Z}_N^*)\}$ and $\{\bar{v}^{\mathsf{Msg}_1 \cdot N^\zeta} \cdot r^{N^\zeta} \mid r \hookleftarrow U(\mathbb{Z}_N^*)\}$ both coincide with the uniform distribution over the subgroup of N^ζ-th residues.

EQUIVOCATION UNDER LOSSY TAGS. We know that, for any $K \in \mathcal{K}$, any injective keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{Keygen}(\Gamma, K)$ and any message $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$, the distribution $\{\mathsf{ct} = \mathsf{Encrypt}(\mathsf{pk}, K, \mathsf{Msg}; r) \mid r \hookleftarrow D_R^{\mathsf{LPKE}}\}$ is equal to $\{(\bar{v}^{\mathsf{Msg}} \cdot r)^{N^\zeta} \bmod N^{\zeta+1} \mid r \hookleftarrow U(\mathbb{Z}_N^\star)\}$, which is nothing but the uniform distribution over the subgroup of N^ζ-th residues in $\mathbb{Z}_{N^{\zeta+1}}^*$ since $D_R^{\mathsf{LPKE}} = U(\mathbb{Z}_N^*)$.

Since Opener is deterministic, it suffices to observe that any N^ζ-th residue ct uniquely determines $r_{\mathsf{ct}} = \mathsf{ct}^{1/N^\zeta} \bmod N \in \mathbb{Z}_N^*$ such that $\mathsf{ct} = r_{\mathsf{ct}}^{N^\zeta} \bmod N^{\zeta+1}$ as its N^ζ-th roots are of the form $\{r_i = r_{\mathsf{ct}} + i \cdot N \mid i \in \mathbb{Z}_{N^\zeta}\}$. For any lossy ciphertext $\mathsf{ct} \in \mathbb{Z}_{N^{\zeta+1}}^*$, Opener thus outputs $r = r_{\mathsf{ct}} \cdot \bar{v}^{-\mathsf{Msg}} \bmod N$, which is the unique $r \in \mathbb{Z}_N^*$ such that $\mathsf{ct} = (u^K \cdot v)^{\mathsf{Msg}} \cdot r^{N^\zeta} \bmod N^{\zeta+1}$. Said otherwise, for any $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$, the support $S_{\mathsf{pk},\mathsf{Msg},\mathsf{ct},K}$ is a singleton and Opener always outputs the correct value.

EQUIVOCATION UNDER LOSSY KEYS. For any possibly distinct t, K, any lossy keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{tk}) \leftarrow \mathsf{LKeygen}(\Gamma, K)$ and any message $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$, the distribution of lossy ciphertexts $\{\mathsf{ct} = \mathsf{Encrypt}(\mathsf{pk}, t, \mathsf{Msg}; r) \mid r \hookleftarrow D_R^{\mathsf{LPKE}}\}$ can be written $\{\big((\bar{u}^{t-K} \cdot \bar{v})^{\mathsf{Msg}} \cdot r\big)^{N^\zeta} \bmod N^{\zeta+1} \mid r \hookleftarrow U(\mathbb{Z}_N^*)\}$, which is the uniform distribution over the subgroup of N^ζ-th residues in $\mathbb{Z}_{N^{\zeta+1}}^*$.

Since $\mathsf{LOpener}$ is deterministic, it suffices to note that, for any lossy ciphertext $\mathsf{ct} \in \mathbb{Z}_{N^{\zeta+1}}^*$, it outputs $r = r_{\mathsf{ct}} \cdot (\bar{u}^{t-K} \cdot \bar{v})^{-\mathsf{Msg}} \bmod N$, where $r_{\mathsf{ct}} = \mathsf{ct}^{1/N^\zeta} \bmod N$ is the unique N^ζ-th root of ct in $\mathbb{Z}_N^*$. For any tag $t \in \{0,1\}^L$ and any message Msg, the support $S_{\mathsf{pk},\mathsf{Msg},\mathsf{ct},t}$ is a singleton and $\mathsf{LOpener}$ always outputs the only $r \in \mathbb{Z}_N^*$ such that $\mathsf{ct} = (u^K \cdot v)^{\mathsf{Msg}} \cdot r^{N^\zeta} \bmod N^{\zeta+1}$. Finally, for the lossy tag $t = K$, Opener outputs the same result as $\mathsf{LOpener}$. □

3.2 The Argument System

Our one-time simulation-sound argument is very similar to the one of [65] which provides unbounded simulation-soundness using a more expensive $\mathcal{R}$-lossy PKE scheme. The construction relies on the following ingredients.

- A trapdoor Σ-protocol $\Pi' = (\mathsf{Gen}'_{\mathsf{par}}, \mathsf{Gen}'_{\mathcal{L}}, \mathsf{P}', \mathsf{V}')$ for an NP language $\mathcal{L}$. This protocol should satisfy the properties of Definition 2.21. In addition, the function $\mathsf{BadChallenge}(\tau_\Sigma, \mathsf{crs}, x, a)$ should be computable within time $T \in \mathsf{poly}(\lambda)$ for any input $(\tau, \mathsf{crs}, x, a)$. Let also $B \in \mathsf{poly}(\lambda)$ the maximal length of the first prover message sent by P'.

- A strongly unforgeable one-time signature scheme $\mathsf{OTS} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys in $\{0,1\}^L$, where $L \in \mathsf{poly}(\lambda)$.
- An $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme $\Pi^{\mathsf{LPKE}} = (\mathsf{Par\text{-}Gen}, \mathsf{Keygen}, \mathsf{LKeygen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Opener}, \mathsf{LOpener})$ with space $\mathcal{K} = \mathcal{T} = \{0,1\}^L$. We assume that its decryption algorithm is computable within time T.
- A correlation intractable hash family $\mathcal{H} = (\mathsf{Gen}, \mathsf{Hash})$ for the class $\mathcal{R}_{\mathsf{CI}}$ of relations that are efficiently searchable within time T.

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

$\mathbf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$**:** Given public parameters par and a language $\mathcal{L}$, the CRS is generated as follows.

1. Generate a CRS $\mathsf{crs}'_{\mathcal{L}} \leftarrow \mathsf{Gen}'_{\mathcal{L}}(\mathsf{par}, \mathcal{L})$ for the trapdoor Σ-protocol Π'.
2. Choose the description a one-time signature scheme $\mathsf{OTS} = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ with verification keys in $\{0,1\}^L$, where $L \in \mathsf{poly}(\lambda)$.
3. Choose public parameters $\Gamma \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Par\text{-}Gen}(1^\lambda, 1^L, 1^B)$ for an $\mathcal{R}_{\mathsf{NEQ}}$-lossy PKE scheme with tag space $\mathcal{K} = \mathcal{T} = \{0,1\}^L$. Then, generate lossy keys $(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}}) \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LKeygen}(\Gamma, 0^L)$.
4. Generate a key $k \leftarrow \mathsf{Gen}(1^\lambda)$ for a correlation intractable hash function with output length $\kappa = \Theta(\lambda)$.

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

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

1. Compute $(a', st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w)$. Then, sample $r \hookleftarrow D_R^{\mathsf{LPKE}}$ in the randomness space R^{LPKE} of Π^{LPKE}. Using the tag $\mathsf{VK} \in \{0,1\}^L$, compute $a \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, a'; r)$.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, a, \mathsf{VK}))$.
3. Compute $z' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, a', \mathsf{Chall}, st')$ by executing the prover of Π'. Define $z = (z', a', r)$.
4. Generate $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, a, z, \mathsf{lbl}))$ and output $\boldsymbol{\pi} = \big(\mathsf{VK}, (a, z), sig\big)$.

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

1. Write $z = (z', a', r)$ and return 0 if any of these does not parse properly or if $a \neq \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, a'; r)$.
2. Let $\mathsf{Chall} = \mathsf{Hash}\big(k, (x, a, \mathsf{VK})\big)$. If $\mathsf{V}'(\mathsf{crs}'_{\mathcal{L}}, x, a', \mathsf{Chall}, z')) = 1$, return 1. Otherwise, return 0.

Theorem 3.3. *The above argument is statistically (resp. computationally) zero-knowledge if: (i) Π^{LPKE} is statistically equivocable under lossy keys; (ii) The trapdoor Σ-protocol Π' is statistically (resp. computationally) special zero-knowledge.* (The proof is given in Supplementary Material B.1.)

Theorem 3.4. *The above construction provides one-time simulation-soundness if: (i)* OTS *is a strongly unforgeable one-time signature; (ii)* Π^{LPKE} *is an* $\mathcal{R}_{\mathsf{NEQ}}$*-lossy PKE scheme; (iii) The hash function family* $\mathcal{H}$ *is correlation-intractable for all relations that are searchable within time* T*, where* T *denotes the maximal running time of algorithms* $\mathsf{BadChallenge}(\cdot,\cdot,\cdot,\cdot)$ *and* $\Pi^{\mathsf{LPKE}}.\mathsf{Decrypt}(\cdot,\cdot,\cdot)$*.* (The proof is given in Supplementary Material B.2.)

4 An Adaptively Secure CCA2-Secure Threshold Encryption Scheme Based on Paillier and LWE

Our construction combines a one-time simulation-sound argument of composite residuosity with a threshold variant of an Elgamal-Paillier combination due to Camenisch and Shoup [22]. As in [73], we use a generalization of the Camenisch-Shoup system based on ideas from Damgård and Jurik [37].

For simplicity, we first present a non-robust version of the scheme. In Supplementary Material C, we will explain how to obtain robustness against malicious adversaries by having each server prove that its decryption share is consistent with some public commitment to its corresponding secret key share.

$\mathsf{KeyGen}(1^\lambda, \mathbb{A})$: The dealer conducts the following steps:

1. Choose a safe-prime product $N = pq$, of which the prime factors are of the form $p = 2p' + 1$, $q = 2q' + 1$ for some primes primes $p', q' > 2^{l(\lambda)}$, where $l : \mathbb{N} \to \mathbb{N}$ is a polynomial. Choose an integer $\zeta \geq 1$ so as to define the message space as $\mathsf{MsgSp} = \mathbb{Z}_{N^\zeta}$. Then, define the language
$$\mathcal{L}^{\mathsf{DCR}} := \{x \in \mathbb{Z}^*_{N^{\zeta+1}} \mid \exists w \in \mathbb{Z}^\star_N : x = w^{N^\zeta} \bmod N^{\zeta+1}\}$$
and choose $g_0 \hookleftarrow U(\mathbb{Z}^*_N)$.
2. Generate a common reference string $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ for the one-time simulation-sound argument system $\Pi^{\mathsf{OTSS}} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ of Section 3.
3. Let $\sigma > \sqrt{\lambda \cdot e} \cdot N^\zeta$ be a Gaussian parameter, where $e = \Omega(\ell^{(1+\sqrt{2})/2})$ is the dimension of the matrix $\mathbf{M}$ in Section 2.3. Sample a secret key $x \hookleftarrow D_{\mathbb{Z},\sigma}$ and compute $h = g_0^{4N^\zeta \cdot x} \bmod N^{\zeta+1}$. Define the public key $\mathsf{pk} := (N, \zeta, g_0, h, \mathsf{crs})$ whereas the centralized secret key $\mathsf{sk} := x \in \mathbb{Z}$.
4. Share sk using a LISS scheme. To this end, sample $\bar{\boldsymbol{\rho}} = (\rho_2, \ldots, \rho_e)^\top \hookleftarrow (D_{\mathbb{Z},\sigma})^{(e-1)}$, define $\boldsymbol{\rho} = [x \mid \bar{\boldsymbol{\rho}}^\top]^\top \in \mathbb{Z}^e$ and compute
$$\boldsymbol{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_d \end{bmatrix} = \mathbf{M} \cdot \boldsymbol{\rho} \in \mathbb{Z}^d,$$
where $\mathbf{M} \in \mathbb{Z}^{d \times e}$ is the share-generating matrix of Section 2.3, which computes the Boolean formula associated with the threshold access structure $\mathbb{A}$. Then, define the private key shares as
$$\mathsf{sk}_i = \big(s_j\big)_{j \in \psi^{-1}(i)} = \big(\mathbf{M}_j \cdot \boldsymbol{\rho}\big)_{j \in \psi^{-1}(i)} \in \mathbb{Z}^{d_i} \qquad \forall i \in [\ell],$$

where $\mathbf{M}_j \in \mathbb{Z}^{1\times e}$ denotes the j-th row of $\mathbf{M}$ while d_i stands for the number of rows assigned by the LISS scheme to server i.

Finally, output the public key $\mathsf{pk} = (N, \zeta, g_0, h, \mathsf{crs})$ and the vector of secret-key shares $(\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$.

Encrypt$(\mathsf{pp}, \mathsf{pk}, \mathsf{Msg})$**:** To encrypt $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$, choose $r \hookleftarrow U(\{0, \ldots, \lfloor N/4 \rfloor\})$ and compute

$$C_0 = {g_0}^{2N^\zeta \cdot r} \bmod N^{\zeta+1} \qquad C_1 = (1+N)^{\mathsf{Msg}} \cdot h^r \bmod N^{\zeta+1}$$

Then, using the witness $w = g_0^{2r} \bmod N$, compute a simulation-sound NIZK argument $\boldsymbol{\pi} \leftarrow \mathsf{P}\big(\mathsf{crs}, C_0, g_0^{2r} \bmod N, \mathsf{lbl}\big)$ that $C_0 \in \mathcal{L}^{\mathsf{DCR}}$ using the label $\mathsf{lbl} = C_1$. Then, return the ciphertext $\mathsf{ct} := (C_0, C_1, \boldsymbol{\pi})$.

PartDec$\big(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}\big)$**:** On input of its share $\mathsf{sk}_i = \{s_j = \mathbf{M}_j \cdot \boldsymbol{\rho}\}_{j\in\psi^{-1}(i)}$ and a ciphertext $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})$, the i-th server does the following.

1. If $\mathsf{V}(\mathsf{crs}, C_0, \boldsymbol{\pi}, \mathsf{lbl}) = 0$, return $\perp$.
2. For each $j \in \psi^{-1}(i) = \{j_1, \ldots, j_{d_i}\}$, compute $\mu_{i,j} = C_0^{2\cdot s_j} \bmod N^{\zeta+1}$ and return

$$\begin{aligned}\boldsymbol{\mu}_i &= (\mu_{i,j_1}, \ldots, \mu_{i,j_{d_i}}) \\ &= \big(C_0^{2\cdot s_{j_1}} \bmod N^{\zeta+1},\ \ldots,\ C_0^{2\cdot s_{j_{d_i}}} \bmod N^{\zeta+1}\big) \ \in (\mathbb{Z}^*_{N^{\zeta+1}})^{d_i}.\end{aligned}$$

Combine$\big(\mathsf{pp}, \mathcal{B} = (\mathcal{S} \in \mathbb{A}, \{\boldsymbol{\mu}_i\}_{i\in\mathcal{S}}), \mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})\big)$**:** First, parse the set $\mathcal{S}$ as $\mathcal{S} = \{j_1, \ldots, j_t\}$ and find a vector $\boldsymbol{\lambda}_{\mathcal{S}} = [\boldsymbol{\lambda}_{j_1}^\top \mid \ldots \mid \boldsymbol{\lambda}_{j_t}^\top]^\top \in \{-1, 0, 1\}^{d_{\mathcal{S}}}$ such that $\boldsymbol{\lambda}_{\mathcal{S}} \cdot \mathbf{M}_{\psi^{-1}(\mathcal{S})} = (1, 0, \ldots, 0)$, where $d_{\mathcal{S}} = \sum_{i\in\mathcal{S}} d_i$ and $\boldsymbol{\lambda}_{j_i} = (\lambda_{j_i,1}, \ldots, \lambda_{j_i,d_{j_i}}) \in \{-1, 0, 1\}^{d_i}$ for all $i \in [t]$. Then, do the following:

1. Compute
$$\hat{\mu} \triangleq \prod_{i\in[t]} \prod_{k\in[d_{j_i}]} \mu_{j_i,k}^{\lambda_{j_i,k}} \bmod N^{\zeta+1}.$$
2. Compute $\hat{C}_1 = C_1/\hat{\mu} \bmod N^{\zeta+1}$ and return $\perp$ if $\hat{C}_1 \neq 1 \bmod N$. Otherwise, run the algorithm of [37, Section 3] so as to recover $\mathsf{Msg} \in \mathbb{Z}_{N^\zeta}$ by extracting the discrete logarithm of $\hat{C}_1$ w.r.t. the base $(1+N)$.

In the dealing phase, the matrix $\mathbf{M} \in \mathbb{Z}^{d\times e}$ has $O(\log \ell)$ non-zero entries for threshold access structures. If we apply the LISS scheme based on the Benaloh-Leichter secret sharing [11] and the result of Hoory *et al.* [58], $\mathbf{M}$ has dimensions $d, e = O(\ell^{1+\sqrt{2}})$, so that its rows have norm $\|\mathbf{M}_j\| = O(\sqrt{e}\log \ell)$, which leads to share units of magnitude $|s_j| = O(\sigma e \cdot \log \ell)$.

The scheme thus provides compactness in the sense of Definition 2.13 since the size of ciphertexts and public keys only depends on λ. By increasing the exponent $\zeta > 1$, the ratio between ciphertext and plaintext sizes can approach

1, which was not possible in [70,67].[11] We now prove security in the sense of Definition 2.16.

Theorem 4.1. *The above scheme provides IND-CCA security in the adaptive corruption setting assuming that: (i) The* DCR *assumption holds; (ii) The argument system* Π^{OTSS} *provides one-time simulation-soundness.*

Proof. We consider a sequence of games where, for each i, we call W_i the event that the adversary wins in Game_i.

Game_0: This is the real IND-CCA game. The challenger faithfully answers all queries. In the challenge phase, the adversary $\mathcal{A}$ chooses two messages $\mathsf{Msg}_0, \mathsf{Msg}_1 \in \mathbb{Z}_{N^\zeta}$. The challenger flips a coin $b \hookleftarrow U(\{0,1\})$ and computes the challenge ciphertext $\mathsf{ct}^* = (C_0^*, C_1^*, \boldsymbol{\pi}^\star)$ by running the real encryption algorithm. When $\mathcal{A}$ halts, it outputs $b' \in \{0,1\}$ and we denote by W_0 the event that $b' = b$. By definition, $\mathbf{Adv}^{\text{ind-cca}}(\mathcal{A}) := |\Pr[W_0] - 1/2|$.

Game_1: This game is identical to Game_0 except that we change the generation of the common reference string and the generation of $\boldsymbol{\pi}^\star$ in the challenge ciphertext. In the key generation phase, the challenger runs $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Sim}_0(1^\lambda, \mathcal{L}^{\mathsf{DCR}})$. In the challenge ciphertext $\mathsf{ct}^\star = (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$, the NIZK argument $\boldsymbol{\pi}^\star$ is simulated as $\boldsymbol{\pi}^\star \leftarrow \mathsf{Sim}_1\big(\mathsf{crs}, \tau_{\mathsf{zk}}, C_0^\star, C_1^\star\big)$ without using the witness. From the perfect zero-knowledge property of Π^{OTSS}, Game_1 is indistinguishable from Game_0 and $\Pr[W_1] = \Pr[W_0]$.

Game_2: This game is identical to Game_1 except that we change the generation of the challenge ciphertext $\mathsf{ct}^\star = (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$. Now, the challenger first samples $z_0 \hookleftarrow U(\mathbb{Z}_N^*)$, which is used to compute $z = z_0^{N^\zeta} \bmod N^{\zeta+1}$ and then

$$C_0^\star = z^2 \bmod N^{\zeta+1}, \qquad C_1^\star = (1+N)^{\mathsf{Msg}_b} \cdot C_0^{\star 2x} \bmod N^{\zeta+1}, \tag{5}$$

before simulating $\boldsymbol{\pi}^\star \leftarrow \mathsf{Sim}_1\big(\mathsf{crs}, \tau_{\mathsf{zk}}, C_0^\star, C_1^\star\big)$ as in Game_1. Since the subgroup of $2N^\zeta$-th residues is a cyclic group of order $p'q'$, the distribution of $(C_0^\star, C_1^\star)$ is statistically close to that of Game_1. Indeed, the distribution of $C_0^\star$ is now perfectly (instead of statistically) uniform in the subgroup of $2N^\zeta$-th residues. Hence, $|\Pr[W_2] - \Pr[W_1]| < 2^{-\Omega(\lambda)}$.

Game_3: This game is like Game_2 except that, in order to construct the challenge ciphertext, we now sample $z \hookleftarrow U(\mathbb{Z}^\star_{N^{\zeta+1}})$ uniformly in $\mathbb{Z}^*_{N^{\zeta+1}}$ instead of sampling it from the subgroup of N^ζ-th residues. Then, $(C_0^\star, C_1^\star)$ are still computed as per (5). Under the DCR assumption, this change goes unnoticed and a straightforward reduction shows that $|\Pr[W_3] - \Pr[W_2]| \leq \mathbf{Adv}^{\mathsf{DCR}}(\lambda)$.

At this point, we are done with the DCR assumption and we can henceforth use the factorization of N in subsequent games.

[11] While the rate can be optimized via hybrid encryption, this would ruin the voting-friendly property of the scheme [14]. Moreover, the KEM/DEM framework does not immediately work in the threshold setting (see, e.g., [4]).

Game_4: In this game, the challenger rejects all pre-challenge partial decryption queries $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})$ such that C_0 is not an N^ζ-th residue (note that this can be efficiently checked using the factorization of N). The soundness of the argument system (which is implied by its simulation-soundness) implies that the probability to reject a ciphertext that would not have been rejected in Game_3 is negligible: we have $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}^{\mathsf{OTSS}}(\lambda)$.

Game_5: We modify the partial decryption oracle and now reject post-challenge queries $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})$ such that $(C_0, C_1, \boldsymbol{\pi}) \neq (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$ and C_0 is not an N^ζ-th residue. By doing so, the challenger does not reject a ciphertext that would not have been rejected in Game_4 until the event F_5 that $\mathcal{A}$ queries the partial decryption of a ciphertext $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi}) \neq (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$ such that $\mathsf{V}(\mathsf{crs}, C_0, \boldsymbol{\pi}, C_1) = 1$ although $C_0^{2p'q'} \bmod N^{\zeta+1} \neq 1$. Clearly, event F_5 would contradict the one-time simulation-soundness of the NIZK argument system Π^{OTSS}. We have $|\Pr[W_5] - \Pr[W_4]| \leq \mathbf{Adv}^{\mathsf{OTSS}}(\lambda)$.

Game_6: We finally modify the challenge ciphertext and now compute $(C_0^\star, C_1^\star)$ by sampling $C_0^\star \hookleftarrow \mathbb{QR}_{N^{\zeta+1}}$ as a random quadratic residue in $\mathbb{Z}^*_{N^{\zeta+1}}$ and computing $C_1^\star = (1+N)^{\mathsf{Msg}^\star} \cdot C_0^{\star 2x} \bmod N^{\zeta+1}$ for a random $\mathsf{Msg}^\star \hookleftarrow U(\mathbb{Z}_{N^\zeta})$. Lemma 4.2 shows that Game_6 and Game_5 are negligibly far apart in terms of statistical distance, so that $|\Pr[W_6] - \Pr[W_5]| \leq 2^{-\lambda}$.

In Game_6, we have $\Pr[W_6] = 1/2$ since $\mathsf{ct}^\star$ is completely independent of the challenger's bit $b \sim U(\{0,1\})$. □

Lemma 4.2. Game_6 *and* Game_5 *are statistically indistinguishable.*

Proof. The proof uses similar arguments to [6, Theorem 5]. In Game_5, the challenge ciphertext has components $(C_0^\star, C_1^\star)$ of the form

$$C_0^\star = (1+N)^{\alpha_z} \cdot g^{\beta_z} \bmod N^{\zeta+1},$$
$$C_1^\star = (1+N)^{\mathsf{Msg}_b + 2\alpha_z \cdot (x \bmod N^\zeta)} \cdot g^{2\beta_z \cdot (x \bmod p'q')} \bmod N^{\zeta+1},$$

with $g = g_0^{2N^\zeta} \bmod N^{\zeta+1}$ and for uniform $\alpha_z \sim U(\mathbb{Z}_{N^\zeta})$, $\beta_z \sim U(\mathbb{Z}_{p'q'})$. Since $\gcd(2\alpha_z, N^\zeta) = 1$ with overwhelming probability $\varphi(N)/N$, we only need to show that, from $\mathcal{A}$'s view, $x \bmod N^\zeta$ is statistically uniform over $\mathbb{Z}_{N^\zeta}$ in order to prove that the distribution of $(C_0^\star, C_1^\star)$ is statistically close to that of Game_6.

