Adaptively CCA-Secure Threshold Decryption from Lattices with Polynomial Modulus

Yuxin Zhang[1,2], Yunxiao Zhou[1,2], Shuai Han[1,2(✉)], Shengli Liu[1,2(✉)], and Xinyi Huang[3]

[1] School of Computer Science,
Shanghai Jiao Tong University, Shanghai 200240, China
{yuxin.zhang,cloudzhou,dalen17,slliu}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
huangxinyi@nuaa.edu.cn

Abstract. Threshold Decryption (TD) enables a set of decryptors, each holding a secret key share, to collaboratively decrypt ciphertexts. In this paper, we resolve the open problem raised by Devevey et al. [PKC 2021] of constructing an *adaptively* secure TD with *polynomial* modulus from lattices. As identified by Hwang et.al. [ePrint 2026/1585], solving this open problem is a particularly challenging task. The most challenging part is how to limit the leakage of secret key shares from decryption queries while keeping the modulus a polynomial.

In this paper, we resolve the above open problem with two TD schemes $\mathsf{TD}_1, \mathsf{TD}_2$ from lattices in the random oracle model.

- TD_1 is a non-interactive TD scheme achieving adaptive CCA-security with polynomial modulus for a small number of decryptors and bounded decryption queries. It can be extended to achieve robustness and works well in the asynchronous settings.
- TD_2 is a non-interactive TD scheme achieving adaptive CCA-security with polynomial modulus for bounded decryption queries. Compared with TD_1, it supports any polynomial number of decryptors and works well in the synchronous settings.

To our best knowledge, our TD_1 and TD_2 are the *first* non-interactive TD schemes achieving adaptive security and polynomial modulus from lattices. The core techniques behind our schemes include the polynomial noise flooding technique and the coupled *refined Leftover Hash Lemma* (LHL) working for adaptive noisy linear leakage, which may be of independent interest.

1 Introduction

Threshold Decryption (TD) is a type of public-key encryption (PKE) in which multiple decryptors jointly perform decryption. With a (t, N)-TD scheme, the secret (decryption) key is pre-shared among N decryptors, each holding a secret key share. To decrypt a ciphertext, each decryptor computes a partial decryption with its own secret key share via a decryption protocol. Then t partial

decryptions can be combined to recover the underlying plaintext from the ciphertext. Since its introduction [17], TD has been widely deployed as a useful cryptographic tool to support applications in blockchains [2, 11], multi-party computation [9, 53], e-voting [13], etc.

TD schemes can be categorized into asynchronous and synchronous settings [54, 33]. In the *synchronous* setting, the set of t participating decryptors is fixed at the outset of the decryption protocol. In the *asynchronous* setting, any t decryptors can contribute their partial decryptions to recover the underlying plaintext, so we do not have to fix the set of t decryptors at the beginning of the decryption protocol. We study both types of TD schemes in this paper.

In the decryption protocol of TD, if the partial decryptions do not need interactions among the decryptors, then decryption protocol is non-interactive and hence round-optimal. Such a TD becomes a *non-interactive* one as well. In this paper, we focus on constructing *non-interactive* TD schemes.

Adaptive CCA Security for TD. Recall that the de facto security notion for PKE is IND-CCA security. This naturally extends to the security of TD. Note that TD is deployed in a setting of multiple (N) decryptors, so its IND-CCA security allows the adversary $\mathcal{A}$ to obtain partial decryptions from N decryption oracles. Moreover, in the multi-decryptor setting, $\mathcal{A}$ may corrupt some decryptors to obtain their secret key shares and internal ephemeral randomness through a corruption oracle. To avoid trivial attacks, $\mathcal{A}$ cannot obtain/compute t (or more) valid partial decryptions for the challenge ciphertext from the decryption oracles and the corruption oracle.

There are two types of corruptions. With *static corruptions*, $\mathcal{A}$ obtains beforehand a fixed set of (at most $t-1$) corrupted secret key shares when the CCA experiment of TD begins. In contrast, with *adaptive corruptions*, $\mathcal{A}$ can adaptively issue corruption queries and obtain decryptors' secret key shares and ephemeral randomness throughout the CCA experiment. Adaptive corruptions captures the corruption attacks in the real world more accurately than static corruptions. Therefore, the expected security for TD is *CCA security under adaptive corruptions*, which is called *adaptive CCA security* for short.

Most of the existing TD schemes [3, 7, 43, 12, 8, 33, 45, 46, 54, 35, 6, 28] consider only static corruptions. Only few works [37, 18, 14, 52] achieve adaptive CCA security: the TD scheme [37] is based on the subgroup membership assumptions from pairings, the TD schemes [18] are from DCR or LWE, and the recent TD schemes [14, 52] are from the DDH assumption in pairing-free groups. Recently, Boudgous et al. [6] mentioned that the 3-round TD scheme in [35] may also achieve adaptive CCA security, but a formal proof is missing.

Post-Quantum Secure TD. NIST has been calling for and working on post-quantum transition. There are many efforts on post-quantum secure TD schemes, most of which are constructed from lattices. We list the features and securities of the existing non-interactive TD from lattices in Table 1.

As shown in Table 1, most of the existing works [3, 7, 43, 12, 8, 33, 45, 46, 54] obtain only CPA-security under static corruptions. Recently, Brzuska et al. [10] proposed a generic CPA-to-CCA transform. However, this transform does not

result in adaptive CCA-secure TD schemes, since lattice-based adaptively CPA-secure TD scheme is not available to satisfy the requirements asked in the transform [10]. Up to now, there is only one lattice-based TD scheme proposed by Devevey et al. [18] enjoying adaptive CCA security based on the LWE assumption. However, their modulus q in the LWE assumption has to be a super-polynomial one. It is known that the LWE problem is quantumly linked to the "short vector" problems on n-dimensional lattices with a factor of $\tilde{O}(q\sqrt{n})$. The super-polynomial size of modulus q weakens the hardness of LWE assumption [47]. Therefore, Devevey et al. [18] left the following open problem.

"It remains an open problem to prove adaptive security under a more common LWE assumption with polynomial approximation factor."

Addressing this open problem is regarded as a particularly challenging task in [28]. Many works tried to reduce the modulus size to polynomial. For example, the (t, N)-TD scheme [7] has a modulus $q = O(\sqrt{\ell} \cdot \mathsf{poly}(\lambda))$, where ℓ indicates the number of encryption queries. The work in [12] has modulus $q = O(N^t \cdot \sqrt{\ell} \cdot \mathsf{poly}(\lambda))$. However, both of works only achieves CPA security under static corruptions. A progress by Micciancio and Suhl [43] made the modulus not only polynomial but also independent of the number of encryption queries ℓ. But their TD scheme again achieves only CPA security under static corruptions and it only supports the special (N, N)-TD scheme. The most recent progress owes to Hwang et.al [28], who presented a TD scheme with polynomial modulus $q = N^{O(t)} \cdot \sqrt{\ell} \cdot \mathsf{poly}(\lambda)$ achieving simulation-based CCA security under static corruptions.

Therefore, each of the existing lattice-based TD schemes either has adaptive security with super-polynomial modulus, or has a polynomial modulus but only achieve static security. Thus the problem of constructing lattice-based TD schemes with adaptive security left by Devevey et al. [18] is still open up to now.

Our Contributions. We resolve the above open problem with two TD schemes TD_1 and TD_2, which are the *first* non-interactive TD schemes that enjoy not only adaptive CCA security but also polynomial modulus[1] from lattices.

- Our TD_1 is an asynchronous (t, N)-TD scheme, and its adaptive CCA security is based on the MLWE assumption with modulus $q = O(N^{t/2} \cdot \sqrt{\ell} \cdot \mathsf{poly}(\lambda))$ where ℓ denotes the number of partial decryption queries. The modulus q is polynomial for settings of small user number N and bounded query number ℓ. We also extend TD_1 to $\mathsf{TD}_1^{\mathsf{rob}}$ to achieve robustness. The robustness of $\mathsf{TD}_1^{\mathsf{rob}}$ is guaranteed by the public verifiability of partial decryptions to identify misbehaving decryptors.
- Our TD_2 is a synchronous (t, N)-TD scheme, and its adaptive CCA security is based on the MLWE and MSIS assumptions with modulus $q = O(\sqrt{\ell} \cdot$

[1] Following the convention of prior works [7, 12, 28], we take $q = O(\sqrt{\ell})$ as a polynomial modulus. This is also justified in the asymptotic setting, where the number of adversarial queries ℓ is polynomially bounded and so is $q = O(\sqrt{\ell})$. In the concrete setting, many recent works [16, 32, 48] also consider $q = O(\sqrt{\ell})$ as an acceptable modulus for $\ell \leq 2^{64}$, as suggested by NIST [44, 4.A.2].

Table 1. Comparison with existing *non-interactive* threshold decryption schemes from lattices (a) in the asynchronous setting and (b) in the synchronous setting. Here the column "**Security**" indicates whether the TD scheme has CPA or CCA security, where the symbol CPA*/CCA* indicates that the CPA/CCA security is a simulation-based one. "**Adaptive Corruptions**" indicates whether the security is under adaptive corruptions. "**Modulus**" shows the asymptotic growth of the modulus q, where ℓ denotes the number of encryption and/or partial decryption queries. "**Robustness**" indicates whether the TD scheme satisfies robustness, where "✓*" for our TD_1 indicates that it can be extended to $\mathsf{TD}_1^{\mathsf{rob}}$ achieving robustness. **"Assumption"** specifies the underlying hardness assumptions.

(a) Non-interactive TD schemes from lattices in the *asynchronous* setting.

TD Schemes	**Security**	**Adaptive Corruptions**	**Modulus**	**Robustness**	**Assumption**
BGG+18 [3]	CPA*	×	$\mathsf{superpoly}(\lambda)$	×	LWE
BS23 [7]	CPA	×	$\sqrt{\ell} \cdot \mathsf{poly}(\lambda)$	✓	MLWE
MS24 [43]	CPA*	×	$\mathsf{poly}(\lambda)$	×	LWE
CLW25 [12]	CPA*	×	$N^t \cdot \sqrt{\ell} \cdot \mathsf{poly}(\lambda)$	×	LWE
BFM+25 [8]	CPA*	×	$\mathsf{superpoly}(\lambda)$	✓	RLWE
KLL+26 [33]	CPA*	×	$\mathsf{superpoly}(\lambda)$	×	RLWE
HKM+26 [28]	CCA*	×	$N^{O(t)} \cdot \sqrt{\ell} \cdot \mathsf{poly}(\lambda)$	×	MLWE
DLN+21 [18]	CCA	✓	$\mathsf{superpoly}(\lambda)$	✓	LWE
Our TD_1	CCA	✓	$N^{t/2} \cdot \sqrt{\ell} \cdot \mathsf{poly}(\lambda)$	✓*	MLWE

(b) Non-interactive TD schemes from lattices in the *synchronous* setting.

TD Schemes	**Security**	**Adaptive Corruptions**	**Modulus**	**Robustness**	**Assumption**
OT25 [45], OT26 [46]	CPA^*	×	$\mathsf{superpoly}(\lambda)$	×	LWE/MLWE
CPS26 [54]	CPA^*	×	$\mathsf{superpoly}(\lambda)$	×	LWE
Our TD_2	CCA	✓	$\sqrt{\ell} \cdot \mathsf{poly}(\lambda)$	×	MLWE & MSIS

$\mathsf{poly}(\lambda)$). The modulus q is polynomial if the number ℓ of the decryption queries is bounded. Compared to TD_1, TD_2 supports any polynomial user number N but works in a synchronous setting.

- The core techniques for our TD_1 and TD_2 to support adaptive CCA security with polynomial modulus include polynomial noise flooding and the coupled *refined Leftover Hash Lemma* (LHL) working for adaptive noisy linear leakage. Our core techniques may have wide applications dealing with leakages dependent on the extractor in LHL, which is of independent interest.

In Table 1, we compare our TD_1 and TD_2 with existing non-interactive TD schemes from lattices. We further provide concrete parameter choices and efficiency estimates for both schemes at the 128-bit security level in Appendix E. For $N \leq 16$, TD_1 achieves ciphertext sizes of 119∼178 KB. For $N = 1024$,TD_2 achieves ciphertexts sizes of 101 KB, while each user's partial decryption is only 1.87 KB. These demonstrate the practical feasibility of our schemes.

Technical Overview. Our threshold decryption schemes stem from the dual Regev public-key encryption (PKE) scheme [26]. With the basic CPA-secure dual Regev scheme, one can use public key $\mathsf{pk} = \mathbf{Ak}$ to encrypt a message μ

and get a ciphertext consisting of $(\mathbf{c}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top,\ d = \mathbf{s}^\top\mathsf{pk} + e' + \mathsf{Encode}(\mu))$, where $\mathbf{c}$ is an MLWE instance, e' is an MLWE error, and $\mathsf{Encode}(\mu)$ helps to remove noises in the decryption. To leverage it to a CCA-secure PKE, we can resort to a tag-based non-interactive zero-knowledge (NIZK) proof with one-time simulation soundness (OTSS) [50, 15] to prove that $\mathbf{c}$ is an MLWE instance with witness $(\mathbf{s}, \mathbf{e})$. Accordingly, the CCA-secure PKE works as follows.

- The secret key is a small-norm vector $\mathsf{sk} = \mathbf{k}$ and the public key is $\mathsf{pk} = \mathbf{Ak}$.
- The ciphertext for message μ is defined by

$$\mathsf{ct} := \big(\mathbf{c}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top,\ d = \mathbf{s}^\top\mathsf{pk} + e' + \mathsf{Encode}(\mu),\ \pi \leftarrow \mathsf{Prv}(\mathbf{c}, \tau, (\mathbf{s}, \mathbf{e}))\big),$$

  where $\mathbf{s}$, $\mathbf{e}$, e' are of small norm, and $\tau := \mathsf{H}(\mathbf{c}, d)$ serves as a tag of NIZK with H a collision-resistant hash function.
- To decrypt $\mathsf{ct} = (\mathbf{c}, d, \pi)$, the decryptor first verifies π with Vfy of NIZK, computes $h := d - \mathbf{c}^\top\mathsf{sk}$ and obtains $\mu := \mathsf{Decode}(h)$.

We briefly recall its CCA-security proof, which goes with four main steps. The goal is to show that an adversary $\mathcal{A}$ can hardly tell whether a challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}^*, d^*, \pi^*)$ encrypts μ_0 or μ_1, even if $\mathcal{A}$ gets $\mathsf{pk} = \mathbf{Ak}$ and can obtain decryptions of any ciphertexts $\mathsf{ct} = (\mathbf{c}, d, \pi) \neq \mathsf{ct}^*$.

(1) For $\mathsf{ct}^* = (\mathbf{c}^*, d^*, \pi^*)$, d^* can be rewritten as $\mathbf{c}^{*\top}\mathsf{sk}+d'$ for some indepdendent noise d', and then we use the zero-knowledge simulator of NIZK to generate π^*, without using the witness for $\mathbf{c}^*$.
(2) We change $\mathbf{c}^*$ in ct^* to a uniformly random one by the MLWE assumption.
(3) We resort to the OTSS of NIZK to ensure that the decryption of $\mathsf{ct} = (\mathbf{c}, d, \pi)$ succeeds only when $\mathbf{c}$ is an MLWE instance (since $\mathcal{A}$ can hardly forge a valid proof π for a non-MLWE instance $\mathbf{c}$, even if it sees a simulated π^*). This can be used to limit the information about sk to pk in the decryption oracle.
(4) Finally, we argue that given pk, the secret key sk has enough entropy left, and then by the (generalized) leftover hash lemma (LHL) [20], the uniformly random $\mathbf{c}^*$ acts as an extractor so that $\mathbf{c}^{*\top}\mathsf{sk}$ in d^* is uniformly random, and thus ct^* is totally independent of μ_0 and μ_1.

Note that there is only one security reduction with the MLWE assumption (i.e., the second step) and this security reduction does not involve the secret key $\mathsf{sk} = \mathbf{k}$. Therefore, the reduction algorithm can always sample sk by itself. This fact lays a very good foundation for threshold decryption to achieve CCA-security under adaptive corruptions, since the reduction algorithm can always generate sk and the key shares to freely deal with adaptive corruptions from adversaries.

Thresholdize PKE & Limit Leakage of sk from Partial Decryptions. We will construct TD schemes by thresholdizing the above CCA-secure PKE. Observe that its decryption is just a linear operation with sk, so it is friendly to linear secret sharing (LSS). Let us share $\mathsf{sk} = \mathbf{k}$ with an LSS scheme (e.g., the Shamir secret sharing [51]), and produce N secret key shares $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$. For partial decryption, each decryptor i may compute a decryption share

$$h_i := \mathbf{c}^\top\mathsf{sk}_i \tag{1}$$

and then a combiner can combine the decryption shares from set S as

$$\sum_{i\in S}\lambda_{i,S}\cdot h_i=\sum_{i\in S}\lambda_{i,S}\cdot(\mathbf{c}^\top\mathsf{sk}_i)=\mathbf{c}^\top(\sum_{i\in S}\lambda_{i,S}\cdot\mathsf{sk}_i)=\mathbf{c}^\top\mathsf{sk},$$

and then recover μ as before. Here $\{\lambda_{i,S}\}$ are the combining coefficients determined by S depending on the LSS scheme (e.g., $\{\lambda_{i,S}\}$ are just the Lagrange interpolation coefficients for Shamir secret sharing).

Though the decryption works correctly, it is completely insecure, since the partial decryption $h_i:=\mathbf{c}^\top\mathsf{sk}_i$ in (1) leaks too much information about sk_i: collecting enough equations of h_i w.r.t different $\mathbf{c}$ can completely solve sk_i. To circumvent this problem, one needs to find a way to limit the leakage of sk_i while preserving the decryption correctness. There are generically two solutions.

- **Noise flooding: relying on super-polynomial modulus.** One common way to address this problem is adding some noise flooding term y_i to the partial decryption (1), as in [3, 18]. The idea is that as long as $\mathbf{c}^\top=\mathbf{s}^\top\mathbf{A}+\mathbf{e}^\top$ (which can be ensured by the OTSS-NIZK), then in the partial decryption,

$$h_i:=\mathbf{c}^\top\mathsf{sk}_i+y_i=\mathbf{s}^\top\mathbf{A}\mathsf{sk}_i+\mathbf{e}^\top\mathsf{sk}_i+y_i.$$

  We expect noise y_i smudges $\mathbf{e}^\top\mathsf{sk}_i$ so $\mathbf{A}\mathsf{sk}_i$ covers the leakage of sk_i in h_i. However, for smudging, y_i has to been chosen super-polynomial large to hide $\mathbf{e}^\top\mathsf{sk}_i$, leading to a *super-polynomial modulus* q.

