Proving Threshold Regev PKE from Adaptive Hint-MLWE: Efficient, Non-interactive, and CCA Secure

Yisol Hwang[1], Shuichi Katsumata[2,3], Seonhong Min[1]
Guilhem Niot[2], Yongsoo Song[1]

[1] Seoul National University
[2] PQShield
[3] National Institute of Advanced Industrial Science and Technology (AIST)

Abstract. Threshold public-key encryption (tPKE) has recently attracted renewed interest, largely due to NIST's call for Multi-Party Threshold Cryptography. While classical tPKE has approached a high state of maturity, its post-quantum counterpart has not. Indeed, thresholdizing the celebrated lattice-based Regev PKE, which forms the basis of ML-KEM, remains unsatisfactory. Interestingly, *how* to thresholdize Regev PKE has not fundamentally changed in over a decade — the only thing that has gradually progressed is its security analysis. To this day, it remains open whether threshold Regev can be proven secure while simultaneously satisfying a polynomial modulus, non-interactive decryption, and CCA-compatibility, each of which is essential for practical deployment.

We answer this affirmatively, providing the first proof that threshold Regev is secure under the MLWE assumption while satisfying all three requirements. In fact, we prove that it satisfies a very strong form of simulation-based security — even stronger than what was known under a super-polynomial modulus — allowing the adversary to obtain partial decryptions even of the challenge ciphertext. At the technical heart of our result is the *adaptive* hint-MLWE (AHMLWE) problem, an adaptive variant of hint-MLWE where the adversary obtains hints on the MLWE secret with *adaptively* chosen coefficients. We show that AHMLWE reduces tightly to standard MLWE, which may be of independent interest.

1 Introduction

A t-out-of-n threshold public-key encryption (tPKE) strengthens the security of traditional PKE in sensitive deployments by distributing the secret key across n parties so that any t of them can jointly decrypt, while no single participant can act alone. While tPKE has always been a topic of interest, it has seen renewed interest in recent years, notably due to NIST announcing a call for threshold cryptosystems [BP26]. Classical applications include preserving voter privacy in e-voting [CGS97, ABGS23], while more modern applications include mempool privacy for blockchains and decentralized finance [CGPP24, CGPW25].

In the classical setting, tPKE has approached a high state of maturity, with efficient threshold variants of RSA and ElGamal known for decades. In contrast, constructing an efficient and secure tPKE from post-quantum assumptions has proven to be challenging. While the study of lattice-based tPKE was initiated over a decade ago by Bendlin and Damgård [BD10], the community is still making progress, proposing numerous constructions to this date, e.g., [BGG+18, KLO+19, DLN+21, CCMS21, BS23, MS24, CLW25, LP25, BdLP26].

In this work, as with many prior works, our target is thresholdizing the Regev PKE [Reg05], or more precisely, its optimized variant [LPR10, LP11], which forms the basis of ML-KEM [Nat24].[4] Our goal is to satisfy all of the following requirements *simultaneously*, each of which is essential for practical deployment:

1. *Polynomial-sized modulus*: the scheme can be instantiated with concretely efficient parameters;
2. *(Asynchronous) non-interactive decryption*: each party outputs a single partial decryption without knowing the set of decryptors;
3. *CCA-compatible*: the scheme can be generically upgraded to CCA security.

Below, let us take a closer look at why these seemingly modest requirements have been elusive to achieve at the same time.

Blueprint for CPA-secure threshold Regev. While there are subtle differences, all threshold Regev PKEs share the same blueprint. The public key consists of a matrix A and $\vec{b} = -A\vec{s} + \vec{e} \bmod q$ for some short noise vector $\vec{e}$, and a ciphertext of a message m takes the form $(c_0, \vec{c}_1) = (\vec{r}_2^\top \vec{b} + r_0 + \mathsf{Encode}(m),\ A^\top \vec{r}_2 + \vec{r}_1) \bmod q$ for some short noise $r = (r_0, \vec{r}_1, \vec{r}_2)$ and encoding algorithm Encode. The secret $\vec{s}$ is linearly secret-shared among the parties into shares $(\vec{s}_i)_{i\in[n]}$, where for any valid set of decryptors $\mathcal{S}$, there exist constants $\lambda_i^{\mathcal{S}}$ such that $\vec{s} = \sum_{i\in\mathcal{S}} \lambda_i^{\mathcal{S}} \vec{s}_i$.

To perform non-interactive decryption, each party outputs its partial decryption $\vec{c}_1^\top \vec{s}_i + f_i \bmod q$ for "some" noise f_i to be discussed later. Here, note that without the noise f_i, one can trivially recover $\vec{s}_i$ using Gaussian elimination given sufficiently many partial decryptions. Once partial decryptions from a valid set of decryptors $\mathcal{S}$ are collected, we can compute $c_0 + \sum_{i\in\mathcal{S}} \lambda_i^{\mathcal{S}} \cdot \vec{c}_1^\top \vec{s}_i \approx \mathsf{Encode}(m) + \sum_{i\in\mathcal{S}} \lambda_i^{\mathcal{S}} f_i \bmod q$. Therefore, as long as the secret sharing scheme supports small reconstruction coefficients $\lambda_i^{\mathcal{S}}$, we can recover m.[5]

How to prove CPA security. The main difference between all prior works on threshold Regev boils down to how this blueprint is proven secure. While there is a hierarchy of CPA security notions for tPKE [BKW26], at a minimum, the

[4] Throughout the introduction, we do not differentiate between these variants for simplicity, as they are interchangeable in the context of tPKE.

[5] It is well-known that this restriction on $\lambda_i^{\mathcal{S}}$ can be removed in the "synchronous" setting, where the set of decryptors is known at the time of partial decryption (e.g., [dVPS26]).

scheme must tolerate partial decryption queries on honestly generated, known-message ciphertexts ct_m. That is, the adversary is given access to an encryption oracle EncO, which on input m outputs an honestly generated ciphertext ct_m, and it may then obtain partial decryptions of ct_m. This implicitly models the requirement that the secret shares $\vec{s}_i$ do not leak.

The initial proof by Bendlin and Damgård [BD10] showed that threshold Regev is secure in the n-out-of-n setting if f_i is super-polynomially larger than the encryption noise r, leading to a scheme with a super-polynomial modulus q. Later, [BGG$^+$18, DLN$^+$21] showed, among others, that threshold Regev is secure even in the t-out-of-n setting, still assuming a super-polynomial q, to the security parameter λ and the number of queries Q.

The first to prove threshold Regev secure under a *polynomial* modulus was Boudgoust and Scholl [BS23]. However, as they relied on the Rényi divergence [6], which does not directly apply to decision-based games, they were only able to prove OW-CPA security — to achieve IND-CPA security, they had to modify threshold Regev. Later, Micciancio and Suhl [MS24] also proved (a slight variant of) threshold Regev secure under a polynomial modulus, but restricted to the n-out-of-n and unstructured LWE setting; that is, their result does not extend to the structured module setting without assuming a non-standard *known-norm* MLWE assumption. The state-of-the-art is the recent work by Cini, Lai, and Woo [CLW25], who proved that threshold Regev (which they call Pilvi) is secure under a polynomial modulus by introducing an a priori bound Q_{enc} on the number of EncO queries, with their modulus scaling as $q \approx \sqrt{Q_{\mathsf{enc}}}$. Unlike a bound on the number of partial decryptions, however, this is an artificial bound that is unobservable in the real-world. Counterintuitively, even if the adversary does *not* obtain any partial decryption of ct_m, the proof still requires the modulus of threshold Regev to scale with Q_{enc}.

Towards CCA security. Beyond CPA security, achieving CCA security is essential for the practical applicability of tPKE. For standard PKEs, we typically rely on the Fujisaki-Okamoto (FO) transform [FO99] to convert a CPA-secure PKE into a CCA-secure one. However, as the FO transform relies on a re-encryption technique that checks the validity of the ciphertext using the underlying plaintext, it is notoriously difficult to perform in a threshold manner. Indeed, existing works thresholdizing the FO transform rely on generic MPC techniques [KLO$^+$19, CCMS21], leading to a highly inefficient interactive decryption protocol.

Another popular approach is the encrypt-and-prove methodology, where the encryptor attaches a proof π computed via a non-interactive zero-knowledge proof (NIZK), demonstrating knowledge of the ciphertext randomness. Unlike the FO transform, the parties can simply perform threshold decryption of the CPA-secure tPKE after *publicly* checking the proof π, making it "seemingly"

[6] They assume an a priori upper bound Q on the number of partial decryptions the adversary obtains, a convention that has since remained widely used in tPKE. Limiting the number of oracle queries in security proofs is a common convention that predates this work (e.g., [ASY21] for threshold signatures).

compatible with CCA security. Indeed, many prior works (e.g., [BS23, MS24, CLW25]) suggested this as a viable path. However, none of them provided a formal proof, nor a concrete instantiation of a CCA-secure tPKE.

Recently, [BKW26] exposed the pitfall of this approach: they provided a hierarchy of CPA security notions and showed that, depending on the level of CPA security satisfied by the underlying tPKE, the above NIZK transform does *not* result in a CCA-secure scheme. At a high level, this is because the reduction R (i.e., the adversary against the CPA-secure tPKE) must simulate the partial decryptions of the honest parties to the adversary in the CCA security game. Observe that this is unique to the threshold setting: in the non-threshold setting, if R can extract the message from the proof π attached to an adversarially submitted $\mathsf{ct}_{\mathcal{A}}$, R can easily simulate the decryption oracle by returning m.

Let us go back to the minimal CPA security notion discussed above. In that game, the CPA adversary has access to an EncO oracle outputting *honestly generated* ct_m, and a $\mathsf{PartDecO}$ oracle that on input ct_m outputs the partial decryptions of the honest parties. It is clear that this is useless to R, as it offers no interface to obtain partial decryptions of adversarial but well-formed ciphertexts $\mathsf{ct}_{\mathcal{A}}$. Indeed, what R needs is an EncO oracle that allows it to specify the (potentially malicious) randomness used to generate ct_m. [BKW26] coined such security *semi-malicious* CPA security. They also defined a slightly weaker notion where EncO returns the (honestly generated) randomness used to construct ct_m, along with a generic transform relying instead on a non-interactive *proof of randomness* (NIPoR) — as a NIPoR is a stronger primitive than a NIZK, this results in a less efficient CCA transform.

Summary of the situation. It turns out that the only security proofs of threshold Regev covering semi-malicious CPA security are the initial ones [BD10, BGG$^+$18], both relying on a super-polynomial modulus.[7] Notably, all subsequent works focusing on a polynomial modulus [BS23, MS24, CLW25] are *not* compatible with the NIZK transform, effectively *trading security with efficiency.* While [CLW25] claims that the NIPoR transform of [BKW26] can be used on Pilvi to achieve CCA security, this is not the case upon closer inspection, since they achieve a strictly weaker notion of CPA security than what the transform requires. [8]

Therefore, while there has been significant progress in our understanding of threshold Regev, we still do not know the answer to the following fundamental question, which brings us back to the main goal of this work:

Can the Regev PKE be safely and efficiently thresholdized, simultaneously satisfying requirements (1) to (3)?

[7] While these works do not claim semi-malicious CPA security, it is clear that their proofs achieve it. We ignore [DLN$^+$21] as it is a direct construction of a CCA-secure scheme, even though it follows the general NIZK transform.

[8] We informed the authors of this issue, and they have confirmed that their weak notion of CPA security is insufficient for achieving CCA security with polynomial modulus. Accordingly, their concrete parameters in Section 7 is limited to a CPA secure scheme.

1.1 Our Contribution

In this work, we answer the above question affirmatively. We provide the first proof that threshold Regev is secure under the MLWE assumption while satisfying all three requirements: a polynomial modulus (Item 1), non-interactive decryption (Item 2), and CCA-compatibility (Item 3). In fact, we prove that threshold Regev satisfies the strongest *simulation-based* CPA security in the hierarchy of [BKW26] — even stronger than what was proven by [BD10, BGG$^+$18] under a super-polynomial modulus — allowing the adversary to obtain partial decryptions even of the *challenge* ciphertext. A comparison with prior CPA-secure schemes is provided in Table 1.

Work	Polynomial modulus	PartDecO on L_{Chal}	PartDecO on L_{Enc}	rnd.	Model	CCA compatible
[BD10]†	✗	✗	✓	mal.	SIM	✓
[BGG$^+$18]	✗	✗	✓	mal.	SIM	✓
[BS23]	✓	✗	✓	$\perp$	IND	✗
[MS24]†,‡	✓	✗	✓	$\perp$	SIM	✗
[CLW25]	✓	✓$^{(-)}$	✓$^{(-)}$	hon.	SIM	✗
Ours	✓	✓	✓	mal.	SIM	✓

Table 1. Comparison of CPA-secure (Regev-style) non-interactive tPKEs. Oracles L_{Chal} and L_{Enc} provide partial decryptions of challenge ciphertexts ct^* and known-message ciphertexts ct_m, respectively. Column "rnd." indicates ct_m randomness: hidden ($\perp$), honest (hon. and given to $\mathcal{A}$), or malicious (mal. and chosen by $\mathcal{A}$). IND and SIM denote indistinguishability and simulation-based security. A scheme is CCA-compatible if it can be transformed into a CCA-secure one via a generic NIZK transform. †: Supports only the n-out-of-n setting ($t = n$). ‡: [MS24] requires non-standard MLWE assumption. $^{(-)}$: [CLW25] bounds the number of ct_m created, preventing the CCA transform from [BKW26]. They also restrict the number of partial decryptions of ct^* to $t-1$ parties.

At the technical heart of our result is the *adaptive* hint-MLWE (AHMLWE) problem, an adaptive variant of the hint-MLWE problem [KLSS23] that allows the adversary to obtain hints on the MLWE secret with *adaptively* chosen coefficients. We show that AHMLWE reduces tightly to the standard MLWE problem using the noise-rerandomization technique of Katsumata and Yamada [KY16]. Our security proof then consists of rewriting the partial decryptions of the honest parties and carefully layering the hybrid games, invoking several variants of AHMLWE in order.

We further provide the first concrete instantiation of a CCA-secure threshold Regev PKE by instantiating the NIZK with the Lantern proof system [LNP22]. For concrete efficiency, we rely on Shamir secret sharing over the monomi-

als [BFM+25, CLW25, KLL+26], supporting up to $n = 32$ parties [9]. As a result, our CPA-secure variant achieves ciphertext sizes of 19.10KB, 43.97KB, and 19.41KB for the threshold $t = 2, 16, 31$, respectively, and our CCA-secure variant achieves ciphertext sizes of 99.77KB, 172.34KB, and 100.35KB for the threshold $t = 2, 16, 31$, respectively.

1.2 Related Work

Threshold fully homomorphic encryption. A closely related line of work constructs threshold *fully homomorphic* encryption (tFHE). Boneh et al. [BGG+18] proposed the first tFHE under super-polynomial noise. Unlike tPKE, tFHE does not need to consider CCA security. Since then, many papers have followed, e.g., [BS23, PS24, OT25, ZZLP25, OT26]. The recent lines of work [PS24, OT25, OT26] have focused on a modification of [BGG+18] where they optimize the size of the partial decryption. While ciphertext ct still lives over a super-polynomial modulus, they include a "sanitization" step so that the partial decryption lives over a polynomial modulus. Some works additionally introduce multi-party computation (MPC) for the decryption to achieve a polynomially large modulus, however, they are either confined to a 3-party setting [KLO+19] or require a power-of-two modulus to work [ZZLP25]. Most importantly, they do not support a constant-round decryption.

More CCA-secure tPKEs. Devevey et al. [DLN+21] construct the first *adaptively* and CCA secure lattice-based tPKE in the *standard* model (i.e., no random oracles). They rely on lossy arguments, requiring a super-polynomial sized modulus, and mainly serving as a feasibility result. Micciancio and Suhl [MS24] build the only threshold Regev that achieves a polynomial modulus without any restrictions on the number of partial decryptions. Their proof explicitly relies on the weaker CPA model, where the encryption randomness is hidden from the adversary — combining their proof strategy with ours will be an interesting research direction. Recently, Lapiha and Prest [LP25] and Boudgoust et al.[BdLP26] departed from the threshold Regev blueprint and focused directly on constructing a lattice-based CCA-secure tPKE with polynomial modulus. Their construction is based on building a CPA-secure threshold identity-based encryption (tIBE) and converting it into a CCA-secure tPKE via a threshold variant of the BCHK transform [CHK04, BCHK07]. They achieve a small ciphertext size of about 30 kB for $t = 32$ and up to $Q = 2^{46}$ decryption queries, however at the cost of requiring an *interactive* decryption protocol. Moreover, it is unclear if their construction can achieve a strong form of simulation security, supporting partial challenge ciphertext decryptions.

Concurrent Work. Recently, Zhang et al. [ZZH+26] presented two adaptively secure threshold PKE schemes TD_1 and TD_2 with a polynomially large modulus,

[9] Asymptotically, our proof supports an unbounded polynomial number of parties by relying on $\{0, 1\}$-linear secret sharing schemes (e.g., [BGG+18, BS23]). However, since these are concretely inefficient, we did not choose them for our instantiation.

respectively, in the asynchronous and synchronous settings. Their original constructions were designed concurrently with our work. However, the initial version of their work had a proof issue[10], which they fixed in a subsequent update, after the appearance of our work. While their proof fix to TD_1 remains independent of our work, their proof fix to TD_2 is based on our proof techniques and the adaptive Hint-MLWE problem, making TD_2 effectively a follow-up to our work.

TD_1 builds on the adaptively secure tPKE by Devevey et al. [DLN+21] that relies on Dual Regev with a very wide public matrix, and shows that the secret retains enough min-entropy to apply the leftover hash lemma even given the adaptive hints leaked by partial decryptions. Their only computational step, based on (semi-adaptive) Hint-MLWE, does not involve the secret key, so their reduction can sample it and all its shares itself. Due to replicated rather than Shamir secret sharing, their CCA ciphertexts are 1.4–1.6× larger than ours for $n = 16$, $t \in \{2, 8\}$ and up to 2^{60} queries, and their partial decryptions reach 13.1 MB for $t = 8$ whereas ours is only 2.3 KB, thus it is more than 5600 times larger than ours. Moreover, TD_1 is proven in a game-based model and it is unclear if it satisfies unbounded partial decryptions of challenge ciphertexts.

On the other hand, TD_2 uses primal Regev, and corresponds to the construction sketched in Footnote 5 for the *synchronous* setting: it combines our proof techniques and the adaptive Hint-MLWE problem with the zero-share technique of [DKM+24, dVPS26] to mask partial decryptions, thereby supporting an arbitrary polynomial number of users with Shamir sharing. They further show that this construction is adaptively secure, reminiscent of the adaptively secure threshold signature from [KRT24].

1.3 Technical Overview

We now describe our two main technical contributions: the adaptive hint-MLWE problem and its reduction to MLWE, and how they drive the security proof of our tPKE scheme, including the simulation of partial decryptions of *challenge* ciphertexts. Throughout, $\mathcal{R} := \mathbb{Z}[X]/(X^N + 1)$ denotes the ring of integers of the $2N$-th cyclotomic field for a power-of-two integer N, and $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$ for an integer modulus q; vectors and matrices over $\mathcal{R}$ are written in bold.

Adaptive hint-MLWE. The MLWE problem asks to distinguish $(A, [I_k \mid A]\cdot\vec{x})$ from a uniformly random pair, for $A \xleftarrow{\$} \mathcal{R}_q^{k\times\ell}$ and $\vec{x} \leftarrow \chi^{k+\ell}$. Hint-MLWE [KLSS23] strengthens the adversary by additionally revealing linear hints $\vec{v}_i^\top \vec{x} + y_i$ on the secret $\vec{x}$, where the coefficient vectors $\vec{v}_i$ are fixed *before* the game begins and y_i is drawn from an independent noise distribution χ'.

In our setting, each partial decryption is, after a short rewriting, exactly such a hint. Substituting the public-key and ciphertext relations $\vec{b} = -A\vec{s} + \vec{e}$

[10] The proof in their original ePrint version (posted on 2026-08-06) argued that the key distribution can be determined from the hints. However, the distribution of these hints may itself depend on the key when queries are made adaptively, thereby breaking the argument (application of Lemma 13 in Lemma 6). This issue was acknowledged by the authors via email, and addressed in their ePrint update of 2026-09-19.

and $\vec{c}_1 = A^\top \vec{r}_2 + \vec{r}_1$ reveals a linear function of the *short* secret $(\vec{s}, \vec{e})$, with the ciphertext's own encryption randomness $r = (r_0, \vec{r}_1, \vec{r}_2)$ as its coefficient and flooding noise f_i. This fits selective hint-MLWE only when the encryption randomness is fixed in advance — but a semi-malicious adversary chooses it adaptively, so the relevant coefficient is only known when the query is made. We therefore define *adaptive hint-MLWE* (AHMLWE, Def. 8), where the adversary may query an oracle $\mathcal{O}_{\mathsf{hint}}$ on adaptively chosen inputs $\vec{v} \in \mathcal{V}$ and receive $\vec{v}^\top \vec{x} + y$ for fresh $y \leftarrow \chi'$.

Hardness is not a direct corollary of the selective case, since a reduction to hint-MLWE must commit to its hints before seeing the eventual queries. We rely on the noise-rerandomization technique of Katsumata and Yamada [KY16] (Lem. 7): from one selective hint, one can locally compute $M\vec{x} + \vec{y}'$ for any bounded matrix M, with $\vec{y}'$ statistically close to a fresh Gaussian. Our reduction (Thm. 1) asks for one selective hint per secret coordinate and answers each adaptive query by rerandomizing the appropriate linear combination of these hints, keeping the loss relative to the underlying hint-MLWE instance tight.

The threshold Regev blueprint, revisited. Our scheme follows the threshold Regev blueprint (Fig. 3). The public key is $(\vec{b}, A)$ for $A \leftarrow \mathcal{U}(\mathcal{R}_q^{k \times k})$ and $\vec{b} = -A\vec{s} + \vec{e}$, with $\vec{s}, \vec{e} \leftarrow D_{\sigma_s}^k$; a ciphertext of m is $(c_0, \vec{c}_1)$ for $c_0 = \vec{r}_2^\top \vec{b} + r_0 + \gamma^{-1}\lfloor q/p \rceil m$ and $\vec{c}_1 = A^\top \vec{r}_2 + \vec{r}_1$, where $(r_0, \vec{r}_1, \vec{r}_2) \leftarrow D_{\sigma_r}$; and the secret $\vec{s}$ is Shamir-shared at monomial evaluation points into shares $\vec{s}_1, \ldots, \vec{s}_n$, so that $\vec{s} = \sum_{i \in \mathcal{I}} \lambda_i^{\mathcal{I}} \vec{s}_i$ for any size-t set $\mathcal{I}$. To support asynchronous decryption — where the set of decryptors is not known when partial decryptions are produced — each party simply releases the unweighted $d_i = \vec{c}_1^\top \vec{s}_i + f_i$; the Lagrange weights and a clearing factor γ — ensuring $\gamma\lambda_i^{\mathcal{I}}$ is small — only enter at the very end, in FinDec.

Semi-malicious security via adaptive hint-MLWE. The proof proceeds in three stages. First, we show that partial decryptions leak nothing about the secret key itself, even though they may still leak the message — which is harmless at this point, since no ciphertext has yet been declared a challenge. Second, we show that challenge ciphertexts can themselves be replaced by uniformly random values, so that they stop leaking their message. Third, and this is genuinely specific to the threshold setting, we must still answer partial-decryption queries on these now-message-independent challenge ciphertexts: one more application of AHMLWE turns certain virtual shares into one-time masks, after which a purely combinatorial argument about Shamir sharing, taken up in the next paragraph, ensures that the adversary's view of partial decryptions of challenge ciphertexts can be simulated without knowing the message when fewer than t queries are made.