In Game_5, we note that the challenger rejects all ciphertexts $\mathsf{ct} = (C_0, C_1, \boldsymbol{\pi})$ such that $C_0 \notin \mathcal{L}^{\mathsf{DCR}}$ and $(C_0, C_1, \boldsymbol{\pi}) \neq (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$. For each partial decryption query $(i, (C_0, C_1, \boldsymbol{\pi}))$ such that $C_0 \in \mathcal{L}^{\mathsf{DCR}}$, the adversary can only learn the information $\{\mathbf{M}_j \cdot \boldsymbol{\rho} \bmod p'q'\}_{j \in \psi^{-1}(i)}$. As for partial decryption queries involving the challenge ciphertext $\mathsf{ct}^\star = (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$, we can handle them as if they were corruption queries since the latter reveal at least as much information as the former. Let $\mathcal{C}^\star$ the set of parties for which the adversary made either a corruption query or a decryption query on the challenge $\mathsf{ct}^\star = (C_0^\star, C_1^\star, \boldsymbol{\pi}^\star)$. Let $\mathbf{M}_{\mathcal{C}^\star}$ to be the sub-matrix of $\mathbf{M}$ obtained by stacking up the rows assigned to those parties.

Since $\mathcal{C}^\star$ is not authorized in $\mathbb{A}$, there exists $\boldsymbol{\kappa} \in \mathbb{Z}^e$ such that $\kappa_1 = 1$ and $\mathbf{M}_{\mathcal{C}^\star} \cdot \boldsymbol{\kappa} = \mathbf{0}^{d_{\mathcal{C}^\star}}$. Let a matrix $\mathbf{L}$ whose rows form a basis of the lattice

$\{\mathbf{m} \in \mathbb{Z}^e, \langle \mathbf{m}, \boldsymbol{\kappa} \rangle = 0\}$, where the rows of $\mathbf{M}_{\mathcal{C}^\star}$ live. Note that $(\mathbf{L}, \mathbf{L} \cdot \boldsymbol{\rho})$ reveals at least as much information as $(\mathbf{M}_{\mathcal{C}^\star}, \mathbf{M}_{\mathcal{C}^\star} \cdot \boldsymbol{\rho})$, so that we may condition on $(\mathbf{L}, \mathbf{L} \cdot \boldsymbol{\rho})$. When we additionally condition on $(\mathbf{M}_{[\ell] \setminus \mathcal{C}^\star} \cdot \boldsymbol{\rho} \bmod p'q')$, we condition on something that reveals fewer information than $\boldsymbol{\rho} \bmod p'q'$.

Let an arbitrary vector $\boldsymbol{\rho}_0 \in \mathbb{Z}^e$ satisfying

$$\mathbf{L} \cdot \boldsymbol{\rho}_0 = \mathbf{L} \cdot \boldsymbol{\rho}, \qquad \boldsymbol{\rho}_0 \equiv \boldsymbol{\rho} \pmod{p'q'}.$$

The conditional distribution of $\boldsymbol{\rho}$ is $\boldsymbol{\rho}_0 + D_{\Lambda, \sigma, -\boldsymbol{\rho}_0}$, where

$$\Lambda = \{\boldsymbol{x} \in \mathbb{Z}^e \;\mid\; \mathbf{L} \cdot \boldsymbol{x} = 0 \quad \wedge \quad \boldsymbol{x} = \mathbf{0} \bmod p'q'\}$$

is the lattice $\Lambda = \boldsymbol{\kappa} \cdot \mathbb{Z} \cap (p'q' \cdot \mathbb{Z}^e) = (p'q' \cdot \mathbb{Z}) \cdot \boldsymbol{\kappa}$. Let us write $\boldsymbol{\rho}_0 = y \cdot \boldsymbol{\kappa} + (\boldsymbol{\rho}_0^\perp)$, where $y \in \mathbb{R}$ and $\boldsymbol{\rho}_0^\perp \in \mathbb{Z}^e$ is orthogonal to $\boldsymbol{\kappa}$. Conditionally on $\mathbf{L} \cdot \boldsymbol{\rho}$ and $\boldsymbol{\rho} \bmod p'q'$, the distribution of $\boldsymbol{\rho}$ can be written

$$\begin{aligned} \boldsymbol{\rho}_0 + D_{\Lambda, \sigma, -\boldsymbol{\rho}_0} &= (\boldsymbol{\rho}_0^\perp) + y \cdot \boldsymbol{\kappa} + D_{(p'q' \cdot \mathbb{Z}) \cdot \boldsymbol{\kappa}, \sigma, -(\boldsymbol{\rho}_0^\perp) - y \cdot \boldsymbol{\kappa}} \\ &= (\boldsymbol{\rho}_0^\perp) + y \cdot \boldsymbol{\kappa} + \boldsymbol{\kappa} \cdot D_{(p'q' \cdot \mathbb{Z}), \sigma / \|\boldsymbol{\kappa}\|, -y}. \end{aligned}$$

Since $\kappa_1 = 1$, the conditional distribution of $x = \langle (1, 0, \ldots, 0), \boldsymbol{\rho} \rangle$ is thus

$$c + D_{(p'q' \cdot \mathbb{Z}), \sigma / \|\boldsymbol{\kappa}\|, -y},$$

where $c = y + \langle (1, 0, \ldots, 0), \boldsymbol{\rho}_0^\perp \rangle$. We now consider the distribution obtained by reducing the distribution $D_{(p'q' \cdot \mathbb{Z}), \sigma / \|\boldsymbol{\kappa}\|, -y}$ over $\Lambda_0 = p'q' \cdot \mathbb{Z}$ modulo its sublattice $\Lambda_0' = (p'q') \cdot (N^\zeta \mathbb{Z})$. Since $p'q' \cdot N^\zeta < N^{\zeta+1}$, by Lemma 2.7, choosing the standard deviation $\sigma > \sqrt{\lambda \cdot e} \cdot N^{\zeta+1}$ suffices (by [54, Lemma 3.1] which implies $\eta_\epsilon(\Lambda_0') < \lambda^{1/2} N^{\zeta+1}$) to ensure that $x \bmod N^\zeta$ is within distance $2^{-\lambda}$ from $U(\Lambda_0 / \Lambda_0')$ conditionally on $\mathcal{A}$'s view. This completes the proof since $\gcd(p'q', N^\zeta) = 1$ implies $\Lambda_0 / \Lambda_0' \simeq \mathbb{Z}_{N^\zeta}$. □

In Supplementary Material C, we show how to turn the scheme into a robust TPKE system. This is achieved by using trapdoor Σ-protocols to prove the validity of decryption shares. To this end, we need to first construct a standard Σ-protocol with binary challenges in order to apply the generic trapdoor Σ-protocol construction of Ciampi *et al.* [29]. The disadvantage of this approach is that parallel repetitions incur a communication overhead $\Theta(\lambda)$. In applications to voting (where "non-malleable" ciphertext components are removed from ciphertexts before homomorphically processing them), this may be acceptable if proofs of correct partial decryptions are computed by trustees with higher computational resources than voters. It remains an interesting open problem to achieve robustness while proving the correctness of partial decryptions without using parallel repetitions.

5 An Adaptively Secure CCA2-Secure Threshold Encryption Scheme from LWE

We now describe our construction based on the LWE assumption. Our construction combines the standard dual Regev cryptosystem with a LISS scheme and

a one-time simulation-sound NIZK argument $\Pi^{\mathsf{OTSS}} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$ for the gap language $\mathcal{L} = \{\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}}\}$ defined as

$$\mathcal{L}_{\mathsf{zk}} = \{(\mathbf{c}_0, \mathbf{c}_1) : \exists (\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^{n+L} \times \mathbb{Z}^{m+L} : \|\mathbf{e}\| \leq \tilde{d} \ \wedge \ [\mathbf{c}_0^\top | \mathbf{c}_1^\top]^\top = \mathbf{B}\mathbf{s} + \mathbf{e}\}$$

$$\mathcal{L}_{\mathsf{sound}} = \{(\mathbf{c}_0, \mathbf{c}_1) : \exists (\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^{n+L} \times \mathbb{Z}^{m+L} : \|\mathbf{e}\| \leq \gamma\tilde{d} \ \wedge \ [\mathbf{c}_0^\top | \mathbf{c}_1^\top]^\top = \mathbf{B}\mathbf{s} + \mathbf{e}\},$$

for some matrix $\mathbf{B} \in \mathbb{Z}_q^{(m+L)\times(n+L)}$, where $m, n \in \mathsf{poly}(\lambda)$ while $L \in \mathsf{poly}(\lambda)$ defines the message space $\mathbb{Z}_p^L$ for some prime p. Specific constructions of (even unbounded simulation-sound) NIZK arguments for such languages have been recently proposed by Libert *et al.* in [65, Section 3] under on the LWE assumption. Our construction goes as follows:

$\mathsf{KeyGen}(1^\lambda, \mathbb{A})$: The trusted dealer proceeds as follows.

1. Choose a LISS scheme $\mathcal{L}_{\mathsf{LISS}} = (\mathcal{M} = (\mathbf{M} \in \mathbb{Z}^{d\times e}, \psi, \varepsilon), \mathbb{A}, \mathcal{R}, \mathcal{K})$, as defined in Definition 2.11. Here, $\mathcal{M}$ is an ISP for $\mathbb{A}$, $\mathcal{R}$ is the reconstruction vector space and $\mathcal{K}$ is the sweeping vector space. Define $d_i := |\psi^{-1}(i)|$ for all $i \in [\ell]$. Recall that, in the case $\mathbb{A} = T_{t,\ell}$, we have $e = O(\ell^{1+\sqrt{2}})$. Let also $\delta = O(\log \ell)$ the maximal number of non-zero entries on a row of the ISP matrix $\mathbf{M}$.
2. Select public parameters $\mathsf{pp} = \{m, n, q, p, L, \mathcal{L}_{\mathsf{LISS}}\}$, consisting of a prime p; a modulus $q = p \cdot K$, for some $K \in \mathbb{N}^*$; dimensions $n = \Omega(\lambda^2)$, $m = \Omega(n \log q)$; a standard deviation $\sigma = \Omega(\sqrt{e \cdot m})$; and an integer $L \in \mathsf{poly}(\lambda)$. These are chosen so that all prime factors of q are larger than $2\sqrt{m}\sigma$.[12] The message space is defined as $\mathcal{M} = \mathbb{Z}_p^L$.
3. Set $\gamma = m^{0.5+\Omega(1)}$. Next, pick two Gaussian parameters $\beta, \beta_{\mathsf{s}} \in (0, 1)$ such that

   $$\beta_{\mathsf{s}} > \left(2^\lambda (\max_{i\in[\ell]} d_i)\delta\sigma^2\gamma m\sqrt{m+L}\right) \cdot \beta \quad \text{and} \quad 2p \cdot d_{\mathcal{S}} \cdot \beta_{\mathsf{s}} < 1/2 \quad \forall \mathcal{S} \in \mathbb{A}.$$

   Also, choose a real $B^\star > 0$ such that $B^\star < q$ and such that $B^\star$ and β_{s} satisfy

   $$2\left(\log(d_i(m+1)) \cdot (m+1) \cdot d_i^{3/2} \cdot \sqrt{\gamma^2\tilde{d}^2 + 1}\right) \cdot \beta_{\mathsf{s}} < B^\star/q \qquad \forall i \in [\ell].$$

   Here, $B^\star$ is only relevant in the modified version of the scheme (deferred to Supplementary Material D), which provides consistency.

   Then, defining the norm bound $\tilde{d} = 2\beta q \cdot \sigma\sqrt{m(m+L)}$, the lower bound on β_{s} can be restated as

   $$\beta_{\mathsf{s}} q > 2^{\lambda-1}(\max_{i\in[\ell]} d_i) \cdot \delta\sigma\sqrt{m} \cdot (\gamma \cdot \tilde{d}).$$

[12] In particular, this implies that all prime factors of q are $\Omega(\sqrt{e} \cdot \lambda^2 \cdot \log q)$. In the case of threshold access structures $\mathbb{A} = T_{t,\ell}$, this becomes $\Omega(\ell^{(1+\sqrt{2})/2} \cdot \lambda^2 \cdot \log q)$.

4. Sample $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$, $\mathbf{R} \hookleftarrow D_{\mathbb{Z},\sigma}^{m\times L}$ and compute $\mathbf{U} := \mathbf{AR} \in \mathbb{Z}_q^{n\times L}$. Define the following matrix $\mathbf{B} \in \mathbb{Z}_q^{(m+L)\times(n+L)}$ as:

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}^\top & \mathbf{0}^{m\times L} \\ \mathbf{U}^\top & K \cdot \mathbf{I}_L \end{bmatrix}$$

and define the language $\mathcal{L} = \{\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}}\}$:

$$\mathcal{L}_{\mathsf{zk}} = \left\{(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^L : \exists \mathbf{s} \in \mathbb{Z}_q^{n+L} : \left\|[\mathbf{c}_0^\top | \mathbf{c}_1^\top]^\top - \mathbf{B}\cdot\mathbf{s}\right\| \leq \tilde{d}\right\}$$
$$\mathcal{L}_{\mathsf{sound}} = \left\{(\mathbf{c}_0, \mathbf{c}_1) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^L : \exists \mathbf{s} \in \mathbb{Z}_q^{n+L} : \left\|[\mathbf{c}_0^\top | \mathbf{c}_1^\top]^\top - \mathbf{B}\cdot\mathbf{s}\right\| \leq \gamma\tilde{d}\right\}$$

Then, generate a common reference string $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ for a one-time simulation-sound proof system $\Pi^{\mathsf{OTSS}} = (\mathsf{Setup}, \mathsf{P}, \mathsf{V})$. Define the public key $\mathsf{pk} := (\mathbf{A}, \mathbf{U}, \mathsf{crs})$ and the centralized secret key $\mathsf{sk} := \mathbf{R}$.

5. Share each column of $\mathsf{sk} = \mathbf{R}$ using the LISS scheme $\mathcal{L}_{\mathsf{LISS}}$. To this end, parse $\mathbf{R}$ as $\mathbf{R} = \left[\mathbf{r}_1 \mid \mathbf{r}_2 \mid \cdots \mid \mathbf{r}_L\right] \in \mathbb{Z}^{m\times L}$ and share each column independently. Namely, for each $\tau \in [L]$, sample $\hat{\mathbf{\Phi}}_\tau \hookleftarrow (D_{\mathbb{Z},\sigma})^{(e-1)\times m}$ and set

$$\mathbf{R}_\tau = \mathbf{M} \cdot \underbrace{\begin{bmatrix} \mathbf{r}_\tau^\top \\ \hline \hat{\mathbf{\Phi}}_\tau \end{bmatrix}}_{\triangleq \mathbf{\Phi}_\tau} \in \mathbb{Z}^{d\times m},$$

where $\mathbf{M}$ is the matrix computing the Boolean formula associated with the access structure $\mathbb{A}$. Then, define the private key shares as

$$\mathsf{sk}_i = \left\{\mathbf{R}_{\tau,\psi^{-1}(i)} = \mathbf{M}_{\psi^{-1}(i)} \cdot \mathbf{\Phi}_\tau \in \mathbb{Z}^{d_i\times m}\right\}_{\tau\in[L]} \qquad \forall i \in [\ell].$$

Finally, return $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$.

$\mathsf{Encrypt}(\mathsf{pp}, \mathsf{pk}, \mathsf{Msg})$: To encrypt $\mathsf{Msg} \in \mathbb{Z}_p^L$, first sample $\mathbf{s} \hookleftarrow \mathbb{Z}_q^n, \mathbf{e}_0 \hookleftarrow D_{\mathbb{Z}^m,\beta q}$, $\mathbf{e}_1 \hookleftarrow D_{\mathbb{Z}^L,2\beta\cdot\sqrt{m}\sigma\cdot q}$ compute:

$$\begin{aligned} \mathbf{c}_0 &= \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}_0 \in \mathbb{Z}_q^m \\ \mathbf{c}_1 &= \mathbf{U}^\top \cdot \mathbf{s} + \mathbf{e}_1 + K \cdot \mathsf{Msg} \in \mathbb{Z}_q^L \\ \boldsymbol{\pi} &\leftarrow \mathsf{P}\big(\mathsf{crs}, (\mathbf{c}_0^\top \mid \mathbf{c}_1^\top)^\top, (\bar{\mathbf{s}}, \bar{\mathbf{e}})\big), \end{aligned} \tag{6}$$

using the witnesses $\bar{\mathbf{s}} = (\mathbf{s}^\top \mid \mathsf{Msg}^\top)^\top \in \mathbb{Z}_q^{n+L}$, $\bar{\mathbf{e}} = (\mathbf{e}_0^\top \mid \mathbf{e}_1^\top)^\top \in \mathbb{Z}^{m+L}$ and return the ciphertext $\mathsf{ct} := (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$.

$\mathsf{PartDec}\big(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct}\big)$: Given a ciphertext $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ and its secret key share $\mathsf{sk}_i = \{\mathbf{R}_{\tau,\psi^{-1}(i)}\}_{\tau\in[L]}$, the i-th server does the following.

1. If $\mathsf{V}(\mathsf{crs}, (\mathbf{c}_0, \mathbf{c}_1), \boldsymbol{\pi}) = 0$, return $\perp$.
2. Otherwise, for every $\tau \in [L]$, compute $\bar{\boldsymbol{\mu}}_{i,\tau} = \mathbf{R}_{\tau,\psi^{-1}(i)} \cdot \mathbf{c}_0 \in \mathbb{Z}_q^{d_i}$. Then, for each $\tau \in [L]$, sample $\mathbf{e'}_{i,\tau} \hookleftarrow D_{\mathbb{Z}_q^{d_i},\beta_{\mathsf{s}}\cdot q}$ and return

$$\boldsymbol{\mu}_i = \{\boldsymbol{\mu}_{i,\tau}\}_{\tau\in[L]} \triangleq \{\bar{\boldsymbol{\mu}}_{i,\tau} + \mathbf{e'}_{i,\tau}\}_{\tau\in[L]}.$$

Combine$\big(\mathsf{pp}, \mathcal{B} = (\mathcal{S} \in \mathbb{A}, \{\boldsymbol{\mu}_i = \{\boldsymbol{\mu}_{i,\tau}\}_{\tau\in[L]}\}_{i\in\mathcal{S}}), (\mathbf{c}_0, \mathbf{c}_1)\big)$**:** First, find a vector $\boldsymbol{\lambda}_\mathcal{S} = [\boldsymbol{\lambda}_{j_1}^\top \mid \ldots \mid \boldsymbol{\lambda}_{j_t}^\top]^\top \in \{-1,0,1\}^{d_\mathcal{S}}$ such that $\boldsymbol{\lambda}_\mathcal{S} \cdot \mathbf{M}_{\psi^{-1}(\mathcal{S})} = (1,0,\ldots,0)$, where $d_\mathcal{S} = \sum_{i\in\mathcal{S}} d_i$, $\boldsymbol{\lambda}_{j_i} \in \{-1,0,1\}^{d_{j_i}}$ for all $i \in [t]$ and $\mathcal{S} := \{j_1,\ldots,j_t\}$. Then, do the following:

1. For each $\tau \in [L]$, compute
$$\boldsymbol{\mu}_\tau \triangleq \sum_{i\in\mathcal{S}} \langle \boldsymbol{\lambda}_i, \boldsymbol{\mu}_{i,\tau}\rangle = \langle \mathbf{r}_\tau, \mathbf{c}_0\rangle + \sum_{i\in\mathcal{S}} \langle \boldsymbol{\lambda}_i, \mathbf{e'}_{i,\tau}\rangle$$
and let $[\boldsymbol{\mu}_1^\top \mid \ldots \mid \boldsymbol{\mu}_L^\top]^\top = \mathbf{R}^\top\mathbf{c}_0 + \mathbf{e}''$, where $\mathbf{e}'' \in \mathbb{Z}^L$ is a short vector with $\mathbf{e}''[\tau] = \sum_{i\in\mathcal{S}} \langle \boldsymbol{\lambda}_i, \mathbf{e'}_{i,\tau}\rangle$.
2. Compute
$$\mathbf{v} := \mathbf{c}_1 - \mathbf{R}^\top\mathbf{c}_0 - \mathbf{e}'' = K\cdot\mathsf{Msg} + \mathbf{e}_1 - \mathbf{R}^\top\mathbf{e}_0 - \mathbf{e}'' \ \in \mathbb{Z}_q^L$$
and find and return the vector $\mathsf{Msg} \in \mathbb{Z}_p^L$ that minimizes the distance $|\mathbf{v}[i] - K\cdot\mathsf{Msg}[i]|$ for each $i \in [L]$.

To store a public key, $n(m+L)\log q$ bits are necessary, plus the size of the common reference string of Π^{OTSS}, which depends on m, n, L and q. Each factor of q is at least $2\sqrt{m}\sigma$, where $\sigma = \Omega(\sqrt{em})$ and $e, d = O(\ell^{1+\sqrt{2}})$. We thus have $\log q = \max(\lambda, c\cdot\log m + 0.5\log e)$ for some constant c. However, we have $\log e = O(\log\ell)$ and we can assume $\log\ell \ll \lambda$. We thus have $\log q = O(\lambda)$, so that the public key is compact in the sense of Definition 2.13. The same goes for ciphertexts. We need $(n+m)L\log q$ bits to represent $(\mathbf{c}_0, \mathbf{c}_1)$, besides the size of the proof which depends on n, m, L, q.

Lemma 5.1. *The above TPKE scheme provides correctness.*

Proof. For any ciphertext such that $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{zk}}$, decryption is correct as long as, for any authorized $\mathcal{S} \in \mathbb{A}$, we have
$$\left|\sum_{i\in\mathcal{S}} \langle \boldsymbol{\lambda}_i, \mathbf{e'}_{i,\tau}\rangle + \mathbf{e}_1[\tau] + \langle \mathbf{r}_\tau, \mathbf{e}_0\rangle\right| < K/2, \qquad \forall\tau \in [L]$$
Note that, according to Lemma 2.3, since we have $\sigma = \omega(\sqrt{\log m})$ (recall that $\sigma = \Omega(\sqrt{e\cdot m})$), we have $\|\mathbf{r}_\tau\| \le \sigma\sqrt{m}$ w.h.p. We also know that $|\mathbf{e}_1[\tau]| \le \tilde{d}$ and $\|\mathbf{e}_0\| \le \tilde{d}$ since $\left\|[\mathbf{e}_0^\top \mid \mathbf{e}_1^\top]^\top\right\| \le \tilde{d}$ with overwhelming probability. Using the Cauchy-Schwartz inequality, the scheme is thus correct as long as we have $d_\mathcal{S}\cdot\beta_\mathsf{s} q + \tilde{d}(1+\sigma\sqrt{m}) < K/2$. With $\tilde{d} = 2\beta q\sigma\sqrt{m(m+L)}$, this condition can be replaced by $d_\mathcal{S}\cdot\beta_\mathsf{s} q + 4\beta q\sigma^2 m\sqrt{m+L} < K/2$ or, equivalently,
$$\left(d_\mathcal{S}\cdot\beta_\mathsf{s} + 4\beta\sigma^2 m\cdot\sqrt{m+L}\right)p < 1/2. \tag{7}$$
Recall that we chose $\beta_\mathsf{s} > 2^{\lambda-2}(\max_{i\in[\ell]} d_i)\delta\gamma\cdot(4\beta\sigma^2 m\sqrt{m+L})$, so that the left-hand-side member of (7) is indeed smaller than $2pd_\mathcal{S}\cdot\beta_\mathsf{s} < 1/2$. □

In our proof of CCA2-security under adaptive corruptions, we will use the following lemmas.