  Then by combining decryption shares from set S, one can get

$$\sum_{i\in S}\lambda_{i,S}\cdot h_i=\sum_{i\in S}\lambda_{i,S}\cdot(\mathbf{c}^\top\mathsf{sk}_i+y_i)=\mathbf{c}^\top\mathsf{sk}+\sum_{i\in S}\lambda_{i,S}\cdot y_i,$$

  The combination incurs an error $\sum_{i\in S}\lambda_{i,S}\cdot y_i$, which might be too large to recover the plaintext when the combining coefficients $\{\lambda_{i,S}\}$ are large.
- **Masking technique: not work for polynomial modulus.** Another possible solution is adding some random mask Δ_i to the partial decryption (1)

$$h_i:=\mathbf{c}^\top\mathsf{sk}_i+\Delta_i=\mathbf{s}^\top\mathbf{A}\mathsf{sk}_i+\mathbf{e}^\top\mathsf{sk}_i+\Delta_i.$$

  This masking technique was introduced in [16] and has been used in the design of many threshold signature schemes [16, 32] and threshold fully homomorphic encryption [54]. With this technique, each mask Δ_i is uniformly random, so it can perfectly hide the term $\mathbf{e}^\top\mathsf{sk}_i$; moreover, the masks $\{\Delta_i\}_{i\in S}$ in set S satisfy $\sum_{i\in S}\Delta_i=0$, so that these masks can be canceled during combination. (Technically, these masks are generated by random oracle on secret seeds pre-shared between each pair of users in their secret key shares.)

  Moreover, to enable decryption correctness, we needs to "move" the combining coefficient $\lambda_{i,S}$ from the combination to partial decryption, namely

$$h_i:=\lambda_{i,S}\cdot\mathbf{c}^\top\mathsf{sk}_i+\Delta_i, \tag{2}$$

  and the combination computes

$$\sum_{i\in S}h_i=\sum_{i\in S}\lambda_{i,S}\cdot\mathbf{c}^\top\mathsf{sk}_i+\sum_{i\in S}\Delta_i=\mathbf{c}^\top\mathsf{sk}. \tag{3}$$

However, this technique in fact *does not solve the leakage problem* — although the mask Δ_i hides the leakage of sk_i in partial decryption (2), the cancellation of the masks during combination (3) completely leaks $\mathbf{c}^\top \mathsf{sk}$. Then sk can be recovered if one gets enough $\mathbf{c}^\top \mathsf{sk}$ w.r.t. different $\mathbf{c}$.

In summary, these existing two solutions either rely on super-polynomial modulus, or do not work at all in limiting the leakage of secret key.

In this work, we propose different new solutions, resulting in our two TD schemes TD_1, TD_2 with polynomial modulus.

Our solution 1: TD_1 via Polynomial Noise Flooding + Adaptive LHL. In our TD_1, we use the replicated secret sharing (RSS) [29] to share sk with short shares $\mathsf{sk}_i = \{\mathbf{k}_I | I \in \mathcal{I}, i \in I\}$ and special combining coefficients of 1s.

Roughly speaking, our scheme TD_1 is designed according to the above noise flooding method, but we extend the noise flooding technique to support polynomial modulus. Namely, the partial decryption of TD_1 is given by

$$h_i = \{h_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i} \quad \text{with} \quad h_{i,I} := \mathbf{c}^\top \mathbf{k}_I + y_{i,I} = \mathbf{s}^\top \mathbf{A} \mathbf{k}_I + \mathbf{e}^\top \mathbf{k}_I + y_{i,I}, \qquad (4)$$

where we use a *polynomial* noise $y_{i,I} \leftarrow D_{\mathcal{R},\sigma_y}$ to smudge the term $\mathbf{e}^\top \mathbf{k}_I$. Any t or more partial decryptions can be combined to

$$\textstyle\sum_{i\in S} \sum_{I\in w_i} h_{i,I} = \sum_{i\in S} \sum_{I\in w_i} \mathbf{c}^\top \mathbf{k}_I + y_{i,I} = \mathbf{c}^\top \mathbf{k} + \sum_{i\in S} \sum_{I\in w_i} y_{i,I},$$

where $(w_i)_{i\in S}$ is determined by the set S via specific algorithm of RSS (cf. Subsect. 2.2). For correctness, the error term $\sum_{i\in S} \sum_{I\in w_i} y_{i,I}$ contains $C := \binom{N}{N-t+1}$ items, so it follows distribution $D_{\mathcal{R},\sigma}$ with $\sigma = \sqrt{C} \cdot \sigma_y$. If N is small, then $\sqrt{C} \cdot \sigma_y$ is polynomial-bounded. This yields polynomial modulus q.

For adaptive CCA security, to exclude trivial attacks, the adversary $\mathcal{A}$ can issue at most $t-1$ corruption/decryption queries to obtain at most $t-1$ secret shares and partial decryptions of the challenge ciphertext ct^*. Then RSS guarantees the existence of $\mathbf{k}_{\tilde{I}}$ (with a specific index $\tilde{I}$) about which $\mathcal{A}$ learns nothing from the $t-1$ queries (cf. Subsect. 2.2). Meanwhile, $\mathcal{A}$ can also choose any $\mathsf{ct} \neq \mathsf{ct}^*$ and issue queries to obtain partial decryptions of ct. Suppose that $\mathcal{A}$ makes at most ℓ decryption queries, and the j-th query is $\mathsf{ct}^{(j)} = (\mathbf{c}^{(j)}, d^{(j)}, \pi^{(j)})$, where $\mathbf{c}^{(j)\top} = \mathbf{s}^{(j)\top} \mathbf{A} + \mathbf{e}^{(j)\top}$ (ensured by the OTSS-NIZK). Then via these queries, by (4), what $\mathcal{A}$ learns about the specific $\mathbf{k}_{\tilde{I}}$ is covered by the projection leakage $\mathbf{A}\mathbf{k}_{\tilde{I}}$ and ℓ noisy linear leakages $\{\mathbf{e}^{(j)\top} \mathbf{k}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}$, where each $y^{(j)}_{i,\tilde{I}} \leftarrow D_{\mathcal{R},\sigma_y}$.

The key observation is that, for achieving CCA-security, we do not need to fully hide the terms $\mathbf{e}^{(j)\top} \mathbf{k}_{\tilde{I}}$ in $\mathbf{e}^{(j)\top} \mathbf{k}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}$ (in which case a super-polynomial $y^{(j)}_{\tilde{I}}$ is inevitable). Instead, it suffices to ensure that, conditioned on the project leakage $\mathbf{A}\mathbf{k}_{\tilde{I}}$ and noisy linear leakages $\{\mathbf{e}^{(j)\top} \mathbf{k}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}$, the term $\mathbf{c}^{*\top} \mathbf{k}_{\tilde{I}}$ in

$$d^* = \mathbf{s}^{*\top} \mathbf{A}\mathbf{k} + e'^* + \mathsf{Encode}(\mu_\beta) = \mathbf{c}^{*\top} \mathsf{sk} + d' = \mathbf{c}^{*\top} \mathbf{k}_{\tilde{I}} + \textstyle\sum_{I\in\mathcal{I}, I\neq\tilde{I}} \mathbf{c}^{*\top} \mathbf{k}_I + d' \quad (5)$$

is (statistically close to) uniform, so as to hide the plaintexts μ_0/μ_1 perfectly.

Achieving the above goal requires two key steps: first change $\mathbf{c}^{*\top} = \mathbf{s}^{*\top}\mathbf{A} + \mathbf{e}^{*\top}$ to a uniformly random vector, and then argue the uniformity of $\mathbf{c}^{*\top}\mathbf{k}_{\tilde{I}}$. To establish the first step with a polynomial modulus, we make use of the Semi-Adaptive Hint-MLWE in [34]. The Semi-Adaptive Hint-MLWE assumption guarantees that $\mathbf{c}^{*\top} = \mathbf{s}^{*\top}\mathbf{A} + \mathbf{e}^{*\top}$ can be replaced by a uniformly random vector, even conditioned on a hint $e'^{*} - \mathbf{e}^{*\top}\mathbf{k}$, where $e'^{*} - \mathbf{e}^{*\top}\mathbf{k}$ is used to simulate the term $\mathbf{s}^{*\top}\mathbf{A}\mathbf{k} + e'^{*} = \mathbf{c}^{*\top}\mathbf{k} + e'^{*} - \mathbf{e}^{*\top}\mathbf{k}$ in d^{*} (5) perfectly.

Below we show how we establish the second step with a polynomial modulus, i.e., conditioned on $\mathbf{A}\mathbf{k}_{\tilde{I}}$ and $\{\mathbf{e}^{(j)\top}\mathbf{k}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}$, it holds $\mathbf{c}^{*\top}\mathbf{k}_{\tilde{I}} \approx_s \mathsf{uniform}$.

Firstly, we deal with the projection leakage $\mathbf{A}\mathbf{k}_{\tilde{I}}$. We set $\mathbf{A} := \mathbf{U}\left[\mathbf{B}\middle|\ \mathbf{B}\mathbf{R} + \mathbf{E}\middle|\ \mathbf{I}\right]$ from an MLWE sample and uniform matrix $\mathbf{U}$, so that $\mathbf{A}\mathbf{D} = \mathbf{0}$ with a small-norm and full rank $\mathbf{D} := \begin{bmatrix}-\mathbf{R}\\ \mathbf{I}\\ -\mathbf{E}\end{bmatrix}$. Then by [19, Lemma 7], we can apply a Gaussian decomposition to get $\mathbf{k}_{\tilde{I}} = \mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} + \mathbf{k}^{(1)}_{\tilde{I}}$, where $\mathbf{k}^{(0)}_{\tilde{I}}$ is independent of $\mathbf{k}^{(1)}_{\tilde{I}}$. As a result, $\mathbf{A}\mathbf{k}_{\tilde{I}} = \mathbf{A}(\mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} + \mathbf{k}^{(1)}_{\tilde{I}}) = \mathbf{A}\mathbf{k}^{(1)}_{\tilde{I}}$, which perfectly hides $\mathbf{k}^{(0)}_{\tilde{I}}$, and $\{\mathbf{e}^{(j)\top}\mathbf{k}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]} = \{\mathbf{e}^{(j)\top}\mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}} + \mathbf{e}^{(j)\top}\mathbf{k}^{(1)}_{\tilde{I}}\}_{j\in[\ell]}$ and $\mathbf{c}^{*\top}\mathbf{k}_{\tilde{I}} = \mathbf{c}^{*\top}\mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} + \mathbf{c}^{*\top}\mathbf{k}^{(1)}_{\tilde{I}}$. Thus, it suffices to show that conditioned on $\{\mathbf{e}^{(j)\top}\mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}$, it holds $\mathbf{c}^{*\top}\mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} \approx_s \mathsf{uniform}$. (In this way, we get rid of the projection leakage.)

Next, we need to deal with the noisy linear leakage. Since $\mathbf{D}$ is a full rank matrix, $\mathbf{u}^{*\top} := \mathbf{c}^{*\top}\mathbf{D}$ is uniformly random and can serve as a seed of extractor. Then one may want to use the leftover hash lemma (LHL) to show that $\mathbf{c}^{*\top}\mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} = \mathbf{u}^{*\top}\mathbf{k}^{(0)}_{\tilde{I}} \approx_s \mathsf{uniform}$ conditioned on the noisy linear leakage $\{\mathbf{e}^{(j)\top}\mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}$. However, we stress that the LHL is *not applicable* here, since the leakage $\{\mathbf{e}^{(j)\top}\mathbf{D}\mathbf{k}^{(0)}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}$ may depend on $\mathbf{u}^{*\top} = \mathbf{c}^{*\top}\mathbf{D}$.[2] To see this, note that $\mathcal{A}$ may issue partial decryption queries $\mathsf{ct}^{(j)} = (\mathbf{c}^{(j)}, d^{(j)}, \pi^{(j)})$ with $\mathbf{c}^{(j)\top} = \mathbf{s}^{(j)\top}\mathbf{A} + \mathbf{e}^{(j)\top}$ after obtaining the challenge ciphertext $\mathsf{ct}^{*} = (\mathbf{c}^{*}, d^{*}, \pi^{*})$, so $\mathbf{e}^{(j)\top}$ can arbitrarily depend on $\mathbf{c}^{*}$ (and hence $\mathbf{u}^{*\top} = \mathbf{c}^{*\top}\mathbf{D}$).

To resolve the problem, we propose a new variant of LHL called *adaptive LHL with noisy linear leakage*, which is coupled with the polynomial noise flooding technique and applies to the setting of adaptively chosen noisy linear leakage. With a careful parameter choice and delicate analysis, our adaptive LHL shows that polynomial flooding noise $\{y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}$ ensures that for uniformly random $\mathbf{u}^{*}$,

$$(\mathbf{u}^{*\top}\mathbf{k}^{(0)}_{\tilde{I}}, \{(\mathbf{e}^{(j)\top}\mathbf{D})\mathbf{k}^{(0)}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}) \approx_s (\mathsf{uniform}, \{(\mathbf{e}^{(j)\top}\mathbf{D})\mathbf{k}^{(0)}_{\tilde{I}} + y^{(j)}_{i,\tilde{I}}\}_{j\in[\ell]}),$$ [3]

[2] Recall that the (generalized) LHL [20] guarantees that if a source $\mathbf{k}$ has enough minimal entropy given some leakage Y, then a uniformly random $\mathbf{u}^{*}$ can extract $\mathbf{u}^{*\top}\mathbf{k}$ from $\mathbf{k}$ s.t. $\mathbf{u}^{*\top}\mathbf{k} \approx_s \mathsf{uniform}$ conditioned on Y. However, the leakage Y must be independent of $\mathbf{u}^{*}$ so that LHL can apply.

[3] Here we apply our adaptive LHL (Lemma 8 in Sect. 3) with $\mathbf{e}^{(j)\top}\mathbf{D}$, which is of small norm since both $\mathbf{e}^{(j)\top}$ and $\mathbf{D}$ are of small norm.

where each $y_{i,\tilde{I}}^{(j)} \leftarrow D_{\mathcal{R},\sigma_y}$, and each $\mathbf{e}^{(j)}$ may adaptively depend on $\mathbf{u}^*$, $\mathbf{u}^{*\top}\mathbf{k}_{\tilde{I}}^{(0)}$/ $\mathsf{uniform}$ and the first $j-1$ leakages $\{\mathbf{e}^{(r)\top}\mathbf{k}_{\tilde{I}}^{(0)}+y_{i,\tilde{I}}^{(r)}\}_{r\in[j-1]}$, as long as each $\mathbf{e}^{(j)\top}\mathbf{D}$ is of small norm. We refer to Sect. 3 for more details of our adaptive LHL. Then we proved the CCA security for TD_1.

As for adaptive corruptions, inheriting from the dual Regev PKE, in the CCA security proof of our TD_1, all the security reductions do not involve the secret key $\mathsf{sk} = \mathbf{k}$ and all the secret shares $\{\mathsf{sk}_i\}$ and all the randomness $\{y_{i,I}\}$ used in partial decryptions, thus the reduction algorithm can always sample them by itself and answer the corruption queries with $\{\mathsf{sk}_i\}$ and $\{y_{i,I}\}$. Therefore, our TD_1 achieves adaptive CCA security.

As for robustness, we further extend our TD_1 to $\mathsf{TD}_1^{\mathsf{rob}}$, by adding adaptively sound NIZK proofs $\pi_{i,I}$ to prove the well-formedness of $h_{i,I}$ in partial decryption (4). Then, one can verify the vaildity of π_i to identify misbehaving decryptors. In this way, our extension $\mathsf{TD}_1^{\mathsf{rob}}$ additionally achieves robustness. The added NIZK does not affect the adaptive CCA security proof since we only use the statistical zero-knowledge property of NIZK for the adaptive CCA security proof.

Our Solution 2: TD_2 via Masking + Polynomial Noise Flooding + Adaptive Hint-MLWE. Due to the use of RSS in our TD_1, its modulus is polynomial only for small user number N. To enable larger N, our second solution is going back to the Shamir secret sharing scheme for sharing sk.

Our construction of TD_2 proceeds in two steps: we first construct an adaptively semi-malicious CPA (smCPA) secure TD scheme $\mathsf{TD}_2^{\mathsf{smCPA}}$, and then upgrade it to the adaptively CCA secure TD_2 via the generic smCPA-to-CCA transform by Brzuska, Klooß and Woo (BKW) [10].

Our $\mathsf{TD}_2^{\mathsf{smCPA}}$ is based on an optimal variant of dual Regev PKE [38], where $\mathsf{pk} = \mathbf{A}\mathbf{k}+\mathbf{f}$ and $\mathsf{sk} = \mathbf{k}$. We share sk with Shamir secret sharing to obtain secret key shares $\mathsf{sk}_1, \ldots, \mathsf{sk}_N$, and integrate polynomial noise flooding technique with the masking technique to upgrade the PKE [38] to $\mathsf{TD}_2^{\mathsf{smCPA}}$. More precisely, the partial decryption of $\mathsf{TD}_2^{\mathsf{smCPA}}$ is similar to (2), but it adds both a random mask Δ_i and a polynomial noise y_i, i.e.,

$$h_{i,S} := \lambda_{i,S} \cdot \mathbf{c}^\top \mathsf{sk}_i + y_i + \Delta_i, \tag{6}$$

where $y_i \leftarrow D_{\mathcal{R},\sigma_y}$. Then by $\sum_{i\in S} \Delta_i = 0$, the combination (3) becomes

$$\begin{aligned}\textstyle\sum_{i\in S} h_{i,S} &= \textstyle\sum_{i\in S} \lambda_{i,S} \cdot \mathbf{c}^\top \mathsf{sk}_i + \sum_{i\in S} y_i = \mathbf{c}^\top \mathbf{k} + \sum_{i\in S} y_i \\ &= \mathbf{s}^\top \mathbf{A}\mathbf{k} + \mathbf{e}^\top \mathbf{k} + \textstyle\sum_{i\in S} y_i = \mathbf{s}^\top \mathsf{pk} - \mathbf{s}^\top \mathbf{f} + \mathbf{e}^\top \mathbf{k} + \sum_{i\in S} y_i.\end{aligned}$$

Thanks to polynomial-sized y_i and small-norm $\mathbf{s}$, $\mathbf{f}$, $\mathbf{e}$ and $\mathbf{k}$, the correctness of $\mathsf{TD}_2^{\mathsf{smCPA}}$ follows with a polynomial modulus q.

[4] As mentioned in (2), the coefficient $\lambda_{i,S}$ is "moved" from the combination to partial decryption to ensure correctness, and accordingly, the partial decryption of our TD_2 needs to know the set S to compute the coefficient $\lambda_{i,S}$.

$\mathsf{TD}_2^{\mathsf{smCPA}}$ achieves adaptive smCPA security, which is weaker than adaptive CCA security by allowing $\mathcal{A}$ to make partial decryption queries only for honestly-generated ciphertext $\mathsf{ct} = (\mathbf{c}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top, d)$ with known encryption randomness $(\mathbf{s}, \mathbf{e})$ or for the challenge ciphertext ct^*. To prove adaptive smCPA security, we can answer the partial decryption queries (i, ct, S) equivalently as follows.

Case (i): If i is not the last honest decryptor in S, the partial decryption is answered with a uniformly random $h_{i,S} \leftarrow\!\!\$\ \mathcal{R}_q$ instead of (6), thanks to the uniform mask Δ_i.