For the first stage, fix a maximal corrupted set $\mathcal{C}' \supseteq \mathcal{C}$ of size $t - 1$. By Lagrange interpolation against $\mathcal{C}'$, the share $\vec{s}_j$ of an honest party $j \notin \mathcal{C}'$ satisfies $\vec{s}_j = \mu_j \cdot (\vec{s} - \sum_{k \in \mathcal{C}'} \lambda_k^j \vec{s}_k)$ for $\mu_j := (\lambda_j^{\mathcal{C}' \cup \{j\}})^{-1}$, so its partial decryption $d_j = \vec{c}_1^\top \vec{s}_j + f_j$ reduces, via $\vec{b} = -A\vec{s} + \vec{e}$ and $\vec{c}_1 = A^\top \vec{r}_2 + \vec{r}_1$, to a known part plus the single term $\mu_j(\vec{r}_1^\top \vec{s} + \vec{r}_2^\top \vec{e}) + f_j$ — exactly a hint on $(\vec{s}, \vec{e})$ with

coefficient $\mu_j \cdot (\vec{r}_1, \vec{r}_2)$. Since a semi-malicious query lets $\mathcal{A}$ pick $\vec{r}$ itself, interleaved with other queries, this coefficient is only fixed once the ciphertext is produced — precisely the adaptivity AHMLWE is built to tolerate, and where selective hint-MLWE would not suffice. Once AHMLWE replaces $\vec{b}$ by uniform, every partial decryption carries no information about the secret, regardless of which ciphertexts it decrypts, challenge or not.

For the second stage, a symmetric argument handles each challenge ciphertext itself: reading it as an MLWE sample with secret $\vec{r}_2$ and small error $(r_0, \vec{r}_1)$, the same partial-decryption term now leaks its randomness as a hint with coefficient $\vec{v} = \mu_j \cdot (1, \vec{s}, \vec{e})$ — the honestly sampled $\vec{s}, \vec{e}$ playing the role that the encryption randomness played above. After applying AHMLWE, the challenge ciphertexts can be replaced by uniform, so that they carry no information about their message.

For the third stage, what remains is to actually answer partial-decryption queries on these challenge ciphertexts independently of the message — and a final application of the same idea (deferred to the proof of Thm. 3) sets this up. It targets the virtual shares of $\mathcal{C}' \setminus \mathcal{C}$: parties that are never corrupted. The key observation is that every dependency on such a share $\vec{s}_k$ — in the partial decryption of *any* ciphertext, challenge or not — appears through exactly two quantities: an MLWE sample $\vec{b}_k = A\vec{s}_k + \vec{e}_k$, playing the role of a "virtual public key" for party k, and, for each challenge ciphertext ct, a value $d_{\mathsf{ct},k} = \vec{c}_1^{\top}\vec{s}_k + f_{\mathsf{ct},k}$, playing the role of party k's own partial decryption of ct. Replacing both by uniform via AHMLWE (after pulling out a factor of γ to keep the relevant coefficients small) turns each $d_{\mathsf{ct},k}$, for $k \in \mathcal{C}' \setminus \mathcal{C}$, into what is effectively a fresh one-time mask on the partial decryptions of honest parties for ct — exactly the missing ingredient for simulating partial decryptions of challenge ciphertexts, which we turn to next.

Simulating challenge partial decryptions. This last step needs no further hardness assumption: it is a purely information-theoretic consequence of Shamir sharing. Fix a challenge ciphertext ct. After the rerandomizations above, the partial decryption (minus flooding noise $f_{\mathsf{ct},i}$) of *every* party $i \in [n]$ — corrupted or not — is exactly the evaluation $Q_{\mathsf{ct}}(\alpha_i)$ of a degree-$<t$ polynomial Q_{ct} over $\mathcal{R}_q$ whose constant term $Q_{\mathsf{ct}}(0)$ encodes the message m. Q_{ct} is pinned at two kinds of points: its value at 0, which depends on m, and its values at $\mathcal{C}$, which the corrupted parties can compute themselves from their own shares. The remaining $t - 1 - |\mathcal{C}|$ points are exactly where the one-time masks $d_{\mathsf{ct},k}$, for $k \in \mathcal{C}' \setminus \mathcal{C}$, introduced above land, and are therefore uniformly distributed and independent of everything else. Shamir's threshold guarantee ensures that this uniformity would hold no matter which $t - 1 - |\mathcal{C}|$ points other than $\mathcal{C}$ and 0 we chose to query. So the simulator never needs to know Q_{ct}: it answers the first $t - 1 - |\mathcal{C}|$ queries of $\mathsf{PartDecO}(\mathsf{ct}, \cdot)$ on distinct honest parties with fresh uniform values. Only once the adversary's queries plus $\mathcal{C}$ reach the threshold t does Q_{ct} become pinned down; at that point, the simulator requests the real m (as SemiMalSimCPA permits, Fig. 1) and interpolates the necessary values. In

short, simulating partial decryptions of a challenge ciphertext amounts to lazily sampling one Shamir-shared polynomial per ciphertext.

Achieving CCA security. We also propose a CCA-secure upgrade of our tPKE scheme, by relying on the BKW transform [BKW26], which can be seen as a threshold variant of the Naor-Yung paradigm [NY90]. This construction allows us to transform our semi-malicious CPA-secure tPKE scheme into a CCA-secure one by leveraging a simulation extractable (SIMEXT) NIZK system. This transform appends a NIZK proof of well-formedness to the ciphertext, and the partial decryption algorithm verifies the proof before running the baseline partial decryption.

Given that the statement to be proven consists only of norm bounds and linear relations, the Lantern proof system [LNP22] is a natural choice to instantiate this NIZK. However, Lantern is simulation-sound [BIMP26] but lacks straightline extraction. We upgrade it to a SIMEXT NIZK system by relying on on a single-key Regev PKE, following the encryption-to-the-sky approach.

We provide concrete parameters and corresponding ciphertext sizes using Lantern via the LaZer library [LSS24].

Round optimality. Our CCA-secure scheme is a direct augmentation of the Regev blueprint: decryption remains non-interactive and asynchronous, with each party independently outputting a single partial decryption. By contrast, the only other practical CCA-secure constructions, based on the BCHK+ transform [LP25, BdLP26], require a synchronous channel and three to five communication rounds.

2 Preliminaries

2.1 Notations

$|\cdot|$ denotes the cardinality of a set. For an integer k, $[k]$ denotes the index set $[k] = \{1, \ldots, k\}$. Let N be a power of two, and let q be an integer. We denote the ring of integers of the $2N$-th cyclotomic field by $\mathcal{R} = \mathbb{Z}[X]/(X^N + 1)$. For some (finite) sets $\mathcal{T}$ and $\mathcal{S}$, if $|\mathcal{T}| = t$ and $\mathcal{T} \subseteq \mathcal{S}$, we write $\mathcal{T} \subseteq_t \mathcal{S}$.

For a vector $\mathbf{v} = (v_i)_{0 \le i < m} \in \mathbb{Z}^m$, the L^p-norm and L^∞-norm are defined as $\|\mathbf{v}\|_p = \left(\sum_{i=0}^{m-1} |v_i|^p\right)^{1/p}$ and $\|\mathbf{v}\|_\infty = \max_{0 \le i < m} |v_i|$, respectively. For a matrix $\mathbf{A} \in \mathbb{Z}^{m \times n}$, the matrix two-norm of $\mathbf{A}$ is defined as $\|\mathbf{A}\|_2 = \max_{\|\mathbf{v}\|_2 = 1} \|\mathbf{A}\mathbf{v}\|_2$. If a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive semidefinite, meaning that $\mathbf{x}^\top \mathbf{A}\mathbf{x} \ge 0$ for any $\mathbf{x} \in \mathbb{R}^n$, we write $\mathbf{A} \succeq 0$. Similarly, for $v = v_0 + v_1 X + \cdots + v_{N-1} X^{N-1} \in \mathcal{R}$, we define the L^p-norm and L^∞-norm as $\|v\|_p = \left(\sum_{i=0}^{N-1} |v_i|^p\right)^{1/p}$ and $\|v\|_\infty = \max_{0 \le i < N} |v_i|$. We extend the L^2-norm to tuples $\vec{v} = (v_i)_{0 \le i < \ell} \in \mathcal{R}^\ell$ via their coefficient embedding into $\mathbb{Z}^{N\ell}$, *i.e.,* $\|\vec{v}\|_2 := \left(\sum_{i=0}^{\ell-1} \|v_i\|_2^2\right)^{1/2}$.

We denote the uniform distribution over a finite set $\mathcal{S}$ as $\mathcal{U}(\mathcal{S})$. For a real $B > 0$, we write $\mathcal{V}_B$ to denote the set of all tuples $\vec{v} = (v_i)_i \in \mathcal{R}^*$ (of any

length, clear from context) such that $\sum_i \|v_i\|_1^2 \le B$. For a distribution $\mathcal{X}$, we write $x \leftarrow \mathcal{X}$ to denote sampling x from $\mathcal{X}$.

We also recall a useful lemma.

Lemma 1 (Cauchy-Schwarz bound for ring inner products). *For any $\vec{u}, \vec{v} \in \mathcal{R}^\ell$, $\|\vec{u}^\top \vec{v}\|_\infty \le \|\vec{u}\|_2 \cdot \|\vec{v}\|_2$.*

Proof. For each coefficient index $0 \le c < N$, reduction modulo $X^N + 1$ makes the c-th coefficient of $X^c \cdot w$ a signed permutation $\tau_c(w)$ of w's coefficients, so the c-th coefficient of $\vec{u}^\top \vec{v}$ equals $\langle \vec{u}, (\tau_c(v_i))_i \rangle$ (viewing both as vectors in $\mathbb{Z}^{N\ell}$), with $\|(\tau_c(v_i))_i\|_2 = \|\vec{v}\|_2$. Cauchy-Schwarz then bounds this by $\|\vec{u}\|_2 \|\vec{v}\|_2$ for every c. □

2.2 Gaussian Distribution

For the covariance matrix $\mathbf{\Sigma} \succeq 0$ and $\mathbf{v}, \mathbf{c} \in \mathbb{Z}^n$, the Gaussian mass function $\rho_{\mu,\sqrt{\mathbf{\Sigma}}}(\mathbf{v}) : \mathbb{Z}^n \to (0, 1]$ is defined as follows:

$$\rho_{\mathbf{c},\sqrt{\mathbf{\Sigma}}}(\mathbf{v}) = \exp(-(\mathbf{v} - \mathbf{c})^\top \mathbf{\Sigma}^{-1} (\mathbf{v} - \mathbf{c})/2).$$

The discrete Gaussian distribution restricts ρ to a full-rank lattice $\Lambda \subset \mathbb{R}^n$ and renormalizes: with center $\mathbf{c} \in \mathbb{R}^n$ and covariance $\mathbf{\Sigma}$, its mass function is

$$D_{\Lambda,\mathbf{c},\sqrt{\mathbf{\Sigma}}}(\mathbf{v}) = \frac{\rho_{\mathbf{c},\sqrt{\mathbf{\Sigma}}}(\mathbf{v})}{\sum_{\mathbf{x}\in\Lambda} \rho_{\mathbf{c},\sqrt{\mathbf{\Sigma}}}(\mathbf{x})}.$$

In the subscripts of both ρ and D, we omit $\mathbf{c}$ when $\mathbf{c} = \mathbf{0}$, write σ for $\sqrt{\mathbf{\Sigma}}$ when $\mathbf{\Sigma} = \sigma^2 \mathbf{I}_n$, and omit Λ when clear from context.

We also recall useful definitions and lemmas.

Definition 1 (Smoothing parameter [MR04]). *For an n-dimensional lattice Λ and positive real $\varepsilon > 0$, the smoothing parameter $\eta_\varepsilon(\Lambda)$ is the smallest s such that $\rho_{1/(s\sqrt{2\pi})}(\Lambda^* \setminus \{\vec{0}\}) \le \varepsilon$.*

Note that η_ε defined above is the usual smoothing parameter of [MR04], where the Gaussian is defined as $\rho(\mathbf{v}) = \exp(-\pi \mathbf{v}^\top \mathbf{\Sigma}^{-1} \mathbf{v})$. Since we use the Gaussian definition with a factor $1/2$ in the exponent instead, a factor of $\sqrt{2\pi}$ appears in our smoothing conditions, e.g. $\sigma \ge \eta_\varepsilon(\mathbb{Z}^N)/\sqrt{2\pi}$ rather than $\sigma \ge \eta_\varepsilon(\mathbb{Z}^N)$.

Lemma 2 (Infinity-norm tail bound, adapted from [GPV08, Lemma 4.2]). *For $x \leftarrow D_{\mathbb{Z},\sigma}$ and $t > 0$, $\Pr[|x| \ge t\sigma] \le 2e^{-t^2/2}$. Consequently, for $s = (s_i)_{0\le i<\ell} \leftarrow D^\ell_{\mathcal{R},\sigma}$ and $\varepsilon \in (0, 1)$, taking a union bound over the $N\ell$ coefficients of s, $\Pr[\max_{0\le i<\ell} \|s_i\|_\infty \ge \rho(\varepsilon, \ell) \cdot \sigma] \le \varepsilon$, where $\rho(\varepsilon, \ell) := \sqrt{2\ln(2N\ell/\varepsilon)}$.*

Lemma 3 (Two-norm tail bound, [DKM+24, Lemma 3.4]). *Let $s \leftarrow D^\ell_{\mathcal{R},\sigma}$. Then, $\Pr\left[\|s\|_2 \ge e^{1/4}\sigma\sqrt{N\ell}\right] \le 2^{-N\ell/10}$.*

Corollary 1. *Let* $s = (s_i)_{0 \le i < \ell} \leftarrow D^{\ell}_{\mathcal{R},\sigma}$. *Then,* $\Pr\left[\sum_{i=0}^{\ell-1} \|s_i\|_1^2 \ge e^{1/2} N^2 \ell \sigma^2\right] \le 2^{-N\ell/10}$, *since Cauchy-Schwarz gives* $\|s_i\|_1 \le \sqrt{N}\|s_i\|_2$ *for each* i, *so* $\sum_i \|s_i\|_1^2 \le N\|s\|_2^2$, *and the claim follows by squaring Lem. 3's bound on* $\|s\|_2$.

For $u \in \mathcal{R}$, recall that one-norm submultiplicativity gives $\|u \cdot s_i\|_1 \le \|u\|_1 \|s_i\|_1$ for every i; combined with Corollary 1, this lets us bound $\sum_i \|u \cdot s_i\|_1^2 \le \|u\|_1^2 \sum_i \|s_i\|_1^2$ whenever u is independent of s.

2.3 Module Learning With Errors

We describe the Module Learning With Errors (MLWE) problem [BGV14, LS15] and its Hint-MLWE problem variant [KLSS23]. The Hint-MLWE problem states that the adversary cannot distinguish the MLWE distribution from uniform, even when given 'linear hints' about the secret.

Definition 2 (MLWE). *Let* k, ℓ, q *be integers, and* χ *be a probability distribution over* $\mathcal{R}$. *The advantage of an adversary* $\mathcal{A}$ *against the MLWE problem* $\mathsf{MLWE}_{q,k,\ell,\chi}$ *is defined as:*

$$\mathsf{Adv}^{\mathsf{MLWE}}_{q,k,\ell,\chi}(\mathcal{A}, 1^\lambda) = \left| \Pr\left[\mathcal{A}\Big(A, [I_k \mid A] \cdot \vec{x}\Big) = 1\right] - \Pr\left[\mathcal{A}\Big(A, \vec{b}\Big) = 1\right] \right|$$

where $A \xleftarrow{\$} \mathcal{R}_q^{k \times \ell}, \vec{b} \xleftarrow{\$} \mathcal{R}_q^k$, *and* $\vec{x} \leftarrow \chi^{k+\ell}$. *The MLWE assumption states that any efficient adversary* $\mathcal{A}$ *has a negligible advantage. We simply write* $\mathsf{MLWE}_{q,k,\ell,\sigma}$ *when* $\mathcal{D}$ *is the discrete Gaussian distribution with parameter* σ.

Definition 3 (Hint-MLWE). *Let* k, ℓ, q, Q *integers and* χ, χ' *probability distributions over* $\mathcal{R}$, *and* $(\vec{v}_i)_{i \in [Q]}$ *vectors in* $\mathcal{R}^{k+\ell}$. *The advantage of an adversary* $\mathcal{A}$ *against the Hint-MLWE problem* $\mathsf{HMLWE}_{q,k,\ell,Q,\chi,\chi',(\vec{v}_i)_{i\in[Q]}}$ *is defined as:*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{HMLWE}}_{q,k,\ell,Q,\chi,\chi',(\vec{v}_i)_{i\in[Q]}}(\mathcal{A}) \\ &= \left| \Pr\left[\mathcal{A}\Big(A, [I_k \mid A] \cdot \vec{x}, (\vec{v}_i, z_i)_{i\in[Q]}\Big)\right] = 1 \;\; - \Pr\left[\mathcal{A}\Big(A, \vec{b}, (\vec{v}_i, z_i)_{i\in[Q]}\Big) = 1\right] \right|\end{aligned}$$

where $A \xleftarrow{\$} \mathcal{R}_q^{k \times \ell}, \vec{b} \xleftarrow{\$} \mathcal{R}_q^k$, $\vec{x} \leftarrow \chi^{k+\ell}$, *and* $z_i = \vec{v}_i^\top \vec{x} + y_i$ *where* $y_i \leftarrow \chi'$. *The* $\mathsf{HMLWE}_{q,k,\ell,Q,\chi,\chi',(\vec{v}_i)_{i\in[Q]}}$ *assumption states that any efficient adversary* $\mathcal{A}$ *has a negligible advantage. We write* $\mathsf{HMLWE}_{q,k,\ell,Q,\sigma_{\vec{x}},\sigma_y,(\vec{v}_i)_{i\in[Q]}}$ *when* $\chi = D_{\mathcal{R},\sigma_{\vec{x}}}$ *and* $\chi' = D_{\mathcal{R},\sigma_y}$.

In this work, we will only rely on a restrained version of the Hint-MLWE problem where the hint vectors $\vec{v}_i$ are standard basis vectors in $\mathcal{R}^{k+\ell}$, and there are Q copies of each standard basis vector. The following lemma directly follows from Theorem 1 in [KLSS23].

Lemma 4 (Hardness of HMLWE). *Let* ℓ, k, q, Q *be integers,* $\mathcal{E}$ *be the tuple of* $(k+\ell)Q$ *vectors containing* Q *copies of each* $(1, 0, \ldots, 0), \ldots, (0, 0, \ldots, 1) \in \mathcal{R}^{k+\ell}$, *and* $\sigma_{\vec{x}}$ *and* σ_y *be some positive reals.*

Let $\sigma > 0$ *defined as* $\sigma^{-2} = 2\left(\sigma_{\vec{x}}^{-2} + Q\sigma_y^{-2}\right)$. *If* $\sigma \ge \eta_\varepsilon(\mathbb{Z}^N)/\sqrt{\pi}$ *for some* $0 < \varepsilon \le 1/2$, *then there exists an efficient reduction from* $\mathsf{MLWE}_{q,k,\ell,\sigma}$ *to* $\mathsf{HMLWE}_{q,k,\ell,(k+\ell)Q,\sigma_{\vec{x}},\sigma_y,\mathcal{E}}$ *that reduces the advantage by at most* $2(k+\ell)\varepsilon$.

2.4 Definitions and Security Models for tPKE

In this section, we recall the syntax and security notions of Threshold Public-key Encryption (tPKE), as recently consolidated in [BKW26]. We focus on the *asynchronous* setting where the set of decryptors is not known at the time of partial decryptions.

Definition 4 (tPKE). *A t-out-of-n tPKE scheme is a tuple of algorithms* $\mathsf{tPKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{PartDec}, \mathsf{FinDec})$ *defined as follows, where $\mathcal{PT}, \mathcal{CT}$ and $\mathcal{RD}$ denote the plaintext, ciphertext, and encryption randomness spaces, respectively, and χ_{rnd} denotes the distribution over $\mathcal{RD}$ from which honest encryption draws its randomness:*

- $\mathsf{KeyGen}(1^\lambda, t, n)$**:** *Upon input the security parameter λ, threshold t, and number of parties n, outputs a public key* pk *and n shares* $\mathsf{sk}_1, \ldots, \mathsf{sk}_n$ *of the secret key.*
- $\mathsf{Enc}(\mathsf{pk}, m)$**:** *Upon input a public key* pk *and a message $m \in \mathcal{PT}$, samples* $\mathsf{rnd} \xleftarrow{\$} \chi_{\mathsf{rnd}}$ *and returns a ciphertext* $\mathsf{ct} \in \mathcal{CT}$*. We write* $\mathsf{Enc}(\mathsf{pk}, m; \mathsf{rnd})$ *to denote the (deterministic) encryption process using the specific encryption randomness* $\mathsf{rnd} \in \mathcal{RD}$*.*
- $\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct})$**:** *Takes as input the i-th key share* sk_i *and a ciphertext* $\mathsf{ct} \in \mathcal{CT}$*, and returns a partial decryption* pd_i*.*
- $\mathsf{FinDec}(\mathsf{ct}, \{\mathsf{pd}_i\}_{i\in\mathcal{I}})$**:** *Takes as input ciphertext* $\mathsf{ct} \in \mathcal{CT}$ *and partial decryptions* $\{\mathsf{pd}_i\}_{i\in\mathcal{I}}$ *of some subset $\mathcal{I} \subseteq [n]$ of size t, and returns a plaintext $m \in \mathcal{PT} \cup \{\perp\}$.*

Definition 5 (Correctness). *A t-out-of-n tPKE scheme* $\mathsf{tPKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{PartDec}, \mathsf{FinDec})$ *is correct if*

$$\Pr\left[\begin{array}{c} \mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, m) \\ m' \neq m : \mathsf{pd}_i \leftarrow \mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct}) \quad \text{for } i \in \mathcal{I} \\ m' \leftarrow \mathsf{FinDec}(\mathsf{ct}, \{\mathsf{pd}_i\}_{i\in\mathcal{I}}) \end{array}\right] \leq \mathrm{negl}(\lambda)$$

for every message $m \in \mathcal{PT}$ and every subset $\mathcal{I} \subseteq [n]$ of size t, where the probability is taken over the random coins used in KeyGen*,* Enc*, and* $\mathsf{PartDec}$*.*

We now define CPA and CCA security for tPKE. As discussed in the introduction, prior works considered various security models making trade-offs adapted to their construction. We already do not wish to rely on any trusted third party assumption, or hiding of the ciphertexts' randomness generated by the encryption oracle. It still remains several dimensions studied in [BKW26]: the type of security (indistinguishability vs simulation security), number of corruptions supported ($= t-1$ or $\leq t-1$), support of partial decryption of challenge ciphertexts, support of *honest* or *semi-malicious* ciphertexts, *static* or *adaptive* corruption of parties.

Semi-malicious Simulation Security. In this model, the adversary may supply any encryption randomness within the valid space—it need not follow the

scheme's prescribed distribution—and security is formalized via a simulator rather than mere indistinguishability. The simulator must produce the public key and the corrupt parties' secret key shares without access to honest parties' keys, and must faithfully simulate all partial decryptions seen by the adversary. This captures the general scenario where the encryptor is one of the adversarial parties or colludes with them, since the public key is public and any party can produce ciphertexts with arbitrary randomness.

In the Chosen-Ciphertext Attacks (CCA) game, the key difference is that the adversary can submit any ciphertext directly to the partial decryption oracle, without the restriction that it must be generated through EncO or ChalO.