Lemma 5.2 ([5]). *Let $\mathcal{H} = \{h : X \to Y\}_{h\in\mathcal{H}}$ be a family of universal hash functions, for countable sets X, Y. For any random variable T taking values in X, we have $\Delta\big((h, h(T)), (h, U(Y))\big) \leq \frac{1}{2} \cdot \sqrt{2^{-H_\infty(T)} \cdot |Y|}$. More generally, let $(T_i)_{i\leq k}$ be independent random variables with values in X, for some $k > 0$. We have $\Delta\big((h, (h(T_i))_{i\leq k}), (h, (U(Y))^{(i)})_{i\leq k}))\big) \leq \frac{k}{2} \cdot \sqrt{2^{-H_\infty(T)} \cdot |Y|}$.*

Lemma 5.3 ([61, Lemma 1]). *Let q, m, ℓ be positive integers and r a positive real satisfying $r > \max(\omega(\sqrt{\log m}), \omega(\sqrt{\log \ell}))$. Let $\mathbf{b} \in \mathbb{Z}_q^m$ and $\mathbf{x} \hookleftarrow D_{\mathbb{Z}^m,r}$. Then, for any $\mathbf{V} \in \mathbb{Z}^{m\times\ell}$ and any positive real $\sigma > \|\mathbf{V}\|$, there exists a* PPT *algorithm* $\mathsf{ReRand}(\mathbf{V}, \mathbf{b} + \mathbf{x}, r, \sigma)$ *that outputs $\mathbf{b}' = \mathbf{V}^\top\mathbf{b} + \mathbf{x}' \in \mathbb{Z}_q^\ell$ where $\mathbf{x}'$ is distributed statistically close to $D_{\mathbb{Z}^\ell,2r\sigma}$.*

Theorem 5.4. *Under the $\mathsf{LWE}_{n,q,\beta,m}$ assumption, the above scheme provides IND-CCA security under adaptive corruptions in the standard model.*

Proof. We are going to prove the theorem by proving that all the following games are computationally indistinguishable for a fixed $b \in \{0, 1\}$.

Game_0: This is the real experiment $\mathsf{Expt}^{\mathsf{CCA}}_{\mathcal{A},\mathsf{TPKE},b}(1^\lambda)$, where the challenger's bit is $b \in \{0, 1\}$. In this game, the challenger faithfully answers all queries (partial decryption and/or corruption). It also computes the challenge ciphertexts $\mathsf{ct}^* = (\mathbf{c}_1^*, \mathbf{c}_0^*)$ by running the real encryption algorithm.

Game_1: This game is identical to Game_0 except that we change the generation of the common reference string and the generation of the proof of the challenge ciphertext. In the key generation phase, the challenger runs $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Sim}_0(1^\lambda, \mathcal{L})$. In the challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \boldsymbol{\pi}^*)$, the NIZK argument $\boldsymbol{\pi}^*$ is simulated as $\pi^* \leftarrow \mathsf{Sim}_1\big(\mathsf{crs}, \tau_{\mathsf{zk}}, (\mathbf{c}_0^*, \mathbf{c}_1^*)\big)$ without using the witnesses. By the zero-knowledge property of the argument system, Game_1 is indistinguishable from Game_0 .

Game_2: We modify the generation of $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \boldsymbol{\pi}^*)$. When the adversary declares the challenge messages $(\mathsf{Msg}_0^*, \mathsf{Msg}_1^*)$, the challenger first defines $\mathbf{c}_0^* = \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}$, where $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$, $\mathbf{e} \hookleftarrow D_{\mathbb{Z}^m,\beta q}$. Then, it uses the ReRand algorithm of Lemma 5.3 to compute

$$\mathbf{c}_1^\star = \mathsf{Rerand}(\mathbf{R}, \mathbf{c}_0^*, \beta \cdot q, \sigma\sqrt{m}) + \mathsf{Msg}_b^\star \cdot K.$$

Lemma 5.3 implies that $\mathcal{A}$'s view remains statistically unchanged since we have $\mathbf{c}_1^* = \mathbf{R}^\top \cdot \mathbf{A}^\top \cdot \mathbf{s} + \mathbf{e}_1 + \mathsf{Msg}_b^\star \cdot K$, where $\mathbf{e}_1 \hookleftarrow D_{\mathbb{Z}^L,2\beta\cdot\sigma\sqrt{m}\cdot q}$. Game_2 is thus statistically indistinguishable from Game_1.

Game_3: We change again the construction of $\mathsf{ct}^* = (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$ in the challenge phase. Instead of sampling $\mathbf{c}_0^\star$ from the LWE distribution, we now sample it uniformly as $\mathbf{c}_0^\star \hookleftarrow U(\mathbb{Z}_q^m)$. Under the $\mathsf{LWE}_{n,q,\beta,m}$ assumption, this game is computationally indistinguishable from Game_2 .

Game_4: We modify again the generation of $\mathsf{ct}^* = (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$. The challenger first samples $\mathbf{c}_0^\star \hookleftarrow U(\mathbb{Z}_q^m)$, $\mathbf{e}^\star \hookleftarrow D_{\mathbb{Z}^m, \beta q}$ and and sets

$$\mathsf{ct}^\star = (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star) = \left(\mathbf{c}_0^\star,\ \mathbf{R}^\top(\mathbf{c}_0^\star - \mathbf{e}^\star) + \mathbf{e}_1 + \mathsf{Msg}_b^\star \cdot K,\ \boldsymbol{\pi}^\star\right).$$

By Lemma 5.3 and our choice of parameters, we obtain that Game_4 is statistically indistinguishable from Game_3 .

Game_5: We now change the generation of $\mathbf{A}$ in the KeyGen. This time, we generate the public matrix via $(\mathbf{A}, \mathbf{T}_\mathbf{A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. The properties of $\mathsf{TrapGen}$ ensure that the distribution of $\mathbf{A}$ is statistically close to $U(\mathbb{Z}_q^{n\times m})$, so that Game_5 and Game_4 are statistically indistinguishable.

Game_6: This is identical to Game_5 except that the challenger rejects all partial decryption queries $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ where $(\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi}) \neq (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$ and $(\mathbf{c}_0, \mathbf{c}_1) \notin \mathcal{L}_{\mathsf{sound}}$. Note that the challenger keeps answering partial decryption queries for the challenge $\mathsf{ct}^\star = (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$ as long as the combination of corruption queries and partial decryption queries on $\mathsf{ct}^\star$ do not allow $\mathcal{A}$ to trivially win. In order to apply the new rejection rule and determine whether $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}$ or not, the challenger can build a trapdoor for $\Lambda^\perp(\mathbf{B})$ by exploiting the special shape of $q = p \cdot K$ and using the trapdoor $\mathbf{T}_\mathbf{A}$ to compute

$$\mathbf{T}_\mathbf{B} = \begin{bmatrix} \mathbf{T}_\mathbf{A}^\top & \mathbf{0}^{m\times L} \\ \mathbf{V}^\top & p \cdot \mathbf{I}_L \end{bmatrix} \in \mathbb{Z}^{(m+L)\times(m+L)}$$

where $\mathbf{V} \in \mathbb{Z}^{m\times L}$ is a small-norm matrix such that $\mathbf{AV} = -p\mathbf{U} \bmod q$. Note that the choice of a large enough K ensures that $p \ll q$ to make sure that the rows of $\mathbf{T}_\mathbf{B}$ have small norm with respect to the modulus q.[13]
Then, if the adversary is able to distinguish between the two games, it is necessarily able to make a valid-looking partial decryption query for a ciphertext outside $\mathcal{L}_{\mathsf{sound}}$. By the OTSS property of the argument system, this can only occur with negligible probability. Lemma 5.5 shows this in details.

Game_7: We modify the decryption oracle and now also reject all ciphertexts $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ such that $\mathbf{c}_0 = \mathbf{c}_0^\star$ and $(\mathbf{c}_1, \boldsymbol{\pi}) \neq (\mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$ regardless of whether $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}$ or not (note that $\mathbf{c}_0^\star$ can be computed at the outset of the game). We call BAD the event that $\mathcal{A}$ makes a valid decryption query for a ciphertext $(\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ where $\mathbf{c}_0 = \mathbf{c}_0^*$. Clearly, Game_7 is identical to Game_6 until BAD occurs. We claim that $\Pr[\mathsf{BAD}] \leq 2^{-\Omega(\lambda)}$, so that Game_6 and Game_7 are statistically indistinguishable. To prove this claim, we distinguish two cases. First, if $\mathbf{c}_0^*$ cannot be written as $\mathbf{A}^\top\mathbf{s} + \mathbf{e}'$ for some $\mathbf{e}' \in \mathbb{Z}^m$ of norm $\|\mathbf{e}'\| \leq \gamma\tilde{d}$, the ciphertext is rejected by the rule introduced in Game_6

[13] In order for the challenger to test if $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}$, we need to have the inequality $\|\mathbf{T}_\mathbf{B} \cdot \mathbf{e}\|_\infty < q$ for any $\mathbf{e} \in \mathbb{Z}^{m+L}$ such that $\|\mathbf{e}\| \leq \gamma\tilde{d}$. The trapdoor $\mathbf{T}_\mathbf{A}$ allows sampling $\mathbf{V} \in \mathbb{Z}^{m\times L}$ in such a way that the Euclidean norm of the lower rows of $\mathbf{T}_\mathbf{B}$ is $O(\sqrt{m^2+p})$. The Cauchy-Schwartz inequality implies $\|\mathbf{T}_\mathbf{B} \cdot \mathbf{e}\|_\infty = O(\sqrt{m^2+p} \cdot (\gamma\tilde{d}))$. Moreover, we have $\sqrt{m^2+p} \leq 2pm$ and KeyGen imposes the constraint $2^\lambda p(\max_{i\in[\ell],\mathcal{S}\in\mathbb{A}} d_i d_\mathcal{S})\delta\sigma\sqrt{m}(\gamma \cdot \tilde{d}) < q/2$ with $\sigma = \Omega(\sqrt{e \cdot m})$.

and the two games are identical. Second, since $\mathbf{c}_0^*$ is uniformly sampled in $\mathbb{Z}_q^m$, the probability that there exist $(\mathbf{s}, \mathbf{e}) \in \mathbb{Z}_q^n \times \mathbb{Z}^m$ such that $\mathbf{c}_0^* = \mathbf{A}^\top \mathbf{s} + \mathbf{e}$ and $\|\mathbf{e}\| \leq \gamma \tilde{d}$ is negligible. Indeed, for a fixed $\mathbf{e}$, the probability that such a pair $(\mathbf{s}, \mathbf{e})$ exists is at most q^{n-m} when $\mathbf{A}$ has full rank (which is the case w.h.p.). Since there are at most $(\gamma \tilde{d})^m$ integer vectors of norm $\leq \gamma \tilde{d}$ and recalling that $\gamma \tilde{d} \leq K/2$, $m \geq n \log q$ and $K/2 = q/2p$, we find that

$$\Pr[\mathsf{BAD}] \leq \frac{q^n}{(2p)^m} \leq \frac{q^n}{(2p)^{n \log q}} \leq \frac{q^n}{q^{n \log 2p}} \leq \frac{1}{q^{n(\log 2p - 1)}},$$

which is negligible since $p \geq 2$ and $n = \mathsf{poly}(\lambda)$. This shows the statistical indistinguishability of Game_7 and Game_6.

Game_8: We modify the partial decryption oracle for queries $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ where $\mathbf{c}_0 \neq \mathbf{c}_0^*$. Recall that these decryption queries must involve a ciphertext of the form $(\mathbf{A}^\top \mathbf{s} + \mathbf{e}, \mathbf{c}_1)$, where $\mathbf{e} \in \mathbb{Z}^m$ has norm $\|\mathbf{e}\| \leq \gamma \tilde{d}$. Note that the challenger can use its trapdoor $\mathbf{T}_\mathbf{A}$ to compute $\mathbf{s}$ and $\mathbf{e}$. Using these, the challenger now ignores $\mathbf{e}$ and handles the query as if it was $(\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{A}^\top \mathbf{s}, \mathbf{c}_1)$. This means that the challenger will return $(\{\mathsf{sk}_{i,\tau} \cdot \mathbf{A}^\top \mathbf{s} + \mathbf{e}_{i,\tau}\}_{\tau \in [L]}, \mathbf{c}_1)$. By noting that $\mathsf{sk}_{i,\tau} \cdot \mathbf{e} = \mathbf{R}_{\tau, \psi^{-1}(i)} \cdot \mathbf{e} \in \mathbb{Z}^{d_i}$ is small compared to $\beta_\mathsf{s} \cdot q$ in Game_7, we can prove that the output distributions of the partial decryption oracle in Game_7 and Game_8 are statistically close. This is formally proved in Lemma 5.6, which shows that Game_7 and Game_8 are statistically close.

In Game_8, we remark that challenger does no longer answer decryption queries $(\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ where $(\mathbf{c}_0, \mathbf{c}_1) \notin \mathcal{L}_\mathsf{sound}$ and $\mathbf{c}_0 \neq \mathbf{c}_0^\star$, nor those where $\mathbf{c}_0 = \mathbf{c}_0^\star$ and $(\mathbf{c}_1, \boldsymbol{\pi}) \neq (\mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$. In the proof of Lemma 5.7, we will show that this prevents the adversary from inferring too much information about $\mathbf{R}$ by making partial decryption queries for maliciously generated ciphertexts. Consequently, $\mathbf{R}$ still retains a lot of entropy after all corruption and partial decryption queries. By applying the Leftover Hash Lemma, Lemma 5.7 then shows that Game_8 is statistically indistinguishable from Game_9.

Game_9: In this game, the challenge ciphertext $(\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$ is now generated by choosing $\mathbf{c}_0^\star \hookleftarrow U(\mathbb{Z}_q^m)$ and $\mathbf{c}_1^\star \hookleftarrow U(\mathbb{Z}_q^L)$ uniformly at random.

In Game_9, the challenge ciphertext is independent of the challenger's bit, so that the adversary's advantage is clearly zero. □

Lemma 5.5. *If the NIZK argument system satisfies the* OTSS *property then* Game_5 *and* Game_6 *are computationally indistinguishable.*

Proof. Assuming that an adversary $\mathcal{A}$ can distinguish between the two games with non-negligible advantage $\mathsf{Adv}_\mathcal{A}^{56}$, we can build an adversary $\mathcal{B}$ that breaks the one-time simulation-soundness of the argument system with the same advantage. We now describe how our adversary $\mathcal{B}$ proceeds in the experiment of Definition A.2.

The OTSS challenger first runs $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$, samples $\mathbf{U} \hookleftarrow U(\mathbb{Z}_q^{n \times L})$ and constructs the matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{A}^\top & \mathbf{0}^{m \times L} \\ \mathbf{U}^\top & K \cdot \mathbf{I}_L \end{bmatrix} \in \mathbb{Z}^{(m+L) \times (n+L)}$$

together with its trapdoor

$$\mathbf{T_B} = \begin{bmatrix} \mathbf{T_A}^\top & \mathbf{0}^{m \times L} \\ \mathbf{V}^\top & p \cdot \mathbf{I}_L \end{bmatrix} \in \mathbb{Z}^{(m+L) \times (n+L)},$$

where $\mathbf{V} \in \mathbb{Z}^{m \times L}$ is a small-norm matrix sampled using the trapdoor $\mathbf{T_A}$ such that $\mathbf{AV} = -p\mathbf{U}$. The matrix $\mathbf{B}$ and the parameters $\tilde{d}, \gamma$ define the language $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$ and a language-recognition trapdoor $\tau_{\mathcal{L}} := \mathbf{T_B}$. The challenger also runs $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Sim}_0(1^\lambda, \mathcal{L})$. The reduction $\mathcal{B}$ is given $(\mathsf{crs}, \tau_{\mathcal{L}}, \mathcal{L})$. The adversary $\mathcal{A}$ receives from the reduction the public key $\mathsf{pk} := (\mathbf{A}, \mathbf{U}, \mathsf{crs})$.

Observe that $\mathcal{B}$ can deduce a trapdoor $\mathbf{T_A}$ for $\mathbf{A}$ from the knowledge of $\tau_{\mathcal{L}} = \mathbf{T_B}$. It can thus use it to sample $\mathbf{R}$ according to the distribution $D_{\mathbb{Z},\sigma}^{m \times L}$ conditionally on $\mathbf{AR} = \mathbf{U}$.

To answer decryption queries $(\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$, $\mathcal{B}$ first checks that $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}$, which it can always do using $\mathbf{T_B}$. If $(\mathbf{c}_0, \mathbf{c}_1)$ is not in the language $\mathcal{L}_{\mathsf{sound}}$, it returns $\perp$. Otherwise, it proceeds as in the real decryption algorithm.

In order to generate the challenge ciphertext, the reduction $\mathcal{B}$ samples vectors $\mathbf{c}_0^\star \hookleftarrow U(\mathbb{Z}_q^m)$, $\mathbf{e}_0^\star \hookleftarrow D_{\mathbb{Z}^m, \beta q}$, $\mathbf{e}_1^\star \hookleftarrow D_{\mathbb{Z}^L, 2\beta \cdot \sigma\sqrt{mq}}$ and computes

$$\mathbf{c}_1^\star = \mathbf{R}^\top \cdot (\mathbf{c}_0^\star - \mathbf{e}_0^\star) + \mathbf{e}_1^\star + K \cdot \mathsf{Msg}_b^\star \in \mathbb{Z}_q^L.$$

Then, it asks the challenger for a simulated proof $\boldsymbol{\pi}^\star \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, (\mathbf{c}_0^\star, \mathbf{c}_1^\star))$. Upon receiving $\boldsymbol{\pi}^\star$, it returns the ciphertext $\mathsf{ct}^\star := (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$ to $\mathcal{A}$. Observe that the reduction $\mathcal{B}$ perfectly simulates the adversary's environment in Game_5.

We define Out to be the event that the adversary $\mathcal{A}$ makes a decryption query $(\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ that passes the verification test (i.e., $\mathsf{Verify}(\mathsf{crs}, (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})) = 1$) although $(\mathbf{c}_0, \mathbf{c}_1) \notin \mathcal{L}_{\mathsf{sound}}$ and $(\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi}) \neq (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$. If Out never happens, the environment simulated by $\mathcal{B}$ is exactly Game_5, hence $\mathsf{Adv}_{\mathcal{A}}^{56} \leq \Pr[\mathsf{Out}]$. When Out happens, $\mathcal{B}$ can use $\mathbf{T_B}$ to recognize the decryption query $(\tilde{\mathbf{c}}_0, \tilde{\mathbf{c}}_1, \tilde{\boldsymbol{\pi}})$ for which it occurs and output $\big((\tilde{\mathbf{c}}_0, \tilde{\mathbf{c}}_1), \tilde{\boldsymbol{\pi}}\big)$, in which case it wins the one-time simulation-soundness experiment. This implies $\mathsf{Adv}_{\mathcal{A}}^{56} \leq \Pr[\mathsf{Out}] \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{otss}}$, as claimed. □

Lemma 5.6. *The distributions of the response to a partial decryption query* $(\mathbf{c}_0, \mathbf{c}_1) = (\mathbf{A}^\top \mathbf{s} + \mathbf{e}, \mathbf{c}_1)$ *in* Game_7 *and* Game_8 *are statistically close.*

Proof. To prove the lemma, we rely on Lemma 2.5. We first need to look at the norm of $\mathsf{sk}_{i,\tau} \cdot \mathbf{e}$. Recall that $\mathsf{sk}_{i,\tau} = \mathbf{M}_{\psi^{-1}(i)} \cdot \boldsymbol{\Phi}_\tau \in \mathbb{Z}^{d_i \times m}$. Since $\mathbf{M} \in \{0,1\}^{d \times e}$ is binary and has at most $\delta = O(\log \ell)$ non-zero entries per row, we have

$$\|\mathsf{sk}_{i,\tau} \cdot \mathbf{e}\|_\infty \leq \delta \cdot \|\boldsymbol{\Phi}_\tau \cdot \mathbf{e}\|_\infty \leq \delta \cdot \big(\sigma\sqrt{m}\big) \cdot (\gamma \tilde{d})$$

by applying Lemma 2.3 over the rows of $\boldsymbol{\Phi}_\tau$ and using the Cauchy-Schwartz inequality. In any ciphertext $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ such that $\mathbf{c}_0 \neq \mathbf{c}_0^*$ and which is not

rejected by the decryption oracle, we know that $\mathbf{c}_0 = \mathbf{A}^\top \mathbf{s} + \mathbf{e}_0$ for some $\mathbf{e}_0 \in \mathbb{Z}^m$ of norm $\|\mathbf{e}_0\| \leq \gamma \tilde{d}$. Since $\tilde{d} = 2\beta q \sigma \cdot \sqrt{m(m+L)}$, Lemma 2.5 implies

$$\begin{aligned} \Delta(\mathsf{sk}_{i,\tau} \cdot \mathbf{e} + \mathbf{e'}_{i,\tau}, \mathbf{e'}_{i,\tau}) &\leq d_i \left(\frac{\delta \cdot \sigma \sqrt{m} \cdot \gamma \tilde{d}}{\beta_{\mathsf{s}} \cdot q} \right) \\ &= d_i \left(\frac{\delta \beta \cdot \sigma^2 \cdot m \sqrt{m+L} \cdot \gamma}{\beta_{\mathsf{s}}} \right) \end{aligned} \tag{8}$$

as the response to a partial decryption query contains d_i times the distribution $D_{\mathbb{Z}_q, \beta_{\mathsf{s}} \cdot q}$. The statistical distance (10) becomes exponentially small as we have set $\beta_{\mathsf{s}} > 2^\lambda (\max_{i \in [\ell]} d_i) \delta \sigma^2 \gamma \beta m \sqrt{m+L}$. □

Lemma 5.7. Game_8 *and* Game_9 *are statistically indistinguishable.*

Proof. The proof is inspired from [68, Lemma 4.4]. We start by computing the min-entropy of $\mathbf{R}$ conditionally on $\mathcal{A}$'s queries and pk. In the case of the partial decryption queries, we observe that, in Game_8, the adversary can compute partial decryption by itself if it knows

$$\{\mathsf{sk}_{i,\tau} \cdot \mathbf{A}^\top\}_{\tau \in [L]} = \left\{\mathbf{R}_{\tau, \phi^{-1}(i)} \cdot \mathbf{A}^\top\right\}_{\tau \in [L]} = \left\{\mathbf{M}_{\psi^{-1}(i)} \cdot \mathbf{\Phi}_\tau \cdot \mathbf{A}^\top\right\}_{\tau \in [L]},$$

which is completely determined by $\mathbf{\Phi}_\tau \cdot \mathbf{A}^\top = \begin{bmatrix} \mathbf{r}_\tau^\top \\ \hat{\mathbf{\Phi}}_j \end{bmatrix} \cdot \mathbf{A}^\top$. To assess the entropy of $\mathbf{R}$, we may condition its distribution on $\mathbf{\Phi}_\tau \cdot \mathbf{A}$ instead of responses to partial decryption queries. Also, we note that $\hat{\mathbf{\Phi}}_j$ is independent of $\mathbf{R}$. This means that only the first row of the matrix is relevant. This implies that decryption queries on ciphertexts $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}$ do not reveal more information about $\mathbf{R}$ than the public key $\mathsf{pk} = (\mathbf{A}, \mathbf{U})$.

In order to prove the lemma, it is sufficient to prove the following fact, where we consider partial decryption queries on the challenge $\mathsf{ct}^\star = (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$ as if they were corruption queries. Indeed, the involved secret key shares reveal at least as much information as the corresponding partial decryptions for ct^*.

Fact 1 *Let $\mathcal{C}^\star$ the set of parties for which the adversary made either a corruption query or a decryption query on the challenge $\mathsf{ct}^\star = (\mathbf{c}_0^\star, \mathbf{c}_1^\star, \boldsymbol{\pi}^\star)$. Let $\mathbf{M}_{\mathcal{C}^\star}$ be the sub-matrix of $\mathbf{M}$ obtained by stacking up the rows assigned to those parties. For each $\tau \in [L]$, conditionally on*

$$\left(\mathbf{A}^\top,\ \mathbf{M}_{[\ell] \setminus \mathcal{C}^\star} \cdot \mathbf{\Phi}_\tau \cdot \mathbf{A}^\top,\ \mathbf{M}_{\mathcal{C}^\star},\ \mathbf{M}_{\mathcal{C}^\star} \cdot \mathbf{\Phi}_\tau\right), \tag{9}$$

the τ-th column $\mathbf{r}_\tau^\top = (1, 0, \ldots, 0)^\top \cdot \mathbf{\Phi}_\tau$ of $\mathbf{R} \in \mathbb{Z}^{m \times L}$ has min-entropy at least

$$m \cdot \log \sigma - \frac{m}{2} \cdot \log e - n \cdot \log q - \frac{m}{2^m}.$$

Since the set of corrupted servers is not an authorized subset of $\mathbb{A}$, there exists $\boldsymbol{\kappa} \in \mathbb{Z}^e$ such that $\kappa_1 = 1$ and $\mathbf{M}_{\mathcal{C}^\star} \cdot \boldsymbol{\kappa} = \mathbf{0}^{d_{\mathcal{C}^\star}}$. Let a matrix $\mathbf{L} \in \mathbb{Z}^{(e-1) \times e}$ whose

rows form a basis of the lattice $\{\mathbf{m} \in \mathbb{Z}^e, \langle \mathbf{m}, \boldsymbol{\kappa} \rangle = 0\}$, where the rows of $\mathbf{M}_{\mathcal{C}^\star}$ live. Note that $(\mathbf{L}, \mathbf{L}\cdot\boldsymbol{\Phi}_\tau)$ reveals at least as much information as $(\mathbf{M}_{\mathcal{C}^\star}, \mathbf{M}_{\mathcal{C}^\star}\cdot\boldsymbol{\Phi}_\tau)$. To obtain a lower bound on the entropy of $\mathbf{r}_\tau$, we condition on $(\mathbf{L}, \mathbf{L}\cdot\boldsymbol{\Phi}_\tau)$. When we additionally condition on $(\mathbf{A}, \mathbf{M}_{[\ell]\setminus\mathcal{C}^\star}\cdot\boldsymbol{\Phi}_\tau\cdot\mathbf{A}^\top)$, we condition on something that reveals fewer information than $(\mathbf{A}, \boldsymbol{\Phi}_\tau\cdot\mathbf{A}^\top)$.