Case (ii): If i is the last honest one in S (denoted by $i = \mathsf{last}$), the queried ciphertext ct must be distinct from ct^* in order to exclude trivial attacks (otherwise $\mathcal{A}$ can recover μ_0/μ_1 itself by combining the partial decryptions), so ct has to be a honestly-generated ciphertext with known randomness $(\mathbf{s}, \mathbf{e})$. Accordingly, the partial decryption is answered with

$$h_{\mathsf{last},S} := \sum_{i\in\mathsf{HS}} y_i + \mathbf{s}^\top\mathsf{pk} - \mathbf{s}^\top\mathbf{f} + \mathbf{e}^\top\mathbf{k} - \sum_{i\in\mathsf{HS}\setminus\{\mathsf{last}\}} h_{i,S} - \sum_{i\in\mathsf{CS}} (\lambda_{i,S}\mathbf{c}^\top\mathbf{k}_i + \Delta_i), \quad (7)$$

where HS records honest decryptors in S and CS records corrupted ones in S ($S = \mathsf{HS}\cup\mathsf{CS}$). This ensures the sum $\sum_{i\in\mathsf{HS}} h_{i,S}$ follows the correct distribution:[5]

$$\begin{aligned}\textstyle\sum_{i\in\mathsf{HS}} h_{i,S} &= \textstyle\sum_{i\in\mathsf{HS}} y_i + \mathbf{s}^\top\mathsf{pk} - \mathbf{s}^\top\mathbf{f} + \mathbf{e}^\top\mathbf{k} - \sum_{i\in\mathsf{CS}} \lambda_{i,S}\mathbf{c}^\top\mathbf{k}_i - \sum_{i\in\mathsf{CS}} \Delta_i \\ &= \textstyle\sum_{i\in\mathsf{HS}} y_i + \sum_{i\in\mathsf{HS}} \lambda_{i,S}\mathbf{c}^\top\mathsf{sk}_i + \sum_{i\in\mathsf{HS}} \Delta_i,\end{aligned}$$

where the last equality holds since $\mathbf{s}^\top\mathsf{pk} - \mathbf{s}^\top\mathbf{f} + \mathbf{e}^\top\mathbf{k} = \mathbf{c}^\top\mathbf{k} = \sum_{i\in S} \lambda_{i,S}\mathbf{c}^\top\mathbf{k}_i$, and $\sum_{i\in S} \Delta_i = 0$.

Overall, for Case (i), $\mathcal{A}$ learns nothing about the secret $\mathbf{k}$ and $\mathbf{f}$ from the partial decryption oracle's answers; for Case (ii), we have $\mathsf{ct} \neq \mathsf{ct}^*$, and $\mathcal{A}$ can obtain (at most) $y_{\tilde{i}} - \mathbf{s}^\top\mathbf{f} + \mathbf{e}^\top\mathbf{k}$ in (7), where $\tilde{i} \in \mathsf{HS}$ is an honest decryptor in S that is never corrupted by $\mathcal{A}$ throughout the game and thus its randomness $y_{\tilde{i}} \leftarrow D_{\mathcal{R},\sigma_y}$ is unknown to $\mathcal{A}$.[6]

Suppose that $\mathcal{A}$ makes at most ℓ decryption queries, and the j-th query is $(i^{(j)}, \mathsf{ct}^{(j)}, S^{(j)})$ with $\mathsf{ct}^{(j)} = (\mathbf{c}^{(j)\top} = \mathbf{s}^{(j)\top}\mathbf{A} + \mathbf{e}^{(j)\top}, d^{(j)})$. Then via these queries, what $\mathcal{A}$ learns about $\mathbf{k}$ and $\mathbf{f}$ is covered by $\{y_{\tilde{i}}^{(j)} - \mathbf{s}^{(j)\top}\mathbf{f} + \mathbf{e}^{(j)\top}\mathbf{k}\}_{j\in[\ell]}$, where $\mathbf{s}^{(j)}$ and $\mathbf{e}^{(j)}$ may be adaptively chosen by $\mathcal{A}$. According to the Adaptive-Hint MLWE (AHMLWE) assumption recently introduced in [28] (cf. Def. 4), we know that

$$(\mathsf{pk} = \mathbf{A}\mathbf{k}+\mathbf{f}, \{y_{\tilde{i}}^{(j)} - \mathbf{s}^{(j)\top}\mathbf{f} + \mathbf{e}^{(j)\top}\mathbf{k}\}_{j=1}^{\ell}) \approx_c (\mathsf{uniform}, \{y_{\tilde{i}}^{(j)} - \mathbf{s}^{(j)\top}\mathbf{f} + \mathbf{e}^{(j)\top}\mathbf{k}\}_{j=1}^{\ell}). \quad (8)$$

Thus $[\mathbf{A} \mid \mathsf{pk}]$ is uniform. Recall that the challenge ciphertext ct^* can be written as $\mathsf{ct}^* = (\mathbf{c}^{*\top}, d^*) = \mathbf{s}^{*\top}[\mathbf{A} \mid \mathsf{pk}] + [\mathbf{e}^{*\top} \mid e'^*] + [\mathbf{0}^\top \mid \mathsf{Encode}(\mu_\beta)]$. Then by the

[5] The equation below shows that $\sum_{i\in\mathsf{HS}} h_{i,S}$ equals the sums of (6) over $i \in \mathsf{HS}$, so it follows the correct distribution.

[6] This is because the decryptor set $S = \mathsf{HS}\cup\mathsf{CS}$ has size $|S| \geq t$ and $\mathcal{A}$ can corrupt at most $t-1$ decryptors in total to avoid trivial attacks. So there must be a decryptor $\tilde{i}$ in S that $\mathcal{A}$ never corrupts (and thus in HS).

MLWE assumption, $\mathbf{s}^{*\top}[\mathbf{A} \mid \mathsf{pk}] + [\mathbf{e}^{*\top} \mid e'^{*}]$ can be replaced by a uniform one, which perfectly hides μ_β in ct^*. This shows the smCPA security of $\mathsf{TD}_2^{\mathsf{smCPA}}$.

However, we note that it is non-trivial for our $\mathsf{TD}_2^{\mathsf{smCPA}}$ to achieve *adaptive* smCPA security. We need to find a way to deal with adaptive corruptions. Note that the above analysis shows that the partial decryption queries leak nothing about the secret key shares sk_i. Therefore, when $\mathcal{A}$ adaptively corrupts a decryptor (say i), the reduction algorithms can output a randomly sample key share sk_i, and explain the already generated partial decryption $h_{i,S}$ with a properly-distributed randomness y_i and set $\Delta_i := h_{i,S} - y_i - \lambda_{i,S} \cdot \mathbf{c}^\top \mathsf{sk}_i$, to ensure the consistency of (6). However, recall that $\mathcal{A}$ can obtain the decryption randomness y_i through corruptions, but the reduction algorithm to AHMLWE gets $(y_{\bar{i}} - \mathbf{s}^\top \mathbf{f} + \mathbf{e}^\top \mathbf{k})$ from its own challenger (cf. (8)) so it cannot reveal the specific $y_{\bar{i}}$ to $\mathcal{A}$. To solve this problem, the reduction algorithm samples $\{y^{(1)}, \cdots, y^{(|\mathsf{HS}|-1)}\}$ and implicitly sets $y^{(|\mathsf{HS}|)} := y_{\bar{i}}$. When a decryptor i is corrupted, the reduction algorithm assigns an unused value from the sequence $\{y^{(1)}, \ldots, y^{(|\mathsf{HS}|-1)}\}$ to y_i. In this way, $y^{(|\mathsf{HS}|)}$ is always hidden from $\mathcal{A}$ and can be set as the $y_{\bar{i}}$.[7] This change is unnoticeable, due to the identical distributions of $\{y_i\}_{i \in \mathsf{HS}}$ and the sum $\sum_{i \in \mathsf{HS}} y_i - \mathbf{s}^\top \mathbf{f} + \mathbf{e}^\top \mathbf{k} = \sum_{k=1}^{|\mathsf{HS}|-1} y^{(k)} + (y_{\bar{i}} - \mathbf{s}^\top \mathbf{f} + \mathbf{e}^\top \mathbf{k})$ suffices for the reduction algorithm to compute partial decryption in (7). This yields the *adaptive* smCPA security for our $\mathsf{TD}_2^{\mathsf{smCPA}}$.

Finally, we upgrade $\mathsf{TD}_2^{\mathsf{smCPA}}$ to our adaptively CCA secure TD_2 via the generic smCPA-to-CCA transform by BKW [10], preserving the polynomial modulus. We refer to Subsect. 5.2 for more details.

Concurrent Work [28]. We are aware of a concurrent work [28], which also constructs a non-interactive CCA-secure TD scheme with polynomial modulus for a small number of users and bounded decryption queries. More precisely, they first establish a threshold Regev scheme with static simulation-based semi-malicious CPA security, relying on their adaptive Hint-MLWE (AHMLWE) assumption, and then upgrade it to static simulation CCA security by applying the generic CPA-to-CCA transform in [10]. Their work is primarily efficiency-oriented and provides optimized concrete parameters for up to $N = 32$ users.

Their security guarantee, however, is restricted to static corruptions. In fact, as pointed in [28, p. 13], the CCA security under adaptive corruptions "appears particularly challenging to achieve with a polynomial moduli".

In contrast, we resolve this challenging task by proposing the first adaptively CCA-secure threshold decryption schemes with polynomial modulus. Moreover, robustness against maliciously generated partial decryptions is not considered in [28], whereas our extended $\mathsf{TD}_1^{\mathsf{rob}}$ achieves robustness.

Our adaptively secure constructions TD_1 and the original version of TD_2 were developed independently and concurrently with [28]. After both works became public, we further incorporated their AHMLWE assumption and Lantern-based simulation-extractable NIZK to improve the concrete efficiency of our TD_2.

[7] As noted in Footnote 6, $\mathcal{A}$ cannot corrupt all decryptors in HS to avoid trivial attacks.

2 Preliminaries

Notations. Let $\lambda \in \mathbb{N}$ denote the security parameter throughout the paper, and all algorithms, distributions, functions and adversaries take 1^λ as an implicit input. If x is defined by y or the value of y is assigned to x, we write $x := y$. For $i, j \in \mathbb{N}$ with $i < j$, define $[i, j] := \{i, i+1, ..., j\}$ and $[j] := \{1, 2, ..., j\}$. For a set $\mathcal{X}$, denote by $x \leftarrow_\$ \mathcal{X}$ the procedure of sampling x from $\mathcal{X}$ uniformly at random. If $\mathcal{D}$ is a distribution, $x \leftarrow \mathcal{D}$ means that x is sampled according to $\mathcal{D}$. All our algorithms are probabilistic unless stated otherwise. We use $y \leftarrow_\$ \mathcal{A}(x)$ to define the random variable y obtained by executing algorithm $\mathcal{A}$ on input x. If $\mathcal{A}$ is deterministic we write $y \leftarrow \mathcal{A}(x)$. "PPT" abbreviates probabilistic polynomial-time. Denote by negl some negligible function.

Let X and Y be two random variables. The statistical distance of X and Y is defined as $\Delta(X, Y) := \frac{1}{2} \sum_x |\Pr[X = x] - \Pr[Y = x]|$. If $\Delta(X, Y) \leq \mathsf{negl}(\lambda)$, X and Y are called statistically indistinguishable, and denoted by $X \approx_s Y$.

In Appendix A, we present some additional preliminaries, including the syntax of collision-resistant hash functions in Appendix A.1 and the syntax of tag-based quasi-adaptive non-interactive zero-knowledge argument (QA-NIZK) for a gap language and its one-time simulation soundness (OTSS) in Appendix A.2.

2.1 Lattice Backgrounds

Let $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$, with representatives in $(-q/2, q/2]$. Let $\mathcal{R} := \mathbb{Z}[X]/(X^d + 1)$ for d a power of 2 and $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$ for prime q satisfying $q \equiv 1 \bmod 2d$. For $a \in \mathcal{R}$, let $\phi : \mathcal{R} \to \mathbb{Z}^d$ be the coefficient embedding of a and $\phi^{-1} : \mathbb{Z}^d \to \mathcal{R}$ be its inverse. ϕ naturally extends to vectors over $\mathcal{R}$. We abuse this notion for $a \in \mathcal{R}_q$ and $\phi(a) \in \mathbb{Z}_q^d$. For any $a \in \mathcal{R}_q$, let $\mathsf{Rot}(a) \in \mathbb{Z}_q^{d \times d}$ denote the negacyclic matrix associated with a, defined by $\phi(ab) = \mathsf{Rot}(a) \cdot \phi(b)$ for all $b \in \mathcal{R}_q$. We extend this notation element-wisely to matrix $\mathbf{A} = (a_{i,j}) \in \mathcal{R}_q^{n \times m}$, i.e. $\mathsf{Rot}(\mathbf{A}) := (\mathsf{Rot}(a_{i,j}))_{i,j} \in \mathbb{Z}_q^{nd \times md}$. Let $K := \mathbb{R}[X]/(X^d + 1)$.

For a vector $\mathbf{v}$, let $\|\mathbf{v}\|$ (resp., $\|\mathbf{v}\|_\infty$) denote the ℓ_2 (resp., ℓ_∞) norm of $\mathbf{v}$. For a matrix $\mathbf{A}$, let $\|\mathbf{A}\|$ (resp., $\|\mathbf{A}\|_\infty$) denote the largest ℓ_2 (resp., ℓ_∞) norm of $\mathbf{A}$'s columns, and let $s_1(\mathbf{A}) := \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\| = \max_{\mathbf{x} \neq \mathbf{0}} \|\mathbf{A}\mathbf{x}\| / \|\mathbf{x}\|$ denote the largest singular value (spectral norm) of $\mathbf{A}$. For a vector $\mathbf{v} \in \mathcal{R}^m$, these norms are interpreted through its coefficient embedding into a standard Euclidean space.

For a symmetric positive definite matrix $\mathbf{\Sigma} \in \mathbb{R}^{m \times m}$, we write $\sqrt{\mathbf{\Sigma}}$ to denote its square root of $\mathbf{\Sigma}$ satisfying $\sqrt{\mathbf{\Sigma}}^\top \sqrt{\mathbf{\Sigma}} = \mathbf{\Sigma}$. The Gaussian function with covariance matrix $\mathbf{\Sigma}$ and center $\mathbf{c} \in \mathbb{R}^m$ is defined as $\rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\mathbf{x}) := e^{-\pi(\mathbf{x}-\mathbf{c})^\top \mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{c})}$ for $\mathbf{x} \in \mathbb{R}^m$. For a countable set $\Lambda \subset \mathbb{R}^m$, the discrete Gaussian distribution $D_{\Lambda,\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\mathbf{x}) := \rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\mathbf{x}) / \rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\Lambda)$ for $\mathbf{x} \in \Lambda$ and $D_{\Lambda,\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\mathbf{x}) := 0$ for $\mathbf{x} \notin \Lambda$, where $\rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\Lambda) := \sum_{\mathbf{x} \in \Lambda} \rho_{\sqrt{\mathbf{\Sigma}},\mathbf{c}}(\mathbf{x})$. When $\mathbf{\Sigma} = \sigma^2 \mathbf{I}_m$ with $\sigma > 0$, we write $\rho_{\sigma,\mathbf{c}}$ and $D_{\Lambda,\sigma,\mathbf{c}}$ for the corresponding spherical Gaussian function and distribution, respectively. We omit $\mathbf{c}$ when $\mathbf{c} = \mathbf{0}$. The Gaussian function naturally extends to $\mathbf{x} \in K^m$ and $\mathbf{c} \in K^m$ through the coefficient embedding ϕ.

Let Λ be a non-zero lattice. The *smoothing parameter* $\eta_\epsilon(\Lambda)$ of Λ is the smallest real $s > 0$ s.t. $\rho_{1/s}(\Lambda^*) \leq 1+\epsilon$, where $\Lambda^* = \{\mathbf{b}^* \in \mathrm{span}_{\mathbb{R}}(\Lambda) : {\mathbf{b}^*}^\top \cdot \Lambda \subseteq \mathbb{Z}\}$ refers to the dual of Λ. We say that $\sqrt{\mathbf{\Sigma}} \geq \eta_\epsilon(\Lambda)$ if $\rho_{\sqrt{\mathbf{\Sigma}^+}}(\Lambda^*) \leq 1+\epsilon$, where $\mathbf{\Sigma}^+$ denotes the Moore-Penrose pseudo-inverse of $\mathbf{\Sigma}$. When the covariance matrix $\mathbf{\Sigma}$ is positive definite and the lattice Λ is full-rank, $\sqrt{\mathbf{\Sigma}} \geq \eta_\epsilon(\Lambda)$ is equivalent to $\lambda_{\min}(\mathbf{\Sigma}) \geq \eta_\epsilon^2(\Lambda)$, where $\lambda_{\min}(\mathbf{\Sigma})$ denotes the minimal eigenvalue of $\mathbf{\Sigma}$.

Definition 1 (MLWE Assumption). *Let $q, m, n \in \mathbb{N}$ and $\sigma > 0$. The $\mathsf{MLWE}_{q,m,n,\sigma}$-assumption holds if for any PPT $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{MLWE}}_{q,m,n,\sigma,\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}^\top) = 1]\right| \leq \mathsf{negl}(\lambda)$, where $\mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n\times m}$, $\mathbf{s} \leftarrow D_{\mathcal{R}^n,\sigma}$, $\mathbf{e} \leftarrow D_{\mathcal{R}^m,\sigma}$ and $\mathbf{u} \leftarrow_{\$} \mathcal{R}_q^m$. Let $Q \in \mathbb{N}$, the multi-secret MLWE Q-$\mathsf{MLWE}_{q,m,n,\sigma}$ considers the case that $\mathbf{S} \leftarrow D_{\mathcal{R},\sigma}^{n\times Q}$, $\mathbf{E} \leftarrow D_{\mathcal{R},\sigma}^{m\times Q}$ and $\mathbf{U} \leftarrow_{\$} \mathcal{R}_q^{m\times Q}$.*

Definition 2 (MSIS Assumption [22]). *Let $q, m, n \in \mathbb{N}$ with $m > n$, and $\beta > 0$. The $\mathsf{MSIS}_{q,m,n,\beta}$-assumption holds, if for any PPT $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{MSIS}}_{q,m,n,\beta,\mathcal{A}}(\lambda) := \Pr\left[\mathbf{x} \leftarrow \mathcal{A}(\mathbf{A}) \ : \ \mathbf{x} \neq \mathbf{0} \wedge \|\mathbf{x}\| \leq \beta \wedge \mathbf{A}\mathbf{x} = \mathbf{0}\right] \leq \mathsf{negl}(\lambda)$, where $\mathbf{A} := [\mathbf{A}' \mid \mathbf{I}_n]$ for $\mathbf{A}' \leftarrow_{\$} \mathcal{R}_q^{n\times(m-n)}$.*

Lai et al. [34] proposed a variant of MLWE assumption, namely leaky MLWE. Below we recall a specific case of their leaky MLWE.