Definition 6 (Semi-malicious CPA and CCA Security). *A threshold PKE scheme* tPKE *is simulation secure against chosen plaintext attacks (CPA) in the presence of a semi-malicious encryptor, denoted* $\mathsf{SemiMalSimCPA}$ *security, (resp. chosen ciphertext attacks (CCA), denoted* SimCCA *security) if for any PPT adversary* $\mathcal{A}$*, there exists a PPT simulator* $\mathcal{S}$ *such that the advantage*

$$\mathsf{Adv}^{\mathsf{SemiMalSimCPA}}_{\mathsf{tPKE}}(\mathcal{A},\mathcal{S}) := \Big|\Pr\Big[\mathsf{SemiMalSimCPA}^{0}_{\mathsf{tPKE},\mathcal{A},\mathcal{S}}(1^\lambda)=1\Big] - \Pr\Big[\mathsf{SemiMalSimCPA}^{1}_{\mathsf{tPKE},\mathcal{A},\mathcal{S}}(1^\lambda)=1\Big]\Big|$$

$$\textit{(resp.}\quad \mathsf{Adv}^{\mathsf{SimCCA}}_{\mathsf{tPKE}}(\mathcal{A},\mathcal{S}) := \Big|\Pr\Big[\mathsf{SimCCA}^{0}_{\mathsf{tPKE},\mathcal{A},\mathcal{S}}(1^\lambda)=1\Big] - \Pr\Big[\mathsf{SimCCA}^{1}_{\mathsf{tPKE},\mathcal{A},\mathcal{S}}(1^\lambda)=1\Big]\Big|\textit{)}$$

is negligible, where the security experiment is defined in Fig. 1 (resp. the same figure, dropping the boxed elements).

2.5 Shamir Secret Sharing

Secret sharings are useful primitives for threshold constructions. They distribute a secret among a set of participants, such that the secret can be reconstructed by a sufficiently large subset of parties (typically any subset of at least t parties). Among the various secret sharing schemes, Shamir Secret Sharing (SSS) is particularly appealing due to its simplicity and small share size. We consider below that the secret shared is stored at the evaluation point 0.

Definition 7 (Shamir Secret Sharing). *Let* $\mathcal{P}=[n]$ *be a set of participants,* $t \le n$ *be a threshold, and* $\mathcal{S}=\{\alpha_1,\ldots,\alpha_n\} \subset \mathcal{R}_q$ *be evaluation points such that* $-\alpha_j$ *and* $\alpha_i-\alpha_j$ *is invertible in* $\mathcal{R}_q$ *for all distinct* $0 \le i,j \le n$*. For* $\mathcal{I} \subseteq_t [n]$ *and* $i \in \mathcal{I}$*, the Lagrange coefficient is* $\lambda_i^{\mathcal{I}} := \prod_{j\neq i\in\mathcal{I}}(-\alpha_j)\cdot(\alpha_i-\alpha_j)^{-1} \bmod q$*. Then,* t*-out-of-*n *threshold Shamir Secret Sharing over* $\mathcal{R}_q$ *consists of the following two algorithms.*

- $\mathsf{SS.Share}_{t,n}(v)$*: Upon input a value* $v \in \mathcal{R}_q$*, the dealer samples* $r_j \leftarrow R_q$ *for* $1 \le j < t$*, and computes* $s_i = v + \sum_{j=1}^{t-1} r_j \cdot \alpha_i^j \bmod q$*. Output* $\{(s_i,\alpha_i)\}_{i\in[n]}$.

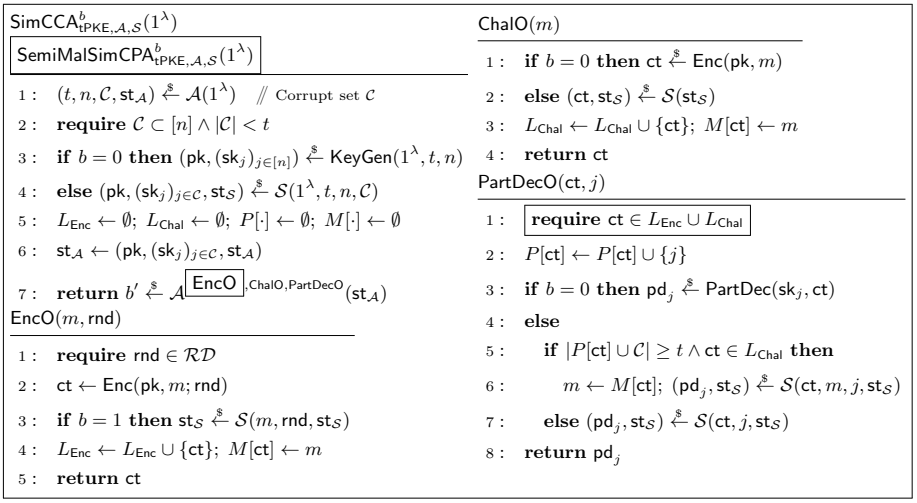

Fig. 1. Simulation-based semi-malicious CPA and CCA security experiments for tPKE. The boxed requirement in PartDecO is part of SemiMalSimCPA$^b_{\mathsf{tPKE},\mathcal{A},\mathcal{S}}$ only.

- $\mathsf{SS.Combine}_{t,n}(\{s_i\}_{i\in\mathcal{I}})$*: Upon input secret shares* $\{s_i\}_{i\in\mathcal{I}}$ *for any subset* $\mathcal{I} \subset [n]$ *such that* $|\mathcal{I}| = t$*, compute and output* $s = \sum_{i\in\mathcal{I}} \lambda_i^\mathcal{I} \cdot s_i \bmod q$

We extend this notation to the secret sharing of vectors as well. More precisely, for a vector $\vec{v} = (v_1, \ldots, v_k) \in \mathcal{R}_q^k$, the notation $\mathsf{SS.Share}_{t,n}(\vec{v})$ means applying $\mathsf{SS.Share}_{t,n}$ component-wise to each coordinate of $\vec{v}$.

Now we discuss how Shamir Secret Sharing can be leveraged in a thresholdized Regev-style encryption scheme. Let $\vec{s}_1, \ldots, \vec{s}_n$ be the shares of the secret vector $\vec{s} \in R^k$, *i.e.,* $\mathsf{SS.Share}_{t,n}(\vec{v}) = \{(\vec{s}_i, \alpha_i)\}_{i\in[n]}$. Then, for some user set $\mathcal{I} \subseteq_t [n]$ to decrypt an MLWE ciphertext $(b, \vec{a}) \in \mathcal{R}_q^{k+1}$, each i-th party outputs the partial decryption $d_i := \langle \vec{a}, \vec{s}_i \rangle + f_i \bmod q$ for some flooding noise f_i.

Upon receiving the partial decryptions $\{d_i\}_{i\in\mathcal{I}}$, we wish to leverage Lagrange interpolation to combine them. Naively, one would compute $b + \sum_{i\in\mathcal{I}} \lambda_i^\mathcal{I} d_i \bmod q$, which would yield $b + \langle \vec{a}, \vec{s} \rangle + \sum_{i\in\mathcal{I}} \lambda_i^\mathcal{I} f_i$. However, the Lagrange coefficients $\lambda_i^\mathcal{I}$ can have large norms, which would amplify the flooding noise $\sum_{i\in\mathcal{I}} \lambda_i^\mathcal{I} f_i$ and prevent correct decryption. To address this issue, prior works have introduced a clearing factor γ during decryption that ensures that the scaled Lagrange coefficients have small norms. Thus, parties compute $\gamma b + \sum_{i\in\mathcal{I}} \gamma \lambda_i^\mathcal{I} d_i \bmod q$, which becomes $\gamma \cdot (b + \langle \vec{a}, \vec{s} \rangle \bmod q) + \sum_{i\in\mathcal{I}} \gamma \lambda_i^\mathcal{I} f_i$ and allows correct decryption as long as $\gamma \lambda_i^\mathcal{I} f_i$ is small enough for all $i \in \mathcal{I}$.

In order to minimize the size of γ and the $\lambda_i^\mathcal{I}$ it is crucial to choose the α_i carefully. Selecting monomials $\alpha_i = X^i$ has so far provided the most compact parameters for this purpose [BFM+25, CLW25, KLL+26]. We adopt this choice and characterize it formally in Lem. 5.

Lemma 5 ([KLL+26, Thm 4.3]). *Assume q is odd. The monomial set $\mathcal{S} = \{\alpha_i = X^{i-1}\}_{1\le i\le 2N}$ over $\mathcal{R}$ satisfies the invertibility property of Def. 7, and for the clearing factor $\gamma = 2^{\lceil \log t\rceil}$, we define the bound B_λ and $B_{\lambda^{-1}}$ as some integers such that $\|\gamma\lambda_i^{\mathcal{I}}\|_1 \le B_\lambda$ and $\|(\lambda_i^{\mathcal{I}})^{-1}\|_1 \le B_{\lambda^{-1}}$ for all $i \in \mathcal{I}$. Then, the following holds for any $\mathcal{I} \subseteq_t [n]$.*

- $B_\lambda \le 2^{-t+\lceil \log t\rceil+1} \cdot \sqrt{N} \cdot \prod_{1\le k\le \lfloor t/2\rfloor} \sin^{-2}\left(\pi \frac{k}{2N}\right)$
- $B_{\lambda^{-1}} \le 2^{t-1} \cdot \sqrt{N} \cdot \prod_{1\le k\le \lfloor t/2\rfloor -1} \cos^{2}\left(\pi \frac{k}{2N}\right)$

3 Adaptive Hint-MLWE

In this section, we introduce Adaptive Hint-MLWE, a variant of Hint-MLWE in which the adversary can choose hint coefficients adaptively, and we provide a reduction to MLWE. Adaptive Hint-MLWE is a natural extension of the Hint-MLWE problem, but previous reduction techniques do not extend to this adaptive setting. We believe the problem is of independent interest and may be more broadly useful. In Def. 8 we give a formal definition of Adaptive Hint-MLWE.

Definition 8 (Adaptive Hint-MLWE). *Let k, ℓ, q, Q be integers, χ, χ' be probability distributions over $\mathcal{R}$, and $\mathcal{V}$ be a set over $\mathcal{R}^{k+\ell}$. The advantage of an adversary $\mathcal{A}$ making at most Q queries to the adaptive hint oracle $\mathcal{O}_{\mathsf{hint}}$ against the Adaptive Hint-MLWE problem $\mathsf{AHMLWE}_{q,k,\ell,Q,\chi,\chi',\mathcal{V}}$ is defined as:*

$$\mathsf{Adv}^{\mathsf{AHMLWE}}_{q,k,\ell,Q,\chi,\chi',\mathcal{V}}(\mathcal{A}) = \left| \Pr\left[\mathsf{Game}^{\mathsf{AHMLWE},0}(\mathcal{A}) = 1\right] - \Pr\left[\mathsf{Game}^{\mathsf{AHMLWE},1}(\mathcal{A}) = 1\right] \right|$$

where $\mathsf{Game}^{\mathsf{AHMLWE},b}(\mathcal{A})$ is shown in Fig. 2. The $\mathsf{AHMLWE}_{q,k,\ell,Q,\chi,\chi',\mathcal{V}}$ assumption states that any efficient adversary $\mathcal{A}$ has only negligible advantage. We may write $\mathsf{AHMLWE}_{q,k,\ell,Q,\sigma_{\vec{x}},\sigma_y,\mathcal{V}}$ as shorthand when χ and χ' are discrete Gaussian distributions with parameters $\sigma_{\vec{x}}$ and σ_y, respectively. We refer to the case $b = 0$ as the 'Random' case, and the case $b = 1$ as the 'Real' case.

Remark 1. Adaptive Hint-MLWE may allow the adversary to choose the hint coefficients from a set of integer matrices, as in [ENP24]. Since we don't need it here, we omit it for simplicity. Our reduction to MLWE in Thm. 1 adapts straightforwardly using the two norm of the matrix to define B_{hint}.

To prove the hardness of Adaptive Hint-MLWE, we adapt the rerandomization lemma from [KY16] into the discrete Gaussian setting. Roughly speaking, the rerandomization lemma allows one to take selective hints and process them to generate adaptive hints on-the-fly. To enable this, we first introduce some useful lemmas. See Sec. A for the proofs.

Lemma 6 (Simplified convolution Lemma, adapted from [dCKK+26]). *Let $\mathbf{M} \in \mathbb{Z}^{m\times n}$ be a matrix, $\sigma > 0$ be a real, and $\mathbf{\Sigma} \succeq 0$ be a matrix such that*

$$\frac{1}{\sigma^2} + \|\mathbf{M}\sqrt{\mathbf{\Sigma}^{-1}}\|_2^2 \le \frac{2\pi}{\eta_\varepsilon(\mathbb{Z}^n)^2}$$

$\mathsf{Game}^{\mathsf{AHMLWE},b}(\mathcal{A})$	$\mathcal{O}_{\mathsf{hint}}(\vec{v})$
1 : $\mathsf{ctr} := 0$	1 : **require** $\vec{v} \in \mathcal{V} \wedge \mathsf{ctr} < Q$
2 : $A \leftarrow \mathcal{R}_q^{k\times\ell}$, $\vec{x} \leftarrow \chi^{k+\ell}$	2 : $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$
3 : **if** $b = 0$ **then** $\vec{b} \leftarrow \mathcal{R}_q^k$	3 : $y \leftarrow \chi'$
4 : **if** $b = 1$ **then** $\vec{b} := [I_k \mid A] \cdot \vec{x}$	4 : **return** $\vec{v}^\top \vec{x} + y$
5 : **return** $\mathcal{A}^{\mathcal{O}_{\mathsf{hint}}}(A, \vec{b})$	

Fig. 2. Security experiments for the AHMLWE problem.

for some $0 < \varepsilon \leq 1/2$*. Then, the following distribution over* $\mathbb{Z}^m$

$$\{\mathbf{Me} + \mathbf{f} \mid \mathbf{e} \leftarrow D_{\mathbb{Z}^n,\sigma}, \mathbf{f} \leftarrow D_{\mathbb{Z}^m,\sqrt{\mathbf{\Sigma}}}\}$$

is within statistical distance 4ε *of* $D_{\mathbb{Z}^m,\sqrt{\sigma^2\mathbf{M}\mathbf{M}^\top+\mathbf{\Sigma}}}$.

Lemma 7 (Gaussian rerandomization, adapted from [KY16]). *Let* n, m *be positive integers,* $\mathbf{M} \in \mathbb{Z}^{m\times n}$*, and* $s, \sigma > 0$ *be reals. Let* $\mathbf{x} \in \mathbb{Z}^n$ *be an arbitrary vector and let* $\mathbf{y}$ *be sampled from* $D_{\mathbb{Z}^n,\sigma}$*. Then, there exists a PPT algorithm* $\mathsf{Rerand}(\mathbf{M}, \mathbf{x} + \mathbf{y}, \sigma, s)$ *that outputs* $\mathbf{Mx} + \mathbf{y}'$ *such that the distribution of* $\mathbf{y}'$ *is within statistical distance* 4ε *of* $D_{\mathbb{Z}^m,\sigma s}$*, if* $\sigma > \eta_\varepsilon(\mathbb{Z}^n)/\sqrt{2\pi}$ *and* $\|\mathbf{M}\|_2 \leq \sqrt{1 - \eta_\varepsilon^2(\mathbb{Z}^n)/(2\pi\sigma^2)} \cdot s$*.*

We show that the Adaptive Hint-MLWE problem is no easier than the MLWE problem. The crux of the argument is to rely on the noise-rerandomization technique from [KY16].

Theorem 1 (Hardness of AHMLWE). *For any integers* ℓ, k, q, N, Q*, set* $\mathcal{V} \subset \mathcal{R}^{k+\ell}$*, and positive reals* $B_{\mathsf{hint}}, \sigma, \sigma_{\vec{x}}, \sigma_y$ *such that*

- $\sum_{0\leq j<k+\ell} \|v_i\|_1^2 \leq B_{\mathsf{hint}}$ *for any* $\vec{v} = (v_i)_{0\leq i<k+\ell} \in \mathcal{V}$*,*
- $\sigma_y \geq \sqrt{2B_{\mathsf{hint}}/\pi} \cdot \eta_\varepsilon(\mathbb{Z}^N)$
- $\sigma \geq \eta_\varepsilon(\mathbb{Z}^N)/\sqrt{\pi}$
- $$\frac{1}{\sigma^2} = 2\left(\frac{1}{\sigma_{\vec{x}}^2} + \frac{\kappa^2 Q}{\sigma_y^2}\right) \quad \textit{for} \quad \kappa := \sqrt{\frac{1-\sqrt{1-2B_{\mathsf{hint}}\eta_\varepsilon^2(\mathbb{Z}^N)/(\pi\sigma_y^2)}}{\eta_\varepsilon^2(\mathbb{Z}^N)/(\pi\sigma_y^2)}}.$$

Then, the $\mathsf{AHMLWE}_{q,k,\ell,Q,\sigma_{\vec{x}},\sigma_y,\mathcal{V}}$ *problem is as hard as the* $\mathsf{MLWE}_{q,k,\ell,\sigma}$ *problem, with an advantage loss at most* $(4Q + 2(k + \ell))\varepsilon$*.*

Proof. Let $\mathcal{E}$ be the set of $(k + \ell)Q$ vectors containing Q copies of each $(1, 0, \ldots, 0), \ldots, (0, 0, \ldots, 1) \in \mathcal{R}^{k+\ell}$. Our proof strategy consists of constructing an adversary $\mathcal{B}$ against the $\mathsf{HMLWE}_{q,k,\ell,(k+\ell)Q,\sigma'_y,\mathcal{E}}$ problem from an adversary $\mathcal{A}$ against the problem $\mathsf{AHMLWE}_{q,k,\ell,Q,\sigma_{\vec{x}},\sigma_y,\mathcal{V}}$ by simulating adaptive hints from selective hints $x_i + y_i$ for $y_i \leftarrow D_{\sigma'_y}$ for some positive real σ'_y to be determined as a function of σ_y. We achieve this by using the rerandomization technique

from [KY16], which provides an algorithm $\mathsf{Rerand} : (\mathbf{M}, \mathbf{x}+\mathbf{y}) \mapsto \mathbf{M}\mathbf{x}+\mathbf{y}'$. We then rely on the reduction between $\mathsf{HMLWE}_{q,k,\ell,(k+\ell)Q,\sigma_{\vec{x}},\sigma'_y,\mathcal{E}}$ and $\mathsf{MLWE}_{q,k,\ell,\sigma}$ from Lem. 4.

Let us start by reducing $\mathsf{AHMLWE}_{q,k,\ell,Q,\sigma_{\vec{x}},\sigma_y,\mathcal{V}}$ to $\mathsf{HMLWE}_{q,k,\ell,(k+\ell)Q,\sigma_{\vec{x}},\sigma'_y,\mathcal{E}}$. Consider an adversary $\mathcal{A}$ against the $\mathsf{AHMLWE}_{q,k,\ell,Q,\sigma_{\vec{x}},\sigma_y,\mathcal{V}}$ problem. Next, let us define an adversary $\mathcal{B}$ against the $\mathsf{HMLWE}_{q,k,\ell,(k+\ell)Q,\sigma_{\vec{x}},\sigma'_y:=\sigma_y/\kappa,\mathcal{E}}$ problem with hints $z_{i,j} := x_j + y_{i,j}$ for $i \in [Q]$ and $0 \le j < k+\ell$ where $y_{i,j} \leftarrow D'_{\sigma_y}$. Now we simulate each i-th call $\mathcal{O}_{\mathsf{hint}}(\mathcal{V})$ by returning a polynomial where the coefficient vector is the output of $\mathsf{Rerand}\big(\mathbf{V}_i,\ \mathbf{z}_i,\ \sigma'_y,\ s,\ \delta\big)$ where $\mathbf{V}_i$ is a concatenation of negacyclic matrices $\mathbf{V}_{i,j}$ corresponding to the j-th entry $v_{i,j}$ of the query vector $\vec{v}_i$ for $0 \le j < k+\ell$, and $\mathbf{z}_i$ is a concatenation of coefficient vectors of each $z_{i,0}, \ldots, z_{i,k+\ell-1}$. We note that $\mathbf{V}_i \cdot \mathbf{z}_i$ is essentially the coefficient vector of the ring element $\vec{v}_i^{\top} \cdot (z_{i,j})_{0 \le j < k+\ell}$.

We observe that

$$\|\mathbf{V}_i\|_2^2 = \| \sum_{j=0}^{k+\ell-1} \mathbf{V}_{i,j}\mathbf{V}_{i,j}^{\top}\|_2 \le \sum_{j=0}^{k+\ell-1} \|\mathbf{V}_{i,j}\mathbf{V}_{i,j}^{\top}\|_2 \le \sum_{j=0}^{k+\ell-1} \|v_{i,j}\|_1^2$$
$$\le B_{\mathsf{hint}} = \kappa^2 \left(1 - \frac{\eta_{\varepsilon}^2(\mathbb{Z}^N)\kappa^2}{2\pi\sigma_y^2}\right)$$

where the last equality is derived from the definition of κ. On the other hand, $\sigma'_y > \eta_{\varepsilon}(\mathbb{Z}^N)/\sqrt{2\pi}$ since $2\sqrt{B_{\mathsf{hint}}} > \kappa$, by definition of κ as well.

Therefore, Lem. 7 applies, and the simulated $\mathcal{O}_{\mathsf{hint}}$ outputs a vector $\mathbf{V}_i\mathbf{x}+\mathbf{y}'_i$ where $\mathbf{x}$ is a concatenation of the coefficient vectors of the secret $\vec{x}$ and the distribution of $\mathbf{y}'_i$ is within statistical distance 4ε from $D_{\mathbb{Z}^N,\sigma_y}$. We remark that this can be seen as a coefficient vector of $\vec{v}^{\top}\cdot\vec{x}+y'_i$ for some ring element y'_i having $\mathbf{y}'_i$ as the coefficient vector. Therefore, $\mathcal{B}$ simulates the view of $\mathcal{A}$ up to statistical distance $Q \cdot 4\varepsilon$, and we conclude that if $\mathcal{A}$ breaks $\mathsf{AHMLWE}_{q,k,\ell,Q,\sigma_{\vec{x}},\sigma_y,\mathcal{V}}$, then $\mathcal{B}$ breaks $\mathsf{HMLWE}_{q,k,\ell,(k+\ell)Q,\sigma_{\vec{x}},\sigma'_y,\mathcal{E}}$.

Now we reduce $\mathsf{HMLWE}_{q,k,\ell,(k+\ell)Q,\sigma_{\vec{x}},\sigma'_y,\mathcal{E}}$ to $\mathsf{MLWE}_{q,k,\ell,\sigma}$. This directly follows from Lem. 4, whose smoothing requirement $\sigma \ge \eta_{\varepsilon}(\mathbb{Z}^N)/\sqrt{\pi}$ is the third assumption of the theorem, since

$$\frac{1}{\sigma^2} = 2\left(\frac{1}{\sigma_{\vec{x}}^2} + \frac{\kappa^2 Q}{\sigma_y^2}\right) = 2\left(\frac{1}{\sigma_{\vec{x}}^2} + \frac{Q}{{\sigma'_y}^2}\right),$$

as all the vectors $e \in \mathcal{E}$ satisfy $\|e\|_1 = 1$. Therefore, the overall advantage loss from Hint-MLWE to MLWE is $(4Q + 2(k+\ell))\varepsilon$, as claimed. □

4 Our Threshold PKE Scheme

In this section, we present our tPKE scheme and analyze its correctness and security. Our construction can be viewed as a direct thresholdization of the standard MLWE-based PKE scheme, which also serves as the underlying CPA-secure component of CRYSTALS-Kyber [BDK+18], combined with Shamir secret sharing

and a standard one-round distributed decryption protocol. To facilitate our security reduction, we introduce minor modifications to this standard template, specifically employing discrete Gaussian distributions for key and encryption randomness generation.