We can now look at the columns of $\boldsymbol{\Phi}_\tau = [\boldsymbol{\rho}_{\tau,1} \mid \ldots \mid \boldsymbol{\rho}_{\tau,m}] \in \mathbb{Z}^{e\times m}$ independently because of the way $\boldsymbol{\Phi}_\tau$ is sampled. For each $i \in [m]$, let a fixed solution $\boldsymbol{\rho}^\star_{\tau,i} \in \mathbb{Z}^e$ of $\mathbf{L}\cdot\mathbf{x} = \mathbf{L}\cdot\boldsymbol{\rho}_{\tau,i}$. We may assume that the adversary knows this specific $\boldsymbol{\rho}^\star_{\tau,i}$ as it can be obtained from $\mathbf{L}\cdot\boldsymbol{\rho}_{\tau,i}$ using de-randomized Gaussian elimination. The conditional distribution of $\boldsymbol{\rho}_{\tau,i}$ is $\boldsymbol{\rho}^\star_{\tau,i} + D_{\Lambda,\sigma,-\boldsymbol{\rho}^\star_{\tau,i}}$, where $\Lambda = \{\mathbf{x} \in \mathbb{Z}^e, \mathbf{L}\cdot\mathbf{x} = 0\}$ is the one-dimensional lattice $\Lambda = \boldsymbol{\kappa}\cdot\mathbb{Z}$.

We thus know that each column of $\boldsymbol{\Phi}_\tau$ is Gaussian over an affine line. Using this observation, we will prove that conditioning on $(\mathbf{A}, \boldsymbol{\Phi}_\tau\cdot\mathbf{A}^\top)$ is not worse than conditioning on $(\mathbf{A}, (1,0,\ldots,0)\cdot\boldsymbol{\Phi}_\tau\cdot\mathbf{A}^\top = \mathbf{r}_\tau^\top\cdot\mathbf{A}^\top)$. More precisely, conditionally on $(\mathbf{L}, \mathbf{L}\cdot\boldsymbol{\Phi}_\tau)$, the last $e-1$ rows of $\boldsymbol{\Phi}_\tau\cdot\mathbf{A}^\top$ do not reveal more information than the first row.

We know that, for each column $\boldsymbol{\rho}_{\tau,i}$ of $\boldsymbol{\Phi}_\tau \in \mathbb{Z}^{e\times m}$, there exists some integer $\xi_{\tau,i}$ such that

$$\boldsymbol{\rho}_{\tau,i} = \xi_{\tau,i}\cdot\boldsymbol{\kappa} + \boldsymbol{\rho}^\star_{\tau,i}.$$

Letting $\boldsymbol{\kappa} = (\kappa_1, \ldots, \kappa_e)^\top$, we can thus write the j-th row of $\boldsymbol{\Phi}_\tau$ as

$$(\boldsymbol{\Phi}_\tau)_j = \kappa_j\cdot(\xi_{\tau,1}, \ldots, \xi_{\tau,m}) + (\rho^\star_{\tau,1,j}, \ldots, \rho^\star_{\tau,m,j}), \qquad \forall j \in [e],$$

where $\boldsymbol{\rho}^\star_{\tau,i} = (\rho^\star_{\tau,i,1}, \ldots, \rho^\star_{\tau,i,e})^\top$ for each $i \in [m]$. Since $\kappa_1 = 1$, this gives us an explicit expression of the $\xi_{\tau,i}$'s:

$$\begin{aligned}(\xi_{\tau,1}, \ldots, \xi_{\tau,m}) &= (\boldsymbol{\Phi}_\tau)_1 - (\rho^\star_{\tau,1,1}, \ldots, \rho^\star_{\tau,m,1})\\ &= \mathbf{r}_\tau^\top - (\rho^\star_{\tau,1,1}, \ldots, \rho^\star_{\tau,m,1}).\end{aligned}$$

This means that we can express the j-th row of $\boldsymbol{\Phi}_\tau$ as

$$(\boldsymbol{\Phi}_\tau)_j = \kappa_j\cdot(\mathbf{r}_\tau^\top - (\rho^\star_{\tau,1,1}, \ldots, \rho^\star_{\tau,m,1})) + (\rho^\star_{\tau,1,j}, \ldots, \rho^\star_{\tau,m,j}).$$

From the above, we find that conditioning on $(\mathbf{A}, \mathbf{r}_\tau^\top\cdot\mathbf{A}^\top, \mathbf{L}, \mathbf{L}\cdot\boldsymbol{\Phi}_\tau)$ does not reveal less information than conditioning on $(\mathbf{A}, \boldsymbol{\Phi}_\tau\cdot\mathbf{A}^\top, \mathbf{L}, \mathbf{L}\cdot\boldsymbol{\Phi}_\tau)$.

We now study $r_{\tau,i} = (1,0,\ldots,0)\cdot\boldsymbol{\rho}_{\tau,i}$. Let us write $\boldsymbol{\rho}^\star_{\tau,i} = c\cdot\boldsymbol{\kappa} + (\boldsymbol{\rho}^\star_{\tau,i})^\perp$, for some $c \in \mathbb{Z}$, with $\langle\boldsymbol{\kappa}, (\boldsymbol{\rho}^\star_{\tau,i})^\perp\rangle = 0$. We can express the distribution of $\boldsymbol{\rho}_{\tau,i}$ as

$$\begin{aligned}\boldsymbol{\rho}^\star_{\tau,i} + D_{\boldsymbol{\kappa}\mathbb{Z},\sigma,-\boldsymbol{\rho}^\star_{\tau,i}} &= \boldsymbol{\rho}^\star_{\tau,i} + D_{\boldsymbol{\kappa}\mathbb{Z},\sigma,-c\cdot\boldsymbol{\kappa}-(\boldsymbol{\rho}^\star_{\tau,i})^\perp}\\ &= \boldsymbol{\rho}^\star_{\tau,i} + \boldsymbol{\kappa}\cdot D_{\mathbb{Z},\sigma/\|\boldsymbol{\kappa}\|,-c}\end{aligned}$$

By taking the inner product with $(1,0,\ldots,0)$, we obtain that the distribution of $r_{\tau,i}$ is $(\boldsymbol{\rho}^\star_{\tau,i})^\perp_1 + c + D_{\mathbb{Z},\sigma/\|\boldsymbol{\kappa}\|,-c}$. Since $\boldsymbol{\kappa} \in \{-1,0,1\}^e$ and due to our choice of σ, we have $\sigma/\|\boldsymbol{\kappa}\| = \Omega(\sqrt{m})$. By Lemma 2.4, we can conclude that, for each $i \in [m]$, the min-entropy of each $r_{\tau,i}$ is at least $\log\left(\frac{\sigma}{\sqrt{e}}\right) - 2^{-m}$. When we additionally

condition on $(\mathbf{A}, \mathbf{r}_\tau \cdot \mathbf{A}^\top)$, we decrease the min-entropy of $\mathbf{r}_\tau$ by at most $n \cdot \log q$. Putting the above altogether, we obtain

$$H_\infty\big(\mathbf{r}_\tau \mid (\mathbf{L}, \mathbf{L} \cdot \mathbf{\Phi}_\tau, \mathbf{A}, \mathbf{r}_\tau^\top \mathbf{A}^\top)\big) \geq m \cdot \left(\log(\sigma) - \frac{1}{2}\log(e)\right) - n \cdot \log q - \frac{m}{2^m}$$

as a lower bound on the entropy of $\mathbf{r}_\tau$ conditionally on responses to all queries.

Let $\mathcal{Q}$ the set of responses to corruption and partial decryption queries. For each $\tau \in [L]$, we thus have

$$H_\infty\big(\mathbf{r}_\tau \mid (\mathcal{Q}, \mathsf{pk})\big) \geq m \cdot \log \sigma - \frac{m}{2} \cdot \log e - \frac{m}{2^m} - n \cdot \log q$$

as an entropy lower bound for the τ-th column of $\mathbf{R} = [\mathbf{r}_1 \mid \ldots \mid \mathbf{r}_L] \in \mathbb{Z}^{m \times L}$. This completes the proof of Fact 1.

To complete the proof of the lemma, we finally have to show that

$$\Delta\big((\mathsf{pk}, \mathbf{a}_0, \mathbf{R}^\top \mathbf{a}_0, \mathcal{Q}), (\mathsf{pk}, \mathbf{a}_0, U(\mathbb{Z}_q^L), \mathcal{Q})\big) = \mathrm{negl}(\lambda).$$

Before applying the Leftover Hash Lemma, we first rewrite this distance as

$$\sum_{(\mathbf{A},\mathbf{V}) \in \mathbb{Z}_q^{n \times m} \times \mathbb{Z}_q^{n \times L}, \mathcal{Q}' \in Q} \Pr\left[\mathsf{pk} = (\mathbf{A}, \mathbf{V}) \cap (\mathcal{Q} = \mathcal{Q}')\right] \cdot \Delta\big((\mathbf{a}_0, \mathbf{R}^\top \mathbf{a}_0), (\mathbf{a}_0, U(\mathbb{Z}_q^L)) \mid (\mathsf{pk} = (\mathbf{A}, \mathbf{V})) \cap \mathcal{Q}'\big),$$

where Q denotes the set of all possible answers to $\mathcal{A}$'s queries.

Since $\mathbf{a}_0$ is uniformly sampled at the beginning of the game, we can use the Leftover Hash Lemma (Lemma 5.2), even with conditioning. We can thus write

$$\Delta\left((\mathbf{a}_0, \mathbf{R}^\top \mathbf{a}_0), (\mathbf{a}_0, U(\mathbb{Z}_q^L)) \mid (\mathsf{pk} = (\mathbf{A}, \mathbf{V})) \cap \mathcal{Q} = \mathcal{Q}'\right) \leq \frac{L}{2}\sqrt{q \cdot 2^{-H_\infty(\mathbf{r}_1 \mid (\mathsf{pk}=(\mathbf{A},\mathbf{V})) \cap \mathcal{Q}=\mathcal{Q}')}} \tag{10}$$

which holds for any set $\mathcal{Q}'$ of possible partial decryption/corruption queries and any possible public key. Note that we can apply Lemma 5.2 since the inner product $\{h_{\mathbf{a}_0} : \mathbb{Z}^m \to \mathbb{Z}_q : \mathbf{r} \mapsto \langle \mathbf{a}_0, \mathbf{r} \rangle \bmod q \mid \mathbf{a}_0 \hookleftarrow U(\mathbb{Z}_q^m)\}$ acts as a universal hash family for sources $\mathbf{r} \in \mathbb{Z}^m$ such that $2\|\mathbf{r}\|_\infty < p'$ for any prime p' dividing q. This is the case here w.h.p. as $2\sqrt{m}\sigma < p'$ for all $p'|q$.

In order to bound the statistical distance (10), we have shown before that

$$q \cdot 2^{-H_\infty(\mathbf{r}_1 \mid (\mathsf{pk}=(\mathbf{A},\mathbf{V})) \cap \mathcal{Q}=\mathcal{Q}')} \leq 2^{\log q - (m \log \sigma - (m/2)\log e - n \log q - m/2^m)}.$$

By our choice of σ and $m = \Omega(n \log q)$, we have $\log \sigma - \frac{1}{2}\log e = \Omega(\log \sqrt{m})$. This implies $m \log(\sigma/\sqrt{e}) \geq n(\log q)(\log \sqrt{m})$, so that

$$q \cdot 2^{-H_\infty(\mathbf{r}_1 \mid (\mathsf{pk}=(\mathbf{A},\mathbf{V})) \cap \mathcal{Q}=\mathcal{Q}')} \leq 2^{m/2^m} \cdot \frac{q^{n+1}}{q^{n \log \sqrt{m}}} < 2 \cdot \frac{q^{n+1}}{q^{n \log \sqrt{m}}} \tag{11}$$

since $m > 4$. Since (11) is bounded by a negligible function of $n(\lambda)$, we obtain

$$\Delta\big((\mathsf{pk}, \mathbf{a}_0, \mathbf{R}^\top \mathbf{a}_0, \mathcal{Q}), (\mathsf{pk}, \mathbf{a}_0, U(\mathbb{Z}_q^L), \mathcal{Q})\big) = \mathrm{negl}(\lambda),$$

which shows that Game_8 and Game_9 are statistically indistinguishable. □

In Supplementary Material D, we show how to modify the scheme to achieve consistency against malicious adversaries.

Acknowledgements

Part of this research was funded by the French ANR ALAMBIC project (ANR-16-CE39-0006). This work was also supported in part by the European Union PROMETHEUS project (Horizon 2020 Research and Innovation Program, grant 780701). Khoa Nguyen was supported in part by the Gopalakrishnan - NTU PPF 2018, by A*STAR, Singapore under research grant SERC A19E3b0099, and by Vietnam National University HoChiMinh City (VNU-HCM) under grant number NCM2019-18-01.

References

1. M. Abe. Robust distributed multiplicaton with out interaction. In *Crypto*, 1999.
2. M. Abe and S. Fehr. Adaptively secure Feldman VSS and applications to universally-composable threshold cryptography. In *Crypto*, 2004.
3. M. Abe and S. Fehr. Perfect NIZK with adaptive soundness. In *TCC*, 2007.
4. M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup. Tag-KEM/DEM: A new framework for hybrid encryption and a new analysis of kurosawa-desmedt KEM. In *Eurocrypt*, 2005.
5. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *Eurocrypt*, 2010.
6. S. Agrawal, B. Libert, and D. Stehle. Fully secure functional encryption for inner products, from standard assumptions. In *Crypto*, 2016.
7. I. Almansa, I. Damgård, and J.-B. Nielsen. Simplified threshold RSA with adaptive and proactive security. In *Eurocrypt*, 2006.
8. J. Alperin-Sheriff and C. Peikert. Circular and KDM security for identity-based encryption. In *PKC*, 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. Bellare, D. Hofheinz, and S. Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *Eurocrypt*, 2009.
11. J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *Crypto*, 1988.
12. R. Bendlin and I. Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In *TCC*, 2010.
13. R. Bendlin, S. Krehbiel, and C. Peikert. How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE. In *ACNS*, 2013.
14. D. Bernhard, V. Cortier, O. Pereira, B. Smyth, and B. Warinschi. Adapting helios for provable ballot privacy. In *ESORICS*, 2011.
15. D. Boneh and X. Boyen. Secure identity based encryption without random oracles. In *Crypto*, 2004.
16. D. Boneh, X. Boyen, and S. Halevi. Chosen ciphertext secure public key threshold encryption without random oracles. In *CT-RSA*, 2006.

17. D. Boneh, R. Gennaro, S. Goldfeder, A. Jain, S. Kim, P. Rasmussen, and A. Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In *Crypto*, 2018.
18. D. Boneh, K. Lewi, H. Montgomery, and A. Raghunathan. Key-homomorphic PRFs and their applications. In *Crypto*, 2013.
19. C. Boyd. Digital multisignatures. In *Cryptography and Coding*, 1989.
20. X. Boyen, Q. Mei, and B. Waters. Direct chosen-ciphertext security from identity-based techniques. In *ACM-CCS*, 2005.
21. E. Boyle, G. Segev, and D. Wichs. Fully leakage-resilient signatures. In *Eurocrypt*, 2011.
22. J. Camenisch and V. Shoup. Practical verifiable encryption and decryption of discrete logarithms. In *Crypto*, 2003.
23. R. Canetti, Y. Chen, J. Holmgren, A. Lombardi, G. Rothblum, R. Rothblum, and D. Wichs. Fiat-Shamir: From practice to theory. In *STOC*, 2019.
24. R. Canetti, R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Adaptive security for threshold cryptosystems. In *Crypto*, 1999.
25. R. Canetti and S. Goldwasser. An efficient threshold public key cryptosystem secure against adaptive chosen-ciphertext attacks. In *Eurocrypt*, 1999.
26. R. Canetti, S. Halevi, and J. Katz. Chosen-ciphertext security from identity-based encryption. In *Eurocrypt*, 2004.
27. 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.
28. D. Chaum and T. Pedersen. Wallet databases with observers. In *Crypto*, 1992.
29. M. Ciampi, R. Parisella, and D. Ventury. On adaptive security of delayed-input Sigma protocols and Fiat-Shamir NIZKs. In *SCN*, 2020.
30. R. Cramer. Modular design of secure, yet practical cryptographic protocols. PhD thesis, University of Amsterdam, 1996.
31. R. Cramer, I. Damgård, and Y. Ishai. Share conversion, pseudorandom secret-sharing and applications to secure computation. In *TCC*, 2005.
32. R. Cramer, I. Damgård, and B. Schoenmaekers. Proofs of partial knowledge and simplified design of witness hiding protocols. In *Crypto*, 1994.
33. R. Cramer and S. Fehr. Optimal black-box secret sharing over arbitrary abelian groups. In *Crypto*, 2002.
34. R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In *Crypto*, 1998.
35. R. Cramer and V. Shoup. Universal hash proofs and a paradigm for chosen ciphertext secure public key encryption. In *Eurocrypt*, 2002.
36. I. Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *Eurocrypt*, 2000.
37. I. Damgård and M. Jurik. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In *PKC*, 2001.
38. I. Damgård and R. Thorbek. Linear integer secret sharing and distributed exponentiation. In *PKC*, 2006.
39. A. De Santis, Y. Desmedt, Y. Frankel, and M. Yung. How to share a function securely. In *STOC*, 1994.
40. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano, and A. Sahai. Robust non-interactive zero-knowledge. In *Crypto*, 2001.
41. Y. Desmedt. Society and group oriented cryptography: A new concept. In *Crypto*, 1987.

42. Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *Crypto*, 1989.
43. Y. Dodis and N. Fazio. Public-key trace and revoke scheme secure against adaptive chosen ciphertext attack. In *PKC*, 2003.
44. Y. Dodis and J. Katz. Chosen-ciphertext security of multiple encryption. In *TCC*, 2005.
45. S. Faust, M. Kohlweiss, G. Marson, and D. Venturi. On the non-malleability of the Fiat-Shamir transform. In *Indocrypt*, 2012.
46. U. Feige, D. Lapidot, and A. Shamir. Multiple non-interactive zero-knowledge under general assumptions. *SIAM J. of Computing*, 29(1), 1999.
47. W. Feller. An introduction to probability theory and its applications. 1968.
48. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Crypto*, 1986.
49. P.-A. Fouque and D. Pointcheval. Threshold Cryptosystems Secure against Chosen-Ciphertext Attacks. In *Asiacrypt*, 2001.
50. Y. Frankel, P. Gemmell, P. MacKenzie, and M. Yung. Optimal-resilience proactive public-key cryptosystems. In *FOCS*, 1997.
51. Y. Frankel, P. MacKenzie, and M. Yung. Adaptively-secure distributed public-key systems. In *ESA*, 1999.
52. J. Garay, P. MacKenzie, and K. Yang. Strengthening zero-knowledge protocols using signatures. In *Eurocrypt*, 2003.
53. C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, 2009.
54. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
55. O. Goldreich. Valiant's polynomial-size monotone formula for majority, 2011.
56. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Eurocrypt*, 2008.
57. B. Hemenway, B. Libert, R. Ostrovsky, and D. Vergnaud. Lossy encryption: Constructions from general assumptions and efficient selective opening chosen ciphertext security. In *Asiacrypt*, 2011.
58. S. Hoory, A. Magen, and T. Pitassi. Monotone circuits for the majority function. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Technique*, pages 410–425. Springer, 2006.
59. S. Jarecki and A. Lysyanskaya. Adaptively secure threshold cryptography: Introducing concurrency, removing erasures. In *Eurocrypt*, 2000.
60. C. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In *Asiacrypt*, 2013.
61. S. Katsumata and S. Yamada. Partitioning via non-linear polynomials functions: More compact ibes from ideal lattices and bilinear maps. In *Asiacrypt*, 2016.
62. J. Katz and V. Vaikuntanathan. Smooth projective hashing and password-based authenticated key-exchange from lattices. In *Asiacrypt*, 2009.
63. E. Kiltz. Chosen-ciphertext security from tag-based encryption. In *TCC*, 2006.
64. A. Lewko and B. Waters. Decentralizing attribute-based encryption. In *Eurocrypt*, 2011.
65. B. Libert, K. Nguyen, A. Passelègue, and R. Titiu. Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In *Asiacrypt*, 2020.
66. B. Libert, K. Nguyen, T. Peters, and M. Yung. One-shot Fiat-Shamir-based NIZK arguments of composite residuosity in the standard model. Cryptology ePrint Archive: Report 2020/1334, 2020.
67. 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.

68. B. Libert, D. Stehlé, and R. Titiu. Adaptively secure distributed PRFs from LWE. In *TCC*, 2018.
69. B. Libert and M. Yung. Adaptively secure non-interactive threshold cryptosystems. In *ICALP*, 2011.
70. B. Libert and M. Yung. Non-interactive cca-secure threshold cryptosystems with adaptive security: New framework and constructions. In *TCC*, 2012.
71. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Asiacrypt*, 2009.
72. V. Lyubashevsky. Lattice signatures without trapdoors. In *Eurocrypt*, 2012.
73. P. Miao, S. Patel, M. Raykova, K. Seth, and M. Yung. Two-sided malicious security for private intersection-sum with cardinality. In *Crypto*, 2020.
74. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *Eurocrypt*, 2012.
75. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.
76. M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, 1990.
77. R. Ostrovsky and M. Yung. How to withstand mobile virus attacks. In *PODC*, 1991.
78. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Eurocrypt*, 1999.
79. C. Peikert and S. Shiehian. Non-interactive zero knowledge for NP from (plain) Learning With Errors. In *Crypto*, 2019.
80. T. Rabin. A simplified approach to threshold and proactive rsa. In *Crypto*, 1999.
81. C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In *Crypto*, 1991.
82. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.
83. A. Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS*, 1999.
84. V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. In *Eurocrypt*, 1998.
85. V. Shoup and R. Gennaro. Securing threshold cryptosystems against chosen ciphertext attack. *J. of Cryptology*, 15(2), 2002.
86. R. Thorbek. *Linear integer secret sharing.* PhD thesis, PhD thesis, Aarhus University, 2009.
87. L. G. Valiant. Short monotone formulae for the majority function. volume 5, pages 363–366. Elsevier, 1984.
88. H. Wee. Threshold and revocation cryptosystems via extractable hash proofs. In *Eurocrypt*, 2011.
89. H. Wee. Dual projective hashing and its applications - lossy trapdoor functions and more. In *Eurocrypt*, 2012.
90. X. Xie, R. Xue, and R. Zhang. Efficient threshold encryption from lossy trapdoor functions. In *PQCrypto*, 2011.
91. A. Young and M. Yung. Questionable encryption and its applications. In *Mycrypt*, 2005.

Supplementary Material

A Other Definitions for Cryptographic Primitives

A.1 Non-Interactive Zero-Knowledge and Simulation-Sound Arguments

We recall the definitions of NIZK proofs. Since it is sufficient for our applications, we allow the common reference string to be generated as a function of the language $\mathcal{L}$ (analogously to quasi-adaptive NIZK proofs [60]). We actually give a slightly different definition than the standard ones, defining NIZK for gap languages. That is, a language is defined by a pair of language $\mathcal{L}_{\mathsf{zk}} \subseteq \mathcal{L}_{\mathsf{sound}}$, and completeness is guaranteed for statements in $\mathcal{L}_{\mathsf{zk}}$ while soundness is guaranteed for statement outside $\mathcal{L}_{\mathsf{sound}}$. This is sufficient for our purpose.

In addition, we consider NIZK argument systems where each argument comes with a label lbl taken as input by both the prover and the verifier. Labels will only be useful when we consider simulation-soundness, which is necessary in our CCA-secure encryption schemes.