Definition 3 (Semi-Adaptive Hint-MLWE, special case of [34, Definition 3]). *Let $q, m, n, \ell \in \mathbb{N}$, $\sigma_0, \sigma_1 > 0$, and $\mathcal{S} \subseteq \mathcal{R}^{\ell\times m}$. The $\mathsf{SA\text{-}HMLWE}^{\ell,\sigma_1,\mathcal{S}}_{q,m,n,\sigma_0}$-assumption holds, if for any PPT $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, it holds that*

$$\mathsf{Adv}^{\mathsf{SA\text{-}HMLWE}}_{\mathsf{para},\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}_2(\mathbf{A}, \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top, \mathbf{K}, \mathbf{h}) = 1] - \Pr[\mathcal{A}_2(\mathbf{A}, \mathbf{u}^\top, \mathbf{K}, \mathbf{h}) = 1]\right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{para} = ((q, m, n, \sigma_0), (\ell, \sigma_1, \mathcal{S}))$, $\mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n\times m}$, $\mathbf{K} \leftarrow_{\$} \mathcal{A}_1(\mathbf{A})$, $\mathbf{K} \in \mathcal{S}$, $\mathbf{s} \leftarrow D_{\mathcal{R}^n,\sigma_0}$, $\mathbf{e} \leftarrow D_{\mathcal{R}^m,\sigma_0}$, $\mathbf{u} \leftarrow_{\$} \mathcal{R}_q^m$, and $\mathbf{h} = \mathbf{K}\mathbf{e} + \mathbf{e}'$ for $\mathbf{e}' \leftarrow D_{\mathcal{R}^\ell,\sigma_1}$.

Lemma 1 (MLWE $\Rightarrow$ SA-HMLWE, adapted from [34, Theorem 3 (Condition 1)]). *Let $q, m, n, \ell \in \mathbb{N}$ and $\mathcal{S} \subseteq \mathcal{R}^{\ell\times m}$. Let $B > 0$ be a real number s.t. $s_1(\mathsf{Rot}(\mathbf{K}))^2 \leq B$ for any $\mathbf{K} \in \mathcal{S}$. Let $\sigma_0, \sigma_1, \sigma > 0$ be real numbers satisfying $\frac{1}{\sigma^2 + 2\eta_\epsilon^2(\mathcal{R}^m)} = \frac{1}{\sigma_0^2} + \frac{B}{\sigma_1^2}$ and $\sigma \geq \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$.*

Then, for any PPT $\mathcal{A}$ against the $\mathsf{SA\text{-}HMLWE}^{\ell,\sigma_1,\mathcal{S}}_{q,m,n,\sigma_0}$ assumption, there exists a PPT $\mathcal{B}$ against the $\mathsf{MLWE}_{q,m,n,\sigma}$ assumption, s.t. $\mathsf{Adv}^{\mathsf{SA\text{-}HMLWE}}_{\mathsf{para},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{MLWE}}_{q,m,n,\sigma,\mathcal{B}}(\lambda) + 4\epsilon/(1-\epsilon)$, where $\mathsf{para} = ((q, m, n, \sigma_0), (\ell, \sigma_1, \mathcal{S}))$.

Very recently, Hwang et al. [28] proposed a variant of Hint-MLWE, named *Adaptive* Hint-MLWE (AHMLWE) to construct TD with simulation-based CCA security. AHMLWE requires the hardness for a PPT $\mathcal{A}$ to distinguish whether $\mathbf{b}$ is an MLWE instance $\mathbf{b} = \mathbf{A}\mathbf{k} + \mathbf{f}$ or a uniformly random vector, even if $\mathcal{A}$ can adaptively choose $(\mathbf{s}, \mathbf{e})$ and obtain the corresponding noisy hint $y + \mathbf{e}^\top\mathbf{k} + \mathbf{s}^\top\mathbf{f}$ after seeing $\mathbf{b}$. Hwang et al. [28] showed that AHMLWE can be reduced to MLWE under appropriate parameter settings, as summarized in Lemma 2.

Definition 4 (Adaptive Hint-MLWE [28]). *Let $q, m, n, \ell \in \mathbb{N}$, $\sigma_k, \sigma_y > 0$, and $\mathcal{V}$ be a set over $\mathcal{R}^{n+m}$. The $\mathsf{AHMLWE}^{\ell,\sigma_y,\mathcal{V}}_{q,m,n,\sigma_k}$ assumption holds, if for any PPT $\mathcal{A}$ making at most ℓ queries to the hint oracle $\mathcal{O}_{\text{HINT}}$, it holds that*

$$\mathsf{Adv}^{\mathsf{AHMLWE}}_{\mathsf{para},\mathcal{A}}(\lambda) := \left|\Pr\left[\mathcal{A}^{\mathcal{O}_{\text{HINT}}}(\mathbf{A}, \mathbf{Ak}+\mathbf{f}) = 1\right] - \Pr\left[\mathcal{A}^{\mathcal{O}_{\text{HINT}}}(\mathbf{A}, \mathbf{u}) = 1\right]\right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{para} := ((q, m, n, \sigma_k), (\ell, \mathcal{V}, \sigma_y))$, $\mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n\times m}, \mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}, \mathbf{f} \leftarrow D_{\mathcal{R}^n,\sigma_k}$, $\mathbf{u} \leftarrow_{\$} \mathcal{R}_q^n$, *and* $\mathcal{O}_{\text{HINT}}$ *on input* $(\mathbf{s}, \mathbf{e}) \in \mathcal{V}$ *outputs* $y + \mathbf{e}^\top\mathbf{k} + \mathbf{s}^\top\mathbf{f}$ *with* $y \leftarrow D_{\mathcal{R},\sigma_y}$.

Lemma 2 (MLWE ⇒ AHMLWE, adapted from [28, Theorem 1]). *Let $q, m, n, d, \ell \in \mathbb{N}$, $\epsilon, \sigma, \sigma_k, \sigma_y, B > 0$ and $\mathcal{V} \subseteq \mathcal{R}^{n+m}$ be parameters satisfying:*

- $\sum_{i\in[n+m]} \|v_i\|_1^2 \leq B$ *for any* $\mathbf{v} = (v_1, \ldots, v_{n+m}) \in \mathcal{V}$;
- $\sigma_y \geq \sqrt{4B} \cdot \eta_\epsilon(\mathbb{Z}^d)$, $\sigma \geq \sqrt{2} \cdot \eta_\epsilon(\mathbb{Z}^d)$,$\epsilon \in (0, \frac{1}{2})$;
- $\frac{1}{\sigma^2} = 2 \cdot (\frac{1}{\sigma_k^2} + \frac{\kappa^2\ell}{\sigma_y^2})$ *for* $\kappa := \sqrt{\frac{1-\sqrt{1-4B\eta_\epsilon^2(\mathbb{Z}^d)/\sigma_y^2}}{2\eta_\epsilon^2(\mathbb{Z}^d)/\sigma_y^2}}$.

Then, for any PPT adversary $\mathcal{A}$ against the $\mathsf{AHMLWE}^{\ell,\sigma_y,\mathcal{V}}_{q,m,n,\sigma_k}$ assumption, there exists a PPT $\mathcal{B}$ against the $\mathsf{MLWE}_{q,n,m,\sigma}$ assumption, s.t. $\mathsf{Adv}^{\mathsf{AHMLWE}}_{\mathsf{para},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{MLWE}}_{q,n,m,\sigma,\mathcal{B}}(\lambda) + (4\ell + 2(n+m)) \cdot \epsilon$, where $\mathsf{para} = ((q, m, n, \sigma_k), (\ell, \sigma_y, \mathcal{V}))$.

Lemma 3 (Tail Bound [40]). *Let $\mathbf{x} \leftarrow D_{\mathbb{Z}^m,\sigma}$. For any $k > 1/\sqrt{2\pi}$, it holds $\Pr[\|\mathbf{x}\| > k\sigma\sqrt{m}] < (\sqrt{2\pi}ke^{(1-2\pi k^2)/2})^m$ and $\Pr[\|\mathbf{x}\|_\infty > k\sigma] < \sqrt{2\pi}mke^{(1-2\pi k^2)/2}$.*

Lemma 4 (Cauchy-Schwarz Bound for Ring Inner Products, adapted from [33]). *For any $\mathbf{u}, \mathbf{v} \in \mathcal{R}^m$, $\|\mathbf{u}^\top\mathbf{v}\|_\infty \leq \|\mathbf{u}\|_2 \cdot \|\mathbf{v}\|_2$.*

Lemma 5 (Convolution Lemma, adapted from [25, Corollary 4.8]). *Let $c \in \mathbb{N}$ and $\sigma > \sqrt{2}\eta_\epsilon(\mathcal{R}^m)$. Then $\left\{\sum_{i=1}^c \mathbf{z}_i \mid \mathbf{z}_i \leftarrow D_{\mathcal{R}^m,\sigma}\right\} \approx_s \left\{\mathbf{z} \mid \mathbf{z} \leftarrow D_{\mathcal{R}^m,\sqrt{c}\cdot\sigma}\right\}$ with statistical distance at most $4c \cdot \epsilon$.*

Lemma 6. *For uniformly sampled $\mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n\times m}$ (w.l.o.g., assume that $m \geq n$), $\mathbf{A}$ has full rank with probability at least $1 - \frac{d}{q^{m-n}(q-1)}$.*

Due to space limitations, the formal proof of Lemma 6 is postponed to Appendix C.1.

Lemma 7 ([19, Lemma 7]). *Let $m \geq n$, $\epsilon \in (0, 1)$, and $\mathbf{D} \in \mathbb{Z}^{md\times nd}$ be an arbitrary matrix. Let $\sigma_k, \sigma_{k_0} > 0$ satisfy $\sigma_{k_0} \geq \eta_\epsilon(\mathcal{R}^n)$ and $\sigma_k > \sqrt{2}s_1(\mathbf{D}) \cdot \sigma_{k_0}$. Let $\mathbf{k}^{(0)} \leftarrow D_{\mathbb{Z}^{nd},\sigma_{k_0}}$ and $\mathbf{k}^{(1)} \leftarrow D_{\mathbb{Z}^{md},\sqrt{\mathbf{\Sigma}}}$ with $\mathbf{\Sigma} := \sigma_k^2\mathbf{I}_{md} - \sigma_{k_0}^2\mathbf{D}\mathbf{D}^\top$. Then the distribution $D_{\mathbf{k}}$ of $\mathbf{k} = \mathbf{D}\mathbf{k}^{(0)} + \mathbf{k}^{(1)}$ satisfies $\Delta(D_{\mathbf{k}}, D_{\mathbb{Z}^{md},\sigma_k}) \leq \frac{2\epsilon}{1-\epsilon}$.*

2.2 Secret Sharing

Replicated Secret Sharing. The replicated secret sharing (RSS) [29] is an additive t-out-of-N threshold secret sharing scheme with *short* secret share, as recalled in $\mathsf{RSS} = (\mathsf{Share}, \mathsf{Select}, \mathsf{Recover})$ in Fig. 1.

More precisely, let $\mathcal{I} := \{I \mid I \subseteq [N], |I| = N-t+1\}$ be the collection of sets $I \subseteq [N]$ of size $N-t+1$ and χ be a distribution over $\mathcal{R}_q^m$ (e.g., $\chi = D_{\mathcal{R}_q^m,\sigma}$). In this construction, the secret $\mathbf{k}$ is the summation of $|\mathcal{I}| = \binom{N}{N-t+1}$ individual keys $\{\mathbf{k}_I \leftarrow\!\!\$\ \chi\}_{I\in\mathcal{I}}$, i.e., $\mathbf{k} = \sum_{I\in\mathcal{I}} \mathbf{k}_I$, and each share sk_i ($i \in [N]$) consists of all individual keys $\mathbf{k}_I$ satisfying $i \in I$ (cf. Share in Fig. 1). Given a set $S \subseteq [N]$ of shares $\{\mathsf{sk}_i\}_{i\in S}$ with $|S| \geq t$, for all $I \in \mathcal{I}$, it must hold that $I \cap S \neq \emptyset$ (since $|I| = N-t+1$ and $|S| \geq t$), so sk_i for any $i \in I \cap S \subseteq S$ contains $\mathbf{k}_I$, and by combining all $\{\mathbf{k}_I\}_{I\in\mathcal{I}}$, one can recover the secret $\mathbf{k}$ (cf. Select and Recover in Fig. 1). Consequently, the RSS scheme satisfies correctness, i.e.,

- ***Correctness.*** For any $S \subseteq [N]$ with $|S| \geq t$, $(\mathbf{k}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N)) \leftarrow \mathsf{Share}(N, t, \chi)$, it holds that $\mathsf{Recover}(S, \{\mathsf{sk}_i\}_{i\in S}) = \mathbf{k}$.

On the other hand, given $\{\mathsf{sk}_i\}_{i\in S}$ for any set S with $|S| < t$, there must be (at least) one $I \in \mathcal{I}$ so that $I \cap S = \emptyset$, so all sk_i for $i \in S$ do not contain this specific individual key $\mathbf{k}_I$, and it fails to recover the whole secret $\mathbf{k}$.

$\underline{\mathsf{Share}(N, t, \chi) \to (\mathbf{k}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N))}$:
$/\!/\chi$ is a distribution over $\mathcal{R}_q^m$
$(\mathsf{sk}_1, \ldots, \mathsf{sk}_N) := (\emptyset, \ldots, \emptyset)$
$\mathcal{I} := \{I \mid I \subseteq [N], |I| = N-t+1\}$
$\mathbf{k} := \mathbf{0}$
For $I \in \mathcal{I}$:
  $\mathbf{k}_I \leftarrow \chi$, $\mathbf{k} := \mathbf{k} + \mathbf{k}_I$
  For each $i \in I$: $\mathsf{sk}_i \leftarrow \mathsf{sk}_i \cup \{\mathbf{k}_I\}$
Return $(\mathbf{k}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N))$

$\underline{\mathsf{Select}(S) \to \mathbf{w}}$: $/\!/S \subseteq [N]$
$\mathcal{I} := \{I \mid I \subseteq [N], |I| = N-t+1\}$
$\mathbf{w} := (w_i)_{i\in S} = (\emptyset, \ldots, \emptyset)$
For $I \in \mathcal{I}$: //choose a party for share $\mathbf{k}_I$
  $i' := \max(I \cap S)$, $w_{i'} := w_{i'} \cup \{I\}$
Return $\mathbf{w}$

$\underline{\mathsf{Recover}(S, \{\mathsf{sk}_i\}_{i\in S}) \to \mathbf{k}}$
If $|S| < t$: Return $\perp$
$(w_i)_{i\in S} \leftarrow \mathsf{Select}(S)$
Return $\mathbf{k} := \sum_{i\in S} \sum_{I\in w_i} \mathbf{k}_I$

Fig. 1. The replicated secret sharing scheme $\mathsf{RSS} = (\mathsf{Share}, \mathsf{Select}, \mathsf{Recover})$ in [29].

Shamir Secret Sharing. The Shamir secret sharing [51] is a linear t-out-of-N threshold secret sharing scheme. Let $q > N$ be prime and $S \subseteq [N]$ be a subset with $|S| \geq t$. For any $i \in S$, the Lagrange coefficient $\lambda_{i,S}$ is defined by $\lambda_{i,S} := \prod_{j\in S\setminus\{i\}} \frac{-j}{i-j}$. Let $k \in \mathcal{R}_q$ be a secret, and let $f \in \mathcal{R}_q[Y]$ be sampled uniformly at random from the set of polynomials of degree at most $t-1$ satisfying $f(0) = k$. Given any $S \subseteq [N]$ with $|S| \geq t$, the secret can be reconstructed via $k = \sum_{i\in S} \lambda_{i,S} \cdot f(i)$ from the corresponding shares $\{f(i)\}_{i\in S}$.

The Shamir secret sharing of $k \in \mathcal{R}_q$ can be easily extended to $\mathbf{k} \in \mathcal{R}_q^m$ by applying the technique component-wisely to $\mathbf{k}$. For simplicity, we denote the Shamir secret sharing algorithm by $\{\mathbf{k}_i\}_{i\in[N]} \leftarrow\!\!\$\ \mathsf{SShare}(N, t, \mathbf{k})$.

2.3 Definition of Threshold Decryption

We present the definition of non-interactive public key threshold decryption (TD) scheme in the asynchronous setting and synchronous setting.

Definition 5 (Threshold Decryption). *Let $t, N \in \mathbb{N}$ with $t < N$. A (t, N)-threshold decryption scheme* TD *for message space $\mathcal{M}$ is defined by a tuple of PPT algorithms* $\mathsf{TD} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{PDec}, \mathsf{Comb})$. *The following syntax is mainly for the asynchronous setting, and the differences in the synchronous setting are highlighted with gray boxes.*

- $\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}$*: The setup algorithm outputs a public parameter* pp*, which serves as an implicit input of other algorithms.*
- $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[N]}) \leftarrow\!\!\$\ \mathsf{KGen}(\mathsf{pp})$*: The key generation algorithm outputs a public key* pk *and a set of secret key shares* $\{\mathsf{sk}_i\}_{i\in[N]}$*, where each party $i \in [N]$ receives* sk_i*.*
- $\mathsf{ct} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, \mu)$*: The encryption algorithm takes as input a public key* pk *and a message $\mu \in \mathcal{M}$, and outputs a ciphertext* ct*.*
- $h_i \leftarrow\!\!\$\ \mathsf{PDec}(\mathsf{sk}_i, \mathsf{ct})$*: The partial decryption algorithm takes as input a secret key share* sk_i *and a ciphertext* ct *, and outputs a partial decryption h_i.*
  In the synchronous setting, PDec *additionally takes as input a subset $S \subseteq [N]$ with $|S| \geq t$, i.e., $h_i \leftarrow\!\!\$\ \mathsf{PDec}(\mathsf{sk}_i, \mathsf{ct}, S)$*
- $\mu \leftarrow \mathsf{Comb}(\mathsf{pk}, \mathsf{ct}, S, \{h_i\}_{i\in S})$*: The combination algorithm takes as input a public key* pk*, a ciphertext* ct*, a subset $S \subseteq [N]$ with $|S| \geq t$, and a set of partial decryption $\{h_i\}_{i\in S}$, and outputs a message μ.*

Correctness. *For $\forall \mu \in \mathcal{M}, \forall S \subseteq [N]$ with $|S| \geq t$, let* $\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup}$, $(\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[N]}) \leftarrow\!\!\$\ \mathsf{KGen}(\mathsf{pp})$, $\mathsf{ct} \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, \mu)$ *and* $\{h_i \leftarrow\!\!\$\ \mathsf{PDec}(\mathsf{sk}_i, \mathsf{ct}, S)\}_{i\in S}$*, it holds that* $\mathsf{Comb}(\mathsf{pk}, \mathsf{ct}, S, \{h_i\}_{i\in S}) = \mu$ *with probability* $1 - \mathsf{negl}(\lambda)$.

We write aTD for asynchronous TD and sTD for synchronous TD.

Next we define the IND-CCA/IND-smCPA security (indistinguishability against chosen-ciphertext/semi-malicious chosen-plaintext attacks under adaptive corruptions). Concretely, below we formalize IND-CCA security for both aTD and sTD, and formalize IND-smCPA for sTD.