Although this design strategy is conceptually natural and similar approaches have been adopted in several prior works, formally proving its security under standard assumptions has remained an open challenge as we discussed in the introduction. This work is the first to formally prove the strongest form of CPA security of a direct thresholdization of the standard MLWE-based PKE scheme under standard assumptions.

4.1 Scheme Description

The construction of our tPKE scheme is given in Figure 3. We assume that the ciphertext modulus q is odd, ensuring in particular that $\gamma^{-1} \bmod q$ exists.

To enable asynchronous decryption, we perform the final recombination in FinDec after multiplying by γ, namely by adding γc_0 and scaling each partial decryption by γ times its corresponding Lagrange coefficient; the point of this factor is that, for the evaluation points we use, these effective coefficients remain small. This normalization trick is standard in this setting and already appears, for instance, in Pilvi [CLW25]. Accordingly, during encryption we compensate for this extra factor by multiplying the message by γ^{-1}. As a result, the original message is correctly reconstructed after final decryption. Crucially, this design allows the partial decryption algorithm to operate without knowing the decryption set in advance, as it does not require the Lagrange coefficients.

Moreover, we make partial decryption deterministic by deriving the flooding term from a PRF keyed by the party's secret state. This removes the need to restrict each party to at most one partial decryption per ciphertext, which Pilvi avoided by imposing exactly such a limitation.

4.2 Correctness and Security

In this section, we analyze the correctness and prove the security of our tPKE scheme. We will rely on the hardness of the Adaptive Hint-MLWE problem.

Theorem 2 (Correctness). *Assuming F is a secure PRF, the tPKE scheme described above is correct, except with negligible probability, if the following holds:*

$$\gamma \left(\rho_r \sigma_r + e^{1/2} \cdot 2kN\sigma_r\sigma_s \right) + tB_\lambda \rho_f \sigma_f < \frac{q}{2p} - \frac{p}{2},$$

where, for the target (negligible) failure probability ε and $\rho(\varepsilon, \cdot)$ as in Lem. 2, $\rho_r := \rho(\varepsilon, 1)$ and $\rho_f := \rho(\varepsilon, t)$, and B_λ is as defined in Lem. 5.

Proof. See Sec. A for the proof.

Threshold PKE

Given a Shamir Secret Sharing scheme $\mathsf{SS} = (\mathsf{Share}, \mathsf{Combine})$, we describe a CPA-secure threshold PKE scheme tPKE as follows.

• $\mathsf{KeyGen}(1^\lambda, t, n)$: Upon input the security parameter λ, threshold t, and number of parties n, sample the secret $\vec{s} \leftarrow D_{\sigma_s}^k$, and let $\{(\vec{s}_i, \alpha_i)\}_{i\in[n]} \leftarrow \mathsf{SS.Share}_{t,n}(\vec{s})$. Sample $A \leftarrow \mathcal{U}(\mathcal{R}_q^{k\times k})$ and $\vec{e} \leftarrow D_{\sigma_s}^k$, and let $\vec{b} = -A\vec{s} + \vec{e} \pmod q$. For $1 \le i \le n$, sample a PRF key K_i and send the public key $\mathsf{pk} = (\vec{b}, A) \in \mathcal{R}_q^{k\times(k+1)}$ and a key share $\mathsf{sk}_i = (\vec{s}_i, \alpha_i, K_i)$ to the i-th party.

• $\mathsf{Enc}(\mathsf{pk}, m)$: Upon input a public key $\mathsf{pk} = (\vec{b}, A)$, and a message $m \in \mathcal{R}_p$, sample $r_0 \leftarrow D_{\sigma_r}$, $\vec{r}_1 \leftarrow D_{\sigma_r}^k$ and $\vec{r}_2 \leftarrow D_{\sigma_r}^k$. Compute $c_0 = \vec{r}_2^\top \vec{b} + r_0 + \gamma^{-1} \cdot \lfloor \frac{q}{p} \rceil m \pmod q$ and $\vec{c}_1 = A^\top \vec{r}_2 + \vec{r}_1 \pmod q$, and output $\mathsf{ct} = (c_0, \vec{c}_1)$.

• $\mathsf{PartDec}(s_i, \mathsf{ct})$: Upon input a secret key share $s_i = (\vec{s}_i, \alpha_i, K_i)$, and a ciphertext $\mathsf{ct} = (c_0, \vec{c}_1) \in \mathcal{R}_q^{k+1}$, compute $f_{\mathsf{ct},i} := F(K_i, \mathsf{ct})$ and output $d_{\mathsf{ct},i} = \vec{c}_1^\top \vec{s}_i + f_{\mathsf{ct},i} \pmod q$. Here, $F : \mathsf{Keys} \times \mathcal{R}_q^{k+1} \to \mathcal{R}$ is a deterministic PRF whose output, for a uniformly random key, is required to be D_{σ_f}-distributed rather than uniform. This makes $f_{\mathsf{ct},i}$ fixed and reproducible for each (s_i, ct), so repeated partial decryptions of the same ciphertext by the same party remain consistent.

• $\mathsf{FinDec}(\mathsf{ct}, \{d_i\}_{i\in\mathcal{I}}, \mathcal{I})$: Upon input a ciphertext $\mathsf{ct} = (c_0, \vec{c}_1)$ and partial decryptions $\{d_i\}_{i\in\mathcal{I}}$ for some subset $\mathcal{I} \subseteq_t [n]$, compute and output the following:

$$m = \left\lfloor \frac{p}{q} \left(\gamma c_0 + \sum_{i\in\mathcal{I}} \gamma \lambda_i^{\mathcal{I}} d_i \bmod q \right) \right\rceil$$

Fig. 3. Our tPKE construction.

Theorem 3 (Security). *The* tPKE *scheme in Fig.3 is semi-malicious simulation secure against CPA attacks under the security of the PRF* F*, and the hardness of the problems* $\mathsf{AHMLWE}_{q,k,k,Q,\sigma_s,\sigma_f,\mathcal{V}_{B_1}}$, $\mathsf{AHMLWE}_{q,k+1,k,n,\sigma_r,\sigma_f,\mathcal{V}_{B_2}}$, *and* $\mathsf{AHMLWE}_{q,k+Q_c,k,Q,\sigma_1,\sigma_f,\mathcal{V}_{B_3}}$ *assumptions, for* $\sigma_1^2 = \frac{{\sigma_f}^2}{2\gamma^2}\left(1 + \sqrt{1 - \frac{2\gamma^2\eta_\varepsilon(\mathbb{Z}^N)^2}{\pi{\sigma_f}^2}}\right)$.

Formally, if there exists an adversary $\mathcal{A}$ *against the* $\mathsf{SemiMalSimCPA}$ *game making at most* Q *queries to the partial decryption oracle* $\mathsf{PartDecO}$ *and* Q_c *queries to the challenge oracle* ChalO*, and corrupting a set* $\mathcal{C}$ *of parties, there exists a simulator* $\mathcal{S}$*, a PRF distinguisher* $\mathcal{D}$*, and* AHMLWE *adversaries* $\mathcal{B}_1$, $\mathcal{B}_2$, *and* $\mathcal{B}_3$*, such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{SemiMalSimCPA}}_{\mathsf{tPKE}}(\mathcal{A},\mathcal{S}) \le & \mathsf{Adv}^{\mathsf{AHMLWE}}_{q,k,k,Q,\sigma_s,\sigma_f,\mathcal{V}_{B_1}}(\mathcal{B}_1) + Q_c \cdot \mathsf{Adv}^{\mathsf{AHMLWE}}_{q,k+1,k,n,\sigma_r,\sigma_f,\mathcal{V}_{B_2}}(\mathcal{B}_2) \\ & + (t-1) \cdot \mathsf{Adv}^{\mathsf{AHMLWE}}_{q,k+Q_c,k,Q,\sigma_1,\sigma_f,\mathcal{V}_{B_3}}(\mathcal{B}_3) + n \cdot \mathsf{Adv}^{\mathsf{PRF}}_F(\mathcal{D}) \\ & + Q_c \cdot 2^{-N(2k+1)/10} + 2^{-2kN/10} + 4(t-1)Q_c\varepsilon,\end{aligned}$$

where $B_1 = e^{1/2}N^2(2k+1)(B_{\lambda^{-1}}\sigma_r)^2$, $B_2 = B_{\lambda^{-1}}^2(1 + 2e^{1/2}kN^2{\sigma_s}^2)$, $B_3 = e^{1/2}N^2(2k+1){\sigma_r}^2\max(\gamma, B_{\lambda^{-1}}B_\lambda)^2$, *and* $B_\lambda, B_{\lambda^{-1}}$ *are as defined in Lem. 5.*

Proof. We use a sequence of games to move from the real world ($b = 0$) to the ideal world ($b = 1$) of the $\mathsf{SemiMalSimCPA}$ definition, where the challenger sim-

ulates the view of the adversary $\mathcal{A}$, bounding $\mathcal{A}$'s change in winning probability between adjacent games. We write $p_i := \Pr[\mathsf{Game}_{i,\mathcal{A}}(1^\lambda) = 1]$ for its winning probability in Game_i.

Throughout the proof, $\mathcal{C} \subseteq [n]$ denotes the set of corrupted parties and $\mathcal{C}' \supseteq \mathcal{C}$ a deterministically chosen extension with $|\mathcal{C}'| = t - 1$. We also fix the valid randomness space for semi-malicious EncO queries to be the two-norm ball $\mathcal{RD} := \{\vec{r} \in \mathcal{R}_q^{2k+1} : \|\vec{r}\|_2 \leq e^{1/4}\sigma_r\sqrt{N(2k+1)}\}$, matching the typical magnitude of an honestly-sampled $\vec{r} \leftarrow D_{\sigma_r}^{2k+1}$.

Game_0: This is the real world, *i.e.,* the case $b = 0$ of the simulation security game $\mathsf{SemiMalSimCPA}$: the public key and secret key shares are generated honestly, and all oracles answer with the real encryption and partial decryption algorithms. Thus $p_0 = \Pr[\mathsf{SemiMalSimCPA}^0_{\mathsf{tPKE},\mathcal{A},\mathcal{S}}(1^\lambda) = 1]$.

Game_1: For every honest party $i \notin \mathcal{C}$, we replace the PRF-derived flooding noise $f_{\mathsf{ct},i} := F(K_i, \mathsf{ct})$ with a value sampled directly from D_{σ_f}, once per queried pair (ct, i) and memoized after that. For $i \in \mathcal{C}$, $f_{\mathsf{ct},i}$ continues to be computed honestly via $F(K_i, \mathsf{ct})$.

Since K_i for $i \notin \mathcal{C}$ is never revealed, this swap is exactly what the security of F guarantees, applied one key at a time over the (at most) n keys:

$$|p_0 - p_1| \leq n \cdot \mathsf{Adv}_F^{\mathsf{PRF}}(\mathcal{D}),$$

for the PRF distinguisher $\mathcal{D}$ that simulates $\mathcal{A}$, answering $\mathsf{PartDecO}$ via its own oracle for the key currently being swapped and honestly otherwise.

Game_2: In this hybrid, we equivalently rewrite the partial decryption oracle $\mathsf{PartDecO}$ in terms of the public key, the shares of the parties in $\mathcal{C}'$, and the secret $\vec{s}$, so that the shares of parties $i \notin \mathcal{C}'$ are no longer directly used.

First, every ciphertext $\mathsf{ct} = (c_0, \vec{c}_1)$ queried to $\mathsf{PartDecO}$ must have been generated either by ChalO or by EncO, so there exist $m \in \mathcal{R}_p$ and $\vec{r} = (r_0, \vec{r}_1, \vec{r}_2) \in \mathcal{R}_q^{2k+1}$ such that $c_0 = \vec{r}_2^\top \vec{b} + r_0 + \gamma^{-1} \cdot \lfloor \frac{q}{p} \rceil m$ and $\vec{c}_1 = A^\top \vec{r}_2 + \vec{r}_1 \pmod q$; the challenger stores them when answering ChalO and EncO, and retrieves them on a $\mathsf{PartDecO}$ query for ct.

Also, as the key shares are generated via Shamir Secret Sharing, any share $\vec{s}_i$, $i \notin \mathcal{C}'$, is a Lagrange interpolation of the secret $\vec{s}$ and the shares of the parties in $\mathcal{C}'$: writing $\mu_i := (\lambda_i^{\mathcal{C}' \cup \{i\}})^{-1}$ and $\lambda_j^i := \lambda_j^{\mathcal{C}' \cup \{i\}}$ for $j \in \mathcal{C}'$ – both relative to the fixed index set $\mathcal{C}' \cup \{i\}$ – we have $\vec{s}_i = \mu_i(\vec{s} - \sum_{j \in \mathcal{C}'} \lambda_j^i \vec{s}_j)$. Consequently, the partial decryption of a ciphertext $\mathsf{ct} = (c_0, \vec{c}_1)$ for an honest party $i \notin \mathcal{C}'$ is $d_{\mathsf{ct},i} = \vec{c}_1^\top \vec{s}_i + f_{\mathsf{ct},i} = \mu_i \vec{c}_1^\top \vec{s} + f_{\mathsf{ct},i} - \mu_i \vec{c}_1^\top \big(\sum_{j \in \mathcal{C}'} \lambda_j^i \vec{s}_j \big)$. Substituting $\vec{b} = -A\vec{s} + \vec{e} \pmod q$ and $\vec{c}_1 = A^\top \vec{r}_2 + \vec{r}_1$ further expands the term involving the secret $\vec{s}$ as $\vec{c}_1^\top \vec{s} = \vec{r}_2^\top(-\vec{b} + \vec{e}) + \vec{r}_1^\top \vec{s}$, where $\vec{r}_2^\top \vec{b}$ is computable from the encryption randomness and the ciphertext, so that we finally obtain

$$d_{\mathsf{ct},i} = -\mu_i \vec{r}_2^\top \vec{b} + \mu_i(\vec{r}_2^\top \vec{e} + \vec{r}_1^\top \vec{s}) + f_{\mathsf{ct},i} - \mu_i \cdot \vec{c}_1^\top \big(\sum_{j \in \mathcal{C}'} \lambda_j^i \vec{s}_j \big).$$

This change is purely syntactic, and thus $p_2 = p_1$.

Game_3: In this game, we modify the key generation algorithm by sampling the shares of the parties in $\mathcal{C}'$ uniformly at random in $\mathcal{R}_q^k$ instead of generating them via Shamir Secret Sharing, and we drop the remaining shares of the honest parties. Since there are exactly $t-1$ parties in $\mathcal{C}'$, their shares are already distributed uniformly at random in $\mathcal{R}_q^k$ and independently of the secret $\vec{s}$ in Game_2, where moreover the shares of parties outside $\mathcal{C}'$ are no longer used and can simply be dropped. The distribution of the adversary's view is therefore unchanged: $p_3 = p_2$.

Game_4: In this game, we replace the public key vector $\vec{b} = -A\vec{s}+\vec{e} \pmod q$ with a uniformly random value in $\mathcal{R}_q^k$, and show that this is indistinguishable under the AHMLWE assumption.

Observe that except for their use in $\vec{b}$, the secret vectors $\vec{s}$ and $\vec{e}$ were only used in the computation of the partial decryptions for honest parties $i \notin \mathcal{C}'$, through the term $\mu_i(\vec{r}_2^\top \vec{e} + \vec{r}_1^\top \vec{s}) + f_{\mathsf{ct},i}$. This term is exactly the response of the hint oracle $\mathcal{O}_{\mathsf{hint}}$ for $\mathsf{AHMLWE}_{q,k,k,Q,\sigma_s,\sigma_f,\mathcal{V}_B}$ on input $\vec{v} := \mu_i \cdot (\vec{r}_2, -\vec{r}_1) \in \mathcal{V}_B$, with B defined below. We then set $\vec{b}$ to the MLWE sample of the AHMLWE instance, identifying $\vec{e}$ with the first half and $-\vec{s}$ with the second half of its secret, since $\vec{b} = -A\vec{s} + \vec{e}$.

Let us now define B. The randomness $\vec{r}$ underlying ct lies in $\mathcal{RD}$: by definition of the game when ct is produced by EncO, and, when ct is produced by ChalO, except with probability at most $2^{-N(2k+1)/10}$ over $\vec{r} \leftarrow D_{\sigma_r}^{2k+1}$ by Lem. 3. Since $(\vec{r}_2, \vec{r}_1)$ is a subset of the coordinates of $\vec{r}$, Corollary 1 gives $\sum_u \|r_u\|_1^2 \le N\|\vec{r}\|_2^2 \le e^{1/2}N^2(2k+1){\sigma_r}^2$ for its $2k$ coordinates r_u. Together with $\|\mu_i\|_1 \le B_{\lambda^{-1}}$ (Lem. 5) and submultiplicativity of the one-norm, this gives $\sum_u \|v_u\|_1^2 \le B_{\lambda^{-1}}^2 e^{1/2}N^2(2k+1){\sigma_r}^2$ for the $2k$ coordinates v_u of $\vec{v}$, so that $\vec{v} \in \mathcal{V}_B$ for $B = e^{1/2}N^2(2k+1)(B_{\lambda^{-1}}\sigma_r)^2$. Note that we crucially rely here on the adaptive nature of the AHMLWE problem, as $\vec{v}$ is determined by the encryption randomness $\vec{r}$, which $\mathcal{A}$ chooses adaptively when it queries EncO – in that case, $\vec{r}$ need not follow the prescribed distribution D_{σ_r}, only the bound enforced by $\mathcal{RD}$, which is exactly what the bound on $\vec{v}$ relies on instead.

Hence, taking a union bound over the (at most Q_c) challenge ciphertexts whose randomness may fall outside $\mathcal{RD}$,

$$|p_3 - p_4| \le \mathsf{Adv}^{\mathsf{AHMLWE}}_{q,k,k,Q,\sigma_s,\sigma_f,\mathcal{V}_B}(\mathcal{B}) + Q_c \cdot 2^{-N(2k+1)/10},$$

where $\mathcal{B}$ internally simulates $\mathcal{A}$ and answers its queries as described above. Note that the games keep sampling $\vec{s}$ and $\vec{e}$, even though they are no longer related to the public key: they remain important for the simulation of the partial decryptions.

Game_5: In this game, we equivalently rewrite the partial decryptions of a challenge ciphertext $\mathsf{ct} = (c_0, \vec{c}_1)$ as a function of the challenge ciphertexts, the shares of parties in $\mathcal{C}'$, and a small hint vector on the ciphertext randomness. For $i \in \mathcal{C}'$, $d_{\mathsf{ct},i} = \vec{c}_1^\top \vec{s}_i + f_{\mathsf{ct},i}$ is already of the desired form. For $i \notin \mathcal{C}'$, we use the fact that $\vec{r}_2^\top \vec{b} = c_0 - r_0 - \gamma^{-1} \cdot \lfloor \frac{q}{p} \rceil m$ and $\vec{c}_1 = A^\top \vec{r}_2 + \vec{r}_1 \pmod q$ to rewrite the partial

decryption as

$$\begin{aligned} d_{\mathsf{ct},i} &= \mu_i(-\vec{r}_2^\top \vec{b} + \vec{r}_2^\top \vec{e} + \vec{r}_1^\top \vec{s}) + f_{\mathsf{ct},i} - \mu_i \cdot \vec{c}_1^\top \big(\sum_{j \in \mathcal{C}'} \lambda_j^i \vec{s}_j \big) \\ &= -\mu_i(c_0 - \gamma^{-1} \cdot \lfloor \frac{q}{p} \rceil m) + \underbrace{\mu_i(r_0 + \vec{r}_2^\top \vec{e} + \vec{r}_1^\top \vec{s}) + f_{\mathsf{ct},i}}_{\text{hint on } (\vec{r}_0, \vec{r}_2, \vec{r}_1)} - \mu_i \cdot \vec{c}_1^\top \big(\sum_{j \in \mathcal{C}'} \lambda_j^i \vec{s}_j \big). \end{aligned}$$

Again, this change is purely syntactic, and thus $p_5 = p_4$.

Game_6: In this game, we replace the challenge ciphertexts produced by ChalO with uniformly random values in $\mathcal{R}_q^{k+1}$. As for the vectors $\vec{s}, \vec{e}$, the game keeps sampling a corresponding randomness $\vec{r} = (\vec{r}_0, \vec{r}_1, \vec{r}_2)$, needed for the simulation of the partial decryptions, but the ciphertexts themselves are now independent of $\vec{r}$.

This change is again indistinguishable under the AHMLWE assumption. Indeed, a challenge ciphertext ct is an MLWE sample with secret $\vec{r}_2$ and error $(r_0, \vec{r}_1)$ for the public matrix $A' = (b, A^\top)$, b uniformly random in $\mathcal{R}_q^k$; and besides the computation of ct, its randomness $\vec{r}$ is only used in the partial decryptions of parties $i \notin \mathcal{C}'$, through the term $\mu_i(r_0 + \vec{r}_2^\top \vec{e} + \vec{r}_1^\top \vec{s}) + f_{\mathsf{ct},i}$ – at most n such terms per ct, as the outputs of $\mathsf{PartDecO}(\mathsf{ct}, i)$ are memoized. This term can be seen as the response of the hint oracle $\mathcal{O}_{\mathsf{hint}}$ for $\mathsf{AHMLWE}_{q,k+1,k,n,\sigma_r,\sigma_f,\mathcal{V}_B}$ on input $\vec{v} := \mu_i \cdot (1, \vec{s}, \vec{e})$.

Here the $2k$ last coordinates of $\vec{v}$ are never adversarially chosen: $\vec{s}, \vec{e} \leftarrow D_{\sigma_s}^k$ are honestly sampled. By Corollary 1, except with probability at most $2^{-2kN/10}$, the $2k$ coordinates s_u of $(\vec{s}, \vec{e})$ satisfy $\sum_u \|s_u\|_1^2 \leq e^{1/2} \cdot 2kN^2{\sigma_s}^2$. Since $\|\mu_i\|_1 \leq B_{\lambda^{-1}}$ by Lem. 5, one-norm submultiplicativity then guarantees that $\vec{v} \in \mathcal{V}_B$ for $B = B_{\lambda^{-1}}^2(1 + 2e^{1/2}kN^2{\sigma_s}^2)$.

Applying this argument to all challenge ciphertexts, and bounding $\mathcal{A}$'s advantage trivially by 1 should the above bound on $(\vec{s}, \vec{e})$ fail, Game_5 is computationally indistinguishable from Game_6 under the AHMLWE assumption:

$$|p_5 - p_6| \leq Q_c \cdot \mathsf{Adv}^{\mathsf{AHMLWE}}_{q,k+1,k,n,\sigma_r,\sigma_f,\mathcal{V}_B}(\mathcal{B}_2) + 2^{-2kN/10},$$

where $\mathcal{B}_2$ internally simulates $\mathcal{A}$ and answers its queries as described above.

Game_7: This hybrid performs syntactical rewrites to prepare for the application of the AHMLWE assumption in Game_9, where we argue that the partial decryptions of challenge ciphertexts by honest parties in $\mathcal{C}' \backslash \mathcal{C}$ are computationally independent of the challenge ciphertexts themselves.