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

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

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

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

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

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

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

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

  Here, $\mathsf{P}(\mathsf{crs}, \cdot, \cdot)$ *is an oracle that outputs* $\perp$ *on input of* $(x, w, \mathsf{lbl}) \notin R_{\mathsf{zk}}$ *and outputs a valid proof* $\pi \leftarrow \mathsf{P}(\mathsf{crs}, x, w, \mathsf{lbl})$ *otherwise;* $\mathcal{O}(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$ *is an oracle that outputs* $\perp$ *on input of* (x, w, lbl) *such that* $(x, w) \notin R_{\mathsf{zk}}$ *and outputs a simulated argument* $\pi \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, x, \mathsf{lbl})$ *on input of* (x, w, lbl) *such that* $(x, w) \in R_{\mathsf{zk}}$*. Note that this simulated proof* π *is generated independently of the witness* w *provided as input.*[14]

The notion of soundness captured by Definition A.1 is *non-adaptive* in that the statement is given as input to the dishonest prover and chosen independently of the common reference string. The stronger notion of *adaptive soundness* allows the target statement to be chosen by the adversary after having received the common reference string. It is known (see, e.g., [3]) that perfect or statistical NIZK arguments cannot provide adaptive soundness under falsifiable assumptions. The reason stems from the impossibility of detecting when the adversary wins and outputs a proof for a false statement. One way to bypass the impossibility results is to consider *trapdoor languages*, where a trapdoor can be used to recognize false statements. In the context of CCA security, we will consider a notion of adaptive soundness for trapdoor languages.

Definition A.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).

Simulation-Soundness. We now recall the definition of simulation-soundness of [83,40], which formalizes 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. This restriction has no impact on our applications where we always have a membership testing trapdoor $\tau_{\mathcal{L}}$ at disposal.

Definition A.2 ([83,40]). *Let a language* $\mathcal{L} = (\mathcal{L}_{\mathsf{zk}}, \mathcal{L}_{\mathsf{sound}})$*. A NIZK argument system for* $\mathcal{L}$ *provides* **simulation-soundness** *if no PPT adversary has non-negligible advantage in this game.*

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

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

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

The weaker notion of *one-time* simulation-soundness (which is sufficient for our purpose of constructing CCA-secure TPKE schemes) is defined identically, except that the adversary is only allowed to make one query at stage 2.

A.2 Consistency Definition for TPKE Schemes

This section recalls the definition of consistency considered in [16] for threshold public-key encryption schemes. It informally captures the infeasibility of outputting two distinct sets of valid-looking decryption shares for which the $\mathsf{Combine}$ algorithm reconstructs different messages. It can easily be shown that consistency is implied by robustness.

Definition A.3 (Consistency). *A TPKE scheme provides consistency if, for any* PPT *adversary* $\mathcal{A}$, *the following experiment* $\mathsf{Expt}^{\mathrm{consist}}_{\mathcal{A},\mathsf{TPKE}}(1^\lambda)$ *outputs* 1 *with negligible probability.*

1. *On input the security parameter* λ, $\mathcal{A}$ *chooses an access structure* $\mathbb{A}$.
2. *The challenger samples* $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$ *and provides* $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_\ell)$ *to* $\mathcal{A}$.
3. $\mathcal{A}$ *eventually outputs a forgery* $(\mathsf{ct}^*, \{\mu_i^0, i \in \mathcal{S}_0\}, \mathcal{S}_0, \{\mu_j^1, j \in \mathcal{S}_0\}, \mathcal{S}_1)$ *where* $\mathcal{S}_0 \in \mathbb{A}$ *and* $\mathcal{S}_1 \in \mathbb{A}$.
4. *Outputs* 1 *only if* $\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}^*, \mu_i^b) = 1$ *for all* $b \in \{0,1\}$ *and* $i \in \mathcal{S}_b$ *and* $\mathsf{Combine}(\mathsf{pk}, (\mathcal{S}_0, \{\phi(\mu_i^*)\}_{i \in \mathcal{S}_0}), \mathsf{ct}) \neq \mathsf{Combine}(\mathsf{pk}, (\mathcal{S}_1, \{\phi(\mu_j^*)\}_{j \in \mathcal{S}_1}), \mathsf{ct})$.

Note that ϕ is the identity function for non-robust TPKE.

B Deferred Proofs for the NIZK Argument of Section 3

B.1 Proof of Theorem 3.3

Proof. To prove the result, we describe a simulator $(\mathsf{Sim}_0, \mathsf{Sim}_1)$ which uses the lossy secret key $\tau_{\mathsf{zk}} = \mathsf{tk}_{\mathsf{LPKE}}$ to simulate transcripts (a, Chall, 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 $\mathsf{Gen}_{\mathcal{L}^{\mathrm{DCR}}}$ while Sim_1 is described hereunder.

$\mathbf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, x, \mathsf{lbl})$**:** On input a statement $x \in \mathbb{Z}_{N^2}^*$, a label lbl and the simulation trapdoor $\tau_{\mathsf{zk}} = \mathsf{sk}_{\mathsf{LPKE}}$, algorithm Sim_1 proceeds as follows.

1. Generate a one-time signature key pair $(\mathsf{VK}, \mathsf{SK}) \leftarrow \mathcal{G}(1^\lambda)$. Then, sample random coins $r \hookleftarrow D_R^{\mathsf{LPKE}}$ and compute
$$a \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, 0^{|a'|}; r), \tag{12}$$
where $0^{|a'|}$ denotes the all-zeroes string of the same length as the first prover message in Π'.
2. Compute $\mathsf{Chall} = \mathsf{Hash}(k, (x, \mathbf{a}, \mathsf{VK}))$.
3. Run the ZK simulator $(a', z') \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Chall})$ of Π' to obtain a simulated transcript (a', Chall, z') of Π' for the challenge Chall.
4. Using the trapdoor key $\mathsf{tk}_{\mathsf{LPKE}}$, compute random coins $\tilde{r} \sim D_{\mathsf{pk}_{\mathsf{LPKE}}, a', a, \mathsf{VK}}$ which explain a as a lossy encryption of a' under the tag VK. Namely, compute
$$\tilde{r} \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LOpener}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{VK}, a, 0^{|a'|}, a'; r) \tag{13}$$
which satisfy
$$a = \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, a'; \tilde{r}).$$
Then, set $z = (z', a', \tilde{r})$.
5. Generate a one-time signature $sig \leftarrow \mathcal{S}(\mathsf{SK}, (x, a, z, \mathsf{lbl}))$ and output the proof $\boldsymbol{\pi} = \big(\mathsf{VK}, (a, z), sig\big)$.

We note that Sim_1 in such a way that, in the DCR-based instantiation, the equivocation (13) of Paillier ciphertexts can be conducted without knowing the factorization of the modulus. This will be crucial in the proof of simulation-soundness.

We now prove that the simulation is statistically indistinguishable from proofs generated by the real prover.

The special zero-knowledge property of Π' implies that its simulator produces $(a', z') \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Chall})$ such that (a', Chall, z') is indistinguishable from a real transcript with challenge Chall. This implies that the distribution

$$\begin{aligned}\big\{\big(a, a', \tilde{r}, z'\big) \mid\ & r \hookleftarrow D_R^{\mathsf{LPKE}},\\ & a \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}\big(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, \mathbf{0}^{|a'|}; r\big),\\ & (a', z') \leftarrow \mathsf{ZKSim}(\mathsf{crs}'_{\mathcal{L}}, x, \mathsf{Chall}),\\ & \tilde{r} \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LOpener}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{VK}, a, 0^{|a'|}, a'; r)\ \ \big\},\end{aligned} \tag{14}$$

is indistinguishable from

$$\begin{aligned}\big\{\big(a, a', \tilde{r}, z'\big) \mid\ & r \hookleftarrow D_R^{\mathsf{LPKE}},\\ & a \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}\big(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, \mathbf{0}^{|a'|}; r\big),\\ & (a', st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w),\\ & z' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, a', \mathsf{Chall}, st'),\\ & \tilde{r} \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LOpener}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{VK}, a, 0^{|a'|}, a'; r)\big\},\end{aligned} \tag{15}$$

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

$$\Big\{\big(a, a', \tilde{r}, z'\big) \mid \begin{array}{l} (a', st') \leftarrow \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w), \\ r \hookleftarrow D_R^{\mathsf{LPKE}}, \\ a \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Encrypt}\big(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{VK}, a'; r\big), \\ z' = \mathsf{P}'(\mathsf{crs}'_{\mathcal{L}}, x, w, a', \mathsf{Chall}, st')\Big\}, \end{array} \tag{16}$$

Clearly, the distribution (14) corresponds to proof generated by the simulator while (16) is the distribution generated by the real prover. This implies that simulated proofs are perfectly indistinguishable from real proofs if the simulator of Π' is perfectly special ZK. □

B.2 Proof of Theorem 3.4

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

Game_0: This is the real game of Definition A.2. Namely, the challenger runs $(\mathsf{crs}, \tau_{\mathsf{zk}}) \leftarrow \mathsf{Sim}_0(\mathsf{par})$ and gives $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, \mathsf{pk}_{\mathsf{LPKE}}, \mathsf{OTS})$ to the adversary $\mathcal{A}$. At the same time, the challenger generates a language description $\mathcal{L}$ together with a trapdoor $\tau_{\mathcal{L}}$ in such a way that it can efficiently test if $\mathcal{A}$'s output satisfies the winning condition (ii). The adversary is allowed to make one query to the oracle $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$. After having received $\mathsf{crs}_{\mathcal{L}}$, $\mathcal{A}$ chooses a statement x with a label lbl and the challenger replies by returning a simulated argument $\boldsymbol{\pi} = \big(\mathsf{VK}, (a, z), sig\big) \leftarrow \mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, x, \mathsf{lbl})$. When $\mathcal{A}$ halts, it outputs a triple $(x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star)$, where $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, (a^\star, z^\star), sig^\star\big)$. The Boolean variable W_0 is set to $W_0 = \mathtt{true}$ under the following conditions: (i) $(x^\star, \mathsf{lbl}^\star, \boldsymbol{\pi}^\star) \neq (x, \mathsf{lbl}, \boldsymbol{\pi})$; (ii) $x^\star \notin \mathcal{L}$; and (iii) $V(\mathsf{crs}, x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star) = 1$. We assume w.l.o.g. that the one-time verification key VK is chosen by Sim_1 at the outset of the game. By definition, we have $\mathbf{Adv}^{\text{1-ss}}_{\mathcal{A}}(\lambda) = \Pr[W_0]$.

Game_1: This is like Game_0 except that the challenger sets $W_1 = \mathtt{false}$ if $\mathcal{A}$ outputs a fake proof $(x^\star, \boldsymbol{\pi}^\star, \mathsf{lbl}^\star)$, where $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, (a^\star, z^\star), sig^\star\big)$ recycles the one-time verification key used by Sim_1 (i.e., $\mathsf{VK}^\star = \mathsf{VK}$). The strong unforgeability of OTS implies that $\Pr[W_1]$ cannot significantly differ from $\Pr[W_0]$ as $\mathcal{B}$ readily implies a forger such that $|\Pr[W_1] - \Pr[W_0]| \leq \mathbf{Adv}^{\text{ots}}_{\mathcal{B}}(\lambda)$.

Game_2: We modify the generation of crs. Namely, we now generate the lossy keys of Π^{LPKE} as $(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}}) \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LKeygen}(\Gamma, \mathsf{VK})$ instead of $(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}}) \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LKeygen}(\Gamma, 0^L)$. By the second indistinguishability property of Π^{LPKE}, we know that changing the initialization value does not significantly affect $\mathcal{A}$'s view as the two distributions of $\mathsf{pk}_{\mathsf{LPKE}}$ are statistically close. It follows that $|\Pr[W_2] - \Pr[W_1]| \leq 2^{-\Omega(\lambda)}$.

Game_3: We modify the oracle $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$ and by using the equivocation property of Π^{LPKE} for lossy tags (instead of lossy keys). To simulate an argument for (x, lbl) to $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$ uses the lossy tag VK (recall that lossy

keys are generated as $(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}}) \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LKeygen}(\Gamma, \mathsf{VK}))$ to equivocate a using the trapdoor key $\mathsf{tk}_{\mathsf{LPKE}}$ instead of the lossy secret key $\mathsf{sk}_{\mathsf{LPKE}}$ of Π^{LPKE}. Namely, at step 4 of Sim_1, the modified $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$ computes random coins $\tilde{r} \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Opener}(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}}, \mathsf{VK}, a, 0^{|a'|}, a', r)$ instead of running $\mathsf{LOpener}$ using $\mathsf{sk}_{\mathsf{LPKE}}$. We define the Boolean variable W_3 exactly as W_2. Since Opener and $\mathsf{LOpener}$ sample from the same distribution D_R^{LPKE} over R^{LPKE}, we have $|\Pr[W_3] - \Pr[W_2]| \leq 2^{-\Omega(\lambda)}$.

Game_4: We change the distribution of $\mathsf{pk}_{\mathsf{LPKE}}$ in the common reference string crs. At step 3 of $\mathsf{Gen}_{\mathcal{L}}$, we generate the keys for Π^{LPKE} as injective keys $(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}}) \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{Keygen}(\Gamma, \mathsf{VK})$ instead of generating them as lossy keys $(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{sk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}}) \leftarrow \Pi^{\mathsf{LPKE}}.\mathsf{LKeygen}(\Gamma, \mathsf{VK})$. The indistinguishability property (i) of Π^{LPKE} ensures that $\Pr[W_4]$ and $\Pr[W_3]$ are negligibly far apart. Recall that this indistinguishability property ensures that the distributions of pairs $(\mathsf{pk}_{\mathsf{LPKE}}, \mathsf{tk}_{\mathsf{LPKE}})$ produced by Keygen and $\mathsf{LKeygen}$ are computationally indistinguishable. We can thus easily build a distinguisher $\mathcal{B}$ against Π^{LPKE} that transitions from Game_4 to Game_5 (in particular, the reduction has $\mathsf{tk}_{\mathsf{LPKE}}$ at disposal to simulate $\mathsf{Sim}_1(\mathsf{crs}, \tau_{\mathsf{zk}}, \cdot, \cdot)$). It comes that $|\Pr[W_4] - \Pr[W_3]| \leq \mathbf{Adv}_{\mathcal{B}}^{\mathsf{indist\text{-}LPKE\text{-}1}}(\lambda)$.

Game_5: We change again the distribution of $\mathsf{crs} = (\mathsf{par}, \mathsf{crs}_{\mathcal{L}}, \mathsf{pk}_{\mathsf{LPKE}}, \mathsf{OTS})$ by relying on the CRS indistinguishability property of the trapdoor Σ-protocol Π'. Namely, we use the $\mathsf{TrapGen}'$ algorithm of Definition 2.21 to generate $\mathsf{crs}'_{\mathcal{L}}$ as $(\mathsf{crs}'_{\mathcal{L}}, \tau_{\Sigma}) \leftarrow \mathsf{TrapGen}'(\mathsf{par}, \mathcal{L}^{\mathsf{DCR}}, \tau_{\mathcal{L}})$ 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 be used in later games to compute the $\mathsf{BadChallenge}$ function of the trapdoor Σ-protocol Π'. In order to evaluate $\mathsf{BadChallenge}$, we also use the injective decryption key $\mathsf{sk}_{\mathsf{LPKE}}$ which allows decrypting $a^\star$ for any $\mathsf{VK} \neq \mathsf{VK}^\star$.

Since we are done with the indistinguishability properties of Π^{LPKE}, we are free to use $\mathsf{sk}_{\mathsf{LPKE}}$ (which contains the factorization of the modulus in our DCR-based instantiation) to decrypt the injective LPKE ciphertext $a^\star$ in $\boldsymbol{\pi}^\star$.

Game_6: We now use the decryption algorithm of the lossy PKE scheme Π^{DCR}. We know that the adversary's output $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, (a^\star, z^\star), sig^\star\big)$ involves a verification key $\mathsf{VK}^\star$ such that $\mathsf{VK} \neq \mathsf{VK}^\star$ unless $W_5 = \mathsf{false}$ in Game_5. This implies that $a^\star$ is an extractable commitment to some $a'^\star$, so that $\boldsymbol{\pi}^\star$ uniquely determines the message $a'^\star$ obtained by decrypting $a^\star$. We thus consider the relation R_{bad} defined by

$$\big((x, a, \mathsf{VK}), \mathsf{Chall}\big) \in R_{\mathsf{bad}} \quad \Leftrightarrow \quad x \notin \mathcal{L} \quad \wedge \quad \mathsf{Chall} = \mathsf{BadChallenge}\left(\tau_{\Sigma}, \mathsf{crs}'_{\mathcal{L}}, x, \Pi^{\mathsf{LPKE}}.\mathsf{Decrypt}\big(\mathsf{sk}_{\mathsf{LPKE}}, \mathsf{VK}, a\big)\right). \tag{17}$$

We now set $W_6 = \mathsf{false}$ if

$$\mathsf{Hash}\big(k, (x^\star, a^\star, \mathsf{VK}^\star)\big) \neq \mathsf{BadChallenge}\left(\tau_{\Sigma}, \mathsf{crs}'_{\mathcal{L}}, x^\star, \Pi^{\mathsf{LPKE}}.\mathsf{Decrypt}\big(\mathsf{sk}_{\mathsf{LPKE}}, \mathsf{VK}^\star, a^\star\big)\right). \tag{18}$$

Otherwise, we set $W_6 = W_5$. If $x^\star \notin \mathcal{L}$, we note that, unless Π^{LPKE} fails to correctly decrypt under injective tags, $\boldsymbol{\pi}^\star$ cannot be accepted by the verifier if inequality (18) holds. By the statistical decryption correctness of Π^{LPKE}, we have $|\Pr[W_6] - \Pr[W_5]| \leq 2^{-\Omega(\lambda)}$.

In Game_6, we have $\Pr[W_6] \leq \mathbf{Adv}^{\mathrm{CI}}_{\mathcal{A}}(\lambda)$ because, if the adversary outputs a valid $\boldsymbol{\pi}^\star = \big(\mathsf{VK}^\star, (a^\star, z^\star), sig^\star\big)$ for $x^\star \notin \mathcal{L}$, we necessarily have

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

which breaks the correlation intractability of $\mathcal{H}$ for the relation R_{bad}.

Putting the above altogether, we obtain

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

which completes the proof. □

C Achieving Robustness in the TPKE of Section 4

In this section, we show how to obtain robustness against malicious adversaries using trapdoor Σ-protocols and correlation-intractable hash functions.

C.1 A Σ-Protocol Showing Equality of Two Discrete Logarithms in the DCR Setting

In order to achieve robustness in the DCR-based scheme of Section 4, we need to prove equalities between discrete logarithms in a group of hidden order. To do this, we adapt a standard Σ-protocol due to Chaum and Pedersen in [28] for the language

$$\mathcal{L}^{\log}_i = \big\{(g_1, \{h_{i,j}, \mu_{i,j}\}_{j \in \psi^{-1}(i)}) \in (\mathbb{Z}^\star_{N^{\zeta+1}})^{2d_i+1} \mid \forall j \in \psi^{-1}(i),\ \exists s_j \in [-B^*, B^*] \ :\ h_{i,j} = g_0^{4N^\zeta \cdot s_j} \ \ \wedge \ \ \mu_{i,j} = g_1^{2 \cdot s_j}\big\},$$

where d_i is the size of the set $\psi^{-1}(i) \subset \{1, \ldots, d\}$. This language can be described by $\mathsf{crs}_{\mathcal{L}} := (g_0, N = pq, \zeta, (\mathbf{M}, \psi), B, B^*)$ with p, q two primes chosen as in the KeyGen algorithm of the TPKE scheme and $g_0, g_1 \in \mathbb{Z}^\star_{N^{\zeta+1}}$, and $B^\star \in \mathbb{N}$ is a guaranteed upper bound on the magnitude of witnesses when they actually have magnitude $B < B^*$. In the scheme of Section 4, witnesses are share units $\{s_i\}_{i=1}^d$ of magnitude $B = O(\sigma \cdot e \cdot \log \ell)$, where we may set $e = O(\ell^{1+\sqrt{2}})$ using the LISS scheme of Benaloh and Leichter [11]. Membership of the language $\mathcal{L}^{\log}_i$ thus only guarantees a looser upper bound on $|s_j|$, but it does not affect robustness.

The Σ-protocol is described as follows:

$\mathsf{Gen}(1^\lambda)$: This algorithm sets $\mathsf{crs}_{\mathcal{L}} = (g_0, N = pq, \zeta, (\mathbf{M}, \psi), B, B^*)$ by choosing $\beta > 2^\lambda dB$, where d is the number of rows of $\mathbf{M} \in \mathbb{Z}^{d \times e}$, and $B^* = 2\beta$. It returns $\mathsf{crs} := (\mathsf{crs}_{\mathcal{L}}, \beta)$.

$\mathsf{P}(\mathsf{crs}, x, w) \leftrightarrow \mathsf{V}(\mathsf{crs}, x)$: Given a crs describing a language $\mathcal{L}_i^{\log}$ and a statement

$$x = \left(g_1, \{h_{i,j}, \mu_{i,j}\}_{j \in \psi^{-1}(i)}\right)$$

P and V write $\psi^{-1}(i) = \{j_1, \ldots, j_{d_i}\}$ and define $\mathbf{h}_i = (h_{i,j_1}, \ldots, h_{i,j_{d_i}})$, $\boldsymbol{\mu}_i = (\mu_{i,j_1}, \ldots, \mu_{i,j_{d_i}})$. Then, they proceed as follows:

1. P samples $\mathbf{r} \hookleftarrow D_{\mathbb{Z}^{d_i}, \beta}$ and sends
$$\boldsymbol{a} := (\mathbf{a}_0, \mathbf{a}_1) = \left(g_0^{4N^\zeta \cdot \mathbf{r}}, g_1^{2 \cdot \mathbf{r}}\right) \in (\mathbb{Z}^*_{N^{\zeta+1}})^{d_i} \times (\mathbb{Z}^*_{N^{\zeta+1}})^{d_i}$$
as a first message to V.
2. V chooses a challenge $\mathsf{Chall} \in \{0,1\}$ and returns it to P.
3. P using the witness $w = \{s_j\}_{j \in \psi^{-1}(i)}$ to define $\mathbf{w} = (s_{j_1}, \ldots, s_{j_{d_i}})^\top \in \mathbb{Z}^{d_i}$ and compute $\mathbf{z} = \mathbf{r} + \mathsf{Chall} \cdot \mathbf{w}$ and aborts if $\|\mathbf{z}\| > B^*/2$. Otherwise, it returns $\mathbf{z} := (z_{j_1}, \ldots, z_{j_{d_i}})^\top \in \mathbb{Z}^{d_i}$.
4. V accepts and returns 1 if $\|\mathbf{z}\| \leq B^*/2$ and
$$g_0^{4N^\zeta \cdot \mathbf{z}} = \mathbf{a}_0 \cdot \mathbf{h}_i^{\mathsf{Chall}} \bmod N^{\zeta+1}, \qquad g_1^{2 \cdot \mathbf{z}} = \mathbf{a}_1 \cdot \boldsymbol{\mu}_i^{\mathsf{Chall}} \bmod N^{\zeta+1}. \quad (19)$$
Otherwise, it returns 0.

Assuming that P and V follow the protocol, it is easy to see that V will always accept but with negligible probability. We now prove other properties.

Special Zero-Knowledge: On input a tuple $(\mathsf{crs}, x, \mathsf{Chall})$, where the statement $x = \left(g_1, \{h_{i,j}, \mu_{i,j}\}_{j \in \psi^{-1}(i)}\right) \in \mathcal{L}_i^{\log}$, ZKSim samples $\mathbf{z} \hookleftarrow D_{\mathbb{Z}^{d_i}, \beta}$ and computes

$$\mathbf{a}_0 = g_0^{4N^\zeta \cdot \mathbf{z}} \cdot \mathbf{h}_i^{-\mathsf{Chall}} \bmod N^{\zeta+1}, \qquad \mathbf{a}_1 = g_1^{2 \cdot \mathbf{z}} \cdot \boldsymbol{\mu}_i^{-\mathsf{Chall}} \bmod N^{\zeta+1}.$$

The final transcript is $\left((\mathbf{a}_0, \mathbf{a}_1), \mathsf{Chall}, \mathbf{z}\right)$. The difference between the simulated distribution of $\mathbf{z}$ its distribution in a real transcript is the center of the Gaussian: In the former case the center is $\mathbf{0}^{d_i}$ whereas, in the latter case, the center is $\mathsf{Chall} \cdot \mathbf{w}$. From Lemma 2.5, we can bound the distance between the two distributions as

$$\Delta\left(D_{\mathbb{Z}^{d_i}, \beta}, (D_{\mathbb{Z}^{d_i}, \beta, \mathsf{Chall} \cdot s_j})_{j \in \psi^{-1}(i)}\right) \leq \frac{d_i B}{\beta} \leq \frac{1}{2^\lambda}. \quad (20)$$

We also note that, if g_1 is an N^ζ-th residue and $x \in \mathcal{L}_i^{\log}$, $(\mathbf{a}_0, \mathbf{a}_1)$ is uniquely determined by $\mathsf{Chall} \in \{0,1\}$ and the statement since $g_0^{4N^\zeta}$ and g_1^2 both generate the subgroup of $2N^\zeta$-th residues in $\mathbb{Z}^*_{N^{\zeta+1}}$. We conclude that (20) also bounds the distance between the distribution of a real transcript (for fixed Chall) and the distribution of a transcript produced by ZKSim.