Definition 6 (IND-CCA/IND-smCPA security). *A TD scheme* $\mathsf{TD} \in \{\mathsf{aTD}, \mathsf{sTD}\}$ *is* IND-CCA *secure, if for any PPT $\mathcal{A}$, it holds* $\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{TD},\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{CCA}}_{\mathsf{TD},\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$*, where* $\mathsf{Exp}^{\mathsf{CCA}}_{\mathsf{aTD},\mathcal{A}}$ *for* aTD *is defined in Fig. 2 with solid boxes, and* $\mathsf{Exp}^{\mathsf{CCA}}_{\mathsf{sTD},\mathcal{A}}$ *for* sTD *is defined in Fig. 2 with gray boxes.*

A synchronous TD scheme sTD *is* IND-smCPA *secure, if for any PPT $\mathcal{A}$, it holds* $\mathsf{Adv}^{\mathsf{smCPA}}_{\mathsf{sTD},\mathcal{A}}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{smCPA}}_{\mathsf{sTD},\mathcal{A}} \Rightarrow 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{smCPA}}_{\mathsf{sTD},\mathcal{A}}$ *for* sTD *is defined in Fig. 2 with both gray boxes and dashed boxes.*

For IND-CCA security, we adopt the strongest adaptive security model from [10, Fig. 1] and further strengthen it by assuming no secure erasure. During the experiment, $\mathcal{A}$ is asked to distinguish encryptions of μ_0 and μ_1, and can adaptively query $\mathcal{O}_{\mathrm{CoR}}$ and $\mathcal{O}_{\mathrm{DEC}}$ oracles. Through $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}, S)$, $\mathcal{A}$ can obtain the partial decryption h_i w.r.t. a ciphertext ct chosen by itself. Through $\mathcal{O}_{\mathrm{CoR}}(i)$, $\mathcal{A}$

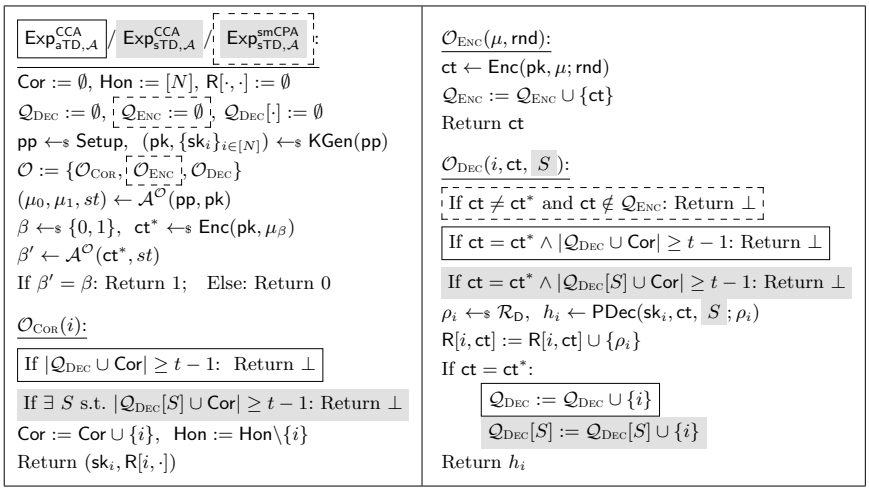

Fig. 2. IND-CCA security experiments $\mathsf{Exp}^{\mathsf{CCA}}_{\mathsf{aTD},\mathcal{A}}$ (with solid boxes) for asynchronous aTD and $\mathsf{Exp}^{\mathsf{CCA}}_{\mathsf{sTD},\mathcal{A}}$ (with gray boxes) for synchronous sTD, and IND-smCPA experiment $\mathsf{Exp}^{\mathsf{smCPA}}_{\mathsf{sTD},\mathcal{A}}$ (with both gray boxes and dashed boxes) for synchronous sTD.

can adaptively obtain user i's the secret key sk_i, together with all ephemeral randomness previously used by user i when answering $\mathcal{O}_{\mathrm{DEC}}$ queries. In particular, our model does not ask randomness erasure, which is a more reasonable.

For IND-smCPA security, we adapt the definition in [10] to the synchronous setting. During the experiment, $\mathcal{A}$ can query $\mathcal{O}_{\mathrm{DEC}}$ with only honestly generated ciphertexts from $\mathcal{O}_{\mathrm{ENC}}$ or the challenge ciphertext ct^*, but $\mathcal{A}$ can choose the randomness rnd that used in encryption. As shown in [10], smCPA security is an intermediate security level between CPA and CCA, and a smCPA secure TD can be upgraded to CCA security by appropriate NIZKs.

To avoid trivial attacks, the total number of corruptions and partial decryptions w.r.t. the challenge ciphertext ct^* queried by $\mathcal{A}$ should be less than t.

Definition 7 (Robustness). TD *has robustness, if for any PPT* $\mathcal{A}$*, it holds* $\mathsf{Adv}^{\mathsf{RB}}_{\mathsf{TD},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{RB}}_{\mathsf{TD},\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*, where* $\mathsf{Exp}^{\mathsf{RB}}_{\mathsf{TD},\mathcal{A}}$ *is defined in Fig. 3.*

For robustness, we adopt the notion of consistency from [18] instead of their definition of robustness, since their robustness is tailored for deterministic decryption schemes and does not apply to lattice-based constructions. During the experiment, $\mathcal{A}$ is given all public keys and secret key shares, and is challenged to generate two partial decryption sets S, S' for the same ciphertext ct that yield inconsistent messages when combined. We note that our robustness is stronger than that in [14], since in our security model, $\mathcal{A}$ can obtain all the secret keys $\{\mathsf{sk}_i\}_{i\in[N]}$ and can designate both decryption sets S, S'.

$\underline{\mathsf{Exp}^{\mathsf{RB}}_{\mathsf{TD},\mathcal{A}}}$:
$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Setup},\ \ (\mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[N]}) \leftarrow\!\!\$\ \mathsf{KGen}(\mathsf{pp})$
$(\mathsf{ct}, S, \{h_i\}_{i\in S}, S', \{h'_i\}_{i\in S'}) \leftarrow \mathcal{A}(\mathsf{pp}, \mathsf{pk}, \{\mathsf{sk}_i\}_{i\in[N]})$
$\mu \leftarrow \mathsf{Comb}(\mathsf{pk}, \mathsf{ct}, S, \{h_i\}_{i\in S}),\quad \mu' \leftarrow \mathsf{Comb}(\mathsf{pk}, \mathsf{ct}, S', \{h'_i\}_{i\in S'})$
If $\mu \neq \mu' \wedge \mu \neq \perp \wedge \mu' \neq \perp$: Return 1; Else: Return 0

Fig. 3. The robustness experiment $\mathsf{Exp}^{\mathsf{RB}}_{\mathsf{TD},\mathcal{A}}$ for TD.

3 Adaptive LHL with Noisy Linear Leakage

The canonical leftover hash lemma (LHL) [20] shows that, if a source $\mathbf{k}$ has enough minimal entropy given some leakage Y, then a uniformly random seed $\mathbf{u} \leftarrow\!\!\$\ \mathbb{Z}_q^m$ is able to extract $\mathbf{u}^\top\mathbf{k}$ from $\mathbf{k}$ s.t.

$$(\mathbf{u}, \mathbf{u}^\top\mathbf{k}, Y) \approx_s (\mathbf{u}, v \leftarrow\!\!\$\ \mathbb{Z}_q, Y).$$

Note that the leakage Y must be independent of $\mathbf{u}$ so that LHL can apply.

In some applications, the leakage Y may contain $\mathbf{Ek}$ for a matrix $\mathbf{E}$ related to seed $\mathbf{u}$, then LHL does not apply anymore, due to the dependence of Y and $\mathbf{u}$. To solve this problem, we require $\mathbf{E}$ to be of small norm, and resort to polynomial noise flooding technique to hide $\mathbf{Ek}$ in Y with a noise vector $\mathbf{y}$, so as to obtain $\mathbf{h} = \mathbf{Ek}+\mathbf{y}$. The polynomial noise flooding $\mathbf{h} = \mathbf{Ek}+\mathbf{y}$ helps to limit the leakage of $\mathbf{k}$ and relax the dependence between $\mathbf{E}$ and $\mathbf{u}$.

More precisely, we develop a refined LHL under *adaptive* noisy linear leakage $\mathbf{h} = \mathbf{Ek}+\mathbf{y}$, as shown in the following Lemma 8, where (each row of) the matrix $\mathbf{E}$ can be adaptively chosen by the adversary/distinguisher after seeing $\mathbf{u}$.[8] With a careful parameter choice and delicate analysis, our refined LHL shows that a polynomial noise $\mathbf{y}$ ensures

$$(\mathbf{u}, \mathbf{u}^\top\mathbf{k}, Y = (\mathbf{E}, \mathbf{Ek}+\mathbf{y})) \approx_s (\mathbf{u}, v \leftarrow\!\!\$\ \mathbb{Z}_q, Y = (\mathbf{E}, \mathbf{Ek}+\mathbf{y})),$$

where $\mathbf{E}$ may arbitrarily depend on $\mathbf{u}$.

The proof of our adaptive LHL goes with three main steps. We first address the challenge that $\mathbf{E}$ may *adaptively* depend on $\mathbf{u}$ via a delicate statistical analysis. Then we analyze the conditional distribution of $\mathbf{k}$ given the noisy linear leakage $\mathbf{h} = \mathbf{Ek}+\mathbf{y}$. Finally, for the conditional distribution of $\mathbf{k}$, we show that the extraction of $\mathbf{u}^\top\mathbf{k}$ is indeed uniform. We recall the useful lemmas needed in the proof of Lemma 8 in Appendix B.

Lemma 8 (Adaptive LHL with Noisy Linear Leakage). *Let $q, m, \ell, d \in \mathbb{N}$ with $m \geq 2$, $\epsilon \in (0, \frac{1}{2})$, and $\sigma_k, \sigma_y, B, \sigma > 0$. Let $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}$. Let $\mathcal{A}$ be an (unbounded) interactive algorithm that has access to a leakage oracle $\mathcal{O}_{\mathbf{k}}$: on each query $\mathbf{e} \in \mathcal{R}^m$ with $s_1(\mathsf{Rot}(\mathbf{e})) \leq B$, $\mathcal{O}_{\mathbf{k}}$ samples $y \leftarrow D_{\mathcal{R},\sigma_y}$ and outputs $h = \mathbf{e}^\top\mathbf{k} + y$ to $\mathcal{A}$. Consider the following two experiments:*

$$\mathsf{Real}:\ \ \mathbf{u} \leftarrow\!\!\$\ \mathcal{R}_q^m,\ \ v_0 := \mathbf{u}^\top\mathbf{k} \bmod q,\ \ \mathsf{Tr} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathbf{k}}}(\mathbf{u}, v_0),$$
$$\mathsf{Ideal}:\ \ \mathbf{u} \leftarrow\!\!\$\ \mathcal{R}_q^m,\ \ v_1 \leftarrow\!\!\$\ \mathcal{R}_q,\ \ \mathsf{Tr} \leftarrow \mathcal{A}^{\mathcal{O}_{\mathbf{k}}}(\mathbf{u}, v_1),$$

[8] Related smoothing-based LHLs for noisy linear leakage were studied in [5, Theorem 8], but their $\mathbf{E}$ is a fixed matrix and independent of the extractor seed $\mathbf{u}$.

where Tr *records all queries to and responses from* $\mathcal{O}_{\mathbf{k}}$*, i.e.,* $\mathsf{Tr} = \{(\mathbf{e}_i, h_i := \mathbf{e}_i^\top \mathbf{k} + y_i)\}_{i\in[\ell]}$ *with each* $y_i \leftarrow D_{\mathcal{R},\sigma_y}$ *for* ℓ *the number of queries made by* $\mathcal{A}$*. For any (all powerful)* $\mathcal{A}$ *that makes* ℓ *queries to* $\mathcal{O}_{\mathbf{k}}$*, if*

$$\tfrac{1}{\sigma_k^2} + \tfrac{\ell B^2}{\sigma_y^2} \le \tfrac{1}{\sigma^2} \text{ and } \sigma \ge \tfrac{8}{\sqrt{\pi}} \cdot q^{\frac{1}{m}} \sqrt{d \ln(2md(1+1/\epsilon))},$$

then it holds that $\Delta\big(((\mathbf{u}, v_0), \mathsf{Tr})_{\mathsf{Real}}, ((\mathbf{u}, v_1), \mathsf{Tr})_{\mathsf{Ideal}}\big) \le 2\epsilon + 2^{-d} + dq^{-m}$.

In particular, for any $\hat{v} \in \mathcal{R}_q$ *and any* $\{(\hat{\mathbf{e}}_i, \hat{h}_i) \in \mathcal{R}^m \times \mathcal{R}\}_{i\in[\ell]}$ *with each* $s_1(\mathsf{Rot}(\hat{\mathbf{e}}_i)) \le B$*, denote* $\hat{\mathbf{E}} = [\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_\ell]^\top$ *and* $\hat{\mathbf{h}} = (\hat{h}_1, \ldots, \hat{h}_\ell)^\top$*, then except with a fraction* $2^{-d} + dq^{-m}$ *of all* $\hat{\mathbf{u}} \in \mathcal{R}_q^m$*, it holds that*

$$\Pr[\hat{\mathbf{u}}^\top \mathbf{k} = \hat{v} \bmod q \mid \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}] \in (1 \pm 4\epsilon) \cdot q^{-d},$$

where the probability is over $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}$ *and* $\mathbf{y} \leftarrow D_{\mathcal{R}^\ell,\sigma_y}$*.*

Proof of Lemma 8. Throughout the proof, for random variables (e.g., $\mathbf{u} \leftarrow\!\!\$\ \mathcal{R}_q^m$, $v_0 := \mathbf{u}^\top \mathbf{k} \bmod q$, $v_1 \leftarrow\!\!\$\ \mathcal{R}_q$, $\mathcal{A}$'s query $\mathbf{e}_i$ and the oracle's response $h_i := \mathbf{e}_i^\top \mathbf{k} + y_i$), we use $\hat{\cdot}$ (*hatted symbols*) to denote *constants* in the corresponding ranges (e.g., $\hat{\mathbf{u}} \in \mathcal{R}_q^m$, $\hat{v} \in \mathcal{R}_q$, $\hat{\mathbf{e}}_i \in \mathcal{R}^m$, $\hat{h}_i \in \mathcal{R}$). Moreover, we use $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_\ell]^\top$ and $\mathbf{h} = (h_1, \ldots, h_\ell)^\top$ to denote $\mathcal{A}$'s queries and the oracle's responses succinctly, and use $\hat{\mathbf{E}} = [\hat{\mathbf{e}}_1, \ldots, \hat{\mathbf{e}}_\ell]^\top$ and $\hat{\mathbf{h}} = (\hat{h}_1, \ldots, \hat{h}_\ell)^\top$ to denote the corresponding constants. Then we have

$$\begin{aligned} &\Delta\big(((\mathbf{u}, v_0), \mathsf{Tr})_{\mathsf{Real}}, ((\mathbf{u}, v_1), \mathsf{Tr})_{\mathsf{Ideal}}\big) \qquad (9)\\ &= \tfrac{1}{2} \cdot \textstyle\sum_{\hat{\mathbf{u}},\hat{v},\hat{\mathbf{E}},\hat{\mathbf{h}}} \big|\Pr[(\mathbf{u}, v_0, \mathbf{E}, \mathbf{h}) = (\hat{\mathbf{u}}, \hat{v}, \hat{\mathbf{E}}, \hat{\mathbf{h}})] - \Pr[(\mathbf{u}, v_1, \mathbf{E}, \mathbf{h}) = (\hat{\mathbf{u}}, \hat{v}, \hat{\mathbf{E}}, \hat{\mathbf{h}})]\big|. \end{aligned}$$

Firstly, we analyze $\Pr[(\mathbf{u}, v_b, \mathbf{E}, \mathbf{h}) = (\hat{\mathbf{u}}, \hat{v}, \hat{\mathbf{E}}, \hat{\mathbf{h}})]$ for any fixed constants $\hat{\mathbf{u}}, \hat{v}, \hat{\mathbf{E}}, \hat{\mathbf{h}}$ and for any $b \in \{0, 1\}$:

$$\begin{aligned} &\Pr[(\mathbf{u}, v_b, \mathbf{E}, \mathbf{h}) = (\hat{\mathbf{u}}, \hat{v}, \hat{\mathbf{E}}, \hat{\mathbf{h}})] = \Pr[\mathbf{u} = \hat{\mathbf{u}}] \cdot \Pr[v_b = \hat{v} | \mathbf{u} = \hat{\mathbf{u}}] \cdot \\ &\prod_{i=1}^{\ell} \Big(\underbrace{\Pr[\mathbf{e}_i = \hat{\mathbf{e}}_i \mid \mathbf{u} = \hat{\mathbf{u}}, v_b = \hat{v}, (\mathbf{e}_j = \hat{\mathbf{e}}_j, h_j = \hat{h}_j)_{j\in[i-1]}]}_{:=\alpha_i} \\ &\quad \cdot \underbrace{\Pr[h_i = \hat{h}_i \mid \mathbf{u} = \hat{\mathbf{u}}, v_b = \hat{v}, (\mathbf{e}_j = \hat{\mathbf{e}}_j, h_j = \hat{h}_j)_{j\in[i-1]}, \mathbf{e}_i = \hat{\mathbf{e}}_i]}_{:=\beta_i} \Big). \qquad (10) \end{aligned}$$

We make two observations:

- The term α_i in (10) is the probability that $\mathcal{A}$ queries $\hat{\mathbf{e}}_i$ in its i-th query, given that $\mathcal{A}$ obtains $(\hat{\mathbf{u}}, \hat{v})$, queries $\hat{\mathbf{e}}_j$ and gets $\hat{h}_j$ in its first $i-1$ queries.
  In particular, for each $i \in [\ell]$, this term α_i is the same for $b = 0$ and $b = 1$ since it is conditioned on the same constant $\hat{v}$.
- The term β_i in (10) is the probability that the oracle outputs $h_i := \hat{\mathbf{e}}_i^\top \mathbf{k} + y_i = \hat{h}_i$ for $\mathcal{A}$'s i-th query, given that $\mathcal{A}$ obtains $(\hat{\mathbf{u}}, \hat{v})$, queries $\hat{\mathbf{e}}_j$ and obtains $\hat{h}_j$ in its first $j \le i-1$ queries, and queries $\hat{\mathbf{e}}_i$ in its i-th query.