Recall that such a partial decryption is $d_{\mathsf{ct},i} = \vec{c}_1^\top \vec{s}_i + f_{\mathsf{ct},i}$, for the uniformly sampled $\vec{c}_1$. In order to later interpret it as an AHMLWE sample (using $\vec{c}_1$ as public matrix, and $\vec{s}_i$ as secret), we need to gather all dependencies on the secret shares $\vec{s}_i$ of parties in $\mathcal{C}' \backslash \mathcal{C}$ – all of which lie within the partial decryption oracle $\mathsf{PartDecO}$ – and ensure each can be interpreted either as an MLWE sample, or as a hint. We shall rely on two families of samples derived from $\vec{s}_i$, $i \in \mathcal{C}'$: the samples $\vec{b}_i = A \cdot \vec{s}_i + \gamma \vec{e}_i$, where $\vec{e}_i \leftarrow \mathcal{U}(\mathcal{R}_q^k)$ is a fresh uniformly random error term (this uniform noise is explicitly handled in Game_9 when invoking AHMLWE); and the

partial decryptions $d_{\mathsf{ct},i} = \vec{c}_1^{\top}\vec{s}_i + f_{\mathsf{ct},i}$ of the challenge ciphertexts ct produced by ChalO.

We distinguish four cases, according to whether ct was generated by EncO or by ChalO, and whether $i \in \mathcal{C}'$ or not. In Cases 2 and 4, we use the linearity $\vec{c}_1^{\top}\big(\sum_{j\in\mathcal{C}'}\lambda_j^i\vec{s}_j\big) = \sum_{j\in\mathcal{C}'}\lambda_j^i\vec{c}_1^{\top}\vec{s}_j$ and substitute $\vec{c}_1^{\top}\vec{s}_j$ either by the MLWE sample of Case 1, or by $d_{\mathsf{ct},j} - f_{\mathsf{ct},j}$.

Case 1, ct generated by EncO and $i \in \mathcal{C}'$: using $\vec{c}_1 = A^{\top}\vec{r}_2 + \vec{r}_1$ for some $\vec{r} \in \mathcal{RD}$,

$$d_{\mathsf{ct},i} = (A^{\top}\vec{r}_2 + \vec{r}_1)^{\top}\vec{s}_i + f_{\mathsf{ct},i} = \vec{r}_2^{\top}(\underbrace{A\cdot\vec{s}_i + \gamma\vec{e}_i}_{\vec{b}_i}) + (\vec{r}_1^{\top}\vec{s}_i - \vec{r}_2^{\top}\gamma\vec{e}_i + f_{\mathsf{ct},i}).$$

Case 2, ct generated by EncO and $i \notin \mathcal{C}'$: recall from Game_2 that $d_{\mathsf{ct},i} = -\mu_i\vec{r}_2^{\top}\vec{b} + \mu_i(\vec{r}_2^{\top}\vec{e} + \vec{r}_1^{\top}\vec{s}) + f_{\mathsf{ct},i} - \mu_i\cdot\vec{c}_1^{\top}\big(\sum_{j\in\mathcal{C}'}\lambda_j^i\vec{s}_j\big)$. Substituting Case 1's expression for $\vec{c}_1^{\top}\vec{s}_j$, $j \in \mathcal{C}'$, its last term becomes $\vec{c}_1^{\top}\big(\sum_{j\in\mathcal{C}'}\lambda_j^i\vec{s}_j\big) = \vec{r}_2^{\top}\sum_{j\in\mathcal{C}'}\lambda_j^i\vec{b}_j + \sum_{j\in\mathcal{C}'}\lambda_j^i(\vec{r}_1^{\top}\vec{s}_j - \vec{r}_2^{\top}\gamma\vec{e}_j)$. $d_{\mathsf{ct},i}$ is thus a function of $\vec{b}$, $\vec{r}$, $\vec{e}$, $\vec{s}$ and of the MLWE samples $\vec{b}_j = A\cdot\vec{s}_j + \gamma\vec{e}_j$, $j \in \mathcal{C}'$, to which we add a hint vector on the ciphertext randomness: $-\mu_i\cdot\sum_{j\in\mathcal{C}'}\lambda_j^i(\vec{r}_1^{\top}\vec{s}_j - \vec{r}_2^{\top}\gamma\vec{e}_j) + f_{\mathsf{ct},i}$.

Case 3, ct generated by ChalO and $i \in \mathcal{C}'$: the partial decryption $d_{\mathsf{ct},i} = \vec{c}_1^{\top}\vec{s}_i + f_{\mathsf{ct},i}$ is directly of the desired form, as ct is a challenge ciphertext.

Case 4, ct generated by ChalO and $i \notin \mathcal{C}'$: Game_5 computes

$$d_{\mathsf{ct},i} = -\mu_i(c_0 - \gamma^{-1}\cdot\lfloor\frac{q}{p}\rceil m) + \mu_i(r_0 + \vec{r}_2^{\top}\vec{e} + \vec{r}_1^{\top}\vec{s}) + f_{\mathsf{ct},i} - \mu_i\vec{c}_1^{\top}\big(\sum_{j\in\mathcal{C}'}\lambda_j^i\vec{s}_j\big),$$

whose last term becomes $\sum_{j\in\mathcal{C}'}\lambda_j^i(d_{\mathsf{ct},j} - f_{\mathsf{ct},j})$ once we substitute $\vec{c}_1^{\top}\vec{s}_j = d_{\mathsf{ct},j} - f_{\mathsf{ct},j}$. Hence $d_{\mathsf{ct},i}$ is a function of $\vec{b}$, $\vec{r}$, $\vec{e}$, $\vec{s}$, c_0, m and the partial decryptions $d_{\mathsf{ct},j}$ of the challenge ciphertext for parties $j \in \mathcal{C}'$, to which we add $\mu_i\sum_{j\in\mathcal{C}'}\lambda_j^i f_{\mathsf{ct},j} + f_{\mathsf{ct},i}$.

This change is purely syntactic, and thus $p_7 = p_6$.

Game_8: This game is again just a preparation to apply AHMLWE in the next hybrid. For each $i \in \mathcal{C}'\backslash\mathcal{C}$, we rewrite the share $\vec{s}_i$ and the error terms $(f_{\mathsf{ct},i})_{\mathsf{ct}}$ (ct produced by ChalO) introduced in Game_7, pulling out a factor of γ from each so that the coefficient $\mu_i(\gamma\lambda_j^i)$ multiplying them in the partial decryption simulation (see Game_9) admits a small representative.

First, since $\vec{s}_i$ is sampled uniformly in $\mathcal{R}_q^k$ and multiplication by the invertible γ is a bijection of $\mathcal{R}_q$, we may equivalently sample a fresh $\vec{s}_i' \leftarrow \mathcal{U}(\mathcal{R}_q^k)$ and set $\vec{s}_i := \gamma\vec{s}_i'$.

Second, provided $\sigma_f \geq \sqrt{2/\pi}\cdot\gamma\eta_\varepsilon(\mathbb{Z}^N)$, set $\delta := \sqrt{1 - 2\gamma^2\eta_\varepsilon(\mathbb{Z}^N)^2/(\pi{\sigma_f}^2)}$ and $\sigma_1^2 := \frac{{\sigma_f}^2}{2\gamma^2}(1+\delta)$, $\sigma_2^2 := \frac{{\sigma_f}^2}{2}(1-\delta)$, so that $1/\sigma_1^2 + \gamma^2/\sigma_2^2 = 2\pi/\eta_\varepsilon(\mathbb{Z}^N)^2$ and $\gamma^2\sigma_1^2 + \sigma_2^2 = {\sigma_f}^2$ – this is the largest value of σ_1 for which such a σ_2 exists. By Lem. 6 applied with $\mathbf{M} = \gamma\mathbf{I}_N$, the distribution of $\gamma g_1 + g_2$, for independent $g_1 \leftarrow D_{\sigma_1}$ and $g_2 \leftarrow D_{\sigma_2}$, is within statistical distance 4ε of D_{σ_f}. We thus replace each coefficient of each $f_{\mathsf{ct},i}$, $i \in \mathcal{C}'\backslash\mathcal{C}$, by a fresh $f_{\mathsf{ct},i} := \gamma f'_{\mathsf{ct},i} + f''_{\mathsf{ct},i}$ with

$f'_{\mathsf{ct},i} \leftarrow D_{\sigma_1}$ and $f''_{\mathsf{ct},i} \leftarrow D_{\sigma_2}$, independently across parties and ciphertexts. A union bound over the at most $(t-1)Q_c$ coefficients so replaced gives

$$|p_7 - p_8| \leq 4(t-1)Q_c\varepsilon.$$

Game_9: In this game, we replace the MLWE samples $\vec{b}_i$ and $d_{\mathsf{ct},i}$ defined in Game_7, for $i \in \mathcal{C}'\backslash\mathcal{C}$ and challenge ciphertexts ct produced by ChalO, with uniformly random values in $\mathcal{R}_q^k$ and $\mathcal{R}_q$ respectively.

This change is indistinguishable under the AHMLWE assumption, replacing these samples iteratively for each $i \in \mathcal{C}'\backslash\mathcal{C}$. Indeed, writing $\vec{b}_i = \gamma(A\vec{s}'_i + \vec{e}_i)$ and $d_{\mathsf{ct},i} = \gamma(\vec{c}_1^{\top}\vec{s}'_i + f'_{\mathsf{ct},i}) + f''_{\mathsf{ct},i}$, we can see the vector $(\vec{b}_i, (d_{\mathsf{ct},i})_{\mathsf{ct}\text{ produced by }\mathsf{ChalO}})$ as γ times an MLWE sample with secret $\vec{s}'_i$ and error $(\vec{e}_i, (f'_{\mathsf{ct},i})_{\mathsf{ct}})$ for the public matrix $A' = (A, (\vec{c}_1)_{\mathsf{ct}})$, shifted by the independent, freshly-sampled $(f''_{\mathsf{ct},i})_{\mathsf{ct}}$. Since $i \notin \mathcal{C}$, the shares $\vec{s}'_i$ and $\vec{e}_i$ are never revealed to the adversary, and besides the above samples they, together with $(f'_{\mathsf{ct},i})_{\mathsf{ct}}$, are only used in the simulation of the partial decryptions, through the terms **(i)** $\gamma(\vec{r}_1^{\top}\vec{s}'_i - \vec{r}_2^{\top}\vec{e}_i) + f_{\mathsf{ct},i}$ for ciphertexts generated by EncO and parties $i \in \mathcal{C}'$; **(ii)** $-\mu_i(\gamma\lambda_j^i)(\vec{r}_1^{\top}\vec{s}'_j - \vec{r}_2^{\top}\vec{e}_j) + f_{\mathsf{ct},i}$ for ciphertexts generated by EncO and parties $i \notin \mathcal{C}'$; and **(iii)** $\mu_i(\gamma\lambda_j^i)f'_{\mathsf{ct},j} + \mu_i\lambda_j^i f''_{\mathsf{ct},j} + f_{\mathsf{ct},i}$ for ciphertexts generated by ChalO and parties $i \notin \mathcal{C}'$.

This is almost directly an instance of the AHMLWE problem, except that we defined the latter with a single distribution for all secret polynomials, while here $\vec{s}'_i, \vec{e}_i$ are uniform and the errors $f'_{\mathsf{ct},i}$ are sampled from D_{σ_1}. This is not an issue: since $\mathcal{U}(\mathcal{R}_q) = \mathcal{U}(\mathcal{R}_q) + D_{\sigma_1} \bmod q$, adding a fresh uniform shift to the secret of a Gaussian-secret sample turns it into a uniform-secret one, so the uniform-secret game is no easier than AHMLWE.

Each of (i)–(iii) is itself a hint query on the secret, which has two parts: $(\vec{s}'_i, \vec{e}_i)$, and the Q_c coordinates $(f'_{\mathsf{ct},i})_{\mathsf{ct}}$, one for each challenge ciphertext. Queries (i) and (ii) concern the $(\vec{s}'_i, \vec{e}_i)$ part, and only arise for ciphertexts produced by EncO, where $\vec{r}_1, \vec{r}_2$ are supplied by the adversary from $\mathcal{RD}$: as in Game_4, the same Cauchy-Schwarz step gives $\sum_u \|r_u\|_1^2 \leq e^{1/2}N^2(2k+1){\sigma_r}^2$ for the $2k$ coordinates r_u of $(\vec{r}_1, \vec{r}_2)$, so that, by submultiplicativity and since the scalars γ and $\mu_i(\gamma\lambda_j^i)$ have one-norm at most γ and $B_{\lambda^{-1}}B_\lambda$ respectively ($B_\lambda, B_{\lambda^{-1}}$ as in Lem. 5), $\sum_u \|v_u\|_1^2 \leq B := e^{1/2}N^2(2k+1){\sigma_r}^2 \max(\gamma, B_{\lambda^{-1}}B_\lambda)^2$ in both cases. Query (iii) instead concerns a single coordinate $f'_{\mathsf{ct},j}$ of the second part, with scalar coefficient $\mu_i(\gamma\lambda_j^i)$, so that $\|v\|_1^2 \leq (B_{\lambda^{-1}}B_\lambda)^2 \leq B$, as $e^{1/2}N^2(2k+1){\sigma_r}^2 \geq 1$ for any meaningful choice of parameters. Hence $\mathcal{V}_B$ covers every hint query that arises in this game.

Formally, there exists an adversary $\mathcal{B}_3$ against $\mathsf{AHMLWE}_{q,k+Q_c,k,Q,\sigma_1,\sigma_f,\mathcal{V}_B}$, making up to Q hint queries and internally simulating $\mathcal{A}$ as described above, such that

$$|p_8 - p_9| \leq (t-1) \cdot \mathsf{Adv}^{\mathsf{AHMLWE}}_{q,k+Q_c,k,Q,\sigma_1,\sigma_f,\mathcal{V}_B}(\mathcal{B}_3).$$

Game_{10}: In this game, we focus on the partial decryptions of the challenge ciphertexts. For the first $t-1-|\mathcal{C}|$ queries for parties $j \notin \mathcal{C}$, we return a uniformly random value in $\mathcal{R}_q$ instead of the simulated partial decryption, and we answer the remaining queries so as to preserve consistency with the previous game.

Recall that, by construction, Game_9 produces the partial decryptions of a challenge ciphertext $\mathsf{ct} = (c_0, \vec{c}_1)$ for *every* party $i \in [n]$ as follows:

- $d_{\mathsf{ct},i} = \vec{c}_1^{\top}\vec{s}_i + f_{\mathsf{ct},i}$ for $i \in \mathcal{C}$ – the real partial decryption, computable by the adversary itself from its corrupted share $\vec{s}_k$.
- $d_{\mathsf{ct},i} \leftarrow \mathcal{R}_q$ for $i \in \mathcal{C}'\backslash\mathcal{C}$.
- $d_{\mathsf{ct},i} = -\mu_i(c_0 - \gamma^{-1} \cdot \lfloor \frac{q}{p} \rceil m) + \mu_i(r_0 + \vec{r}_2^{\top}\vec{e} + \vec{r}_1^{\top}\vec{s}) + f_{\mathsf{ct},i} - \mu_i \sum_{j\in\mathcal{C}'} \lambda_j^i(d_{\mathsf{ct},j} - f_{\mathsf{ct},j})$ for $i \notin \mathcal{C}'$.

We notice that the $d_{\mathsf{ct},i} - f_{\mathsf{ct},i}$ are the evaluations of a single polynomial Q_{ct} of degree $< t$: $Q_{\mathsf{ct}}(0) = c_0 - \gamma^{-1} \cdot \lfloor \frac{q}{p} \rceil m + r_0 + \vec{r}_2^{\top}\vec{e} + \vec{r}_1^{\top}\vec{s}$, $Q_{\mathsf{ct}}(\alpha_i) = \vec{c}_1^{\top}\vec{s}_i$ for $i \in \mathcal{C}$, and $Q_{\mathsf{ct}}(\alpha_i) = d_{\mathsf{ct},i} - f_{\mathsf{ct},i}$ for $i \in \mathcal{C}'\backslash\mathcal{C}$. Indeed, for $i \notin \mathcal{C}'$, since $Q_{\mathsf{ct}}(0) = \sum_{j\in\mathcal{C}'} \lambda_j^i Q_{\mathsf{ct}}(\alpha_j) + \mu_i^{-1} Q_{\mathsf{ct}}(\alpha_i)$, we can solve for $Q_{\mathsf{ct}}(\alpha_i)$ and obtain exactly the formula for $d_{\mathsf{ct},i} - f_{\mathsf{ct},i}$ above.

Of these three conditions, only the first two are actual constraints: $Q_{\mathsf{ct}}(0)$ and $\{Q_{\mathsf{ct}}(\alpha_i)\}_{i\in\mathcal{C}}$ are fixed by the vectors $\vec{e}, \vec{s}, \vec{r}$, the message, and the corrupted shares. The third one is not: it samples Q_{ct} uniformly at the $t - 1 - |\mathcal{C}|$ further points $\mathcal{C}'\backslash\mathcal{C}$. Since Q_{ct} has degree $< t$, this is exactly the number of degrees of freedom it has left after fixing its value at 0 and at $\mathcal{C}$, and since $\mathcal{S}$ gives valid Lagrange coefficients for every index set of size t, its evaluations at *any* $t-1-|\mathcal{C}|$ further points are in fact uniform in $\mathcal{R}_q$ and mutually independent.

The points $\mathcal{C}'\backslash\mathcal{C}$ therefore play no distinguished role: they can equally well be replaced by the first $t-1-|\mathcal{C}|$ honest parties the adversary happens to query, exactly as Game_{10} does. This only changes which evaluations of Q_{ct} are sampled directly and which are recovered by interpolation, leaving the distribution of Q_{ct} unchanged – and hence also that of the partial decryptions $d_{\mathsf{ct},i} = Q_{\mathsf{ct}}(\alpha_i) + f_{\mathsf{ct},i}$. Therefore $p_{10} = p_9$.

The simulator and conclusion. By Game_{10}, every oracle response is produced without ever using an honest party's secret key share, and the message m of a challenge ciphertext is needed only to answer a partial decryption query once the threshold has effectively been met – exactly the requirements on the simulator $\mathcal{S}$ of $\mathsf{SemiMalSimCPA}^1$. We define $\mathcal{S}$ to match the adversary's view in Game_{10}: on input $(1^\lambda, t, n, \mathcal{C})$, it samples $\mathsf{pk} = (\vec{b}, A)$ (c.f. Game_4) and $(\vec{s}_k, \alpha_k)_{k\in\mathcal{C}}$ (c.f. Game_3) uniformly at random, plus $\vec{s}, \vec{e} \leftarrow D_{\sigma_s}^k$ for its own internal use only.

On an $\mathsf{EncO}(m, \mathsf{rnd})$ query, $\mathcal{S}$ records $(m, \vec{r})$; on a ChalO query, it samples a uniformly random ciphertext $\mathsf{ct} = (c_0, \vec{c}_1)$ (c.f. Game_6), together with bookkeeping randomness $\vec{r} = (r_0, \vec{r}_1, \vec{r}_2)$, and records $(\mathsf{ct}, \vec{r})$. On $\mathsf{PartDecO}(\mathsf{ct}, i)$ with $i \notin \mathcal{C}$, $\mathcal{S}$ answers as follows. If $\mathsf{ct} \in L_{\mathsf{Enc}}$, it uses the AHMLWE-hint-style formula of $\mathsf{Game}_5/\mathsf{Game}_7$ applied to the recorded $\vec{r}$ (and to $\vec{s}, \vec{e}$, and to the corrupted shares $\vec{s}_i, \vec{e}_i$, $i \in \mathcal{C}'$). If $\mathsf{ct} \in L_{\mathsf{Chal}}$, it samples $f_{\mathsf{ct},i}$ once and reuses it on any later (ct, i) query, and returns $\vec{c}_1^{\top}\vec{s}_i + f_{\mathsf{ct},i}$ if $i \in \mathcal{C}$; a uniform value in $\mathcal{R}_q$, again sampled once and reused, if i is among the first $t-1-|\mathcal{C}|$ honest parties queried on ct (c.f. $\mathsf{Game}_9/\mathsf{Game}_{10}$); and otherwise, *i.e.,* once $|P[\mathsf{ct}] \cup \mathcal{C}| \geq t$ (which is exactly when $\mathsf{SemiMalSimCPA}^1$ itself hands $\mathcal{S}$ the message $M[\mathsf{ct}]$) the forced interpolated value $d_{\mathsf{ct},i}$ of Game_{10}, computed from $m = M[\mathsf{ct}]$ and the now-fixed values $\{d_{\mathsf{ct},j}\}_{j\in\mathcal{C}'}$.

By construction, the view of $\mathcal{A}$ when run against this $\mathcal{S}$ in $\mathsf{SemiMalSimCPA}^1$ is identically distributed to the view of $\mathcal{A}$ in Game_{10}. By definition, this means that $\mathsf{Adv}^{\mathsf{SemiMalSimCPA}}_{\mathsf{tPKE}}(\mathcal{A},\mathcal{S}) = |p_0 - p_{10}|$, and the triangle inequality over $|p_0 - p_1|, \ldots, |p_9 - p_{10}|$, all of which are either 0 or bounded above in the corresponding game, then recovers the theorem statement. □

5 Achieving CCA Security

In this section, we extend our SemiMalSimCPA-secure tPKE to achieve SimCCA security. We rely on the BKW transform [BKW26], which generically upgrades a semi-malicious secure tPKE to a CCA-secure one by leveraging a simulation-extractable Non-Interactive Zero-Knowledge (NIZK) proof. Roughly, this transform appends a NIZK proof of well-formedness to each ciphertext, which is verified before partial decryption. Simulation extractability ensures that we can (1) Simulate the proofs attached to challenge ciphertexts so that they do not leak any secret information, and (2) Extract the randomness and message underlying all other ciphertexts, which is necessary to simulate their partial decryptions in the SemiMalSimCPA model.

5.1 Non-interactive Proof System

We first recall the notion of *non-interactive zero-knowledge proof* (NIZK) systems in Def. 9.

Definition 9 (NIZK). *A non-interactive zero-knowledge proof system for a language $\mathcal{L}$ in the common reference string (CRS) model is a tuple* $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *with the following syntax:*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{crs}$ *Generates a common reference string.*
- $\mathsf{Prove}(\mathsf{crs}, x, w) \to \pi$ *Given a CRS* crs*, and $(x,w) \in \mathcal{L}$, outputs a proof π.*
- $\mathsf{Verify}(\mathsf{crs}, x, \pi) \to b$ *Given a CRS* crs*, a statement x, and a proof π, outputs a verdict $b \in \{0,1\}$.*

We require NIZK *to be perfectly correct for $\mathcal{L}$, meaning that for all $(x,w) \in \mathcal{L}$, and all* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1$ *where* $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, x, w)$.

We now introduce the property of *simulation extractability* for a NIZK, required by the CPA-to-CCA tPKE transform. Informally, a simulation-extractable NIZK is equipped with a simulator that produces convincing proofs without knowing any witness—so that proofs attached to challenge ciphertexts leak no secret information—while an extractor is still able to recover a valid witness from any proof that the simulator did not itself produce. This single notion thus captures both the (zero-knowledge) simulation-soundness and the straight-line extractability that the transform relies on. Note that there exists a weak variant of simulation extractability, which in particular tolerates proof malleability. However, since it is insufficient for the BKW transform, we do not recall it here.