Special Soundness: Let us assume that g_1 is an N^ζ-th residue in $\mathbb{Z}^*_{N^{\zeta+1}}$. Let us assume that two valid transcripts $\big((\mathbf{a}_0, \mathbf{a}_1), 0, \mathbf{z}\big)$, $\big((\mathbf{a}_0, \mathbf{a}_1), 1, \mathbf{z}'\big)$ exist for the challenges 0 and 1 with $\|\mathbf{z}\|, \|\mathbf{z}'\| \leq B^*/2$. Then, the following holds:

$$\mathbf{h}_i = g_0^{4N^\zeta(\mathbf{z}'-\mathbf{z})} \quad \wedge \quad \boldsymbol{\mu}_i = g_1^{2(\mathbf{z}'-\mathbf{z})}$$

This implies $(g_1, \{h_{i,j}, \mu_{i,j}\}_{j\in\psi^{-1}(i)}) \in \mathcal{L}_i^{\log}$ since $\|\mathbf{z}'-\mathbf{z}\|_\infty \leq \|\mathbf{z}'-\mathbf{z}\| \leq B^*$. Hence, there can be at most one bad challenge if $x \notin \mathcal{L}_i^{\log}$.

Witness-indistinguishability: We rely on Lemma C.1 which shows that the Σ-protocol provides statistical witness-indistinguishability.

In order to obtain robustness without assuming erasures on behalf of the servers, we rely on the statistical WI property of the scheme. The reason is that, if we were to use the NIZK simulator in the proof of CCA security under adaptive corruptions, the simulator would have to explain simulated proofs as if they had been honestly generated. Instead of relying on erasures as a loophole, we exploit the statistical WI property to argue that, even if WI proofs of correct partial decryptions are faithfully generated by the challenger, they do not decrease the entropy of shares from the adversary's view.

Lemma C.1. *The above Σ-protocol for the language $\mathcal{L}_i^{\log}$ is statistically witness-indistinguishable assuming that the order of g_1^2 modulo $N^{\zeta+1}$ divides $p'q'$.*

Proof. Let $x = (g_1, \{h_{i,j}, \mu_{i,j}\}_{j\in\psi^{-1}(i)}) \in \mathcal{L}_i^{\log}$ with $\mathrm{ord}(g_1^2)|p'q'$ be a statement such that $\mathsf{sk}_i^0 = \{s_j^0\}_{j\in\psi^{-1}(i)}$ and $\mathsf{sk}_i^1 = \{s_j^1\}_{j\in\psi^{-1}(i)}$ are two witnesses for x. In particular, for all $j \in \psi^{-1}(i)$, $h_{i,j} = g_0^{4N^\zeta s_j^0} = g_0^{4N^\zeta s_j^1}$ and $\mu_{i,j} = g_1^{2s_j^0} = g_1^{2s_j^1}$ and then $s_j^0 - s_j^1 \in p'q'\mathbb{Z}$. We define the following probability distributions, for $b = 0, 1$, where V is an adversary that chooses challenges.

$$D_b = \Big\{ \big(\boldsymbol{a} := \{g_0^{4N^\zeta r_j}, g_1^{2r_j}\}_{j\in\psi^{-1}(i)}, \mathsf{Chall}, \{r_j + \mathsf{Chall}\cdot s_j^b\}_{j\in\psi^{-1}(i)} \big) \;\big|\; r_j \hookleftarrow D_{\mathbb{Z},\beta}, \mathsf{Chall} \hookleftarrow \mathsf{V}(\mathsf{crs}, x, \boldsymbol{a}) \Big\}$$

To prove witness-indistinguishability, we need to prove that D_1 is indistinguishable from D_2. Since the distributions restricted to a are equal, Chall is independent of the witness. Therefore, it is easy to see that if $\mathsf{Chall} = 0$, the distributions are the same and we have

$$\Delta(D_0, D_1) = \sum_{(\alpha,1,\gamma)} \sum_{j\in\psi^{-1}(i)} |p_{0,j}(\alpha, 1, \gamma) - p_{1,j}(\alpha, 1, \gamma)| \;,$$

where $p_{b,j}(\alpha, c, \gamma) = \Pr_{(\boldsymbol{a},\mathsf{Chall},\mathbf{z})\hookleftarrow D_b}[(a_j, \mathsf{Chall}, z_j) = (\alpha_j, c, \gamma_j)]$ is defined for any triple $\big(\alpha = \{\alpha_{j,0}, \alpha_{j,1}\}_{j\in\psi^{-1}(i)}, c, \gamma = \{\gamma_j\}_{j\in\psi^{-1}(i)}\big)$ such that $g_0^{4N^\zeta\gamma_j} = \alpha_{j,0}h_j^c$ and $g_1^{2\gamma_j} = \alpha_{j,1}\mu_j^c$, for all $j \in \psi^{-1}(i)$. By looking at the components,

$$p_{b,j}(\alpha, 1, \gamma) = \Pr_{r_j\hookleftarrow D_{\mathbb{Z},\beta}}[\alpha_j = \{g_0^{r_j}, g_1^{r_j}\}] \cdot \Pr_{\mathsf{Chall}\hookleftarrow\mathsf{V}(\mathsf{crs},x,\alpha)}[\mathsf{Chall} = c] \cdot \Pr_{(\boldsymbol{a},\mathsf{Chall},\mathbf{z})\hookleftarrow D_b}[z_j = \gamma_j \mid a_j = \alpha_j \wedge \mathsf{Chall} = c] \,,$$

for all $i \in \psi^{-1}(i)$ and $b \in \{0,1\}$. However, as long as $\alpha = \boldsymbol{a}$ and $1 = \mathsf{V}(\mathsf{crs}, x, \alpha)$ we have $\gamma_j = r_j + s_j^b \bmod p'q'$ (because we assume that $\mathrm{ord}(g_1^2)$ divides $p'q'$) for all $j \in \psi^{-1}(i)$, which means that the last probability can be computed as $D_{\mathbb{Z},\beta}(\gamma_j - s_j^b)/\Pr_{r_j}[r_j = \gamma_j - s_j^b \mod p'q']$. By definition of (α, c, γ), the condition "$\boldsymbol{a} = \alpha$" can be replaced with "$r_j = \gamma_j - s_j^b \mod p'q'$ for all $j \in \psi^{-1}(i)$" and

$$\Delta(D_0, D_1) \leq \sum_{\gamma} \sum_{j \in \psi^{-1}(i)} \left| \left(D_{\mathbb{Z},\beta,s_j^0}(\gamma) - D_{\mathbb{Z},\beta,s_j^1}(\gamma) \right) \right|$$

Finally, we notice that giving $\{\gamma_j\}_{j \in \psi^{-1}(i)}$ is equivalent to giving a full transcript $(\alpha, 1, \gamma)$ as α can be computed with $\alpha_{j,0} = g_0^{4N^\zeta(\gamma_j - s_j^0)} = g_0^{4N^\zeta(\gamma_j - s_j^1)}$ and $\alpha_{j,1} = g_1^{2(\gamma_j - s_j^0)} = g_1^{2(\gamma_j - s_j^1)}$. This implies the following bound:

$$\Delta(D_0, D_1) \leq \sum_{j \in \psi^{-1}(i)} \Delta(D_{\mathbb{Z},\beta,s_j^0}, D_{\mathbb{Z},\beta,s_j^1})$$

Since $|s_j^0 - s_j^1| \in 2B$, we get $\Delta(D_0, D_1) \leq 2d_iB/\beta \leq 2^{1-\lambda}$. □

In order to obtain a trapdoor version of the above Σ-protocol, we can apply the generic transformation of Ciampi *et al.* [29, Section 4.2], which uses a semantically secure public-key encryption scheme. In [29], they show how to turn any standard Σ-protocol with binary challenges into a trapdoor Σ-protocol by having the prover encrypt the two possible responses and send the resulting ciphertexts along with its first message.

In order for this transformation to preserve *statistical* WI, we need the underlying PKE scheme to be lossy in order to guarantee that the unopened ciphertext leaks no information on the unopened response at each iteration. More precisely, the underlying public key can be set up in lossy mode when we want to ensure statistical WI and in injective mode when we need to argue soundness. Fortunately, Paillier's cryptosystem can be used for this purpose as it is known [91,57] to be lossy and it does not introduce any additional assumption in our proof of robustness.

The resulting trapdoor Σ-protocol is denoted $\Pi_{\log}^i$ with prover $\mathsf{P}_{\log}^i$ and verifier $\mathsf{V}_{\log}^i$, for all $i = 1$ to ℓ. However, a single CRS denoted $\mathsf{crs}_{\log}$ can be used by all the ℓ protocols. Finally, we assume that each $\Pi_{\log}^i$ already repeats the internal arguments with the binary challenge space $\kappa \in O(\lambda)$ times, and is made non-interactive using a correlation-intractable hash function for the induced bad-challenge relation.

C.2 Achieving Robustness in the DCR/LWE-based Scheme

We use the ℓ trapdoor Σ-protocols $\Pi_{\log}^i$, one for each server, to turn the CCA-secure TPKE $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{PartDec}, \mathsf{Combine})$ of Section 4 into a robust TPKE. To this end, we simply augment KeyGen with the global CRS $\mathsf{crs}_{\log}$ of the $\Pi_{\log}^i$'s and $\mathsf{PartDec}$ with the $\mathsf{P}_{\log}^i$'s, and add the algorithm $\mathsf{PartVerify}$ based on the $\mathsf{V}_{\log}^i$'s. More precisely,

$\mathsf{KeyGen}'(1^\lambda, \mathbb{A})$**:** Run $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{KeyGen}(1^\lambda, \mathbb{A})$, where the secret key shares are contained in the components of $\boldsymbol{s} = \mathbf{M} \cdot \boldsymbol{\rho} = (s_1, \ldots, s_d)^\top$. Compute the verification key $\mathsf{vk} := \boldsymbol{v} := g_0^{4N^\zeta \boldsymbol{s}} = (v_1, \ldots, v_d)^\top$. Define the language $\mathcal{L}_i^{\log}$ for every $1 \leq i \leq \ell$ using pp and pk, and let $\mathsf{crs}_{\log}$ be their global CRS. Update the public key $\mathsf{pk}' = (\mathsf{pk}, \mathsf{vk}, \mathsf{crs}_{\log})$ and return

$$\big(\mathsf{pp}, \mathsf{pk}', \mathsf{sk}_1, \ldots, \mathsf{sk}_\ell\big)$$

$\mathsf{PartDec}'(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct} := (C_0, C_1, \pi))$**:** If $\perp = \mathsf{PartDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct})$, return $\perp$. Else, let $\boldsymbol{\mu}_i = \mathsf{PartDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct})$ and run $\pi_i \leftarrow \mathsf{P}_{\log}^i(\mathsf{crs}_{\log}, (C_0, \{v_j, \mu_j\}_{j \in \psi^{-1}(i)}), \mathsf{sk}_i)$ with $\boldsymbol{\mu}_i = \{\mu_j\}_{j \in \psi^{-1}(i)}$ to prove that the partial decryption was honestly computed. In that case, return $\boldsymbol{\mu}_i' = (\boldsymbol{\mu}_i, \pi_i)$.

$\mathsf{PartVerify}(\mathsf{pp}, \mathsf{pk}', \mathsf{ct} = (C_0, C_1, \pi), \boldsymbol{\mu}_i' = (\boldsymbol{\mu}_i, \pi_i))$**:** Given a partial decryption $\boldsymbol{\mu}_i$ from server i, verify it by running $\mathsf{V}_{\log}^i(\mathsf{crs}_{\log}, (C_0, \{v_j, \mu_j\}_{j \in \psi^{-1}(i)}), \pi_i)$ and $\mathsf{V}(\mathsf{crs}, C_0, \pi, \mathsf{lbl})$. Return 1 if both returned 1, and 0 otherwise.

The CCA security of the TPKE is not affected by this augmentation as the additional arguments of membership for the languages $\mathcal{L}_i^{\log}$ are always run honestly or simulated on true statements in the reductions. Nevertheless, such a strategy would require assuming reliable erasure of the decryption-servers' tape for the indistinguishable transition to hold under adaptive corruption. However, by relying on lemma C.1, the statistical witness indistinguishability is enough as none of the uncorrupted decryption parties' bits of $s_j \bmod N^\zeta$ is revealed in partial decryptions but with negligible probability. Therefore, we can still rely on the fact that the secret key x keeps its entropy until $x \bmod N^\zeta$ is used to emulate the challenge ciphertext in Equation (5) of the security proof.

Theorem C.2. *Let* $\mathsf{TPKE}' = (\mathsf{KeyGen}', \mathsf{Encrypt}, \mathsf{PartDec}', \mathsf{PartVerify}, \mathsf{Combine})$ *with* $\phi(\boldsymbol{\mu}_i, \pi_i) := \boldsymbol{\mu}_i$*. Then,* TPKE' *is robust (according to definition 2.17).*

Proof. Let $\mathcal{A}$ be a PPT adversary against the robustness of TPKE'. Given $(\mathsf{pp}, \mathsf{pk}', \mathsf{sk}_1, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{KeyGen}'(1^\lambda, \mathbb{A})$, the goal of the adversary is to build of forgery $(\mathsf{ct} = (C_0, C_1, \pi), (\boldsymbol{\mu}_i, \pi_i), i)$, which means that $\boldsymbol{\mu}_i$ is not in the range of $\mathsf{PartDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct} := (C_0, C_1, \pi))$ (since $\phi \circ \mathsf{PartDec}' = \mathsf{PartDec}$) while we also have $\mathsf{PartVerify}(\mathsf{pp}, \mathsf{pk}', \mathsf{ct}, (\boldsymbol{\mu}_i, \pi_i)) = 1$.

We consider a sequence of games where, for each i, we call W_i the event that the adversary wins in Game_i.

Game$_0$: This is the real game of robustness as recalled above. More precisely, $\boldsymbol{\mu}_i \neq C_0^{2 \cdot \mathsf{sk}_i} \bmod N^{\zeta+1}$ but $\mathsf{V}_{\log}^i(\mathsf{crs}_{\log}, (C_0, \{v_j, \mu_j\}_{j \in \psi^{-1}(i)}), \pi_i) = 1$ and $\mathsf{V}(\mathsf{crs}, C_0, \pi, \mathsf{lbl}) = 1$. By definition $\Pr[\mathsf{Expt}_{\mathcal{A}, \mathsf{TPKE}'}^{\mathrm{robust}}(1^\lambda) = 1] = \Pr[W_0]$.

Game$_1$: This game is like the previous game except that we keep the factorization of N during the generation of pk. Moreover, we no more consider $\mathcal{A}$ to be successful if it outputs a forgery such that $C_0^{2p'q'} \bmod N^{\zeta+1} \neq 1$. Clearly, $|\Pr[W_1] - \Pr[W_0]| \leq \mathbf{Adv}^{\mathsf{OTSS}}(\lambda)$ since producing a valid ciphertext such that $C_0 \notin \mathcal{L}^{\mathsf{DCR}}$ would contradict the soundness (and thus the one-time simulation-soundness) of the NIZK argument system Π^{OTSS}.

Game$_2$: In this game we only add yet another restriction on the success of the adversary. We also reject a forgery that would have been accepted in the previous game if there is some $j \in \psi^{-1}(i)$ such that $\mu_j \neq C_0^{2s_j} \bmod N^{\zeta+1}$. Since a forgery that would be rejected in this game but accepted in the previous game implies $(C_0, \{v_j, \mu_j\}_{j\in\psi^{-1}(i)}) \notin \mathcal{L}_i^{\log}$ while π_i is valid, $|\Pr[W_2] - \Pr[W_1]|$ is negligible by the soundness of $\Pi_{\log}^i$.

In Game_2, we do not know whether the adversary uses sk_i to compute π_i. Indeed, for $j \in \psi^{-1}(i)$ it might be the case that $\mathcal{A}$ runs $\mathsf{P}_{\log}^i$ with the witness $s_j + p'q'$ or even manages to build the argument without any explicit witness. However, we know that

$$\mu_j = C_0^{2[s_j \bmod p'q']}$$

holds over $\mathbb{Z}^*_{N^{\zeta+1}}$ for all these indexes as $\mathrm{ord}(C_0^2)$ divides $p'q'$. But, independently of how the argument π_i is performed, $\boldsymbol{\mu}_i$ is equal to the honest execution of (the deterministic) $\mathsf{PartDec}$. Consequently, so is the output of $\phi \circ \mathsf{PartDec}'$. We conclude that $\Pr[W_2] = 0$. □

D Achieving Consistency for the LWE-Based Construction

In order to achieve consistency, we need a non-interactive argument showing that partial decryptions are consistent with public commitments to the secret key shares. To obtain such an argument, we apply the $\mathsf{BadChallenge}$ function paradigm to a trapdoor Σ-protocol showing the validity of partial decryptions.

We note that a trapdoor Σ-protocol for the same language can be obtained by applying a generic construction due to Ciampi *et al.* [29]. Recall that the construction of [29] turns any standard Σ-protocol with binary challenges into a trapdoor Σ-protocol by having the prover encrypt the two possible responses and send the encrypted responses along with its first message. Here, we provide a direct construction that avoids the overhead of encrypting the two possible responses. In addition, our application to consistency requires the protocol to be statistically witness indistinguishable (we do not rely on NIZK simulation since it would imply to assume erasures at the servers). In this case, using the transformation of [29] would require the prover to encrypt its two responses using a lossy encryption scheme in lossy mode to ensure statistical WI. With the protocol in this section, we can avoid the additional encryption layer.

D.1 A Trapdoor Σ-Protocol Showing Correctness of Partial Decryptions of the LWE-Based TPKE Scheme

The trapdoor Σ-protocol presented in this section makes use of rejection sampling techniques, which rely on a core technical lemma from [71,72].

Lemma D.1 ([72, Th. 4.6]). *Let V be a subset of $\mathbb{Z}^m$ in which all elements have norms less than T, let σ be a real number such that $\sigma = \omega(T\sqrt{\log m})$, and*

$h : V \to \mathbb{R}$ be a probability distribution. Then, there exists a real number M such that the distribution of the following algorithm $\mathcal{A}$:

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

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

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

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

In order to achieve consistency in the LWE-based scheme of Section 5, the Setup algorithm additionally computes

$$\mathbf{V}_{i,\tau} = \mathbf{A} \cdot \mathbf{R}^{\top}_{\tau,\psi^{-1}(i)} \in \mathbb{Z}_q^{n \times d_i} \qquad \forall (i,\tau) \in [\ell] \times [L],$$

so that each server $i \in [\ell]$ is assigned a verification key $\mathsf{vk}_i = \{\mathbf{V}_{i,\tau}\}_{\tau\in[L]}$, which will serve as a commitment to the secret key share $\mathsf{sk}_i = \{\mathbf{R}_{\tau,\psi^{-1}(i)}\}_{\tau\in[L]}$. In order to enforce consistency, each partial decryption will come with a NIZK argument of consistency with the matrices $\{\mathbf{R}_{\tau,\psi^{-1}(i)}\}_{\tau\in[L]}$ hidden in vk_i. Such a NIZK/NIWI argument can be obtained from the trapdoor Σ-protocol described hereafter.

Consider a ciphertext $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$. We will give a trapdoor Σ-protocol for the language

$$\begin{aligned}\mathcal{L}^{\mathrm{dec}}_{\mathsf{zk}} := \Big\{ \big(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{n\times d_i} \times \mathbb{Z}_q^{d_i} \;\mid\; \exists \mathbf{R}_{\tau,\psi^{-1}(i)} \in \mathbb{Z}^{d_i\times m}, \mathbf{e}'_{i,\tau} \in \mathbb{Z}^{d_i} :\\ \|\mathbf{e}'_{i,\tau}\| \le B_e \quad \wedge \quad \max_{j\in[m]} \|\big((\mathbf{R}^{\top}_{\tau,\psi^{-1}(i)})_j\big)\| \le B_r \quad \forall j \in [m]\\ \wedge \quad \left[\frac{\mathbf{V}_{i,\tau}}{\boldsymbol{\mu}^{\top}_{i,\tau}}\right] = \left[\frac{\mathbf{A}}{\mathbf{c}_0^{\top}}\right] \cdot \mathbf{R}^{\top}_{\tau,\psi^{-1}(i)} + \left[\frac{\mathbf{0}^n}{\mathbf{e}'^{\top}_{i,\tau}}\right] \quad \Big\}, \quad (21)\end{aligned}$$

where $B_r < B_e \le \beta_{\mathsf{s}} q\sqrt{d_i}$. To ease the description, let us define matrices

$$\mathbf{A}_{\mathsf{ct}} = \left[\begin{array}{c|c} \mathbf{A} & \mathbf{0}^n \\ \hline \mathbf{c}_0^{\top} & 1 \end{array}\right] \in \mathbb{Z}_q^{(n+1)\times(m+1)}$$

and

$$\mathbf{A}_{\mathrm{dec}} = \mathbf{I}_{d_i} \otimes \mathbf{A}_{\mathsf{ct}} \in \mathbb{Z}_q^{d_i(n+1)\times d_i(m+1)}. \quad (22)$$

Then, let

$$\mathbf{R}_{\tau,\psi^{-1}(i)}^{\top} = [\mathbf{r}_{i,\tau,1} \mid \ldots \mid \mathbf{r}_{i,\tau,d_i}] \in \mathbb{Z}^{m \times d_i}, \qquad \mathbf{V}_{i,\tau} = [\mathbf{v}_{i,\tau,1} \mid \ldots \mid \mathbf{v}_{i,\tau,d_i}] \in \mathbb{Z}_q^{n \times d_i}.$$

We also define

$$\bar{\mathbf{v}}_{i,\tau,k}^{\top} = \big(\mathbf{v}_{i,\tau,k}^{\top} \mid \boldsymbol{\mu}_{i,\tau}[k]\big)^{\top} \in \mathbb{Z}^{n+1}, \qquad \mathbf{w}_{i,\tau,k}^{\top} = \big(\mathbf{r}_{i,\tau,k}^{\top} \mid \mathbf{e}'_{i,\tau}[k]\big)^{\top} \in \mathbb{Z}^{m+1} \quad (23)$$

for each $k \in [d_i]$, as well as

$$\bar{\mathbf{v}}_{i,\tau}^{\top} = [\bar{\mathbf{v}}_{i,\tau,1}^{\top} \mid \ldots \mid \bar{\mathbf{v}}_{i,\tau,d_i}^{\top}] \in \mathbb{Z}_q^{d_i(n+1)}. \quad (24)$$

To prove membership of $\mathcal{L}_{\mathsf{zk}}^{\mathrm{dec}}$, a natural idea is to prove the existence of a small-norm $\mathbf{w}_{i,\tau}^{\top} = [\mathbf{w}_{i,\tau,1}^{\top} \mid \ldots \mid \mathbf{w}_{i,\tau,d_i}^{\top}]^{d_i(m+1)}$ such that

$$\bar{\mathbf{v}}_{i,\tau} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{w}_{i,\tau} \mod q. \quad (25)$$

We now observe that, if the pair $(\mathbf{c}_0, \mathbf{c}_1)$ belongs to the language $\mathcal{L}_{\mathsf{sound}}$ of Section 5 (i.e., if there exist vectors $\mathbf{s} \in \mathbb{Z}_q^n$ and $\mathbf{e}_0 \in \mathbb{Z}^m$ such that $\|\mathbf{e}_0\| \leq \gamma\tilde{d}$ and $\mathbf{c}_0 = \mathbf{A}^{\top} \cdot \mathbf{s} + \mathbf{e}_0$), then $\mathcal{L}_{\mathsf{zk}}^{\mathrm{dec}}$ is contained in the language

$$\begin{aligned}
\mathcal{L}_{\mathsf{sound}}^{\mathrm{dec}} := \Big\{ & \big(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{n \times d_i} \times \mathbb{Z}_q^{d_i} \mid \exists \mathbf{s} \in \mathbb{Z}_q^n \ : \ \|\mathbf{c}_0 - \mathbf{A}^{\top} \cdot \mathbf{s}\| \leq \gamma\tilde{d} \\
& \wedge \quad \forall k \in [d_i] \ : \ \big|([-\mathbf{s}^{\top} \mid 1] \otimes \boldsymbol{e}_k^{\top}) \cdot \bar{\mathbf{v}}_{i,\tau}\big| \leq B^* \quad \Big\}, \\
= \Big\{ & \big(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{n \times d_i} \times \mathbb{Z}_q^{d_i} \ \mid \ \exists \mathbf{s} \in \mathbb{Z}_q^n, \ \mathbf{y}_{i,\tau} \in \mathbb{Z}^{d_i} \ : \\
& \|\mathbf{c}_0 - \mathbf{A}^{\top} \cdot \mathbf{s}\| \leq \gamma\tilde{d} \ \wedge \ \|\mathbf{y}_{i,\tau}\|_{\infty} \leq B^* \\
& \wedge \quad \left[\frac{\mathbf{V}_{i,\tau}}{\boldsymbol{\mu}_{i,\tau}^{\top}}\right] = \left[\frac{\mathbf{V}_{i,\tau}}{\mathbf{s}^{\top} \cdot \mathbf{V}_{i,\tau} + \mathbf{y}_{i,\tau}^{\top}}\right] \quad \Big\},
\end{aligned} \quad (26)$$

with $B^* > 2\sigma_{\mathrm{dec}} d_i \sqrt{(\gamma^2\tilde{d}^2 + 1)(m+1)}$ and where $\boldsymbol{e}_k \in \mathbb{Z}^{d_i}$ stands for the k-th unit vector while $\bar{\mathbf{v}}_{i,\tau} \in \mathbb{Z}_q^{d_i(n+1)}$ is the encoding of $[\mathbf{V}_{i,\tau}^{\top} \mid \boldsymbol{\mu}_{i,\tau}] \in \mathbb{Z}_q^{d_i \times (n+1)}$ defined in (23)-(24).