In particular, the probability in β_i is taken over the randomness of $\mathbf{k}$ and $y_i \leftarrow D_{\mathcal{R},\sigma_y}$, and we have

$$\begin{aligned}\beta_i &= \Pr\left[h_i := \mathbf{e}_i^\top \mathbf{k} + y_i = \hat{h}_i \;\middle|\; \begin{array}{c}\mathbf{u}=\hat{\mathbf{u}}, v_b = \hat{v},\\ (\mathbf{e}_j = \hat{\mathbf{e}}_j, h_j := \mathbf{e}_j^\top \mathbf{k} + y_j = \hat{h}_j)_{j\in[i-1]}, \mathbf{e}_i = \hat{\mathbf{e}}_i\end{array}\right] \\ &= \Pr\left[\hat{\mathbf{e}}_i^\top \mathbf{k} + y_i = \hat{h}_i \;\middle|\; \mathbf{u}=\hat{\mathbf{u}}, v_b = \hat{v}, (\hat{\mathbf{e}}_j^\top \mathbf{k} + y_j = \hat{h}_j)_{j\in[i-1]}\right].\end{aligned}$$

So the product $\prod_{i=1}^{\ell} \beta_i$ in (10) becomes

$$\begin{aligned}\prod_{i=1}^{\ell} \beta_i &= \prod_{i=1}^{\ell} \Pr\left[\hat{\mathbf{e}}_i^\top \mathbf{k} + y_i = \hat{h}_i \;\middle|\; \mathbf{u}=\hat{\mathbf{u}}, v_b = \hat{v}, (\hat{\mathbf{e}}_j^\top \mathbf{k} + y_j = \hat{h}_j)_{j\in[i-1]}\right] \\ &= \Pr\left[(\hat{\mathbf{e}}_i^\top \mathbf{k} + y_i = \hat{h}_i)_{i\in[\ell]} \;\middle|\; \mathbf{u}=\hat{\mathbf{u}}, v_b = \hat{v}\right] \\ &= \Pr[\hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}} \mid \mathbf{u}=\hat{\mathbf{u}}, v_b = \hat{v}],\end{aligned}$$

where $\mathbf{y} = (y_1, \ldots, y_\ell)^\top \leftarrow D_{\mathcal{R}^\ell,\sigma_y}$.

Taking the above two observations back to (10), we get

$$\begin{aligned}&\Pr[(\mathbf{u}, v_b, \mathbf{E}, \mathbf{h}) = (\hat{\mathbf{u}}, \hat{v}, \hat{\mathbf{E}}, \hat{\mathbf{h}})] \\ =\ &\Pr[\mathbf{u}=\hat{\mathbf{u}}] \cdot \Pr[v_b = \hat{v} | \mathbf{u}=\hat{\mathbf{u}}] \cdot \textstyle\prod_{i=1}^{\ell} \alpha_i \cdot \Pr[\hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}} \mid \mathbf{u}=\hat{\mathbf{u}}, v_b = \hat{v}] \\ =\ &\Pr[\mathbf{u}=\hat{\mathbf{u}}] \cdot \textstyle\prod_{i=1}^{\ell} \alpha_i \cdot \underbrace{\Pr[v_b = \hat{v}, \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}} \mid \mathbf{u}=\hat{\mathbf{u}}]}_{:=\gamma_b}, \qquad (11)\end{aligned}$$

where the product $\prod_{i=1}^{\ell} \alpha_i$ is the same for $b=0$ and $b=1$.

Next, we analyze the term $\gamma_b := \Pr[v_b = \hat{v}, \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}} \mid \mathbf{u}=\hat{\mathbf{u}}]$ in (11). Note that here $\hat{\mathbf{E}}$, $\hat{\mathbf{h}}$ are fixed constants, and the probability is taken over $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}$ and $\mathbf{y} \leftarrow D_{\mathcal{R}^\ell,\sigma_y}$, so the analysis of this term would be purely static (recall that all randomness related to $\mathcal{A}$ in choosing its oracle queries have been absorbed into the product $\prod_{i=1}^{\ell} \alpha_i$, which is the same for $b=0$ and $b=1$).

- For $b=1$, the analysis of the term is easy, since $v_1 \leftarrow\!\!\$\ \mathcal{R}_q$ is independent of $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}$ and $\mathbf{y} \leftarrow D_{\mathcal{R}^\ell,\sigma_y}$:

$$\begin{aligned}\gamma_1 =\ &\Pr[v_1 = \hat{v}, \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}} \mid \mathbf{u}=\hat{\mathbf{u}}] \\ =\ &\Pr[v_1 = \hat{v}, \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}] = \Pr[\hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}] \cdot q^{-d}.\end{aligned}$$

- For $b=0$, since $v_0 := \mathbf{u}^\top \mathbf{k} \bmod q$, we have

$$\begin{aligned}\gamma_0 =\ &\Pr[\mathbf{u}^\top \mathbf{k} = \hat{v} \bmod q, \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}} \mid \mathbf{u}=\hat{\mathbf{u}}] \\ =\ &\Pr[\hat{\mathbf{u}}^\top \mathbf{k} = \hat{v} \bmod q, \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}] \\ =\ &\Pr[\hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}] \cdot \Pr[\hat{\mathbf{u}}^\top \mathbf{k} = \hat{v} \bmod q \mid \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}].\end{aligned}$$

Dividing the above two equations, we get that

$$\gamma_0/\gamma_1 = q^d \cdot \Pr[\hat{\mathbf{u}}^\top \mathbf{k} = \hat{v} \bmod q \mid \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}]. \qquad (12)$$

Below we analyze this division in more details by first analyzing the distribution of $\mathbf{k}$ conditioned on $\hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}$, and then analyzing the conditional probability.

– By applying Lemma 10 in Appendix B with the embedding $\phi : \mathcal{R} \to \mathbb{Z}^d$ (which extends to vectors over $\mathcal{R}$), for $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}$ and $\mathbf{y} \leftarrow D_{\mathcal{R}^\ell,\sigma_y}$, we get

$$(\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}, \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}) \equiv (\mathbf{k}' \leftarrow D_{\mathcal{R}^m,\sqrt{\mathbf{\Sigma}},\mathbf{c}}, \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}),$$

where $\mathbf{\Sigma} := (\mathbf{\Sigma}_k^{-1} + \hat{\mathbf{\Gamma}}^\top \mathbf{\Sigma}_y^{-1} \hat{\mathbf{\Gamma}})^{-1}$ for $\mathbf{\Sigma}_k := \sigma_k^2 \mathbf{I}_{md}$, $\mathbf{\Sigma}_y := \sigma_y^2 \mathbf{I}_{\ell d}$, $\hat{\mathbf{\Gamma}} := \mathsf{Rot}(\hat{\mathbf{E}})$, and $\mathbf{c} := \phi^{-1}(\mathbf{\Sigma}\hat{\mathbf{\Gamma}}^\top \mathbf{\Sigma}_y^{-1} \phi(\hat{\mathbf{h}}))$. Thus, we have $\mathbf{k} \mid_{\hat{\mathbf{E}}\mathbf{k}+\mathbf{y}=\hat{\mathbf{h}}} \sim D_{\mathcal{R}^m,\sqrt{\mathbf{\Sigma}},\mathbf{c}}$.
– Then we analyze $\lambda_{\min}(\mathbf{\Sigma})$ for the $\mathbf{\Sigma}$ defined above. Note that $\mathbf{I}_{md}$ and $\hat{\mathbf{\Gamma}}^\top \hat{\mathbf{\Gamma}}$ are symmetric, by Lemma 9 in Appendix B, we have

$$\lambda_{\max}(\mathbf{\Sigma}^{-1}) \le \tfrac{1}{\sigma_k^2} \cdot \lambda_{\max}(\mathbf{I}_{md}) + \tfrac{1}{\sigma_y^2} \cdot \lambda_{\max}\big(\hat{\mathbf{\Gamma}}^\top \hat{\mathbf{\Gamma}}\big) = \tfrac{1}{\sigma_k^2} + \tfrac{1}{\sigma_y^2} \cdot s_1^2(\hat{\mathbf{\Gamma}}).$$

Since $s_1^2(\hat{\mathbf{\Gamma}}) \le \ell \cdot \max_{i\in[\ell]} s_1(\mathsf{Rot}(\hat{\mathbf{e}}_i))^2 \le \ell B^2$, and by the condition that $\frac{1}{\sigma_k^2} + \frac{\ell B^2}{\sigma_y^2} \le \frac{1}{\sigma^2}$, it follows that $\lambda_{\max}(\mathbf{\Sigma}^{-1}) \le \frac{1}{\sigma_k^2} + \frac{\ell B^2}{\sigma_y^2} \le \frac{1}{\sigma^2}$ and

$$\lambda_{\min}(\mathbf{\Sigma}) = \tfrac{1}{\lambda_{\max}(\mathbf{\Sigma}^{-1})} \ge \sigma^2. \tag{13}$$

– Let $\mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp := \mathit{\Lambda}_q^\perp(\hat{\mathbf{u}}^\top) = \{\mathbf{x} \in \mathcal{R}^m \mid \hat{\mathbf{u}}^\top \mathbf{x} = 0 \bmod q\}$. Since $\mathbf{u} \leftarrow\!\!\$\ \mathcal{R}_q^m$ is uniformly random, by Lemma 11 in Appendix B and the condition $m \ge 2$, except with probability at most 2^{-d} over $\mathbf{u} = \hat{\mathbf{u}}$, we have $\eta_\epsilon(\mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp) \le \frac{8}{\sqrt{\pi}} \cdot q^{\frac{1}{m}} \sqrt{d \ln(2md(1+1/\epsilon))} \le \sigma$, according to the condition of the lemma.

  Moreover, since $\mathbf{u} \leftarrow\!\!\$\ \mathcal{R}_q^m$ is uniform, by Lemma 6, except with probability at most dq^{-m} over $\mathbf{u} = \hat{\mathbf{u}}$, $\hat{\mathbf{u}}$ is full rank. In this case, the cosets of $\mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp$ in $\mathcal{R}^m$ are indexed by $\mathbf{x} \mapsto \hat{\mathbf{u}}^\top \mathbf{x} \bmod q$ for $\mathbf{x} \in \mathcal{R}^m \bmod \mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp$, which yields an isomorphism $\mathcal{R}^m \bmod \mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp \simeq \mathcal{R}_q$.

  Let Good denote the event that $\eta_\epsilon(\mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp) \le \sigma$ holds and $\hat{\mathbf{u}}$ is full-rank. Below we take Good for granted.
– By the event Good and (13), we get that $\sqrt{\lambda_{\min}(\mathbf{\Sigma})} \ge \sigma \ge \eta_\epsilon(\mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp)$. So we can apply Lemma 12 in Appendix B to $\mathbf{k} \mid_{\hat{\mathbf{E}}\mathbf{k}+\mathbf{y}=\hat{\mathbf{h}}} \sim D_{\mathcal{R}^m,\sqrt{\mathbf{\Sigma}},\mathbf{c}}$ w.r.t. the sublattice $\mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp \subseteq \mathcal{R}^m$, and obtain that for any $\hat{\mathbf{k}} \in \mathcal{R}^m \bmod \mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp$,

$$\begin{array}{rcl}
\Pr[\mathbf{k} = \hat{\mathbf{k}} \bmod \mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp \mid \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}] & \in & (1 \pm 4\epsilon)/|\mathcal{R}^m/\mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp| \\
\| & & \| \\
\Pr[\hat{\mathbf{u}}^\top \mathbf{k} = \hat{\mathbf{u}}^\top \hat{\mathbf{k}} \bmod q \mid \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}] & & (1 \pm 4\epsilon)/|\mathcal{R}_q| \\
\| & & \| \\
\Pr[\hat{\mathbf{u}}^\top \mathbf{k} = \hat{v} \bmod q \mid \hat{\mathbf{E}}\mathbf{k} + \mathbf{y} = \hat{\mathbf{h}}] & & (1 \pm 4\epsilon) \cdot q^{-d},
\end{array} \tag{14}$$

where the first equality in the left column follows from the definition of $\mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp$, and the second equality in the left column is by the isomorphism $\mathcal{R}^m \bmod \mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp \simeq \mathcal{R}_q$, which suggests that when the constant $\hat{\mathbf{k}}$ ranges over all values in $\mathcal{R}^m \bmod \mathit{\Lambda}_{\hat{\mathbf{u}}}^\perp$, $\hat{\mathbf{u}}^\top \hat{\mathbf{k}} \bmod q$ ranges over all values $\hat{v}$ in $\mathcal{R}_q$, once for each value.

Consequently, for the division (12), we get that $\gamma_0/\gamma_1 \in [1 - 4\epsilon, 1 + 4\epsilon]$.

Finally, by plugging (11) to (9), together with the facts that the product $\prod_{i=1}^{\ell} \alpha_i$ is the same for $b = 0$ and $b = 1$ and $\gamma_0/\gamma_1 \in [1 - 4\epsilon, 1 + 4\epsilon]$, we have

$$\begin{aligned}
&\Delta\big(((\mathbf{u}, v_0), \mathsf{Tr})_{\mathsf{Real}}, ((\mathbf{u}, v_1), \mathsf{Tr})_{\mathsf{Ideal}}\big) \\
&= \tfrac{1}{2} \cdot \textstyle\sum_{\hat{\mathbf{u}},\hat{v},\hat{\mathbf{E}},\hat{\mathbf{h}}} \left| \Pr[\mathbf{u} = \hat{\mathbf{u}}] \cdot \prod_{i=1}^{\ell} \alpha_i \cdot \gamma_0 - \Pr[\mathbf{u} = \hat{\mathbf{u}}] \cdot \prod_{i=1}^{\ell} \alpha_i \cdot \gamma_1 \right| \\
&= \tfrac{1}{2} \cdot \textstyle\sum_{\hat{\mathbf{u}},\hat{v},\hat{\mathbf{E}},\hat{\mathbf{h}}} \underbrace{\Pr[\mathbf{u} = \hat{\mathbf{u}}] \cdot \prod_{i=1}^{\ell} \alpha_i \cdot \gamma_1}_{=\Pr[(\mathbf{u},v_1,\mathbf{E},\mathbf{h})=(\hat{\mathbf{u}},\hat{v},\hat{\mathbf{E}},\hat{\mathbf{h}})]} \cdot \underbrace{\left|\gamma_0/\gamma_1 - 1\right|}_{\leq 4\epsilon} \leq 2\epsilon.
\end{aligned}$$

The above analysis takes Good for granted. Taking the probability that Good does not happen into account, it affects the statistical distance by at most $2^{-d} + dq^{-m}$. Therefore, (14) holds except with a fraction $2^{-d} + dq^{-m}$ of all $\hat{\mathbf{u}} \in \mathcal{R}_q^m$. This completes the proof of Lemma 8. □

4 Adaptively CCA-Secure Asynchronous TD Scheme

Below we present an adaptively CCA-secure *asynchronous* TD scheme TD_1 for *small user number* N in Subsect. 4.1, and an extension $\mathsf{TD}_1^{\mathsf{rob}}$ which additionally has robustness in Subsect. 4.2. These schemes have polynomial modulus and their securities are based on the MLWE assumption in the RO model.

4.1 TD_1: Adaptively CCA-Secure TD Scheme for Small N

Our asynchronous (t, N)-TD scheme TD_1 for small N uses the following parameters and building blocks. (In Table 2 in Appendix E.1, we present a suggested parameter setting which satisfy all conditions required by the correctness and security of TD_1 in this subsection, and provide an estimate efficiency of TD_1.)

- Let $p, q, n, m \in \mathbb{N}$ with $p < q$. Let $C := \binom{N}{t-1}$.
- Let $\sigma_k, \sigma_0, \sigma_1, \sigma_y > 0$ be Gaussian widths and $B_k, B_1, B_y, B', B'_k \in \mathbb{R}$, where B_k, B_1, B_y are ℓ_∞-norm bounds of $D_{\mathcal{R}^m,\sigma_k}, D_{\mathcal{R},\sigma_1}, D_{\mathcal{R},\sigma_y}$, and B', B'_k are the ℓ_2-norm bounds of $D_{\mathcal{R}^{(m+n)},\sigma_0}$, $D_{\mathcal{R}^m,\sigma_k}$, respectively.
- The RSS scheme $\mathsf{RSS} = (\mathsf{Share}, \mathsf{Select}, \mathsf{Recover})$ (cf. Fig. 1 in Subsect. 2.2).
- An OTSS tag-based QA-NIZK $\Pi = (\mathsf{CRS}, \mathsf{Prv}, \mathsf{Vfy}, \mathsf{Sim})$ (cf. Def. 10 in App. A.2) for tag space $\mathcal{T}$ and language $\mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$ with parameter $\mathbf{A} \in \mathcal{R}_q^{n \times m}$:

$$\mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})} := \left\{ \mathbf{c} \,\middle|\, \exists\, \mathbf{s} \in \mathcal{R}_q^n, \mathbf{e} \in \mathcal{R}_q^m \text{ s.t. } \mathbf{c}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top \wedge \|(\mathbf{s}, \mathbf{e})\|_2 \leq B' \right\}.$$

  We refer to Appendix D.2 for an instantiation of Π for the language $\mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$.
- A family of collision-resistant hash functions $\mathcal{H} = \{\mathsf{H}\colon \{0,1\}^* \to \mathcal{T}\}$.

Our scheme $\mathsf{TD}_1 = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{PDec}, \mathsf{Comb})$ for message space $\mathcal{M} = \mathcal{R}_p$ is shown in Fig. 4 (without gray boxes).

Correctness. The correctness of TD_1 follows from the completeness of Π, the correctness of RSS, together with the condition $B_1 + \sqrt{C}(B'B'_k + B_y) \leq \lfloor \frac{q}{p} \rfloor / 2$.

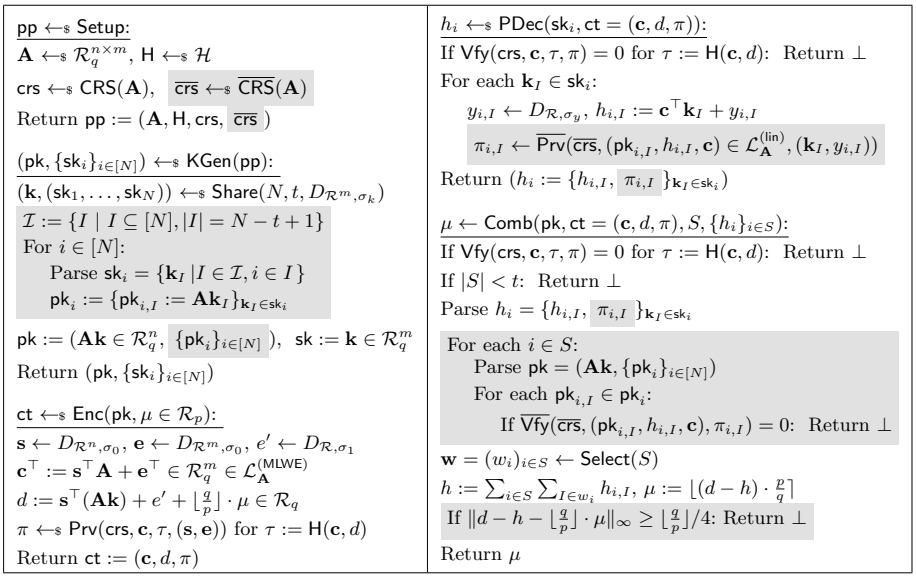

Fig. 4. Our adaptively CCA-secure asynchronous (t, N)-TD scheme TD_1 (without gray boxes) and its robust extension $\mathsf{TD}_1^{\mathsf{rob}}$ (with gray boxes) for small user number N. We explain TD_1 in Subsect. 4.1 and explain $\mathsf{TD}_1^{\mathsf{rob}}$ in Subsect. 4.2.

Firstly, the Vfy algorithms in PDec and Comb will output 1 except with negligible probability, by the completeness of Π.