Definition 10 (Simulation Extractability). *A NIZK scheme* $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *for a language* $\mathcal{L}$ *with associated relation* $\mathcal{R}$ *is* simulation extractable *if there exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim.Setup}, \mathsf{Sim.Sim}, \mathsf{Sim.Ext})$ *such that*

$$\left|\Pr\left[\mathsf{SIMEXT}^0_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}(1^\lambda) = 1\right] - \Pr\left[\mathsf{SIMEXT}^1_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}(1^\lambda) = 1\right]\right|$$

is negligible for any PPT adversary $\mathcal{A}$*, where the experiment* $\mathsf{SIMEXT}^b_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}$ *is defined in Fig. 4.*

$\mathsf{SIMEXT}^b_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}(1^\lambda)$

1 : **if** $b = 0$ **then** $\mathsf{crs} \xleftarrow{\$} \mathsf{NIZK.Setup}(1^\lambda)$
2 : **if** $b = 1$ **then** $(\mathsf{crs}, \mathsf{st}) \xleftarrow{\$} \mathsf{Sim.Setup}(1^\lambda)$
3 : $\mathcal{W} \leftarrow \emptyset$
4 : **return** $\mathcal{A}^{\mathsf{ProveO},\mathsf{VerifyO}}(\mathsf{crs})$

$\mathsf{ProveO}(x, w)$

1 : **assert** $(x, w) \in \mathcal{R}$
2 : **if** $b = 0$ **then** $\pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{crs}, x, w)$
3 : **if** $b = 1$ **then**
4 : $\quad(\pi, \mathsf{st}) \xleftarrow{\$} \mathsf{Sim.Sim}(x, \mathsf{st})$
5 : $\quad\mathcal{W} \leftarrow \mathcal{W} \cup \{(x, \pi)\}$
6 : **return** π

$\mathsf{VerifyO}(x, \pi)$

1 : **if** $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 0$ **then return** 0
2 : **if** $b = 0$ **then return** 1
3 : **if** $(x, \pi) \in \mathcal{W}$ **then return** 1
4 : $(w, \mathsf{st}) \xleftarrow{\$} \mathsf{Sim.Ext}(x, \pi, \mathsf{st})$
5 : **assert** $(x, w) \in \mathcal{R}$ // no extraction failure
6 : **return** 1

Fig. 4. Experiment for the (strong) simulation extractability of a NIZK system.

5.2 Threshold CPA-to-CCA Transform

The Naor-Yung paradigm [NY90] transforms a CPA-secure PKE scheme into a CCA-secure one with the help of a simulation-sound NIZK [DNR04]. In its original form, a ciphertext consists of two encryptions $\mathbf{c}_1$ and $\mathbf{c}_2$ of the same message under two independent key pairs, together with a NIZK proof that they indeed encrypt the same plaintext; decryption verifies the proof and then decrypts either of the two components. This paradigm has been lifted to the threshold setting under additional conditions [FP01, BKW26]. We rely on the variant of Brzuska et al. [BKW26], which upgrades a SemiMalSimCPA-secure tPKE into a SimCCA-secure one by attaching to each ciphertext a *simulation-extractable* NIZK proof of well-formedness, without resorting to double encryption. The transform is as follows.

Let $\mathsf{tPKE}_0 = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{PartDec}, \mathsf{FinDec})$ be a SemiMalSimCPA-secure tPKE with message space $\mathcal{M}$, public key pk_0, and encryption randomness space

$\mathcal{RD}$. The transform attaches to each ciphertext a proof that it is well-formed, i.e., that it lies in the image of $\mathsf{tPKE}_0.\mathsf{Enc}$. Formally, this is captured by the relation

$$\mathcal{R}_{\mathrm{NY}} = \{(\mathbf{c}, \mathsf{rnd}) : \exists\, m \in \mathcal{M},\ \mathbf{c} = \mathsf{tPKE}_0.\mathsf{Enc}(\mathsf{pk}_0, m; \mathsf{rnd})\}\,,$$

where the statement $\mathbf{c}$ is a ciphertext and the witness $\mathsf{rnd} \in \mathcal{RD}$ is the randomness that produced it. Given a simulation-extractable NIZK $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ for $\mathcal{R}_{\mathrm{NY}}$, the transform yields the SimCCA-secure tPKE $\mathsf{tPKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{PartDec}, \mathsf{FinDec})$ described in Fig. 5, where the CRS crs is treated as part of the public key and is available to all algorithms. We refer to [BKW26] for formal proofs of the security of this transform.

$\mathsf{KeyGen}(1^\lambda, t, n)$

1 : $(\mathsf{pk}_0, \mathsf{sk}) \leftarrow \mathsf{tPKE}_0.\mathsf{KeyGen}(1^\lambda, t, n)$
2 : $\mathsf{crs} \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$
3 : **return** $\big(\mathsf{pk} := (\mathsf{pk}_0, \mathsf{crs}),\ \mathsf{sk}\big)$

$\mathsf{Enc}(\mathsf{pk} = (\mathsf{pk}_0, \mathsf{crs}), m)$

1 : $\mathsf{rnd} \xleftarrow{\$} \chi_{\mathsf{rnd}}$
2 : $\mathbf{c} \leftarrow \mathsf{tPKE}_0.\mathsf{Enc}(\mathsf{pk}_0, m; \mathsf{rnd})$
3 : $\pi \leftarrow \mathsf{NIZK.Prove}(\mathsf{crs}, \mathbf{c}, \mathsf{rnd})$
4 : **return** $\mathsf{ct} := (\mathbf{c}, \pi)$

$\mathsf{PartDec}(\mathsf{sk}_i, \mathsf{ct} = (\mathbf{c}, \pi))$

1 : **if** $\mathsf{NIZK.Verify}(\mathsf{crs}, \mathbf{c}, \pi) = 0$ **then**
2 : **return** $\perp$
3 : $d_i \leftarrow \mathsf{tPKE}_0.\mathsf{PartDec}(\mathsf{sk}_i, \mathbf{c})$
4 : **return** d_i

$\mathsf{FinDec}(\mathsf{ct} = (\mathbf{c}, \pi), \{d_i\}_{i \in \mathcal{I}})$

// $\mathcal{I} \subseteq [n],\ |\mathcal{I}| = t$
1 : **return** $\mathsf{tPKE}_0.\mathsf{FinDec}(\mathbf{c}, \{d_i\}_{i \in \mathcal{I}})$

Fig. 5. A SimCCA-secure tPKE tPKE from a SemiMalSimCPA-secure tPKE tPKE_0 via the BKW transform [BKW26].

Remark 2. [BKW26] introduces a second transform for schemes that are only CPA-secure and do not support semi-malicious security (i.e., adversarially chosen randomness). In that case, their transform requires a non-interactive proof of randomness (NIPoR) to certify well-formedness and honest sampling of the encryption randomness. This, however, appears much more costly to instantiate.

5.3 CCA-secure threshold PKE

We now discuss how to concretely instantiate a CCA-secure scheme from our tPKE scheme. We first outline our construction of a simulation extractable NIZK, which is the only missing ingredient in the BKW transform.

Simulation Extractable NIZK from Lattices. The Lantern proof system [LNP22] is a strong contender for instantiating our NIZK, as it excels at proving native lattice relationships and often provides the most compact sizes for

small tasks. Moreover, Lantern has recently been shown to be *strongly* simulation sound [BIMP26, Theorem 3], as required here.

However, Lantern does not support *straight-line*[11] witness extraction. Fortunately, several transformations add straight-line extractability to a NIZK, including Fischlin's transform [Fis05], the (folklore) encryption-to-the-sky technique (see, e.g., [AKSY22] for a lattice-based instantiation), and Katsumata's transform [Kat21]. In this work, we rely on the encryption-to-the-sky paradigm since it is a black-box transform that does not require modifying the underlying NIZK system.

In this paradigm, the prover provides a public key encryption of the witness alongside the NIZK proof, while extending the proven relationship to ensure that the ciphertext correctly encrypts the witness. Assuming access to the PKE's secret key, one can easily extract the witness.

The simulation soundness of the NIZK is preserved by using a simulation-secure PKE (i.e., its ciphertexts can be simulated) whose public keys can be indistinguishably generated by a random oracle, ensuring that the adversary does not get access to the PKE's private key.

We formally recall and prove this upgrade in Sec. B. We then instantiate it for Lantern, using a Regev public-key encryption for witness extraction.

Our Scheme. By combining our tPKE scheme with the simulation-extractable NIZK above, we finally obtain a CCA-secure threshold PKE. Below, we briefly summarize the specifications and optimizations for our scheme.

In our CCA scheme, the encryptor needs to provide a NIZK proof that proves the ciphertext is generated from the encryption algorithm using a message from the plaintext space and encryption randomness drawn from the valid randomness space. As stated in the proof of Thm. 3, the randomness space for our semi-malicious CPA-secure tPKE is given by a closed two-norm ball with radius $e^{1/4}\sigma_r\sqrt{N(2k+1)}$. While this is a sufficient condition to prove the ciphertext well-formedness, we additionally assume that the randomness is confined within an infinity-norm ball with radius $B_r > 0$ to minimize the plaintext space for the witness encryption, hence a compact proof size. This bound B_r is chosen using Lem. 2, following from the fact that $\vec{r}$ is drawn from a Gaussian distribution.

For straight-line extractability, the encryption-to-the-sky transform appends Regev encryptions of the witness to the proof. One could instantiate these with a compact Regev variant such as CRYSTALS-Kyber [BDK+18], but its ciphertext modulus would then differ from the one of the tPKE, forcing the proved relation to deal with two moduli and thereby increasing the proof size. We therefore reuse the tPKE's ciphertext modulus for the witness encryptions. This choice is also beneficial in that it allows to batch encryptions. Since our modulus is already larger than that of a conventional single-key Regev encryption (it grows as $\sqrt{Q}$), we can take a large plaintext modulus and batch several witnesses digit-by-digit

[11] [BIMP26] shows that Lantern satisfies a form of simulation extractability. This notion is however weaker than the one our construction requires, as extraction rewinds the adversary for each extracted proof instead of being straight-line.

into a single ciphertext, rather than encrypting each of them separately. We write L for the resulting number of witness encryptions (see Sec. C), and suggest encrypting them under distinct key pairs sharing a common public matrix, *i.e.,* shared-A MLWE encryption.

Putting these discussions altogether, we can construct a concrete CCA-secure PKE scheme.

6 Evaluation

We now move on to the concrete instantiation of our construction. The parameter selection strategy for our scheme is given in Sec. C.

Recommended Parameters. Now, we choose and provide candidate parameters for our tPKE scheme and corresponding ciphertext sizes. We elaborate how to choose the parameters in Sec. C. We fix the ring degree $N = 256$ and use the lattice estimator [APS15] to measure the security of the underlying hardness problems. We describe recommended parameter sets in Table 2 for the following parameter regimes: maximum number of users $n \in \{16, 32\}$, threshold $t \in \{2, n/2, n-1\}$. Our proposed parameters achieve a 128-bit security level.

t	Q	q	k	σ_r	σ_s	σ_f	L
2	1	$2^{28.7}$	4	$2^{2.3}$	$2^{2.2}$	$2^{18.9}$	9
8	1	$2^{41.9}$	6	$2^{2.2}$	$2^{2.2}$	$2^{27.1}$	5
15	1	$2^{30.9}$	5	$2^{3.5}$	$2^{2.4}$	$2^{20.2}$	11
2	2^{32}	$2^{45.0}$	7	$2^{2.2}$	$2^{2.2}$	$2^{35.2}$	4
8	2^{32}	$2^{58.1}$	9	$2^{2.2}$	$2^{2.2}$	$2^{43.3}$	4
15	2^{32}	$2^{45.9}$	7	$2^{2.2}$	$2^{2.4}$	$2^{35.2}$	4
2	2^{60}	$2^{59.1}$	9	$2^{2.2}$	$2^{2.2}$	$2^{49.3}$	4
8	2^{60}	$2^{72.3}$	11	$2^{2.2}$	$2^{2.2}$	$2^{57.5}$	4
15	2^{60}	$2^{60.1}$	9	$2^{2.2}$	$2^{2.4}$	$2^{49.3}$	4

t	Q	q	k	σ_r	σ_s	σ_f	L
2	1	$2^{30.8}$	5	$2^{2.3}$	$2^{2.2}$	$2^{20.0}$	6
16	1	$2^{63.5}$	9	$2^{2.2}$	$2^{2.2}$	$2^{40.9}$	4
31	1	$2^{33.4}$	5	$2^{3.9}$	$2^{2.4}$	$2^{21.7}$	11
2	2^{32}	$2^{47.0}$	7	$2^{2.2}$	$2^{2.2}$	$2^{36.2}$	4
16	2^{32}	$2^{79.7}$	12	$2^{2.2}$	$2^{2.2}$	$2^{57.1}$	4
31	2^{32}	$2^{47.9}$	7	$2^{2.2}$	$2^{2.4}$	$2^{36.2}$	4
2	2^{60}	$2^{61.1}$	9	$2^{2.2}$	$2^{2.2}$	$2^{50.3}$	4
16	2^{60}	$2^{93.8}$	14	$2^{2.2}$	$2^{2.2}$	$2^{71.2}$	3
31	2^{60}	$2^{62.1}$	9	$2^{2.2}$	$2^{2.4}$	$2^{50.3}$	4

Table 2. Recommended parameter sets for (a) $n = 16$, (b) $n = 32$ parties. t, Q, q, k, σ_r, σ_s, and σ_f denote the threshold, the number of queries, ciphertext modulus, MLWE dimension, discrete Gaussian parameters for the encryption, key generation and partial decryption, respectively. L denotes the number of witness encryptions.

Ciphertext Sizes. We compute the expected ciphertext sizes for the parameter sets presented in Table 2 and summarize them in Table 3. We use the `LaZer` library to compute the expected proof size in the CCA setting.

Currently, the size of CCA-secure PKE ciphertexts is dominated by the NIZK proof. We emphasize that the ciphertext size can be further reduced with opti-

		CPA	CCA		
t	Q	|ctxt|	NIZK	WENC	|ctxt|
2	1	4.49	39.77	11.68	55.94
8	1	9.16	44.05	14.39	67.60
15	1	5.80	40.86	15.47	62.13
2	2^{32}	11.24	46.94	15.46	73.64
8	2^{32}	18.17	55.15	23.62	96.94
15	2^{32}	11.47	46.77	15.77	74.01
2	2^{60}	18.48	55.37	24.02	97.87
8	2^{60}	27.10	64.23	33.88	125.21
15	2^{60}	18.77	55.67	24.40	98.84

		CPA	CCA		
t	Q	|ctxt|	NIZK	WENC	|ctxt|
2	1	5.78	40.86	10.59	57.23
16	1	19.84	55.92	25.79	101.55
31	1	6.27	45.57	16.71	68.55
2	2^{32}	11.74	47.08	16.14	74.96
16	2^{32}	32.37	68.33	39.84	140.54
31	2^{32}	11.98	47.29	16.48	75.75
2	2^{60}	19.10	55.84	24.83	99.77
16	2^{60}	43.97	78.54	49.83	172.34
31	2^{60}	19.41	55.71	25.23	100.35

Table 3. The ciphertext sizes in KB when (a) $n = 16$, (b) $n = 32$ parties. For the CCA-secure variant, NIZK denotes the size of the zero-knowledge proof, and WENC denotes the size of the witness encryption for the straight-line extractability.

mizations in NIZK construction, as our security proof only requires the underlying NIZK scheme to be simulation extractable.

Acknowledgement. We thank Katharina Boudgoust for pointing out a result that was proven in [BIMP26] after our initial version, that Lantern is strongly simulation sound, which directly simplifies our simulation extractable NIZK construction and removes the need for a one-time signature. We also thank Damien Stehlé for valuable comments on prior art and bibliographic suggestions.

This work was supported in part by JST CREST Grant Number JPMJCR22M1 and JST K Program Grant Number JPMJKP24U3.

Guilhem Niot is supported by the French National Research Agency (ANR), through the project RELATE (reference: ANR-25-CE39-4214-01).

The authors used Claude, Gemini 3 Flash, and ChatGPT for language editing and presentation. These tools did not generate technical results, proofs, algorithms, experimental data, or conclusions. All scientific contributions are solely by the authors, and the authors have reviewed and validated all AI-assisted edits and accept responsibility for the final manuscript.

References

ABGS23. Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, and Tjerand Silde. Verifiable mix-nets and distributed decryption for voting from lattice-based assumptions. In Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda, editors, *ACM CCS 2023*, pages 1467–1481. ACM Press, November 2023.

AGHS13. Shweta Agrawal, Craig Gentry, Shai Halevi, and Amit Sahai. Discrete Gaussian leftover hash lemma over infinite domains. In Kazue Sako and

Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 97–116. Springer, Berlin, Heidelberg, December 2013.

AKSY22. Shweta Agrawal, Elena Kirshanova, Damien Stehlé, and Anshu Yadav. Practical, round-optimal lattice-based blind signatures. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *ACM CCS 2022*, pages 39–53. ACM Press, November 2022.

APS15. Martin Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9, 10 2015.

ASY21. Shweta Agrawal, Damien Stehlé, and Anshu Yadav. Towards practical and round-optimal lattice-based threshold and blind signatures. Cryptology ePrint Archive, Report 2021/381, 2021.

BCHK07. Dan Boneh, Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. *SIAM Journal on Computing*, 36(5):1301–1328, 2007.

BD10. Rikke Bendlin and Ivan Damgård. Threshold decryption and zero-knowledge proofs for lattice-based cryptosystems. In Daniele Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 201–218. Springer, Berlin, Heidelberg, February 2010.

BDK+18. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367. IEEE Computer Society Press, April 2018.

BdLP26. Katharina Boudgoust, Rafael del Pino, Oleksandra Lapiha, and Thomas Prest. IND-CCA lattice threshold KEM under 30 KiB. In Shi Bai and Edoardo Persichetti, editors, *PKC 2026, Part I*, volume 16551 of *LNCS*, pages 38–69. Springer, Cham, May 2026.

BFM+25. Zvika Brakerski, Offir Friedman, Avichai Marmor, Dolev Mutzari, Yuval Spiizer, and Ni Trieu. Threshold FHE with efficient asynchronous decryption. Cryptology ePrint Archive, Report 2025/712, 2025.

BGG+18. Dan Boneh, Rosario Gennaro, Steven Goldfeder, Aayush Jain, Sam Kim, Peter M. R. Rasmussen, and Amit Sahai. Threshold cryptosystems from threshold fully homomorphic encryption. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part I*, volume 10991 of *LNCS*, pages 565–596. Springer, Cham, August 2018.

BGV14. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

BIMP26. Katharina Boudgoust, Laurent Imbert, Loïc Masure, and Laz Panard. Naor-yung transform for IND-CCA probing security with lattice instantiations. Cryptology ePrint Archive, Paper 2026/459, 2026.

BKW26. Chris Brzuska, Michael Klooß, and Ivy K. Y. Woo. Threshold public-key encryption: Definitions, relations, and CPA-to-CCA transforms. In Shi Bai and Edoardo Persichetti, editors, *PKC 2026, Part IV*, volume 16554 of *LNCS*, pages 162–197. Springer, Cham, May 2026.

BP26. Luís T. A. N. Brandão and René Peralta. Nist first call for multi-party threshold schemes. (National Institute of Standards and Technology) NIST Internal Report (NISTIR) 8214C, 2026.

BS23. Katharina Boudgoust and Peter Scholl. Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus. In Jian Guo and

Ron Steinfeld, editors, *ASIACRYPT 2023, Part I*, volume 14438 of *LNCS*, pages 371–404. Springer, Singapore, December 2023.

CCMS21. Kelong Cong, Daniele Cozzo, Varun Maram, and Nigel P. Smart. Gladius: LWR based efficient hybrid public key encryption with distributed decryption. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 125–155. Springer, Cham, December 2021.

CGPP24. Arka Rai Choudhuri, Sanjam Garg, Julien Piet, and Guru-Vamsi Policharla. Mempool privacy via batched threshold encryption: Attacks and defenses. In Davide Balzarotti and Wenyuan Xu, editors, *USENIX Security 2024*. USENIX Association, August 2024.

CGPW25. Arka Rai Choudhuri, Sanjam Garg, Guru-Vamsi Policharla, and Mingyuan Wang. Practical mempool privacy via one-time setup batched threshold encryption. In Lujo Bauer and Giancarlo Pellegrino, editors, *USENIX Security 2025*, pages 3477–3495. USENIX Association, August 2025.

CGS97. Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A secure and optimally efficient multi-authority election scheme. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 103–118. Springer, Berlin, Heidelberg, May 1997.

CHK04. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-ciphertext security from identity-based encryption. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 207–222. Springer, Berlin, Heidelberg, May 2004.

CLW25. Valerio Cini, Russell W. F. Lai, and Ivy K. Y. Woo. Pilvi: Lattice threshold PKE with small decryption shares and improved security. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part VI*, volume 16250 of *LNCS*, pages 539–569. Springer, Singapore, December 2025.

dCKK+26. Leo de Castro, Duhyeong Kim, Miran Kim, Keewoo Lee, Seonhong Min, and Yongsoo Song. More efficient lattice-based OLE: from circuit-private linear HE with polynomial overhead. *IACR Communications in Cryptology*, 3(1), 2026.

DKM+24. Rafaël Del Pino, Shuichi Katsumata, Mary Maller, Fabrice Mouhartem, Thomas Prest, and Markku-Juhani O. Saarinen. Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In Marc Joye and Gregor Leander, editors, *EUROCRYPT 2024, Part II*, volume 14652 of *LNCS*, pages 219–248. Springer, Cham, May 2024.

DLN+21. Julien Devevey, Benoît Libert, Khoa Nguyen, Thomas Peters, and Moti Yung. Non-interactive CCA2-secure threshold cryptosystems: Achieving adaptive security in the standard model without pairings. In Juan Garay, editor, *PKC 2021, Part I*, volume 12710 of *LNCS*, pages 659–690. Springer, Cham, May 2021.

DNR04. Cynthia Dwork, Moni Naor, and Omer Reingold. Immunizing encryption schemes from decryption errors. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 342–360. Springer, Berlin, Heidelberg, May 2004.

Dun44. W.J. Duncan. Lxxviii. some devices for the solution of large sets of simultaneous linear equations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 35(249):660–670, 1944.

dVPS26. François Colin de Verdière, Alain Passelègue, and Damien Stehlé. On threshold fully homomorphic encryption with synchronized decryptors. Cryptology ePrint Archive, Paper 2026/031, 2026.

ENP24. Thomas Espitau, Guilhem Niot, and Thomas Prest. Flood and submerse: Distributed key generation and robust threshold signature from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 425–458. Springer, Cham, August 2024.

Fis05. Marc Fischlin. Communication-efficient non-interactive proofs of knowledge with online extractors. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 152–168. Springer, Berlin, Heidelberg, August 2005.

FO99. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, Berlin, Heidelberg, August 1999.

FP01. Pierre-Alain Fouque and David Pointcheval. Threshold cryptosystems secure against chosen-ciphertext attacks. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 351–368. Springer, Berlin, Heidelberg, December 2001.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

Kat21. Shuichi Katsumata. A new simple technique to bootstrap various lattice zero-knowledge proofs to QROM secure NIZKs. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 580–610, Virtual Event, August 2021. Springer, Cham.

KLL$^+$26. Won Kim, Changmin Lee, JeongHwan Lee, Alain Passelègue, and Damien Stehlé. Asynchronous lagrange-based threshold FHE with smaller modulus overhead. To appear at CRYPTO '26, 2026.

KLO$^+$19. Michael Kraitsberg, Yehuda Lindell, Valery Osheter, Nigel P. Smart, and Younes Talibi Alaoui. Adding distributed decryption and key generation to a ring-LWE based CCA encryption scheme. In Julian Jang-Jaccard and Fuchun Guo, editors, *ACISP 19*, volume 11547 of *LNCS*, pages 192–210. Springer, Cham, July 2019.

KLSS23. Duhyeong Kim, Dongwon Lee, Jinyeong Seo, and Yongsoo Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In Helena Handschuh and Anna Lysyanskaya, editors, *CRYPTO 2023, Part V*, volume 14085 of *LNCS*, pages 549–580. Springer, Cham, August 2023.

KRT24. Shuichi Katsumata, Michael Reichle, and Kaoru Takemure. Adaptively secure 5 round threshold signatures from MLWE/MSIS and DL with rewinding. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VII*, volume 14926 of *LNCS*, pages 459–491. Springer, Cham, August 2024.