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

$\mathsf{Gen}_{\mathcal{L}}(\mathsf{par}, \mathcal{L}^{\mathrm{dec}})$: Takes in global parameters par and the description of a language $\mathcal{L}^{\mathrm{dec}} = (\mathcal{L}_{\mathsf{zk}}^{\mathrm{dec}}, \mathcal{L}_{\mathsf{sound}}^{\mathrm{dec}})$ specifying a real $B^* > 0$ such that $B^* > 2\sigma_{\mathrm{dec}} d_i \sqrt{(\gamma^2\tilde{d}^2 + 1)(m+1)}$, and a matrix $\mathbf{A}_{\mathrm{dec}}$ from the distribution (22). It defines the language-dependent $\mathsf{crs}_{\mathcal{L}} = \{\mathbf{A}_{\mathrm{dec}}, B^*\}$. The global CRS is

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

$\mathsf{TrapGen}(\mathsf{par}, \mathcal{L}, \tau_{\mathcal{L}})$: Given par and a language description $\mathcal{L}^{\mathrm{dec}}$ that specifies $B^* > 0$ satisfying the same constraint as in $\mathsf{Gen}_{\mathcal{L}}$, a matrix $\mathbf{A}_{\mathrm{dec}}$ sampled from the distribution (22), as well as a membership-testing trapdoor $\tau_{\mathcal{L}} = \mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ for $\mathcal{L}^{\mathrm{dec}}_{\mathsf{sound}}$, output $\mathsf{crs}_{\mathcal{L}} = \{\mathbf{A}_{\mathrm{dec}}, B^*\}$. The global CRS is

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

and the trapdoor $\tau_{\Sigma} = \mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$.

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

$$\mathbf{x} = \big(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{n \times d_i} \times \mathbb{Z}_q^{d_i},$$

P and V encode $\mathbf{x}$ as a vector $\bar{\mathbf{v}}_{i,\tau} = [\bar{\mathbf{v}}_{i,\tau,1}^{\top} \mid \ldots \mid \bar{\mathbf{v}}_{i,\tau,d_i}^{\top}] \in \mathbb{Z}_q^{d_i(n+1)}$ of the form described in (23)-(24). Then, they conduct the following steps.

1. P encodes the witness $\mathbf{w} = (\mathbf{R}_{\tau,\psi^{-1}(i)}, \mathbf{e}'_{i,\tau}) \in \mathbb{Z}^{d_i \times m} \times \mathbb{Z}^{d_i}$ as a short vector $\mathbf{w}_{i,\tau} = [\mathbf{w}_{i,\tau,1}^{\top} \mid \ldots \mid \mathbf{w}_{i,\tau,d_i}^{\top}]^{d_i(m+1)}$ satisfying (25). Then, it samples a Gaussian vector $\mathbf{r}_w \hookleftarrow D_{\mathbb{Z}^{d_i(m+1)}, \sigma_{\mathrm{dec}}}$. It computes the following which is sent to V:
$$\mathbf{a} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{r}_w \ \in \mathbb{Z}_q^{d_i(n+1)}.$$
2. V sends a random challenge $\mathsf{Chall} \in \{0,1\}$ to P.
3. P computes $\mathbf{z}_w = \mathbf{r}_w + \mathsf{Chall} \cdot \mathbf{w}_{i,\tau} \ \in \mathbb{Z}^{d_i(m+1)}$. It sends $\mathbf{z}_w$ to V with probability $\theta = \min\Big(\frac{D_{\mathbb{Z}^{d_i(m+1)}, \sigma_{\mathrm{dec}}}(\mathbf{z}_w)}{M \cdot D_{\mathbb{Z}^{d_i(m+1)}, \sigma_{\mathrm{dec}}, \mathsf{Chall} \cdot \mathbf{w}_{i,\tau}}(\mathbf{z}_w)}, 1\Big)$, where
$$M = e^{12/\log(d_i(m+1)) + 1/(2\log^2(d_i(m+1)))}.$$
With probability $1-\theta$, P aborts.
4. Given $\mathbf{z}_w \in \mathbb{Z}^{d_i(m+1)}$, V checks if $\|\mathbf{z}_w\| \le \sigma_{\mathrm{dec}}\sqrt{d_i(m+1)}$ and
$$\mathbf{a} + \mathsf{Chall} \cdot \bar{\mathbf{v}}_{i,\tau} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{z}_w \bmod q. \tag{27}$$

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

$\mathsf{BadChallenge}\big(\mathsf{par}, \tau_{\Sigma}, \mathsf{crs}, \mathbf{x}, \mathbf{a}\big)$: Given the trapdoor $\tau_{\Sigma} = \mathbf{T}_{\mathbf{A}} \in \mathbb{Z}^{m \times m}$ and a statement $\mathbf{x} = \big(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}\big) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{d_i \times n} \times \mathbb{Z}_q^{d_i}$, parse the first prover message as $\mathbf{a} \in \mathbb{Z}_q^{d_i(n+1)}$ and conduct the following steps.

1. Using $\mathbf{T}_{\mathbf{A}}$, compute the unique $\mathbf{s} \in \mathbb{Z}_q^n$ satisfying $\|\mathbf{c}_0 - \mathbf{A}^{\top} \cdot \mathbf{s}\| \le \gamma\tilde{d}$. If no such $\mathbf{s}$ exists, return $\perp$.
2. If there exists $k \in [d_i]$ and $d \in \{0,1\}$ such that
$$\big|([-\mathbf{s}^{\top} \mid 1] \otimes \boldsymbol{e}_k^{\top}) \cdot (\mathbf{a} + d \cdot \bar{\mathbf{v}}_{i,\tau}) \bmod q\big| > B^*/2, \tag{28}$$
then return $\mathsf{Chall} = 1 - d$. Otherwise, return $\mathsf{Chall} = \perp$.

Lemma D.2. *The above construction is a trapdoor Σ-protocol for $\mathcal{L}^{\mathrm{dec}}$ if we set $\sigma_{\mathrm{dec}} \geq \log(d_i(m+1)) \cdot \beta_{\mathsf{s}} q \cdot \sqrt{d_i(m+1)}$ and $B^* > 2\sigma_{\mathrm{dec}} \cdot d_i \sqrt{(\gamma^2 \tilde{d}^2 + 1)(m+1)}$.*

Proof. By construction, the protocol has completeness with probability negligibly close to θ. We now prove that it satisfies special soundness, correctness of $\mathsf{BadChallenge}$ and special zero-knowledge.

Special Soundness. Let us assume that $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}$ (so that there exists $\mathbf{s} \in \mathbb{Z}_q^n$ such that $\|\mathbf{c}_0 - \mathbf{A}^\top \mathbf{s}\| \leq \gamma \tilde{d}$) and, for a given $\mathbf{a} \in \mathbb{Z}_q^{d_i(n+1)}$, there exist two valid responses $\mathbf{z}_{w,b} \in \mathbb{Z}^{d_i(m+1)}$ with $\|\mathbf{z}_{w,b}\| \leq \sigma_{\mathrm{dec}} \cdot \sqrt{d_i(m+1)}$ for each $b \in \{0,1\}$ and such that

$$\mathbf{a} + b \cdot \bar{\mathbf{v}}_{i,\tau} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{z}_{w,b} \mod q. \tag{29}$$

Subtracting them yields

$$\bar{\mathbf{v}}_{i,\tau} = \mathbf{A}_{\mathrm{dec}} \cdot (\mathbf{z}_{w,1} - \mathbf{z}_{w,0}) \bmod q,$$

where $\|\mathbf{z}_{w,1} - \mathbf{z}_{w,0}\| \leq 2\sigma_{\mathrm{dec}} \cdot \sqrt{d_i(m+1)}$. Moreover, we also have

$$([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot \bar{\mathbf{v}}_{i,\tau} = ([\mathbf{e}_0^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot (\mathbf{z}_{w,1} - \mathbf{z}_{w,0}) \qquad \forall k \in [d_i],$$

which implies that $(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}) \in \mathcal{L}^{\mathrm{dec}}_{\mathsf{sound}}$ since we assumed $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}$ and

$$([\mathbf{e}_0^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot (\mathbf{z}_{w,1} - \mathbf{z}_{w,0}) < 2\sigma_{\mathrm{dec}} \sqrt{d_i(\gamma^2 \tilde{d}^2 + 1)(m+1)} < B^*.$$

Correctness of BadChallenge. We now show that $\mathsf{BadChallenge}$ provides the correct output. Assuming that $(\mathbf{c}_0, \mathbf{c}_1) \in \mathcal{L}_{\mathsf{sound}}$, the trapdoor $\mathbf{T}_{\mathbf{A}}$ allows computing the vector $\mathbf{s} \in \mathbb{Z}_q^n$ such that $\|\mathbf{c}_0 - \mathbf{A}^\top \mathbf{s}\| \leq \gamma \tilde{d}$. We also assume $(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}) \notin \mathcal{L}^{\mathrm{dec}}_{\mathsf{sound}}$. For a given $\mathbf{a} \in \mathbb{Z}_q^{d_i(n+1)}$, we cannot simultaneously have

$$\left|([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot (\mathbf{a} + d \cdot \bar{\mathbf{v}}_{i,\tau}) \bmod q\right| \leq B^*/2 \qquad \forall k \in [d_i],\ \forall d \in \{0,1\}$$

as this would imply

$$\begin{aligned}
&\left| ([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot \bar{\mathbf{v}}_{i,\tau} \bmod q \right| \\
&= \left| \left(([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot (\mathbf{a} + \bar{\mathbf{v}}_{i,\tau}) \bmod q\right) + \left(([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot (-\mathbf{a}) \bmod q\right) \right| \\
&\leq \left|([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot (\mathbf{a} + \bar{\mathbf{v}}_{i,\tau}) \bmod q\right| + \left|([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot (-\mathbf{a}) \bmod q)\right| \\
&\leq B^*,
\end{aligned}$$

for all $k \in [d_i]$, which would imply $(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}) \in \mathcal{L}^{\mathrm{dec}}_{\mathsf{sound}}$. If the statement is false and $(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}) \notin \mathcal{L}^{\mathrm{dec}}_{\mathsf{sound}}$, there must exist $k \in [d_i]$ and $d \in \{0,1\}$ such that inequality (28) holds.

Now, let us first assume that

$$\left|([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot \mathbf{a} \bmod q\right| > B^*/2.$$

The Cauchy-Schwartz inequality then implies that no sufficiently short vector $\mathbf{z}_{w,0} \in \mathbb{Z}^{d_i(m+1)}$ can satisfy the verification equation (29) for the challenge $b = 0$ since $B^*/2 > \sigma_{\mathrm{dec}}\sqrt{d_i(\gamma^2\tilde{d}^2+1)(m+1)}$. Hence, if a bad challenge exists at all, it can only be $b = 1$.

For the same reason, the condition

$$\exists k \in [d_i] \;:\; \left|([-\mathbf{s}^\top \mid 1] \otimes \boldsymbol{e}_k^\top) \cdot (\mathbf{a} + \bar{\mathbf{v}}_{i,\tau}) \bmod q\right| > B^*/2,$$

is incompatible with the existence of a valid response $\mathbf{z}_{w,1} \in \mathbb{Z}^{d_i(m+1)}$ satisfying (29) for the challenge $b = 1$.

Special Zero-Knowledge. The special ZK property can be shown by applying Lemma D.1. Given a statement

$$\mathbf{x} = \left(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}\right) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{d_i \times n} \times \mathbb{Z}_q^{d_i},$$

and a challenge $\mathsf{Chall}^* \in \{0,1\}$, the simulator first samples $\mathbf{z}_w^* \hookleftarrow D_{\mathbb{Z}^{d_i(m+1)},\sigma_{\mathrm{dec}}}$ and computes

$$\mathbf{a}^* = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{z}_w^* - \mathsf{Chall} \cdot \bar{\mathbf{v}}_{i,\tau} \bmod q.$$

It outputs $(\mathbf{a}^*, \mathsf{Chall}^*, \mathbf{z}_w^*)$ with probability $1/M$. By construction, the triple $(\mathbf{a}^*, \mathsf{Chall}^*, \mathbf{z}_w^*)$ satisfies the verification conditions w.h.p. We show that it is statistically indistinguishable from a real transcript. If $\left(\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}\right) \in \mathcal{L}_{\mathsf{zk}}^{\mathrm{dec}}$, there exists $\mathbf{w}_{i,\tau} \in \mathbb{Z}^{d_i(m+1)}$ of norm $\|\mathbf{w}_{i,\tau}\| \leq \beta_{\mathsf{s}} q \cdot \sqrt{d_i(m+1)}$ such that $\bar{\mathbf{v}}_{i,\tau} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{w}_{i,\tau} \mod q$. The distribution of $\mathbf{z}_w \in \mathbb{Z}^{d_i(m+1)}$ in a real transcript is thus $D_{\mathbb{Z}^{d_i(m+1)},\sigma_{\mathrm{dec}},\mathsf{Chall}\cdot\mathbf{w}_{i,\tau}}$. Then, by Lemma D.1 and our choice of $\sigma_{\mathrm{dec}} \geq \log(d_i(m+1)) \cdot \beta_{\mathsf{s}} q \cdot \sqrt{d_i(m+1)}$, the distribution of the simulated $\mathbf{z}_w^*$ is within statistical distance $2^{-100}/M$ from that of a real non-aborting transcript. Finally, in both the real protocol and the simulation, the statement $\mathbf{x}$, the challenge Chall and the response $\mathbf{z}_w$ uniquely determine $\mathbf{a}$. □

Witness-Indistinguishability. As explained in Section C.1, to obtain consistency without assuming erasures on behalf of the servers, we rely on the statistical WI of the protocol. We prove this property in the following lemma.

Lemma D.3. *The trapdoor Σ-protocol presented above is statistically witness-indistinguishable, assuming the same bounds as in Lemma D.2, i.e., $\sigma_{\mathrm{dec}} \geq \log(d_i(m+1)) \cdot \beta_{\mathsf{s}} q \cdot \sqrt{d_i(m+1)}$ and $B^* > 2\sigma_{\mathrm{dec}} d_i \sqrt{(\gamma^2\tilde{d}^2+1)(m+1)}$.*

Proof. Let $\mathbf{x} = (\mathbf{c}_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}_{i,\tau}) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^{d_i \times n} \times \mathbb{Z}_q^{d_i}$ be a statement interpreted as $\bar{\mathbf{v}}_{i,\tau} \in \mathbb{Z}_q^{d_i(n+1)}$, and let $\mathbf{w}_{i,\tau}^{(1)}, \mathbf{w}_{i,\tau}^{(2)} \in \mathbb{Z}^{d_i(m+1)}$ be two different witnesses for $\mathbf{x}$, *i.e.*, we have:

$$\bar{\mathbf{v}}_{i,\tau} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{w}_{i,\tau}^{(1)} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{w}_{i,\tau}^{(2)} \quad \text{and} \quad \|\mathbf{w}_{i,\tau}^{(1)}\|, \|\mathbf{w}_{i,\tau}^{(2)}\| \leq \beta_{\mathsf{s}} q \cdot \sqrt{d_i(m+1)}.$$

To prove statistical WI, we will demonstrate that if no abort occurs, then the following 2 distributions are statistically indistinguishable:

$$D_1 := \{(\mathbf{a} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{r}_w, \mathsf{Chall},\ \mathbf{z}_w^{(1)} = \mathbf{r}_w + \mathsf{Chall} \cdot \mathbf{w}_{i,\tau}^{(1)}) \mid \\ \mathbf{r}_w \hookleftarrow D_{\mathbb{Z}^{d_i(m+1)},\sigma_{\mathrm{dec}}}, \mathsf{Chall} \leftarrow \mathsf{V}(\mathsf{crs}, \mathbf{x}, \mathbf{a})\},$$

and

$$D_2 := \{(\mathbf{a} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{r}_w, \mathsf{Chall},\ \mathbf{z}_w^{(2)} = \mathbf{r}_w + \mathsf{Chall} \cdot \mathbf{w}_{i,\tau}^{(2)}) \mid \\ \mathbf{r}_w \hookleftarrow D_{\mathbb{Z}^{d_i(m+1)},\sigma_{\mathrm{dec}}}, \mathsf{Chall} \leftarrow \mathsf{V}(\mathsf{crs}, \mathbf{x}, \mathbf{a})\}.$$

Since the distributions are restricted to the same $\mathbf{a}$, the distribution of Chall is independent of the witnesses. Therefore, it can be seen that when $\mathsf{Chall} = 0$, the 2 distributions are identical, and we have $\Delta(D_1, D_2) = \Delta(D_1', D_2')$, where distributions D_1', D_2' are defined as

$$D_1' := \{(\mathbf{a} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{r}_w,\ \mathbf{z}_w^{(1)} = \mathbf{r}_w + \mathbf{w}_{i,\tau}^{(1)}) \mid \mathbf{r}_w \hookleftarrow D_{\mathbb{Z}^{d_i(m+1)},\sigma_{\mathrm{dec}}}\},$$

$$D_2' := \{(\mathbf{a} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{r}_w,\ \mathbf{z}_w^{(2)} = \mathbf{r}_w + \mathbf{w}_{i,\tau}^{(2)}) \mid \mathbf{r}_w \hookleftarrow D_{\mathbb{Z}^{d_i(m+1)},\sigma_{\mathrm{dec}}}\}.$$

To this end, let us consider the following distribution:

$$D := \{(\mathbf{a}' = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{z} - \bar{\mathbf{v}}_{i,\tau},\ \mathbf{z}) \mid \mathbf{z} \hookleftarrow D_{\mathbb{Z}^{d_i(m+1)},\sigma_{\mathrm{dec}}}\}.$$

By Lemma D.1, in the non-aborting case, for a pair $(\mathbf{a}, \mathbf{z}_w^{(1)})$ from D_1', the distribution of $\mathbf{z}_w^{(1)}$ is statistically close to $D_{\mathbb{Z}^{d_i(m+1)},\sigma_{\mathrm{dec}}}$, while $\mathbf{a} = \mathbf{A}_{\mathrm{dec}} \cdot \mathbf{z}_w^{(1)} - \bar{\mathbf{v}}_{i,\tau}$ is uniquely determined by $\mathbf{A}_{\mathrm{dec}}, \bar{\mathbf{v}}_{i,\tau}$ and $\mathbf{z}_w^{(1)}$. Thus, D_1' is statistically close to D, i.e., $\Delta(D_1', D) \leq \mathsf{negl}(\lambda)$.

By the same argument, we also have $\Delta(D_2', D) \leq \mathsf{negl}(\lambda)$ and thus

$$\Delta(D_1, D_2) = \Delta(D_1', D_2') \leq \Delta(D_1', D) + \Delta(D_2', D) \leq \mathsf{negl}(\lambda),$$

which proves the claim.

□

Parallel repetitions and non-interactive versions. We note that the trapdoor Σ-protocol presented in this section works with binary challenge, and hence, admits a soundness error of $1/2$. This error can be made negligibly small via $O(\lambda)$ parallel repetitions. Then, using the transformation suggested in [65], we can obtain a non-interactive protocol in the standard model. For each $i \in [\ell]$ and each $\tau \in [L]$, we will denote the resulting protocol as $\Pi_{i,\tau}^{\mathrm{lwe}}$, with prover $\mathsf{P}_{i,\tau}^{\mathrm{lwe}}$ and verifier $\mathsf{V}_{i,\tau}^{\mathrm{lwe}}$. A global CRS, denoted by $\mathsf{crs}^{\mathrm{lwe}}$, can be used for all $\ell \cdot L$ protocols. We will use these notations in the next section.

D.2 Modifying the LWE-Based Scheme to Achieve Consistency

Let $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{PartDec}, \mathsf{Combine})$ be the Dual-Regev-based CCA2-secure TPKE scheme from Section 5. We use the $\ell \cdot L$ protocols $\Pi^{\mathrm{lwe}}_{i,\tau}$, to transform the scheme into a consistent TPKE. To this end, we augment Keygen with the global CRS $\mathsf{crs}^{\mathrm{lwe}}$ of the $\Pi^{\mathrm{lwe}}_{i,\tau}$'s, augment $\mathsf{PartDec}$ with the $\mathsf{P}^{\mathrm{lwe}}_{i,\tau}$'s, and add algorithm $\mathsf{PartVerify}$ based on the $\mathsf{V}^{\mathrm{lwe}}_{i,\tau}$'s. More concretely, the modifications are as follows.

$\mathsf{Keygen}'(1^\lambda, \mathbb{A})$**:** Run $(\mathsf{pp}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{Keygen}(1^\lambda, \mathbb{A})$. Define the language $\mathcal{L}^i_{\mathrm{lwe}} = (\mathcal{L}^i_{\mathsf{zk}}, \mathcal{L}^i_{\mathsf{sound}})$ for every $i \in [\ell]$ using pp and pk[15], and let $\mathsf{crs}^{\mathrm{lwe}}$ be their global CRS. Update the public key to

$$\mathsf{pk}' = (\mathbf{A}, \mathbf{U}, \mathsf{crs}, \mathsf{crs}^{\mathrm{lwe}}, \{\mathbf{V}_{i,\tau} = \mathbf{A} \cdot \mathbf{R}^\top_{\tau,\psi^{-1}(i)}, (i,\tau) \in [\ell] \times [L]\}),$$

and return $(\mathsf{pp}, \mathsf{pk}', \mathsf{sk}_1, \ldots \mathsf{sk}_\ell)$.

$\mathsf{PartDec}'(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi}))$**:** Run $\boldsymbol{\mu}_i = \{\boldsymbol{\mu}_{i,\tau}\}_{\tau \in [L]} \leftarrow \mathsf{PartDec}(\mathsf{pp}, \mathsf{sk}_i, \mathsf{ct})$, then, for each $\tau \in [L]$, generate

$$\pi_{i,\tau} = \mathsf{P}^{\mathrm{lwe}}_{i,\tau}(\mathsf{crs}^{\mathrm{lwe}}, \boldsymbol{\mu}_{i,\tau}, \mathbf{R}_{\tau,\psi^{-1}(i)}).$$

Return $\boldsymbol{\mu}'_i = \{\boldsymbol{\mu}_{i,\tau}, \pi_{i,\tau}\}_{\tau \in [L]}$.