Next, we show that the Comb algorithm can recover the message correctly except with negligible probability. For any message $\mu \in \mathcal{R}_p$, $\mathsf{Enc}(\mathsf{pk}, \mu)$ generates ciphertext $\mathsf{ct} = (\mathbf{c}, d, \pi)$ with $\mathbf{c}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top$ and $d = \mathbf{s}^\top(\mathbf{Ak}) + e' + \lfloor \frac{q}{p} \rfloor \cdot \mu$, and $\mathsf{PDec}(\mathsf{sk}_i, \mathsf{ct})$ generates partial decryption $h_i = \{h_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i}$ with each $h_{i,I} = \mathbf{c}^\top\mathbf{k}_I + y_{i,I}$. Then for any $S \subseteq [N]$ with $|S| \geq t$, in $\mathsf{Comb}(\mathsf{pk}, \mathsf{ct}, S, \{h_i\}_{i\in S})$, we have $h = \sum_{i\in S}\sum_{I\in w_i} h_{i,I} = \sum_{i\in S}\sum_{I\in w_i}(\mathbf{s}^\top\mathbf{Ak}_I + \mathbf{e}^\top\mathbf{k}_I + y_{i,I}) = \mathbf{s}^\top\mathbf{Ak} + \mathbf{e}^\top\mathbf{k} + \sum_{i\in S}\sum_{I\in w_i} y_{i,I}$, where the last equality follows from $\sum_{i\in S}\sum_{I\in w_i} k_{i,I} = \mathbf{k}$ by the correctness of RSS. Therefore, it holds that

$$d - h = \lfloor \tfrac{q}{p} \rfloor \cdot \mu + e' - \mathbf{e}^\top\mathbf{k} - \textstyle\sum_{i\in S}\sum_{I\in w_i} y_{i,I},$$

where $\mathbf{k}$ and $\sum_{i\in S}\sum_{I\in w_i} y_{i,I}$ are statistically close to $D_{\mathcal{R}^m,\sqrt{C}\cdot\sigma_k}$ and $D_{\mathcal{R},\sqrt{C}\cdot\sigma_y}$, respectively, by Lemma 5. By Cauchy-Schwarz bound (cf. Lemma 4), the noise term $e' - \mathbf{e}^\top\mathbf{k} - \sum_{i\in S}\sum_{I\in w_i} y_{i,I}$ is bounded by $B_1 + \sqrt{C}(B'B'_k + B_y) \leq \lfloor \frac{q}{p} \rfloor/2$ with overwhelming probability. Consequently, $\lfloor (d-h) \cdot \frac{p}{q} \rceil$ correctly recovers μ.

Security. The $\mathsf{IND\text{-}CCA}$ security of TD_1 is shown in the following theorem.

Theorem 1 ($\mathsf{IND\text{-}CCA}$ Security of TD_1). *Let $r, a \in \mathbb{N}$ such that $s := m - r - a \geq 2$, and let $\sigma_{k_0}, \sigma_{\mathsf{MLWE}} > 0$ and let $\epsilon, \epsilon' \in (0, \frac{1}{2})$. If*

- *Π has zero-knowledge and OTSS, and $\mathcal{H}$ is collision-resistant,*

– *the* $\mathsf{SA\text{-}HMLWE}^{1,\sigma_1,\mathcal{S}}_{q,m,n,\sigma_0}$*-assumption holds,*[9] *where* $\mathcal{S} := \{\mathbf{k} \in \mathcal{R}^m : s_1(\mathsf{Rot}(\mathbf{k})) \leq B_{\mathcal{S}}\}$ *and* $B_{\mathcal{S}} > 0$ *is the bound of* $s_1(\mathsf{Rot}(\mathbf{k}))$ *for* $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sqrt{C}\sigma_k}$.
– *the* s-$\mathsf{MLWE}_{q,r,a,\sigma_{\mathsf{MLWE}}}$ *assumption holds,*
– $r = n + O(\log\lambda)$, $\sigma_k \geq \max(\sqrt{2}\cdot B_{\mathbf{D}}\cdot\sigma_{k_0}, \sqrt{2}\cdot\eta_{\epsilon'}(\mathcal{R}^m))$, $\sigma_{k_0} \geq \eta_{\epsilon'}(\mathcal{R}^s)$ *and* $\frac{1}{\sigma_{k_0}^2} + \frac{\ell d B'^2 B_{\mathbf{D}}^2}{\sigma_y^2} \leq \frac{1}{\sigma^2}$ *for* $\sigma \geq \frac{8}{\sqrt{\pi}}\cdot q^{\frac{1}{s}}\sqrt{d\ln(2sd(1+1/\epsilon))}$. *Here* $B_{\mathbf{D}} > 0$ *is the bound of* $s_1(\mathsf{Rot}(\mathbf{D}))$ *for* $\mathbf{D}^\top := [-\mathbf{R}^\top|\mathbf{I}_s| - \mathbf{E}^\top] \in \mathcal{R}^{s\times m}$, *where* $(\mathbf{R},\mathbf{E}) \leftarrow D^{a\times s}_{\mathcal{R},\sigma_{\mathsf{MLWE}}} \times D^{r\times s}_{\mathcal{R},\sigma_{\mathsf{MLWE}}}$.

then the scheme TD_1 *in Fig. 4 is* IND-CCA *secure.*

In Appendix C.2, we also provide an explicit bound of $\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{TD}_1,\mathcal{A}}(\lambda)$.

Due to space limitations, we refer to the technical overview in Sect. 1 for an overview of the proof, and postpone the formal proof to Appendix C.2.

4.2 $\mathsf{TD}_1^{\mathsf{rob}}$: Robustness Extension

Now we extend our TD_1 in Subsect. 4.1 to $\mathsf{TD}_1^{\mathsf{rob}}$ (cf. Fig. 4 with gray boxes) which additionally achieves robustness, with the help of an extra NIZK $\overline{\Pi}$.

– $\overline{\Pi} = (\overline{\mathsf{CRS}}, \overline{\mathsf{Prv}}, \overline{\mathsf{Vfy}}, \overline{\mathsf{Sim}})$ (cf. Def. 9 in Appendix A.2) is a QA-NIZK scheme for gap language $\mathcal{GL}^{(\mathsf{lin})}_{\mathbf{A}} = (\mathcal{L}^{(\mathsf{lin})}_{\mathbf{A}}, \tilde{\mathcal{L}}^{(\mathsf{lin})}_{\mathbf{A}})$ with parameter $\mathbf{A} \in \mathcal{R}_q^{n\times m}$:

$$\mathcal{L}^{(\mathsf{lin})}_{\mathbf{A}} := \left\{(\mathsf{pk}, h, \mathbf{c}) \;\middle|\; \begin{array}{c}\exists(\mathbf{k}\in\mathcal{R}^m,\\ y\in\mathcal{R})\end{array} \text{ s.t. } \begin{array}{l}\mathsf{pk} = \mathbf{A}\mathbf{k} \;\wedge h = \mathbf{c}^\top\mathbf{k} + y\\ \wedge\|\mathbf{k}\|_\infty \leq B_k \wedge \|y\|_\infty \leq B_y\end{array}\right\},$$
$$\tilde{\mathcal{L}}^{(\mathsf{lin})}_{\mathbf{A}} := \left\{(\mathsf{pk}, h, \mathbf{c}) \;\middle|\; \begin{array}{c}\exists(\mathbf{k}\in\mathcal{R}^m,\\ y\in\mathcal{R})\end{array} \text{ s.t. } \begin{array}{l}\mathsf{pk} = \mathbf{A}\mathbf{k} \;\wedge h = \mathbf{c}^\top\mathbf{k} + y\\ \wedge\|\mathbf{k}\|_\infty \leq \tilde{B}_k \wedge \|y\|_\infty \leq \tilde{B}_y\end{array}\right\},$$

where $\tilde{B}_k, \tilde{B}_y > 0$. Note that the relations in $\mathcal{GL}^{(\mathsf{lin})}_{\mathbf{A}} = (\mathcal{L}^{(\mathsf{lin})}_{\mathbf{A}}, \tilde{\mathcal{L}}^{(\mathsf{lin})}_{\mathbf{A}})$ are linear, where both $\mathbf{k}$ and y are short. Hence, in the RO model, $\overline{\Pi}$ can be constructed by applying the Fiat-Shamir transform [24] to a rejection-sampling-based Σ-protocol [39, 40].

The correctness of $\mathsf{TD}_1^{\mathsf{rob}}$ is similar to that of TD_1, expect that it additionally requires the completeness of $\overline{\Pi}$ to ensure $\overline{\mathsf{Vfy}}$ in Comb output 1, and tightens the condition inequality as $B_1 + \sqrt{C}(B'B'_k + B_y) \leq \lfloor\frac{q}{p}\rfloor/4$.

The IND-CCA security of $\mathsf{TD}_1^{\mathsf{rob}}$ is also similar to that of TD_1, as the proof of Theorem 1 (cf. technical overview in Sect. 1 for an overview and Appendix C.2 for the formal proof) also applies to it. This follows from two observations:

– Publishing $\mathsf{pk}_i = \{\mathsf{pk}_{i,I} = \mathbf{A}\mathbf{k}_I\}_{\mathbf{k}_I\in\mathsf{sk}_i}$ in pk does not affect the proof, because our proof of Theorem 1 explicitly allows $\{\mathbf{A}\mathbf{k}_I\}_{I\in\mathcal{I}}$ to be involved in $\mathcal{A}$'s view.

[9] By Lemma 1, the hardness of $\mathsf{SA\text{-}HMLWE}^{1,\sigma_1,\mathcal{S}}_{q,m,n,\sigma_0}$ can be reduced to $\mathsf{MLWE}_{\mathcal{R},q,m,n,\sigma'}$, where $\sigma' > 0$ satisfies $\frac{1}{\sigma_0^2} + \frac{B_{\mathcal{S}}^2}{\sigma_1^2} \leq \frac{1}{\sigma'^2 + 2\eta_\epsilon^2(\mathcal{R}^m)}$ and $\sigma' > \sqrt{2}\cdot\eta_\epsilon(\mathcal{R}^m)$.

– Including $\{\pi_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i}$ in h_i does not affect the proof, because the above rejection-sampling-based NIZK $\overline{\Pi}$ has statistical zero-knowledge property. This property enables us to replace the real proof with a simulated one, and later explain the simulated proof to a specific witness $(\mathbf{k}, y)$ upon corruption with a simulated randomness ρ.[10] By changing both real proofs $\{\pi_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i}$ and randomness ρ to simulated ones, the additional NIZK proofs reveal no extra information, thereby does not affect the proof of Theorem 1.

Finally, we show the robustness of $\mathsf{TD}_1^{\mathsf{rob}}$ via the following theorem.

Theorem 2 (Robustness of $\mathsf{TD}_1^{\mathsf{rob}}$). *Assume that (i) Π has adaptive soundness, (ii) $\overline{\Pi}$ has adaptive soundness, (iii) $2C \cdot (\sqrt{md}B'\tilde{B}_k + \tilde{B}_y) \leq \lfloor \frac{q}{p} \rceil / 2$, then the scheme $\mathsf{TD}_1^{\mathsf{rob}}$ in Fig. 4 (with gray boxes) has robustness.*

Intuitively, NIZKs Π and $\overline{\Pi}$ ensure that only valid ciphertext and partial decryptions are accepted by the challenger, so we can argue that the noises added in the ciphertext and partial decryptions are small by the adaptive soundness of Π and $\overline{\Pi}$. Thus, when combining the two sets of partial decryptions, the aggregated values differ at most $2C(\sqrt{md}B'\tilde{B}_k + \tilde{B}_y) \leq \lfloor \frac{q}{p} \rfloor / 2$, and consequently, they always decrypt to a unique message μ. See Appendix C.3 for a formal proof.

5 Adaptively CCA-Secure Synchronous TD Scheme

In this section, we propose an adaptively CCA-secure *synchronous* TD scheme TD_2 for any *polynomial usr number* N. To this end, we first construct an adaptively smCPA-secure scheme $\mathsf{TD}_2^{\mathsf{smCPA}}$ in Subsect. 5.1 and then show how to upgrade it to CCA-secure TD_2 by the BKW transform [10] in Subsect. 5.2. These schemes have polynomial modulus and their securities are based on the MLWE and MSIS assumptions in the RO model.

5.1 $\mathsf{TD}_2^{\mathsf{smCPA}}$: Adaptively smCPA-Secure Scheme for Polynomial N

Our synchronous (t, N)-TD scheme $\mathsf{TD}_2^{\mathsf{smCPA}}$ for polynomial N uses the following parameters and building blocks.

– Let $q, n, m, p \in \mathbb{N}$ with $p < q$.
– Let $\sigma_k, \sigma_0, \sigma_y > 0$ be Gaussian widths and $B_0, B_y, B'_0, B'_k, B'_{\mathsf{E}} \in \mathbb{R}$, where B_0, B_y are ℓ_∞-norm bounds of $D_{\mathcal{R},\sigma_0}$ and $D_{\mathcal{R},\sigma_y}$, and $B'_0, B'_k, B'_{\mathsf{E}}$ are the ℓ_2-norm bounds of $D_{\mathcal{R}^{m+n},\sigma_0}$, $D_{\mathcal{R}^{m+n},\sigma_k}$ and $D_{\mathcal{R}^{m+n+1},\sigma_0}$, respectively.
– The share algorithm SShare of Shamir secret sharing (cf. Subsect. 2.2).
– Two random oracles $\mathsf{H}_{\mathsf{mask}} : \{0,1\}^* \to \mathcal{R}_q$ and $\mathsf{H}_{\mathsf{rand}} : \{0,1\}^* \to \mathcal{R}$, where the output of $\mathsf{H}_{\mathsf{rand}}$ follows the distribution $D_{\mathcal{R},\sigma_y}$.

Our scheme $\mathsf{TD}_2^{\mathsf{smCPA}} = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{PDec}, \mathsf{Comb})$ for message space $\mathcal{M} = \mathcal{R}_p$ is shown in Fig. 5 (without dashed boxes).

[10] Recall that our CCA model also allows $\mathcal{A}$ to obtain randomness through corruptions.

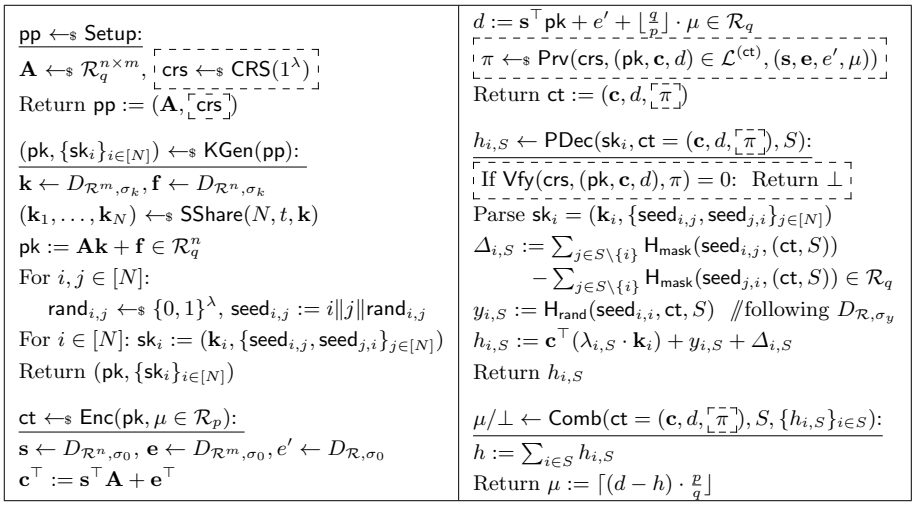

Fig. 5. Our adaptively smCPA-secure synchronous (t, N)-TD scheme $\mathsf{TD}_2^{\mathsf{smCPA}}$ (without dashed boxes) and its adaptively CCA-secure extension TD_2 (with dashed boxes) for polynomial user number N. We explain $\mathsf{TD}_2^{\mathsf{smCPA}}$ in Subsect. 5.1 and TD_2 in Subsect. 5.2.

Correctness. The correctness of $\mathsf{TD}_2^{\mathsf{smCPA}}$ follows from the correctness of Shamir secret sharing (cf. Subsect. 2.2), and the inequality $B'_0B'_k + B_0 + \sqrt{N}B_y \le \lfloor \frac{q}{p} \rfloor/2$. Below we show that the Comb algorithm can recover the message μ correctly except with negligible probability. Note that for S, it holds that $\sum_{i\in S}\Delta_{i,S} = 0$. This is because, according to the computation of $\Delta_{i,S}$ in Fig. 5, we have

$$\begin{aligned}&\textstyle\sum_{i\in S}\Delta_{i,S} = \sum_{i\in S}\sum_{j\in S\setminus\{i\}}(\mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j},(\mathsf{ct},S)) - \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,i},(\mathsf{ct},S)))\\ =&\textstyle\sum_{i\neq j\in S}\mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j},(\mathsf{ct},S)) - \sum_{i\neq j\in S}\mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,i},(\mathsf{ct},S)) = 0.\end{aligned}$$

For any message $\mu \in \mathcal{R}_p$, the encryption algorithm $\mathsf{Enc}(\mathsf{pk}, \mu)$ generates ciphertext $\mathsf{ct} = (\mathbf{c}, d)$ by computing $\mathbf{c}^\top := \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top$, $d := \mathbf{s}^\top(\mathbf{Ak} + \mathbf{f}) + e' + \lfloor \frac{q}{p} \rfloor \cdot \mu$ with $\mathbf{s} \leftarrow D_{\mathcal{R}^n,\sigma_0}, \mathbf{e} \leftarrow D_{\mathcal{R}^m,\sigma_0}, e' \leftarrow D_{\mathcal{R},\sigma_0}$. Then for $h_{i,S}$ generated by the partial decryption algorithm $\mathsf{PDec}(\mathsf{sk}_i, \mathsf{ct}, S)$, we have $h_{i,S} := \mathbf{c}^\top(\lambda_{i,S}\cdot\mathbf{k}_i) + y_{i,S} + \Delta_{i,S}$, where $y_{i,S} = \mathsf{H}_{\mathsf{rand}}(\mathsf{seed}_{i,i}, \mathsf{ct}, S)$ following the distribution $D_{\mathcal{R},\sigma_y}$. Thus, for any $S \subseteq [N]$ with $|S| \ge t$, in the combination algorithm $\mathsf{Comb}(\mathsf{pk}, \mathsf{ct}, S, \{h_{i,S}\}_{i\in S})$, we have

$$h := \textstyle\sum_{i\in S} h_{i,S} = \mathbf{c}^\top(\sum_{i\in S}\lambda_{i,S}\mathbf{k}_i) + \sum_{i\in S} y_{i,S} + \sum_{i\in S}\Delta_{i,S} = \mathbf{c}^\top\mathbf{k} + \sum_{i\in S} y_{i,S},$$

where the last equality holds due to $\sum_{i\in S}\Delta_{i,S} = 0$, and $\sum_{i\in S}\lambda_{i,S}\mathbf{k}_i = \mathbf{k}$ by the correctness of Shamir secret sharing (cf. Subsect. 2.2). Therefore,

$$d - h = \mathbf{s}^\top(\mathbf{Ak}+\mathbf{f}) + e' + \lfloor \tfrac{q}{p} \rfloor\cdot\mu - \mathbf{c}^\top\mathbf{k} - \textstyle\sum_{i\in S} y_{i,S} = \lfloor \frac{q}{p} \rfloor\cdot\mu + \mathbf{s}^\top\mathbf{f} + e' - \mathbf{e}^\top\mathbf{k} - \sum_{i\in S} y_{i,S}.$$

By Lemma 4, it holds that $\|\mathbf{s}^\top\mathbf{f} - \mathbf{e}^\top\mathbf{k}\|_\infty = \|(\mathbf{s},\mathbf{e})^\top(\mathbf{f},-\mathbf{k})\|_\infty \le B'_0B'_k$. By Lemma 5, the distribution of $\sum_{i\in S} y_{i,S}$ is statistically close to $D_{\mathcal{R},\sqrt{|S|}\sigma_y}$, so that $\|\sum_{i\in S} y_{i,S}\|_\infty \le \sqrt{N}B_y$. Consequently, the ℓ_∞-norm of the noise term

$\mathbf{s}^\top \mathbf{f} + e' - \mathbf{e}^\top \mathbf{k} - \sum_{i \in S} y_{i,S}$ is bounded by $B'_0 B'_k + B_0 + \sqrt{N} B_y \leq \lfloor \frac{q}{p} \rfloor / 2$ except with negligible probability. Consequently, $\lceil (d-h) \cdot p/q \rfloor$ correctly recovers μ.