KY16. Shuichi Katsumata and Shota Yamada. Partitioning via non-linear polynomial functions: More compact IBEs from ideal lattices and bilinear maps. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 682–712. Springer, Berlin, Heidelberg, December 2016.

LNP22. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part II*, volume 13508 of *LNCS*, pages 71–101. Springer, Cham, August 2022.

LP11. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for LWE-based encryption. In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Berlin, Heidelberg, February 2011.

LP25. Oleksandra Lapiha and Thomas Prest. A lattice-based IND-CCA threshold KEM from the BCHK+ transform. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part III*, volume 16247 of *LNCS*, pages 461–494. Springer, Singapore, December 2025.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Berlin, Heidelberg, May / June 2010.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 2015.

LSS24. Vadim Lyubashevsky, Gregor Seiler, and Patrick Steuer. The LaZer library: Lattice-based zero knowledge and succinct proofs for quantum-safe privacy. In Bo Luo, Xiaojing Liao, Jun Xu, Engin Kirda, and David Lie, editors, *ACM CCS 2024*, pages 3125–3137. ACM Press, October 2024.

MR04. Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381. IEEE Computer Society Press, October 2004.

MS24. Daniele Micciancio and Adam Suhl. Simulation-secure threshold PKE from LWE with polynomial modulus. *CiC*, 1(4):2, 2024.

Nat24. National Institute of Standards and Technology. Module-Lattice-Based Key-Encapsulation Mechanism Standard. Federal Information Processing Standards Publication (FIPS) FIPS 203, Department of Commerce, Washington, D.C., 2024. NIST has assigned NIST FIPS 203 as the publication identifier for this FIPS.

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *22nd ACM STOC*, pages 427–437. ACM Press, May 1990.

OT25. Hiroki Okada and Tsuyoshi Takagi. Low communication threshold FHE from standard (module-)LWE. In Goichiro Hanaoka and Bo-Yin Yang, editors, *ASIACRYPT 2025, Part VII*, volume 16251 of *LNCS*, pages 165–198. Springer, Singapore, December 2025.

OT26. Hiroki Okada and Tsuyoshi Takagi. Threshold fhe with short decryption shares without a semi-trusted server. In *Australasian Conference on Information Security and Privacy (ACISP)*, 2026. To appear. ePrint Archive: `https://eprint.iacr.org/2026/957`.

PS24. Alain Passelègue and Damien Stehlé. Low communication threshold fully homomorphic encryption. In Kai-Min Chung and Yu Sasaki, editors, *ASIACRYPT 2024, Part I*, volume 15484 of *LNCS*, pages 297–329. Springer, Singapore, December 2024.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

ZZH+26. Yuxin Zhang, Yunxiao Zhou, Shuai Han, Shengli Liu, and Xinyi Huang. Adaptively secure threshold decryption from LWE with polynomial modulus. Cryptology ePrint Archive, Paper 2026/1627, 2026.

ZZLP25. Guy Zyskind, Doron Zarchy, Max Leibovich, and Chris Peikert. High-throughput universally composable threshold FHE decryption. In

Chun-Ying Huang, Jyh-Cheng Chen, Shiuh-Pyng Shieh, David Lie, and Véronique Cortier, editors, *ACM CCS 2025*, pages 2339–2353. ACM Press, October 2025.

A Deferred Proofs

A.1 Proof of Thm. 2

Proof. By the correctness of the baseline Shamir Secret Sharing scheme, the (final) phase of public key encryption is

$$\begin{aligned}\gamma b + \sum_{i\in\mathcal{I}} \gamma\lambda_i^{\mathcal{I}} \cdot d_i &= \gamma\left(\gamma^{-1}\cdot\lfloor\frac{q}{p}\rceil m + r_0 + \vec{r}_1^{\top}\vec{s} + \vec{r}_2^{\top}\vec{e}\right) + \sum_{i\in\mathcal{I}}\gamma\lambda_i^{\mathcal{I}} f_i \pmod q \\ &= \lfloor\frac{q}{p}\rceil m + \gamma(r_0 + \vec{r}_1^{\top}\vec{s} + \vec{r}_2^{\top}\vec{e}) + \sum_{i\in\mathcal{I}}\gamma\lambda_i^{\mathcal{I}} f_i \pmod q.\end{aligned}$$

For this ciphertext to be decrypted correctly, the infinite norm of the total error $e_{\mathsf{ct}} := \gamma(r_0 + \vec{r}_1^{\top}\vec{s} + \vec{r}_2^{\top}\vec{e}) + \sum_{i\in\mathcal{I}}\gamma\lambda_i^{\mathcal{I}} f_i$ needs to be smaller than $\frac{q}{2p} - \frac{p}{2}$.

Since each $f_i = F(K_i, \mathsf{ct})$ is derived from a PRF key never revealed to anyone, the hybrid argument of Game_1 in the proof of Thm. 3 shows that $(f_i)_{i\in\mathcal{I}}$ is, up to advantage $t \cdot \mathsf{Adv}_F^{\mathsf{PRF}}(\mathcal{D})$, distributed as t independent samples from D_{σ_f}, which we assume henceforth.

By Lem. 1 applied to $\vec{u} = (\vec{r}_1, \vec{r}_2)$ and $\vec{v} = (\vec{s}, \vec{e})$, $\|\vec{r}_1^{\top}\vec{s} + \vec{r}_2^{\top}\vec{e}\|_\infty \le \|(\vec{r}_1, \vec{r}_2)\|_2 \cdot \|(\vec{s}, \vec{e})\|_2$. By Lem. 3 (applied with $\ell = 2k$ to $(\vec{r}_1, \vec{r}_2) \leftarrow D_{\sigma_r}^{2k}$ and to $(\vec{s}, \vec{e}) \leftarrow D_{\sigma_s}^{2k}$, each except with probability at most $2^{-2kN/10}$), this is in turn bounded by $e^{1/4}\sigma_r\sqrt{2kN} \cdot e^{1/4}\sigma_s\sqrt{2kN} = e^{1/2} \cdot 2kN\sigma_r\sigma_s$.

By Lem. 2 (applied with $\ell = 1$ to $r_0 \leftarrow D_{\sigma_r}$, and with $\ell = t$ to $(f_i)_{i\in\mathcal{I}} \leftarrow D_{\sigma_f}^t$, each except with negligible probability) and Lem. 5, $\|r_0\|_\infty \le \rho_r\sigma_r$ and $\|\sum_{i\in\mathcal{I}}\gamma\lambda_i^{\mathcal{I}} f_{\mathsf{ct},i}\|_\infty \le tB_\lambda\rho_f\sigma_f$.

Combining these bounds via the triangle inequality, except with negligible probability over the union of the above events (and the PRF advantage), $\|e_{ct}\|_\infty \le \gamma\left(\rho_r\sigma_r + e^{1/2} \cdot 2kN\sigma_r\sigma_s\right) + tB_\lambda\rho_f\sigma_f \le \frac{q}{2p} - \frac{p}{2}$ from the conditions. □

A.2 Proof of Lem. 6

Proof. For any $\mathbf{x} \in \mathbb{Z}^m$, let $\mathcal{D}(\mathbf{x})$ be the probability mass function of the distribution $\{\mathbf{Me} + \mathbf{f} \mid \mathbf{e} \leftarrow D_{\mathbb{Z}^n,\sigma}, \mathbf{f} \leftarrow D_{\mathbb{Z}^m,\sqrt{\mathbf{\Sigma}}}\}$. We have:

$$\begin{aligned}\mathcal{D}(\mathbf{x}) &= \Pr[\mathbf{Me} + \mathbf{f} = \mathbf{x} \mid \mathbf{e} \leftarrow D_{\mathbb{Z}^n,\sigma}, \mathbf{f} \leftarrow D_{\mathbb{Z}^m,\sqrt{\mathbf{\Sigma}}}] \\ &= \sum_{\mathbf{y}\in\mathbb{Z}^n} D_{\mathbb{Z}^n,\sigma}(\mathbf{y}) \cdot D_{\mathbb{Z}^m,\sqrt{\mathbf{\Sigma}}}(\mathbf{x} - \mathbf{My}) \\ &\propto \sum_{\mathbf{y}\in\mathbb{Z}^n} \exp\left[-\frac{1}{2}\left(\frac{1}{\sigma^2}\|\mathbf{y}\|^2 + (\mathbf{x} - \mathbf{My})^{\top}\mathbf{\Sigma}^{-1}(\mathbf{x} - \mathbf{My})\right)\right].\end{aligned}$$

Then, we have

$$\begin{aligned}
&\frac{1}{\sigma^2}\|\mathbf{y}\|^2 + (\mathbf{x}-\mathbf{M}\mathbf{y})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{M}\mathbf{y}) \\
&= \mathbf{y}^\top \left(\frac{1}{\sigma^2}\mathbf{I}_n + \mathbf{M}^\top \mathbf{\Sigma}^{-1}\mathbf{M}\right)\mathbf{y} - 2\mathbf{y}^\top(\mathbf{M}^\top \mathbf{\Sigma}^{-1}\mathbf{x}) + \mathbf{x}^\top \mathbf{\Sigma}^{-1}\mathbf{x} \\
&= (\mathbf{y}-\mathbf{c})^\top \mathbf{\Sigma}_1^{-1}(\mathbf{y}-\mathbf{c}) + \mathbf{x}^\top(\mathbf{\Sigma}^{-1} - \mathbf{\Sigma}^{-1}\mathbf{M}\mathbf{\Sigma}\mathbf{M}^\top\mathbf{\Sigma}^{-1})\mathbf{x} \\
&= (\mathbf{y}-\mathbf{c})^\top \mathbf{\Sigma}_1^{-1}(\mathbf{y}-\mathbf{c}) + \mathbf{x}^\top(\sigma^2\mathbf{M}\mathbf{M}^\top + \mathbf{\Sigma})^{-1}\mathbf{x},
\end{aligned}$$

where $\mathbf{\Sigma}_1^{-1} := \frac{1}{\sigma^2}\mathbf{I}_n + \mathbf{M}^\top\mathbf{\Sigma}^{-1}\mathbf{M}$, $\mathbf{c} := \mathbf{\Sigma}_1\mathbf{M}^\top\mathbf{\Sigma}^{-1}\mathbf{x}$ and the last simplification is derived from the Woodbury Matrix Identity [Dun44]. This implies that

$$\mathcal{D}(\mathbf{x}) \propto \rho_{\sqrt{\sigma^2\mathbf{M}\mathbf{M}^\top+\mathbf{\Sigma}}}(\mathbf{x}) \cdot \sum_{\mathbf{y}\in\mathbb{Z}^n} \rho_{\sqrt{\mathbf{\Sigma}_1}}(\mathbf{y}-\mathbf{c}).$$

From the condition in the lemma, we check the smoothing parameter requirement. The largest singular value of $\sqrt{\mathbf{\Sigma}_1^{-1}}$ satisfies:

$$s_{\max}(\sqrt{\mathbf{\Sigma}_1^{-1}}) = \sqrt{\lambda_{\max}(\mathbf{\Sigma}_1^{-1})} = \sqrt{\lambda_{\max}\left(\frac{1}{\sigma^2}\mathbf{I}_n + \mathbf{M}^\top\mathbf{\Sigma}^{-1}\mathbf{M}\right)} \le \sqrt{\frac{1}{\sigma^2} + \|\mathbf{M}\sqrt{\mathbf{\Sigma}^{-1}}\|_2^2},$$

where $s_{\max}(\cdot)$ denotes the largest singular value and $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue. Given $\frac{1}{\sigma^2} + \|\mathbf{M}\sqrt{\mathbf{\Sigma}^{-1}}\|_2^2 \le \frac{2\pi}{\eta_\varepsilon(\mathbb{Z}^n)^2}$ and noting that $\|\mathbf{M}\sqrt{\mathbf{\Sigma}^{-1}}\|_2^2$ is the largest eigenvalue of $\mathbf{M}^\top\mathbf{\Sigma}^{-1}\mathbf{M}$, this gives $\|\mathbf{\Sigma}_1^{-1}\|_2 \le 2\pi/\eta_\varepsilon(\mathbb{Z}^n)^2$, i.e. $\sqrt{2\pi\mathbf{\Sigma}_1} \ge \eta_\varepsilon(\mathbb{Z}^n)$, which ensures the covariance $\mathbf{\Sigma}_1$ is "wide" enough such that for any offset $\mathbf{c}$, the sum over the lattice $\mathbb{Z}^n$ is nearly constant:

$$\sum_{\mathbf{y}\in\mathbb{Z}^n} \rho_{\sqrt{\mathbf{\Sigma}_1}}(\mathbf{y}-\mathbf{c}) \in \left[\frac{1-\varepsilon}{1+\varepsilon}, 1\right] \rho_{\sqrt{\mathbf{\Sigma}_1}}(\mathbb{Z}^n).$$

due to [AGHS13, Lemma 4]. This leads to the bound:

$$\mathcal{D}(\mathbf{x}) \in [1-4\varepsilon, 1+4\varepsilon] \cdot D_{\mathbb{Z}^m, \sqrt{\sigma^2\mathbf{M}\mathbf{M}^\top+\mathbf{\Sigma}}}(\mathbf{x}),$$

which concludes that the statistical distance is bounded by 4ε. □

A.3 Proof of Lem. 7

Proof. Consider an input $\mathbf{x}+\mathbf{y}$. Then the algorithm $\mathsf{Rerand}(\mathbf{M}, \mathbf{x}+\mathbf{y}, \sigma, s)$ computes the value $\mathbf{M}(\mathbf{x}+\mathbf{y}) + \mathbf{f}$ where $\mathbf{f} \leftarrow D_{\mathbb{Z}^m, \sqrt{\mathbf{\Sigma}}}$ for $\mathbf{\Sigma} := \sigma^2 s^2 \mathbf{I}_m - \sigma^2 \mathbf{M}\mathbf{M}^\top$. Note that $\mathbf{\Sigma}$ is a positive semidefinite matrix since

$$\|\mathbf{M}\|_2 \le \sqrt{1 - \eta_\varepsilon^2(\mathbb{Z}^n)/(2\pi\sigma^2)} \cdot s \le s.$$

Then, since $\sigma^2\mathbf{M}\mathbf{M}^\top + \mathbf{\Sigma} = \sigma^2 s^2 \mathbf{I}_m$, Lem. 6 states that the output of $\mathsf{Rerand}(\mathbf{M}, \mathbf{x}+\mathbf{y}, \sigma, s)$ is within statistical distance 4ε of $D_{\mathbb{Z}^m, \sigma s}$ as long as

$\frac{1}{\sigma^2} + \|\mathbf{M}\sqrt{\mathbf{\Sigma}^{-1}}\|_2^2 \leq \frac{2\pi}{\eta_\varepsilon^2(\mathbb{Z}^n)}$. We claim that this inequality holds given the conditions. Note that the smallest eigenvalue of $\mathbf{\Sigma}$ is at least

$$s^2\sigma^2 - \sigma^2\|\mathbf{M}\|_2^2 \geq \frac{\eta_\varepsilon^2(\mathbb{Z}^n)\sigma^2}{2\pi\sigma^2 - \eta_\varepsilon^2(\mathbb{Z}^n)}\|\mathbf{M}\|_2^2.$$

This directly follows from the condition $\|\mathbf{M}\|_2 \leq \sqrt{\frac{2\pi\sigma^2 - \eta_\varepsilon^2(\mathbb{Z}^n)}{2\pi\sigma^2}} \cdot s$. According to [KLSS23, Lemma 5], this implies $\|\mathbf{\Sigma}^{-1}\|_2 \leq \frac{2\pi\sigma^2 - \eta_\varepsilon^2(\mathbb{Z}^n)}{\eta_\varepsilon^2(\mathbb{Z}^n)\sigma^2}\|\mathbf{M}\|_2^{-2}$. Then, from the property of the matrix two-norm,

$$\|\mathbf{M}\sqrt{\mathbf{\Sigma}^{-1}}\|_2^2 \leq \|\mathbf{M}\|_2^2 \cdot \|\mathbf{\Sigma}^{-1}\|_2 \leq \frac{2\pi\sigma^2 - \eta_\varepsilon^2(\mathbb{Z}^n)}{\eta_\varepsilon^2(\mathbb{Z}^n)\sigma^2} = \frac{2\pi}{\eta_\varepsilon^2(\mathbb{Z}^n)} - \frac{1}{\sigma^2},$$

thereby showing that the claim is true and hence the algorithm outputs the correct value. □

B Constructing a Simulation Extractable NIZK from a Simulation Secure NIZK

This section formalizes the transform from a (strongly) simulation-sound NIZK to a simulation-extractable NIZK, using the encryption-to-the-sky technique for straight-line extractability.

B.1 Building Blocks

The transform combines a simulation-sound NIZK (Def. 11) and a simulation-secure PKE whose public keys can themselves be simulated (Def. 13). Throughout, we write $x \in \mathcal{L}$ (resp. $x \notin \mathcal{L}$) to mean that a witness w with $(x, w) \in \mathcal{R}$ exists (resp. does not).

Simulation-Sound NIZK.

Definition 11 (Simulation Soundness). *A NIZK* $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ *for a relation* $\mathcal{R}$ *is* simulation-sound *if there exists a PPT simulator* $\mathsf{Sim} = (\mathsf{Sim.Setup}, \mathsf{Sim.Sim})$ *such that the following two properties hold, for the experiments defined in Fig. 6.*

Zero-knowledge: *For every PPT adversary* $\mathcal{A}$*,*

$$\left|\Pr\left[\mathsf{ZK}^0_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}(1^\lambda) = 1\right] - \Pr\left[\mathsf{ZK}^1_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}(1^\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda),$$

i.e., a simulated CRS together with simulated proofs (which are produced without a witness) is indistinguishable from an honest CRS with honest proofs.

Simulation-soundness: *For every PPT adversary* $\mathcal{A}$*,*

$$\Pr\left[\mathsf{SS}_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}(1^\lambda) = 1\right] \leq \mathsf{negl}(\lambda),$$

i.e., even after observing simulated proofs, $\mathcal{A}$ *cannot produce a valid proof for a false statement, unless it is a proof returned by the simulator. As the freshness condition is on the pair* (x^*, π^*) *rather than on* x^* *alone, this is the* strong *variant of simulation soundness: in particular, simulated proofs cannot be mauled into new valid proofs for the same (possibly false) statement.*

$\mathsf{ZK}^b_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}(1^\lambda)$

1 : **if** $b = 0$ **then** $\mathsf{crs} \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$
2 : **if** $b = 1$ **then** $(\mathsf{crs}, \mathsf{st}) \xleftarrow{\$} \mathsf{Sim.Setup}(1^\lambda)$
3 : **return** $\mathcal{A}^{\mathsf{ProveO}}(\mathsf{crs})$

$\mathsf{ProveO}(x, w)$

1 : **assert** $(x, w) \in \mathcal{R}$
2 : **if** $b = 0$ **then** $\pi \xleftarrow{\$} \mathsf{Prove}(\mathsf{crs}, x, w)$
3 : **if** $b = 1$ **then** $(\pi, \mathsf{st}) \xleftarrow{\$} \mathsf{Sim.Sim}(x, \mathsf{st})$
4 : **return** π

$\mathsf{SS}_{\mathsf{NIZK},\mathsf{Sim},\mathcal{A}}(1^\lambda)$

1 : $(\mathsf{crs}, \mathsf{st}) \xleftarrow{\$} \mathsf{Sim.Setup}(1^\lambda)$
2 : $\mathcal{Q} \leftarrow \emptyset$
3 : $(x^*, \pi^*) \leftarrow \mathcal{A}^{\mathsf{SimO}}(\mathsf{crs})$
4 : **return** $\mathsf{Verify}(\mathsf{crs}, x^*, \pi^*) = 1$
5 : $\quad \wedge\ (x^*, \pi^*) \notin \mathcal{Q}\ \wedge\ x^* \notin \mathcal{L}$

$\mathsf{SimO}(x)$

1 : $(\pi, \mathsf{st}) \xleftarrow{\$} \mathsf{Sim.Sim}(x, \mathsf{st})$
2 : $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(x, \pi)\}$
3 : **return** π

Fig. 6. Experiments for the simulation security of a NIZK: zero-knowledge (ZK^b) and simulation-soundness (SS).

Simulation-Secure PKE with Uniform Keys.

Definition 12 (PKE). *A public-key encryption scheme is a tuple* $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *with public-key space* $\mathcal{PK}$ *and encryption randomness space* $\mathcal{RD}$*, with the following syntax:*

- $\mathsf{KeyGen}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$ *Generates a public key* $\mathsf{pk} \in \mathcal{PK}$ *and a secret key.*
- $\mathsf{Enc}(\mathsf{pk}, m) \to \mathbf{c}$ *Encrypts a message* m *under* pk*, producing a ciphertext* $\mathbf{c}$*. We write* $\mathsf{Enc}(\mathsf{pk}, m; \mathsf{rnd})$ *for the deterministic encryption using explicit randomness* $\mathsf{rnd} \in \mathcal{RD}$*.*
- $\mathsf{Dec}(\mathsf{sk}, \mathbf{c}) \to m$ *Decrypts* $\mathbf{c}$ *using* sk*.*

It is correct over $\mathcal{RD}$ *if, with overwhelming probability over* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, decryption succeeds for* every *accepted randomness simultaneously:*

$$\Pr_{(\mathsf{pk},\mathsf{sk})\leftarrow\mathsf{KeyGen}(1^\lambda)}\left[\,\exists\, m,\ \mathsf{rnd} \in \mathcal{RD}\ :\ \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m; \mathsf{rnd})) \neq m\,\right] \leq \mathsf{negl}(\lambda).$$

In particular, decryption recovers the message even for an adversarially chosen $\mathsf{rnd} \in \mathcal{RD}$*, not only for honestly sampled randomness.*

Definition 13 (Simulation-Secure PKE with Uniform Keys). *A PKE scheme* $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *with public-key space* $\mathcal{PK}$ *is* simulation-secure with uniform keys *if there exists a PPT ciphertext simulator* $\mathsf{Sim.Enc}$ *such that the following two properties hold. We call a key sampled uniformly from* $\mathcal{PK}$ *an* oblivious *key, i.e. it carries no secret.*

Key simulation: *A key produced by the honest key generation is indistinguishable from a uniformly sampled one:*

$$\{\mathsf{pk} : (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)\} \approx_c \{\mathsf{pk} : \mathsf{pk} \xleftarrow{\$} \mathcal{PK}\}.$$

Ciphertext simulation: *Under an oblivious key, ciphertexts can be produced without the message: for every PPT adversary* $\mathcal{A}$*,*

$$\left|\Pr\left[\mathsf{CS}^0_{\mathsf{PKE},\mathcal{A}}(1^\lambda) = 1\right] - \Pr\left[\mathsf{CS}^1_{\mathsf{PKE},\mathcal{A}}(1^\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda),$$

where the experiment $\mathsf{CS}^b_{\mathsf{PKE},\mathcal{A}}$ *is defined in Fig. 7.*

$\mathsf{CS}^b_{\mathsf{PKE},\mathcal{A}}(1^\lambda)$	$\mathsf{EncO}(m)$
1 : $\mathsf{pk} \xleftarrow{\$} \mathcal{PK}$	1 : **if** $b = 0$ **then** $\mathbf{c} \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, m)$
2 : **return** $\mathcal{A}^{\mathsf{EncO}}(\mathsf{pk})$	2 : **if** $b = 1$ **then** $\mathbf{c} \xleftarrow{\$} \mathsf{Sim.Enc}(\mathsf{pk})$
	3 : **return** $\mathbf{c}$

Fig. 7. Experiment for the ciphertext simulation of a PKE scheme under an oblivious (uniformly sampled) public key.