$\mathsf{PartVerify}(\mathsf{pp}, \mathsf{pk}, \mathsf{ct}, \boldsymbol{\mu}'_i = \{\boldsymbol{\mu}_{i,\tau}, \pi_{i,\tau}\}_{\tau \in [L]})$**:** First, check that $\mathsf{ct} = (\mathbf{c}_0, \mathbf{c}_1, \boldsymbol{\pi})$ is a valid ciphertext by running $\mathsf{V}(\mathsf{crs}, (\mathbf{c}_0, \mathbf{c}_1), \boldsymbol{\pi})$. If it is not, return 0.

For every $\tau \in [L]$, if $\mathsf{V}^{\mathrm{lwe}}_{i,\tau}(\mathsf{crs}^{\mathrm{lwe}}, \boldsymbol{\mu}_{i,\tau}, \pi_{i,\tau}) = 0$, then return 0. If all the L arguments are valid, return 1.

The CCA security of the TPKE scheme is not affected by the above augmentation, as the additional NIZK arguments are always run honestly or simulated on true statements in the reductions. However, as discussed in Section C.2, for the indistinguishable transition to hold under adaptive corruption, we would need to assume reliable erasure of the decryption-servers' tape. Instead of relying on this additional assumption, we use the statistical WI property established in Lemma D.3. It is sufficient for the reductions to go through, as none of the bits of the key share sk_i is revealed in partial decryptions, except for a negligible probability.

Theorem D.4. *Let* $\mathsf{TPKE}' = (\mathsf{KeyGen}', \mathsf{Encrypt}, \mathsf{PartDec}', \mathsf{PartVerify}, \mathsf{Combine})$ *be the augmented scheme, Then,* TPKE' *satisfies the consistency requirement in the sense of Definition A.3.*

Proof. Let $\mathcal{A}$ be a PPT adversary against the consistency of TPKE'. We will show that the advantage of $\mathcal{A}$ in the experiment $\mathsf{Expt}^{\mathrm{consist}}_{\mathcal{A},\mathsf{TPKE}'}(1^\lambda)$ from Definition A.3 is negligible, based on the soundness of Π^{OTSS} and of $\Pi^{\mathrm{lwe}}_{i,\tau}$, for $(i,\tau) \in [\ell] \times [L]$. To this end, we consider a sequence of games where, for each j, we call W_j the event that the adversary wins in $\mathbf{Game}_j$.

[15] Note that $\mathcal{L}^i_{\mathrm{lwe}}$ is the same for all L sub-protocols corresponding to the i-th server.

Game$_0$: This is the real consistency game. The adversary first chooses an access structure $\mathbb{A}$, then it receives $(\mathsf{pp}, \mathsf{pk}', \mathsf{sk}_1, \ldots, \mathsf{sk}_\ell) \leftarrow \mathsf{KeyGen}'(1^\lambda, \mathbb{A})$ from the challenger. Eventually it outputs a forgery consisting of:

- A ciphertext $\mathsf{ct}^* = (\mathbf{c}_0^*, \mathbf{c}_1^*, \boldsymbol{\pi}^*)$;
- A set $\mathcal{S}_0 \in \mathbb{A}$ along with partial decryptions $\{\mu_i^0, i \in \mathcal{S}_0\}$;
- A set $\mathcal{S}_1 \in \mathbb{A}$ along with partial decryptions $\{\mu_j^1, j \in \mathcal{S}_1\}$.

Assume that $\mathsf{PartVerify}(\mathsf{pk}, \mathsf{ct}^*, \mu_i^b) = 1$ for all $b \in \{0,1\}$ and $i \in \mathcal{S}_b$ and $\mathsf{Combine}(\mathsf{pk}, (\mathcal{S}_0, \{\mu_i^*\}_{i\in\mathcal{S}_0}), \mathsf{ct}^*) \neq \mathsf{Combine}(\mathsf{pk}, (\mathcal{S}_1, \{\mu_j^*\}_{j\in\mathcal{S}_1}), \mathsf{ct}^*)$. By definition, we have

$$\Pr\left[\mathsf{Expt}^{\mathrm{consist}}_{\mathcal{A},\mathsf{TPKE}'}(1^\lambda) = 1\right] = \Pr[W_0].$$

Game$_1$: In this game, we change the generation of matrix $\mathbf{A}$ in the KeyGen'. Instead of sampling $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times m})$, we generate it together with a trapdoor $\mathbf{T_A}$ via $(\mathbf{A}, \mathbf{T_A}) \leftarrow \mathsf{TrapGen}(1^n, 1^m, q)$. The properties of $\mathsf{TrapGen}$ ensure that the distribution of $\mathbf{A}$ is statistically close to $U(\mathbb{Z}_q^{n\times m})$. Therefore, we have $|\Pr[W_1] - \Pr[W_0]| \leq \mathsf{negl}(\lambda)$.

Game$_2$: In this game, we use the trapdoor $\mathbf{T_A}$ to test whether there exists $\mathbf{s} \in \mathbb{Z}_q^n$ satisfying $\|\mathbf{c}_0^* - \mathbf{A}^\top\mathbf{s}\| \leq \gamma\tilde{d}$, and if this is not the case, we no longer consider $\mathcal{A}$ to be successful. Clearly,

$$|\Pr[W_2] - \Pr[W_1]| \leq \mathbf{Adv}^{\mathsf{OTSS}}_{\mathcal{A}}(\lambda) \leq \mathsf{negl}(\lambda),$$

since producing $\mathsf{ct} = (\mathbf{c}_0^*, \mathbf{c}_1^*, \boldsymbol{\pi}^*)$ such that $\mathsf{V}(\mathsf{crs}, (\mathbf{c}_0^*, \mathbf{c}_1^*), \boldsymbol{\pi}^*) = 1$ so that $(\mathbf{c}_0^*, \mathbf{c}_1^*) \notin \mathcal{L}_{\mathsf{sound}}$ would contradict the soundness of the NIZK argument system Π^{OTSS}.

Game$_3$: Recall that, in the previous games, for all $i \in \mathcal{S}_0$ and $j \in \mathcal{S}_1$, we have

$$\mu_i^0 = \{(\boldsymbol{\mu}_{i,\tau}^0, \pi_{i,\tau}^0), \tau \in [L]\}, \quad \mu_j^1 = \{(\boldsymbol{\mu}_{j,\tau}^1, \pi_{j,\tau}^1), \tau \in [L]\},$$

such that all the pairs $(\boldsymbol{\mu}_{i,\tau}^0, \pi_{i,\tau}^0)$, $(\boldsymbol{\mu}_{j,\tau}^1, \pi_{j,\tau}^1)$ pass the verifications of algorithm $\mathsf{PartVerify}$. In particular, we have

$$\mathsf{V}_{i,\tau}^{\mathrm{lwe}}(\mathsf{crs}^{\mathrm{lwe}}, \boldsymbol{\mu}_{i,\tau}^0, \pi_{i,\tau}^0) = 1, \quad \mathsf{V}_{j,\tau}^{\mathrm{lwe}}(\mathsf{crs}^{\mathrm{lwe}}, \boldsymbol{\mu}_{j,\tau}^1, \pi_{j,\tau}^1) = 1,$$

for all $i \in \mathcal{S}_0$, $j \in \mathcal{S}_1$, $\tau \in [L]$.
In this game we add yet another restriction on the success of the adversary. We also reject a forgery that would have been accepted in the previous game if there exists $\mathbf{s} \in \mathbb{Z}_q^n$ satisfying $\|\mathbf{c}_0^* - \mathbf{A}^\top\mathbf{s}\| \leq \gamma\tilde{d}$, but

- Either there exists a pair $(i, \tau) \in \mathcal{S}_0 \times [L]$, for which there does not exist $\mathbf{y}_{i,\tau} \in \mathbb{Z}^{d_i}$ satisfying

$$\|\mathbf{y}_{i,\tau}\|_\infty \leq B^* \quad \wedge \quad (\boldsymbol{\mu}_{i,\tau}^0)^\top = \mathbf{s}^\top \cdot \mathbf{V}_{i,\tau} + \mathbf{y}_{i,\tau}^\top;$$

- Or there exists a pair $(j, \tau) \in \mathcal{S}_1 \times [L]$, for which there does not exist $\mathbf{y}_{j,\tau} \in \mathbb{Z}^{d_j}$ satisfying

$$\|\mathbf{y}_{j,\tau}\|_\infty \leq B^* \quad \wedge \quad (\boldsymbol{\mu}_{i,\tau}^1)^\top = \mathbf{s}^\top \cdot \mathbf{V}_{j,\tau} + \mathbf{y}_{j,\tau}^\top.$$

Note that, the existence of such a pair (i,τ) (resp., (j,τ)) would contradict the soundness of the argument system $\Pi^{\mathrm{lwe}}_{i,\tau}$ (resp., $\Pi^{\mathrm{lwe}}_{j,\tau}$) , since it would imply that $(\mathbf{c}^*_0, \mathbf{V}_{i,\tau}, \boldsymbol{\mu}^0_{i,\tau}) \notin \mathcal{L}^i_{\mathsf{sound}}$ (resp., $(\mathbf{c}^*_0, \mathbf{V}_{j,\tau}, \boldsymbol{\mu}^1_{j,\tau}) \notin \mathcal{L}^j_{\mathsf{sound}}$), yet $\mathsf{V}^{\mathrm{lwe}}_{i,\tau}(\mathsf{crs}^{\mathrm{lwe}}, \boldsymbol{\mu}^0_{i,\tau}, \pi^0_{i,\tau}) = 1$ (resp., $\mathsf{V}^{\mathrm{lwe}}_{j,\tau}(\mathsf{crs}^{\mathrm{lwe}}, \boldsymbol{\mu}^1_{j,\tau}, \pi^1_{j,\tau}) = 1$) at the same time. Therefore, by the soundness of $\Pi^{\mathrm{lwe}}_{i,\tau}$ (resp., $\Pi^{\mathrm{lwe}}_{j,\tau}$), we have

$$\begin{aligned}|\Pr[W_3] - \Pr[W_2]| &\leq \sum_{i\in\mathcal{S}_0,\tau\in[L]} \mathbf{Adv}^{\Pi^{\mathrm{lwe}}_{i,\tau}}_{\mathcal{A}}(\lambda) + \sum_{j\in\mathcal{S}_1,\tau\in[L]} \mathbf{Adv}^{\Pi^{\mathrm{lwe}}_{j,\tau}}_{\mathcal{A}}(\lambda)\\ &\leq \mathsf{negl}(\lambda),\end{aligned}$$

since we have $|\mathcal{S}_0|, |\mathcal{S}_1|, L \in \mathsf{poly}(\lambda)$.

We now demonstrate that the adversary cannot win in $\mathbf{Game}_3$. Indeed, since $\mathcal{S}_0$ and $\mathcal{S}_1$ are authorized sets, by construction, there must exist coefficients $\{\boldsymbol{\lambda}^b_i \in \{-1,0,1\}^{d_i}\}_{i\in\mathcal{S}_b}$, where $b \in \{0,1\}$, such that

$$\sum_{i\in\mathcal{S}_b} \mathbf{V}_{i,\tau} \cdot \boldsymbol{\lambda}^b_i = \mathbf{u}_\tau,$$

where $\mathbf{u}_\tau \in \mathbb{Z}^n_q$ is the τ-th column of $\mathbf{U} = [\mathbf{u}_1 \mid \ldots \mid \mathbf{u}_L] = \mathbf{A}\cdot\mathbf{R}$. Then, based on the norm bounds of $\mathbf{y}_{i,\tau}$, $\mathbf{y}_{j,\tau}$, we obtain that

$$\sum_{i\in\mathcal{S}_b} \left(\boldsymbol{\mu}^b_{i,\tau}\right)^\top \cdot \boldsymbol{\lambda}^b_i = \mathbf{s}^\top \cdot \mathbf{u}_\tau + \bar{y},$$

for some $\bar{y}$ satisfying $|\bar{y}| < B^* \sum_{i\in\mathcal{S}_b} d_i = B^* d_{\mathcal{S}_b}$, and for each $b \in \{0,1\}$. It then follows that

$$\mathsf{Combine}(\mathsf{pk}, (\mathcal{S}_0, \{\mu^*_i\}_{i\in\mathcal{S}_0}), \mathsf{ct}^*) = \mathsf{Combine}(\mathsf{pk}, (\mathcal{S}_1, \{\mu^*_j\}_{j\in\mathcal{S}_1}), \mathsf{ct}^*),$$

which contradicts the winning condition of $\mathcal{A}$.

In other words, we have $\Pr[W_3] = 0$. Then, by the triangle inequalities, we obtain that

$$\Pr[W_0] \leq \mathsf{negl}(\lambda).$$

This concludes the lemma. □

E On the Soundness of a Σ-Protocol Proving Plaintext Equalities Between Paillier Ciphertexts

In this section, we analyze the soundness of a Fiat-Shamir-based non-interactive proof system showing plaintext equalities between Paillier ciphertexts generated for distinct moduli. This proof system was used in a Paillier-based threshold cryptosystem proposed by Fouque and Pointcheval [49, Section 4.2].

Let $N_1 = p_1 q_1$ and $N_2 = p_2 q_2$ be RSA moduli and let $g_1 \in \mathbb{Z}^*_{N_1^2}$, $g_2 \in \mathbb{Z}^*_{N_2^2}$ be elements of order at least N_1 and N_2, respectively. Let an integer $M < \min(N_1, N_2)$ defining the message space $\mathsf{MsgSp} = [0, M]$. Let also the language

$$\mathcal{L}^{\text{eq-dcr}} := \{(C_1, C_2) \in \mathbb{Z}^*_{N_1^2} \times \mathbb{Z}^*_{N_2^2} \mid \exists m \in [0, M],\ w_1 \in \mathbb{Z}^\star_{N_1}, w_2 \in \mathbb{Z}^\star_{N_2} : \\ C_1 = g_1^m \cdot w_1^{N_1} \bmod N_1^2 \ \ \wedge \ \ C_2 = g_2^m \cdot w_2^{N_2} \bmod N_2^2\}.$$

We assume that the challenge space is $\{0, \ldots, 2^\lambda - 1\}$ and that $p, q > 2^{l(\lambda)}$, for some polynomial $l : \mathbb{N} \to \mathbb{N}$ such that $l(\lambda) > \lambda$ for any sufficiently large $\lambda \in \mathbb{N}$.

$\mathbf{P}\big(\mathsf{crs}, \boldsymbol{x}, \boldsymbol{w}\big) \leftrightarrow \mathbf{V}(\mathsf{crs}, \boldsymbol{x})$: Given a crs, a statement $\boldsymbol{x} = (C_1, C_2) \in \mathbb{Z}^*_{N_1^2} \times \mathbb{Z}^*_{N_2^2}$, P (who has the witness $\boldsymbol{w} = (m, w_1, w_2) \in [0, M] \times \mathbb{Z}^\star_{N_1} \times \mathbb{Z}^\star_{N_2}$) and V interact as follows:

1. P chooses random $a \hookleftarrow U([0, R])$, $r_1 \hookleftarrow U(\mathbb{Z}^*_{N_1})$, $r_2 \hookleftarrow U(\mathbb{Z}^*_{N_2})$ and sends $A_1 = g_1^a \cdot r_1^{N_1} \bmod N_1^2$, $A_2 = g_2^a \cdot r_2^{N_2} \bmod N_2^2$ to V.
2. V sends a random challenge $\mathsf{Chall} \hookleftarrow U(\{0, \ldots, 2^\lambda - 1\})$ to P.
3. P aborts if $a + \mathsf{Chall} \cdot m \notin [0, R]$. Otherwise, it sends V the response
$$z = a + \mathsf{Chall} \cdot m, \quad z_1 = r_1 \cdot w_1^{\mathsf{Chall}} \bmod N_1, \quad z_2 = r_2 \cdot w_2^{\mathsf{Chall}} \bmod N_2$$
4. V checks if $z \in [0, R]$ and
$$A_1 \cdot C_1^{\mathsf{Chall}} \equiv z_1^{N_1} \cdot g_1^z \pmod{N_1^2}, \qquad A_2 \cdot C_2^{\mathsf{Chall}} \equiv z_2^{N_2} \cdot g_2^z \pmod{N_2^2}$$
and returns 0 if this condition is not satisfied.

We show that the above protocol does not provide soundness in general when $N_1 \neq N_2$ as a cheating prover can create a fake proof for messages $m_1 \in \mathbb{Z}_{N_1}$, $m_2 \in \mathbb{Z}_{N_2}$ such that $N_2 < m_1 < N_1$ (we assume w.l.o.g. that $N_1 > N_2$).

Consider a false statement $(C_1, C_2) \notin \mathcal{L}^{\text{eq-dcr}}$ together with a valid transcript $(\boldsymbol{a} = (A_1, A_2), \mathsf{Chall}, \boldsymbol{z} = (z, z_1, z_2))$ of the Σ-protocol (note that such a transcript can be generated by running the HVZK simulator). For each $j \in \{1, 2\}$, Let $m_j = \mathcal{D}_{\mathsf{sk}_j}(C_j) \in \mathbb{Z}_{N_j}$ and $a_j = \mathcal{D}_{\mathsf{sk}_j}(A_j) \in \mathbb{Z}_{N_j}$ the plaintexts obtained by decrypting $\boldsymbol{x} = (C_1, C_2) \in \mathbb{Z}^*_{N_1^2} \times \mathbb{Z}^*_{N_2^2}$ and $\boldsymbol{a} = (A_1, A_2) \in \mathbb{Z}^*_{N_1^2} \times \mathbb{Z}^*_{N_2^2}$, respectively.

Note that, since $(C_1, C_2) \notin \mathcal{L}^{\text{eq-dcr}}$, we have $m_1 \neq m_2$. Moreover, a valid transcript must contain $(\mathsf{Chall}, z) \in \{0, \ldots, 2^\lambda - 1\} \times [0, R]$ satisfying the following equations over $\mathbb{Z}$:

$$\begin{aligned} a_1 &= z - m_1 \cdot \mathsf{Chall} + k_1 \cdot N_1 \\ a_2 &= z - m_2 \cdot \mathsf{Chall} + k_2 \cdot N_2, \end{aligned} \tag{30}$$

for some $k_1, k_2 \in \mathbb{Z}$. We remark that valid solutions $(z, \mathsf{Chall}, k_1, k_2) \in \mathbb{Z}^4$ of the system (30) live in a coset of the two-dimensional lattice

$$\Lambda = \left\{ (z_1, z_2, z_3, z_4) \in \mathbb{Z}^4 : \begin{bmatrix} 1 & -m_1 & N_1 & 0 \\ 1 & -m_2 & 0 & N_2 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \mathbf{0} \right\}.$$

Let us further assume that $m_1 = N_1 - 1$ and $m_2 = N_2 - 1$ (in which case $(C_1, C_2) \notin \mathcal{L}^{\text{eq-dcr}}$ since $N_1 \neq N_2$). Then, $\boldsymbol{z} \triangleq (z_1, z_2, z_3, z_4) = (-1, 1, 1, 1)$ belongs to Λ.

A cheating prover can thus initially encrypt $C_1 = g_1^{m_1} \cdot w_1^{N_1} \bmod N_1^2$ and $C_2 = g_2^{m_2} \cdot w_2^{N_2} \bmod N_2^2$, with $w_1 \hookleftarrow U(\mathbb{Z}_{N_1}^*)$, $w_2 \hookleftarrow U(\mathbb{Z}_{N_2}^*)$. Then, it can run the HVZK simulator by choosing $\widetilde{\mathsf{Chall}} = 2^\lambda - 1$, $\tilde{z} \hookleftarrow U([0, R - 2^\lambda])$, $\tilde{z}_1 \hookleftarrow U(\mathbb{Z}_{N_1}^*)$, $\tilde{z}_2 \hookleftarrow U(\mathbb{Z}_{N_2}^*)$ and computing

$$A_1 = \tilde{z}_1^{N_1} \cdot g_1^{\tilde{z}} \cdot C_1^{-\widetilde{\mathsf{Chall}}} \bmod N_1^2, \qquad A_2 = \tilde{z}_2^{N_2} \cdot g_2^{\tilde{z}} \cdot C_2^{-\widetilde{\mathsf{Chall}}} \bmod N_2^2,$$

which implicitly yields a solution $(\tilde{z}, \widetilde{\mathsf{Chall}}, \tilde{k}_1, \tilde{k}_2) \in \mathbb{Z}^4$ to the system (30). For a given challenge $\mathsf{Chall} \in \{0, \ldots, 2^\lambda - 1\}$ sent by the verifier, the cheating prover can then use the vector $(\mathsf{Chall} - \widetilde{\mathsf{Chall}}) \cdot (-1, 1, 1, 1) \in \Lambda$ to generate a solution $(z, \mathsf{Chall}, k_1, k_2) \in \mathbb{Z}^4$ of (30) for the given challenge Chall by setting

$$\begin{aligned} z &= \tilde{z} - (\mathsf{Chall} - \widetilde{\mathsf{Chall}}) \\ z_1 &= \tilde{z}_1 \cdot g_1^{\mathsf{Chall} - \widetilde{\mathsf{Chall}}} \cdot w_1^{\mathsf{Chall} - \widetilde{\mathsf{Chall}}} \bmod N_1 \\ z_2 &= \tilde{z}_2 \cdot g_2^{\mathsf{Chall} - \widetilde{\mathsf{Chall}}} \cdot w_2^{\mathsf{Chall} - \widetilde{\mathsf{Chall}}} \bmod N_2 \end{aligned}$$

which implicitly sets

$$\begin{aligned} k_1 &= \tilde{k}_1 + (\mathsf{Chall} - \widetilde{\mathsf{Chall}}) \\ k_2 &= \tilde{k}_2 + (\mathsf{Chall} - \widetilde{\mathsf{Chall}}). \end{aligned}$$

Note that $((A_1, A_2), \mathsf{Chall}, (z, z_1, z_2))$ forms an accepting transcript. Indeed, we have $-(\mathsf{Chall} - \widetilde{\mathsf{Chall}}) < 2^\lambda$ (so that $z \in [0, R]$) and

$$\begin{aligned} z_1^{N_1} \cdot g_1^z \cdot C_1^{-\mathsf{Chall}} &\equiv \left(\tilde{z}_1 \cdot g_1^{\mathsf{Chall} - \widetilde{\mathsf{Chall}}} \cdot w_1^{\mathsf{Chall} - \widetilde{\mathsf{Chall}}}\right)^{N_1} \cdot g_1^{\tilde{z} - (\mathsf{Chall} - \widetilde{\mathsf{Chall}})} \cdot \left(g_1^{m_1} \cdot w_1^{N_1}\right)^{-\mathsf{Chall}} \\ &\equiv \tilde{z}_1^{N_1} \cdot g_1^{\tilde{z} - (1 - N_1) \cdot (\mathsf{Chall} - \widetilde{\mathsf{Chall}})} \cdot g_1^{-m_1 \cdot \mathsf{Chall}} \cdot w_1^{-N_1 \cdot \widetilde{\mathsf{Chall}}} \\ &\equiv \tilde{z}_1^{N_1} \cdot g_1^{\tilde{z} + m_1 \cdot (\mathsf{Chall} - \widetilde{\mathsf{Chall}})} \cdot g_1^{-m_1 \cdot \mathsf{Chall}} \cdot w_1^{-N_1 \cdot \widetilde{\mathsf{Chall}}} \\ &\equiv \tilde{z}_1^{N_1} \cdot g_1^{\tilde{z} - m_1 \cdot \widetilde{\mathsf{Chall}}} \cdot w_1^{-N_1 \cdot \widetilde{\mathsf{Chall}}} \equiv \tilde{z}_1^{N_1} \cdot g_1^{\tilde{z}} \cdot C_1^{-\widetilde{\mathsf{Chall}}} \equiv A_1 \pmod{N_1^2}, \end{aligned}$$

while we obtain

$$z_2^{N_2} \cdot g_2^z \cdot C_2^{-\mathsf{Chall}} \equiv \tilde{z}_2^{N_2} \cdot g_2^{\tilde{z}} \cdot C_2^{-\widetilde{\mathsf{Chall}}} \equiv A_2 \pmod{N_2^2}$$

in the same way.

This invalidates the soundness of the Σ-protocol and the proof that the Paillier-based threshold cryptosystem from [49] provides IND-CCA2 security in the random oracle model. We insist that we did not find a concrete chosen-ciphertext attack against the scheme. Still, the security proof cannot apply a Naor-Yung-based strategy since, in a sequence of hybrid games, the adversary can always distinguish which one of the secret keys $\mathsf{sk}_1 = (p_1, q_1)$, $\mathsf{sk}_2 = (p_2, q_2)$ is used to answer decryption queries. A simple solution to this problem is to add a range proof showing that the plaintext is smaller than $\min(N_1, N_2)$.