B.2 The Transform

Fix a relation $\mathcal{R}$ with language $\mathcal{L}$. The transform combines the two building blocks:

- a simulation-secure PKE $\mathsf{PKE} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with uniform keys (Def. 13), public-key space $\mathcal{PK}$, ciphertext simulator $\mathsf{PKE.Sim.Enc}$, and encryption randomness space $\mathcal{RD}$ over which it is correct (Def. 12);
- a simulation-sound NIZK $\mathsf{NIZK}_0 = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ (Def. 11), with simulator $\mathsf{Sim}_0 = (\mathsf{Sim}_0.\mathsf{Setup}, \mathsf{Sim}_0.\mathsf{Sim})$, for the augmented relation

$$\mathcal{R}_0 = \left\{ \big((x, \mathsf{pk}, \mathbf{c}),\ (w, \mathsf{rnd})\big) : \begin{array}{c} (x, w) \in \mathcal{R} \ \wedge\ \mathsf{rnd} \in \mathcal{RD} \\ \wedge\ \ \mathbf{c} = \mathsf{PKE.Enc}(\mathsf{pk}, w; \mathsf{rnd}) \end{array} \right\},$$

  with language $\mathcal{L}_0$.

The resulting NIZK $\mathsf{NIZK} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ for $\mathcal{R}$ is described in Fig. 8. It is defined in the random oracle model, using a hash function $H : \{0,1\}^* \to \mathcal{PK}$ modeled as a random oracle to derive the PKE public key $\mathsf{pk} = H(\mathsf{seed})$ from a public seed placed in the CRS; as H is a random oracle, this key is an oblivious (uniform) key that no party can invert to a trapdoor. Intuitively, each proof encrypts the witness under pk and uses NIZK_0 to prove that this ciphertext encrypts a valid witness. The trapdoor of pk then enables straight-line extraction, while the strong simulation soundness of NIZK_0 prevents the adversary from mauling simulated proofs.

$\mathsf{Setup}(1^\lambda)$

1 : $\mathsf{crs}_0 \leftarrow \mathsf{NIZK}_0.\mathsf{Setup}(1^\lambda)$
2 : $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda$
3 : **return** $\mathsf{crs} := (\mathsf{crs}_0, \mathsf{seed})$

$\mathsf{Verify}(\mathsf{crs} = (\mathsf{crs}_0, \mathsf{seed}), x, \pi = (\mathbf{c}, \pi_0))$

1 : $\mathsf{pk} := H(\mathsf{seed})$
2 : $b \leftarrow \mathsf{NIZK}_0.\mathsf{Verify}(\mathsf{crs}_0, (x, \mathsf{pk}, \mathbf{c}), \pi_0)$
3 : **return** b

$\mathsf{Prove}(\mathsf{crs} = (\mathsf{crs}_0, \mathsf{seed}), x, w)$

1 : $\mathsf{pk} := H(\mathsf{seed})$
2 : $(\mathbf{c}, \mathsf{rnd}) \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, w)$
3 : $\pi_0 \leftarrow \mathsf{NIZK}_0.\mathsf{Prove}(\mathsf{crs}_0, (x, \mathsf{pk}, \mathbf{c}), (w, \mathsf{rnd}))$
4 : **return** $\pi := (\mathbf{c}, \pi_0)$

Fig. 8. The simulation-extractable NIZK NIZK from a simulation-sound NIZK NIZK_0 and a simulation-secure PKE PKE.

Theorem 4. *In the random-oracle model, if* NIZK_0 *is a simulation-sound NIZK for* $\mathcal{R}_0$ *and* PKE *is simulation-secure with uniform keys, then the NIZK* NIZK *of Fig. 8 is simulation-extractable (Def. 10) for* $\mathcal{R}$*, with the simulator* Sim *of Fig. 9.*

$\mathsf{Sim.Setup}(1^\lambda)$

1 : $(\mathsf{crs}_0, \mathsf{st}_0) \leftarrow \mathsf{NIZK}_0.\mathsf{Sim.Setup}(1^\lambda)$
2 : $(\mathsf{pk}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$
3 : $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda$
4 : $H[\mathsf{seed}] := \mathsf{pk}$ // program the oracle
5 : **return** $(\mathsf{crs} := (\mathsf{crs}_0, \mathsf{seed}), \mathsf{st} := (\mathsf{st}_0, \mathsf{pk}, \mathsf{sk}_{\mathsf{PKE}}))$

$\mathsf{Sim.Sim}(x, \mathsf{st} = (\mathsf{st}_0, \mathsf{pk}, \mathsf{sk}_{\mathsf{PKE}}))$

1 : $\mathbf{c} \leftarrow \mathsf{PKE.Sim.Enc}(\mathsf{pk})$
2 : $(\pi_0, \mathsf{st}_0) \leftarrow \mathsf{NIZK}_0.\mathsf{Sim.Sim}((x, \mathsf{pk}, \mathbf{c}), \mathsf{st}_0)$
3 : **return** $(\pi := (\mathbf{c}, \pi_0), \mathsf{st})$

$\mathsf{Sim.Ext}(x, \pi = (\mathbf{c}, \pi_0), \mathsf{st})$

1 : **return** $w := \mathsf{PKE.Dec}(\mathsf{sk}_{\mathsf{PKE}}, \mathbf{c})$

Fig. 9. The simulator $\mathsf{Sim} = (\mathsf{Sim.Setup}, \mathsf{Sim.Sim}, \mathsf{Sim.Ext})$ for the NIZK of Fig. 8.

Proof. We show that SIMEXT^0 and SIMEXT^1 (Fig. 4) are indistinguishable through a sequence of hybrids. Fix a PPT adversary $\mathcal{A}$ making at most q queries to ProveO.

H_0: the real experiment SIMEXT^0. Here $\mathsf{crs} = (\mathsf{crs}_0, \mathsf{seed})$ with $\mathsf{crs}_0 \leftarrow \mathsf{NIZK}_0.\mathsf{Setup}(1^\lambda)$ and $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda$; the PKE key $\mathsf{pk} = H(\mathsf{seed})$ is uniform over $\mathcal{PK}$, since H is a random oracle and seed is fresh. The oracle ProveO answers with honest proofs of Fig. 8, and VerifyO returns $\mathsf{Verify}(\mathsf{crs}, x, \pi)$.

H_1: as H_0, but we simulate the base CRS and every base proof: $(\mathsf{crs}_0, \mathsf{st}_0) \leftarrow \mathsf{NIZK}_0.\mathsf{Sim.Setup}(1^\lambda)$, and each ProveO call produces $\pi_0 \leftarrow \mathsf{NIZK}_0.\mathsf{Sim.Sim}((x, \mathsf{pk}, \mathbf{c}), \mathsf{st}_0)$ rather than an honest proof (note that $\mathbf{c} \leftarrow \mathsf{PKE.Enc}(\mathsf{pk}, w)$ still encrypts the real witness). Since $(x, w) \in \mathcal{R}$ and $\mathsf{rnd} \in \mathcal{RD}$ imply $((x, \mathsf{pk}, \mathbf{c}), (w, \mathsf{rnd})) \in \mathcal{R}_0$, this is a legitimate use of the zero-knowledge oracle.

We can define an adversary $\mathcal{B}_1$ against the zero-knowledge of NIZK_0 that simulates H_0 and H_1 for $\mathcal{A}$. It uses the crs and ProveO oracle of its zero-knowledge challenger to answer $\mathcal{A}$'s queries, and outputs whatever $\mathcal{A}$ outputs. Then,

$$|\Pr[\mathsf{H}_1 = 1] - \Pr[\mathsf{H}_0 = 1]| = \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{NIZK}_0}(\mathcal{B}_1),$$

H_2: as H_1, but each ProveO ciphertext is simulated, $\mathbf{c} \leftarrow \mathsf{PKE.Sim.Enc}(\mathsf{pk})$ instead of $\mathsf{PKE.Enc}(\mathsf{pk}, w)$. The oracle no longer uses w and now coincides with Sim.Sim from Fig. 9.

By the simulation security of PKE with uniform keys, we can define an adversary $\mathcal{B}_2$ against the ciphertext simulation of PKE such that $|\Pr[\mathsf{H}_2 = 1] - \Pr[\mathsf{H}_1 = 1]| = \mathsf{Adv}^{\mathsf{cs}}_{\mathsf{PKE}}(\mathcal{B}_2)$.

H_3: as H_2, but $\mathsf{pk} = H(\mathsf{seed})$ is now *trapdoored*: we sample $(\mathsf{pk}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and program $H[\mathsf{seed}] := \mathsf{pk}$, exactly as Sim.Setup does.

Distinguishing this programming from a uniform pk directly breaks the key simulation of PKE. Formally, we can define an adversary $\mathcal{B}_3$ against the key simulation of PKE such that $|\Pr[\mathsf{H}_3 = 1] - \Pr[\mathsf{H}_2 = 1]| = \mathsf{Adv}^{\mathsf{ks}}_{\mathsf{PKE}}(\mathcal{B}_3)$.

At this point, crs and ProveO are exactly those of Sim.Setup and Sim.Sim, while VerifyO still returns $\mathsf{Verify}(\mathsf{crs}, x, \pi)$.

H_4: the ideal experiment SIMEXT^1. Compared to H_3, ProveO now additionally records each simulated pair (x, π) in a list $\mathcal{W}$, initially empty. Moreover, VerifyO uses $\mathcal{W}$: on a query (x, π) with $\mathsf{Verify}(\mathsf{crs}, x, \pi) = 1$ and $(x, \pi) \notin \mathcal{W}$, it extracts $w' \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}_{\mathsf{PKE}}, \mathbf{c})$ and reports an extraction failure when $(x, w') \notin \mathcal{R}$ instead of returning 1. The two hybrids agree unless $\mathcal{A}$ makes a VerifyO query $(x, \pi = (\mathbf{c}, \pi_0))$ triggering the following *bad event* Bad:

1. $\mathsf{NIZK}_0.\mathsf{Verify}(\mathsf{crs}_0, (x, \mathsf{pk}, \mathbf{c}), \pi_0) = 1$ (the proof verifies);
2. $(x, \pi) \notin \mathcal{W}$; and
3. $(x, w') \notin \mathcal{R}$ for $w' \leftarrow \mathsf{PKE.Dec}(\mathsf{sk}_{\mathsf{PKE}}, \mathbf{c})$.

It remains to bound $\Pr[\mathsf{Bad}]$, which we do through a reduction $\mathcal{B}_4$ against the simulation soundness of NIZK_0. On input crs_0 from its challenger, $\mathcal{B}_4$ samples $(\mathsf{pk}, \mathsf{sk}_{\mathsf{PKE}}) \leftarrow \mathsf{PKE.KeyGen}(1^\lambda)$ and $\mathsf{seed} \xleftarrow{\$} \{0,1\}^\lambda$, programs $H[\mathsf{seed}] := \mathsf{pk}$, and runs $\mathcal{A}$ on $\mathsf{crs} := (\mathsf{crs}_0, \mathsf{seed})$. It answers each $\mathsf{ProveO}(x, w)$ query by sampling $\mathbf{c} \leftarrow \mathsf{PKE.Sim.Enc}(\mathsf{pk})$, querying $\pi_0 \leftarrow \mathsf{SimO}((x, \mathsf{pk}, \mathbf{c}))$, adding $(x, \pi := (\mathbf{c}, \pi_0))$ to $\mathcal{W}$, and returning π; it answers each $\mathsf{VerifyO}(x, \pi)$ query with $\mathsf{Verify}(\mathsf{crs}, x, \pi)$.

This perfectly simulates H_3, and thus H_4 until Bad occurs. Since $\mathcal{B}_4$ knows $\mathsf{sk}_{\mathsf{PKE}}$ (and $\mathcal{R}$ is efficiently decidable), it can detect the first $\mathsf{VerifyO}$ query $(x, (\mathbf{c}, \pi_0))$ triggering Bad, upon which it outputs $(x^*, \pi^*) := ((x, \mathsf{pk}, \mathbf{c}), \pi_0)$. We show that this is a winning output whenever Bad occurs:

- *Validity:* $\mathsf{NIZK}_0.\mathsf{Verify}(\mathsf{crs}_0, x^*, \pi^*) = 1$ by condition 1.
- *Freshness:* the set $\mathcal{Q}$ of the SS experiment contains exactly the pairs $((x_i, \mathsf{pk}, \mathbf{c}_i), \pi_{0,i})$ for the ProveO queries x_i of $\mathcal{A}$, answered with $(\mathbf{c}_i, \pi_{0,i})$. As pk is fixed throughout the game , $(x^*, \pi^*) \in \mathcal{Q}$ would imply $(x, (\mathbf{c}, \pi_0)) \in \mathcal{W}$, contradicting condition 2.
- *Falsity:* $x^* \notin \mathcal{L}_0$. Suppose otherwise: then $\mathbf{c} = \mathsf{PKE}.\mathsf{Enc}(\mathsf{pk}, w; \mathsf{rnd})$ for some w with $(x, w) \in \mathcal{R}$ and some $\mathsf{rnd} \in \mathcal{RD}$. Unless $(\mathsf{pk}, \mathsf{sk}_{\mathsf{PKE}})$ fails to be correct over $\mathcal{RD}$, which happens with probability at most δ_{PKE} (Def. 12), decryption then recovers $w' = \mathsf{PKE}.\mathsf{Dec}(\mathsf{sk}_{\mathsf{PKE}}, \mathbf{c}) = w$, hence $(x, w') \in \mathcal{R}$, contradicting condition 3.

Therefore $\Pr[\mathsf{Bad}] \leq \mathsf{Adv}^{\mathsf{ss}}_{\mathsf{NIZK}_0}(\mathcal{B}_4) + \delta_{\mathsf{PKE}}$, and since H_3 and H_4 agree unless Bad occurs, $|\Pr[\mathsf{H}_4 = 1] - \Pr[\mathsf{H}_3 = 1]| \leq \Pr[\mathsf{Bad}]$. Chaining the four transitions and using $\mathsf{H}_0 = \mathsf{SIMEXT}^0$ and $\mathsf{H}_4 = \mathsf{SIMEXT}^1$ yields

$$\begin{aligned}\left|\Pr\left[\mathsf{SIMEXT}^0 = 1\right] - \Pr\left[\mathsf{SIMEXT}^1 = 1\right]\right| \leq\ & \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{NIZK}_0}(\mathcal{B}_1) + \mathsf{Adv}^{\mathsf{cs}}_{\mathsf{PKE}}(\mathcal{B}_2) + \mathsf{Adv}^{\mathsf{ks}}_{\mathsf{PKE}}(\mathcal{B}_3) \\ & + \mathsf{Adv}^{\mathsf{ss}}_{\mathsf{NIZK}_0}(\mathcal{B}_4) + \delta_{\mathsf{PKE}},\end{aligned}$$

which is negligible under the theorem's assumptions and concludes the proof. □

C Parameter Selection

C.1 Choosing the Parameters

B_λ and $B_{\lambda^{-1}}$ We first bound the bounds B_λ and the inverses $B_{\lambda^{-1}}$, as both the hint coefficients and the noise term contain these terms. While we have a mathematically proven bound for them in Lem. 5, there are some discrepancies compared to the real bounds, introducing extra overhead to the final ciphertext moduli.

Therefore, we instead use empirically computed bounds of $\gamma\lambda_i^{\mathcal{I}}$ and $(\lambda_i^{\mathcal{I}})^{-1}$, following the analysis given in [KLL+26]. In [KLL+26], the authors reported that $\gamma\lambda_j^{\mathcal{I}}$ is maximized when choosing the evaluation points as follows:

$$\mathcal{S} = \begin{cases} \{X^i\}_{-(t-1)/2 \leq i \leq (t-1)/2} & \text{if t is odd} \\ \{X^i\}_{-t/2 < i \leq t/2} & \text{if t is even} \end{cases}.$$

On the other hand, $(\lambda_j^{\mathcal{I}})^{-1}$ is maximized when the evaluation point set is given as follows:

$$\mathcal{S} = \begin{cases} \{0\} \cup \{X^i\}_{-(t-1)/2 \leq i < (t-1)/2} & \text{if t is odd} \\ \{0\} \cup \{X^i\}_{-t/2 < i < t/2} & \text{if t is even} \end{cases}.$$

We also use the well-known trick of using evaluation points over the subring $\mathbb{Z}[Y]/(Y^{N'}+1)$ of $\mathcal{R}=\mathbb{Z}[X]/(X^N+1)$, where $N'=2^{\lceil \log n \rceil -1}$ and $Y=X^{N/N'}$. Then, the size of the Lagrange coefficients is only dependent on N', rather than N. Based on this observation, we always choose the minimal subring for the Lagrange coefficients to minimize the parameters.

Gaussian Parameters. We elaborate on how to choose parameters $\mathsf{pp}=(p,q,N,k,\sigma_s,B_r,\sigma_f)$ for our tPKE scheme and then provide a recommended parameter set and corresponding ciphertext sizes. First, we begin by explaining how to choose the (adaptive) hint-MLWE parameters. Recall that Thm. 1 states that the problem $\mathsf{AHMLWE}_{q,k,\ell,Q,\sigma_{\vec{x}},\sigma_y,\mathcal{V}_B}$ reduces to $\mathsf{MLWE}_{q,k,\ell,\sigma}$ where σ is a positive real number defined as $\sigma^{-2}=2(\sigma_{\vec{x}}^{-2}+\kappa^2 Q\sigma_y^{-2})$ for a constant $\kappa:=\sqrt{\frac{1-\sqrt{1-2B\eta_\varepsilon^2(\mathbb{Z}^N)/(\pi\sigma_y^2)}}{\eta_\varepsilon^2(\mathbb{Z}^N)/(\pi\sigma_y^2)}}$. Note that we simply chose the hint set to be $\mathcal{V}_B$, so we do not need to consider the first condition in the theorem. Next, following the convention for MLWE security, we fix $\sigma=3.2$. Note that this will naturally satisfy the third condition required for the reduction; $\sigma\geq\eta_\varepsilon(\mathbb{Z}^N)/\sqrt{\pi}$ for practical parameters with $N\leq 2^{16}$. We also observe that κ is essentially a monotonically decreasing function with respect to σ_y, which outputs $\sqrt{2B}$ if σ_y has the smallest possible value, $\sqrt{2B/\pi}\cdot\eta_\varepsilon(\mathbb{Z}^N)$, and outputs $\approx\sqrt{B}$ if σ_y is sufficiently large. Since having κ larger than the real value only makes the value of σ_y increase by less than half a bit, which has no negative effect on security at all, we can simply set $\kappa=\sqrt{2B}$. Then, the final condition reduces to $\sigma^{-2}=2(\sigma_{\vec{x}}^{-2}+2BQ\sigma_y^{-2})$. Since $\sigma_{\vec{x}}$ is real, we have $\sigma^{-2}>4BQ\sigma_y^{-2}$. Note that this implies $\sigma_y>2\sqrt{BQ/\pi}\cdot\eta_\varepsilon(\mathbb{Z}^N)$ combined with the third condition, thereby satisfying the condition $\sigma_y\geq\sqrt{2B/\pi}\cdot\eta_\varepsilon(\mathbb{Z}^N)$ naturally. Consequently, we only need to consider the final condition $\sigma^{-2}=2(\sigma_{\vec{x}}^{-2}+2BQ\sigma_y^{-2})$ when choosing the parameters for the adaptive Hint-MLWE problem.

Now, for our scheme to be correct and secure, it needs to satisfy Thm. 2 and Thm. 3. Given the threshold and the number of parties (t,n), these theorems state that the following conditions need to be satisfied:

1. $\gamma(\rho_r\sigma_r+e^{1/2}\cdot 2kN\sigma_r\sigma_s)+tB_\lambda\rho_f\sigma_f<\frac{q}{2p}-\frac{p}{2}$,
2. $\mathsf{AHMLWE}_{q,k,k,Q,\sigma_s,\sigma_f,\mathcal{V}_{B_1}}$ is hard for $B_1=e^{1/2}N^2(2k+1)(B_{\lambda^{-1}}\sigma_r)^2$,
3. $\mathsf{AHMLWE}_{q,k+1,k,n,\sigma_r,\sigma_f,\mathcal{V}_{B_2}}$ is hard for $B_2=B_{\lambda^{-1}}^2(1+2e^{1/2}kN^2{\sigma_s}^2)$,
4. $\mathsf{AHMLWE}_{q,k+Q_c,k,Q,\sigma_1,\sigma_f,\mathcal{V}_{B_3}}$ is hard for $\sigma_1^2=\frac{{\sigma_f}^2}{2\gamma^2}\left(1+\sqrt{1-\frac{2\gamma^2\eta_\varepsilon(\mathbb{Z}^N)^2}{\pi{\sigma_f}^2}}\right)$ and $B_3=e^{1/2}N^2(2k+1){\sigma_r}^2\max(\gamma,B_{\lambda^{-1}}B_\lambda)^2$.

Our strategy is to choose all the Gaussian parameters from the last three conditions and select q accordingly using the first condition. However, since all the Gaussian parameters σ_r, σ_s, and σ_f are convolved in the conditions for adaptive hint problems above, it is hard to formulate the optimization problem to minimize these Gaussian parameters. Therefore, we first heuristically obtain the Gaussian parameters by ignoring insignificant terms and then check if they satisfy the security conditions afterwards.

We first investigate the condition for the last adaptive Hint-MLWE problem (4). Observe that $\sigma_1^2 \leq {\sigma_f}^2/\gamma^2$ by definition, we obtain

$$\begin{aligned}\sigma^{-2} &\geq 2(\gamma^2 + 2B_3Q){\sigma_f}^{-2} \approx 4B_3Q{\sigma_f}^{-2} \\ &= (4e^{1/2}N^2(2k+1)\max(\gamma, B_{\lambda^{-1}}B_\lambda)^2Q){\sigma_r}^2{\sigma_f}^{-2}.\end{aligned}$$

Based on this inequality, we set σ_f the smallest possible value as

$$\sigma_f = \sqrt{4e^{1/2}N^2(2k+1)\max(\gamma, B_{\lambda^{-1}}B_\lambda)^2Q} \cdot \sigma\sigma_r.$$

Now, by substituting this into the conditions for the first two adaptive-Hint MLWE problems (2 and 3), we obtain the following equalities:

- $\sigma^{-2} = 2\left({\sigma_s}^{-2} + 0.5B_{\lambda^{-1}}^2 \max(\gamma, B_{\lambda^{-1}}B_\lambda)^{-2}\sigma^{-2}\right)$
- $\sigma^{-2} = 2\left(1 + 2B_2 n/\left(4e^{1/2}N^2(2k+1)\max(\gamma, B_{\lambda^{-1}}B_\lambda)^2Q\sigma^2\right)\right) \cdot {\sigma_r}^{-2}$

Since the first equality does not contain any unknown variables, we obtain an explicit formula for $\sigma_s := \sigma\sqrt{2/\left(1 - B_{\lambda^{-1}}^2/\max(\gamma, B_{\lambda^{-1}}B_\lambda)^2\right)}$. Then, as B_2 can be computed with known constants and σ_s, we also get the value for σ_r by substituting σ_s into the second equality. Recall that we used an approximate equality when computing σ_f, which may have reduced the size of σ_f. Therefore, we check if the condition for the final adaptive Hint-MLWE problem (4) actually holds, *i.e.,* the resulting σ is larger than 3.2. If not, we can simply increase the size of σ_f by setting it $\sqrt{2(\gamma^2 + 2B_3Q)} \cdot \sigma$ without changing σ_r and σ_s, since a larger σ_f results in an increased security level for the first two adaptive Hint-MLWE problems (2 and 3), with larger σ. We note this last condition will likely hold nevertheless, as we already use a naïve bound σ_f/γ for σ_1. Finally, it remains to choose the smallest q that satisfies the first condition.