The following Theorem 3 shows the smCPA security of $\mathsf{TD}_2^{\mathsf{smCPA}}$. Due to space limitations, we postpone the full smCPA security proof to Appendix C.4. (See technical overview in Sect. 1 for an intuition.)

Theorem 3 (IND-smCPA Security of $\mathsf{TD}_2^{\mathsf{smCPA}}$). *Let $\mathsf{H}_{\mathsf{mask}}, \mathsf{H}_{\mathsf{rand}}$ be RO. If*

- *the $\mathsf{MSIS}_{q,n,m+n,2B'_{\mathsf{E}}}$ assumption and $\mathsf{MLWE}_{q,m+1,n,\sigma_0}$ assumption hold,*
- *the $\mathsf{AHMLWE}^{\ell,\mathcal{V},\sigma_y}_{q,m,n,\sigma_k}$ assumption holds, where $\mathcal{V} := \{\mathbf{v} = (v_i)_{i\in[n+m]} \in \mathcal{R}^{n+m} : \sum_{i\in[n+m]} \|v_i\|_1^2 \leq B_{\mathsf{hint}}\}$ for $B_{\mathsf{hint}} := d \cdot B'^2_{\mathsf{E}}$,*[11]

then the scheme $\mathsf{TD}_2^{\mathsf{smCPA}}$ in Fig. 5 is IND-smCPA secure.

In Appendix C.4, we also provide an explicit bound of $\mathsf{Adv}^{\mathsf{smCPA}}_{\mathsf{TD}_2^{\mathsf{smCPA}},\mathcal{A}}(\lambda)$.

5.2 TD_2: Achieving Adaptive CCA-Security via BKW Transform

Now we upgrade our smCPA-secure $\mathsf{TD}_2^{\mathsf{smCPA}}$ in Subsect. 5.1 to a CCA-secure TD_2 (cf. Fig. 5 with dashed boxes) by applying the smCPA-to-CCA transform by Brzuska, Klooß and Woo (BKW) in [10], with the help of a NIZK Π.

- $\Pi = (\mathsf{CRS}, \mathsf{Prv}, \mathsf{Vfy}, \mathsf{Sim})$ is a simulation-extractable NIZK scheme (cf. Def. 11 in Appendix A.2) for language $\mathcal{L}^{(\mathsf{ct})}$:

$$\mathcal{L}^{(\mathsf{ct})} := \left\{ (\mathsf{pk}, \mathbf{c}, d) \;\middle|\; \exists (\mathsf{rnd} = (\mathbf{s}, \mathbf{e}, e'), \mu), \text{ s.t. } \begin{array}{c} \|\mathsf{rnd}\| \leq B'_{\mathsf{E}} \;\wedge\; \mathbf{c}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top \\ d = \mathbf{s}^\top \mathsf{pk} + e' + \lfloor \frac{q}{p} \rfloor \cdot \mu \end{array} \right\},$$

  which captures that $(\mathbf{c}, d)$ is a ciphertext of $\mathsf{TD}_2^{\mathsf{smCPA}}$ under public key pk and randomness $\mathsf{rnd} = (\mathbf{s}, \mathbf{e}, e')$, with B'_{E} the ℓ_2-norm bound of $D_{\mathcal{R}^{m+n+1},\sigma_0}$.

For an instantiation of above Π, we use the construction in [28], as recalled in Appendix D.3 for completeness. In Table 3 in Appendix E.2, we present a suggested parameter setting which satisfy all conditions required by the correctness and security of TD_2 in this section, and provide an estimate efficiency of TD_2.

The adaptive CCA security of our TD_2 follows from the following theorem.

Theorem 4 (IND-CCA Security of TD_2, adapted from [10, Theorem 7]). *If $\mathsf{TD}_2^{\mathsf{smCPA}}$ is an adaptively IND-smCPA secure synchronous TD with deterministic PDec algorithm and Π is a simulation-extractable NIZK for relation $\mathcal{R}^{(\mathsf{ct})}$, then TD_2 in Fig. 5 (with dashed boxes) is adaptively IND-CCA secure.*

Proof Sketches. In [10, Theorem 7], BKW prove that their smCPA-to-CCA transform works well in the asynchronous setting and in a model where corruptions reveals only the secret key. Their proof extends to our synchronous setting for the following two reasons.

[11] By Lemma 2, under appropriate parameters, AHMLWE can be reduced to MLWE.

First, the two settings differ only in that the synchronous PDec additionally takes a set S as input. Roughly speaking, during the smCPA-to-CCA reduction, to answer $\mathcal{O}_{\mathrm{DEC}}^{(\mathsf{CCA})}(i, \mathsf{ct}, S)$ with $\mathsf{ct} = (\mathbf{c}, d, \pi)$ from a CCA adversary, the reduction can extract (rnd, μ) from the NIZK proof π, query $\mathcal{O}_{\mathrm{ENC}}(\mu, \mathsf{rnd})$ to its own $\mathsf{IND\text{-}smCPA}$ challenger to obtain an honestly generated ct', and then query $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}', S)$ to its own $\mathsf{IND\text{-}smCPA}$ challenger and return the decryption result to the CCA adversary. This is the same simulation as in their asynchronous proof [10, Theorem 7], with S simply passed to the partial-decryption oracle $\mathcal{O}_{\mathrm{DEC}}$. All other parts of the proof remain unchanged.

Second, their corruption oracle reveals sk_i, whereas ours also reveals the randomness used by PDec. Since our PDec is deterministic, its randomness is empty, so the two corruption oracles return equivalent information.

Therefore, the proof of [10, Theorem 7] also applies to our setting to show the $\mathsf{IND\text{-}CCA}$ security of our TD_2, and Theorem 4 follows. □

References

[1] Blömer, J., Bobolz, J., Porzenheim, L.: A generic construction of an anonymous reputation system and instantiations from lattices. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part II. LNCS, vol. 14439, pp. 418–452. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8724-5_13

[2] Boneh, D., Bünz, B., Nayak, K., Rotem, L., Shoup, V.: Context-dependent threshold decryption and its applications. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part VI. LNCS, vol. 16250, pp. 506–538. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5119-4_16

[3] Boneh, D., Gennaro, R., Goldfeder, S., Jain, A., Kim, S., Rasmussen, P.M.R., Sahai, A.: Threshold cryptosystems from threshold fully homomorphic encryption. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part I. LNCS, vol. 10991, pp. 565–596. Springer, Cham (Aug 2018). https://doi.org/10.1007/978-3-319-96884-1_19

[4] Boschini, C., Takahashi, A., Tibouchi, M.: MuSig-L: Lattice-based multi-signature with single-round online phase. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 276–305. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_10

[5] Boudgoust, K., Lapiha, O.: Leftover hash lemma(s) over cyclotomic rings. In: Daemen, J., Thomé, E. (eds.) Advances in Cryptology – EUROCRYPT 2026. pp. 62–91. Springer Nature Switzerland, Cham (2026)

[6] Boudgoust, K., Lapiha, O., del Pino, R., Prest, T.: IND-CCA lattice threshold KEM under 30 KiB. In: PKC 2026 (May 2026), https://eprint.iacr.org/2026/021

[7] Boudgoust, K., Scholl, P.: Simple threshold (fully homomorphic) encryption from LWE with polynomial modulus. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part I. LNCS, vol. 14438, pp. 371–404. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8721-4_12

[8] Brakerski, Z., Friedman, O., Marmor, A., Mutzari, D., Spiizer, Y., Trieu, N.: Threshold FHE with efficient asynchronous decryption. Cryptology ePrint Archive, Report 2025/712 (2025), https://eprint.iacr.org/2025/712

[9] Braun, L., Damgård, I., Orlandi, C.: Secure multiparty computation from threshold encryption based on class groups. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part I. LNCS, vol. 14081, pp. 613–645. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38557-5_20

[10] Brzuska, C., Klooß, M., Woo, I.K.Y.: Threshold public-key encryption: Definitions, relations, and cpa-to-cca transforms. In: Bai, S., Persichetti, E. (eds.) PKC 2026, Part IV. LNCS, vol. 16554, pp. 162–197. Springer (2026). https://doi.org/10.1007/978-3-032-26740-5_6, https://doi.org/10.1007/978-3-032-26740-5_6

[11] Choudhuri, A.R., Garg, S., Piet, J., Policharla, G.V.: Mempool privacy via batched threshold encryption: Attacks and defenses. In: Balzarotti, D., Xu, W. (eds.) USENIX Security 2024. USENIX Association (Aug 2024), https://www.usenix.org/conference/usenixsecurity24/presentation/choudhuri

[12] Cini, V., Lai, R.W.F., Woo, I.K.Y.: Pilvi: Lattice threshold PKE with small decryption shares and improved security. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part VI. LNCS, vol. 16250, pp. 539–569. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5119-4_17

[13] Cramer, R., Gennaro, R., Schoenmakers, B.: A secure and optimally efficient multi-authority election scheme. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 103–118. Springer, Berlin, Heidelberg (May 1997). https://doi.org/10.1007/3-540-69053-0_9

[14] Das, S., Ren, L., Yang, Z.: Adaptively secure threshold ElGamal decryption from DDH. Cryptology ePrint Archive, Report 2025/1477 (2025), https://eprint.iacr.org/2025/1477

[15] De Santis, A., Di Crescenzo, G., Ostrovsky, R., Persiano, G., Sahai, A.: Robust non-interactive zero knowledge. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 566–598. Springer, Berlin, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8_33

[16] Del Pino, R., Katsumata, S., Maller, M., Mouhartem, F., Prest, T., Saarinen, M.J.O.: Threshold raccoon: Practical threshold signatures from standard lattice assumptions. In: Joye, M., Leander, G. (eds.) EUROCRYPT 2024, Part II. LNCS, vol. 14652, pp. 219–248. Springer, Cham (May 2024). https://doi.org/10.1007/978-3-031-58723-8_8

[17] Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) CRYPTO'89. LNCS, vol. 435, pp. 307–315. Springer, New York (Aug 1990). https://doi.org/10.1007/0-387-34805-0_28

[18] Devevey, J., Libert, B., Nguyen, K., Peters, T., Yung, M.: Non-interactive CCA2-secure threshold cryptosystems: Achieving adaptive security in the standard model without pairings. In: Garay, J. (ed.) PKC 2021, Part I. LNCS, vol. 12710, pp. 659–690. Springer, Cham (May 2021). https://doi.org/10.1007/978-3-030-75245-3_24

[19] Devevey, J., Passelègue, A., Stehlé, D.: G+G: A Fiat-Shamir lattice signature based on convolved gaussians. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023, Part VII. LNCS, vol. 14444, pp. 37–64. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8739-9_2

[20] Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008). https://doi.org/10.1137/060651380, https://doi.org/10.1137/060651380

[21] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Dilithium: A lattice-based digital signature scheme. IACR TCHES **2018**(1), 238–268 (2018). https://doi.org/10.13154/tches.v2018.i1.238-268, https://tches.iacr.org/index.php/TCHES/article/view/839

[22] Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(1), 238–268 (Feb 2018). https://doi.org/10.13154/tches.v2018.i1.238-268, https://tches.iacr.org/index.php/TCHES/article/view/839

[23] Espitau, T., Niot, G., Prest, T.: Flood and submerse: Distributed key generation and robust threshold signature from lattices. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part VII. LNCS, vol. 14926, pp. 425–458. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68394-7_14

[24] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Berlin, Heidelberg (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12

[25] Genise, N., Micciancio, D., Peikert, C., Walter, M.: Improved discrete gaussian and subgaussian analysis for lattice cryptography. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part I. LNCS, vol. 12110, pp. 623–651. Springer, Cham (May 2020). https://doi.org/10.1007/978-3-030-45374-9_21

[26] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Ladner, R.E., Dwork, C. (eds.) 40th ACM STOC. pp. 197–206. ACM Press (May 2008). https://doi.org/10.1145/1374376.1374407

[27] Groth, J.: Simulation-sound NIZK proofs for a practical language and constant size group signatures. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 444–459. Springer, Berlin, Heidelberg (Dec 2006). https://doi.org/10.1007/11935230_29

[28] Hwang, Y., Katsumata, S., Min, S., Niot, G., Song, Y.: Proving threshold Regev PKE from adaptive hint-MLWE: Efficient, non-interactive, and CCA secure. Cryptology ePrint Archive, Paper 2026/1585 (2026), https://eprint.iacr.org/2026/1585

[29] Ito, M., Saito, A., Nishizeki, T.: Secret sharing schemes realizing general access structure. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '87). pp. 99–102. IEEE (1987)

[30] Jiang, K., Wee, H., Zhu, C.: Oriole: Adaptively secure partially non-interactive threshold signatures from lattices. Cryptology ePrint Archive, Paper 2026/793 (2026), https://eprint.iacr.org/2026/793

[31] Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 1–20. Springer, Berlin, Heidelberg (Dec 2013). https://doi.org/10.1007/978-3-642-42033-7_1

[32] Katsumata, S., Reichle, M., Takemure, K.: Adaptively secure 5 round threshold signatures from MLWE/MSIS and DL with rewinding. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part VII. LNCS, vol. 14926, pp. 459–491. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68394-7_15

[33] Kim, W., Lee, C., Lee, J., Passelègue, A., Stehlé, D.: Asynchronous lagrange-based threshold FHE with smaller modulus overhead. CRYPTO 2026 (Aug 2026), https://eprint.iacr.org/2026/973

[34] Lai, R.W.F., Swarnakar, M., Woo, I.K.Y.: Leaky LWE: Learning with errors with semi-adaptive secret- and error-leakage. CiC **2**(3), 21 (2025). https://doi.org/10.62056/ah89ksuc2

[35] Lapiha, O., Prest, T.: A lattice-based IND-CCA threshold KEM from the BCHK+ transform. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part III. LNCS, vol. 16247, pp. 461–494. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5099-9_15

[36] Libert, B., Nguyen, K., Passelègue, A., Titiu, R.: Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 128–158. Springer, Cham (Dec 2020). https://doi.org/10.1007/978-3-030-64837-4_5

[37] Libert, B., Yung, M.: Adaptively secure non-interactive threshold cryptosystems. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part II. LNCS, vol. 6756, pp. 588–600. Springer, Berlin, Heidelberg (Jul 2011). https://doi.org/10.1007/978-3-642-22012-8_47

[38] Lindner, R., Peikert, C.: Better key sizes (and attacks) for LWE-based encryption. In: Kiayias, A. (ed.) CT-RSA 2011. LNCS, vol. 6558, pp. 319–339. Springer, Berlin, Heidelberg (Feb 2011). https://doi.org/10.1007/978-3-642-19074-2_21

[39] Lyubashevsky, V.: Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 598–616. Springer, Berlin, Heidelberg (Dec 2009). https://doi.org/10.1007/978-3-642-10366-7_35

[40] Lyubashevsky, V.: Lattice signatures without trapdoors. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 738–755. Springer, Berlin, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_43

[41] Lyubashevsky, V., Nguyen, N.K., Plançon, M.: Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 71–101. Springer, Cham (Aug 2022). https://doi.org/10.1007/978-3-031-15979-4_3

[42] Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 700–718. Springer, Berlin, Heidelberg (Apr 2012). https://doi.org/10.1007/978-3-642-29011-4_41

[43] Micciancio, D., Suhl, A.: Simulation-secure threshold PKE from LWE with polynomial modulus. CiC **1**(4), 2 (2024). https://doi.org/10.62056/a0zogy4e-

[44] National Institute of Standards and Technology (NIST): Post-quantum cryptography pqc - security (evaluation criteria). NIST Computer Security Resource Center (CSRC) (2025), https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria)

[45] Okada, H., Takagi, T.: Low communication threshold FHE from standard (module-)LWE. In: Hanaoka, G., Yang, B.Y. (eds.) ASIACRYPT 2025, Part VII. LNCS, vol. 16251, pp. 165–198. Springer, Singapore (Dec 2025). https://doi.org/10.1007/978-981-95-5122-4_6

[46] Okada, H., Takagi, T.: Threshold FHE with short decryption shares without a semi-trusted server. ACISP 2026 (2026), https://eprint.iacr.org/2026/957

[47] Peikert, C.: A decade of lattice cryptography. Cryptology ePrint Archive, Report 2015/939 (2015), https://eprint.iacr.org/2015/939

[48] Pino, R.D., Katsumata, S., Niot, G., Reichle, M., Takemure, K.: Unmasking TRaccoon: A lattice-based threshold signature with an efficient identifiable abort protocol. In: Kalai, Y.T., Kamara, S.F. (eds.) CRYPTO 2025, Part VI. LNCS, vol. 16005, pp. 423–456. Springer, Cham (Aug 2025). https://doi.org/10.1007/978-3-032-01887-8_14

[49] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC. pp. 84–93. ACM Press (May 2005). https://doi.org/10.1145/1060590.1060603

[50] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: 40th FOCS. pp. 543–553. IEEE Computer Society Press (Oct 1999). https://doi.org/10.1109/SFFCS.1999.814628

[51] Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11), 612–613 (Nov 1979). https://doi.org/10.1145/359168.359176

[52] Shoup, V.: Back to the future: simple threshold decryption secure against adaptive corruptions. CiC 2026 (2025), https://eprint.iacr.org/2025/1578

[53] Urban, A., Rambaud, M.: Robust multiparty computation from threshold encryption based on RLWE. In: Mouha, N., Nikiforakis, N. (eds.) ISC 2024, Part I. LNCS, vol. 15257, pp. 294–314. Springer, Cham (Oct 2024). https://doi.org/10.1007/978-3-031-75757-0_15

[54] de Verdière, F.C., Passelègue, A., Stehlé, D.: On threshold fully homomorphic encryption with synchronized decryptors. Cryptology ePrint Archive, Paper 2026/031 (2026), https://eprint.iacr.org/2026/031

[55] Wang, H., Steinfeld, R., Saarinen, M.J.O., Esgin, M.F., Yiu, S.M.: mmCipher: Batching post-quantum public key encryption made bandwidth-optimal. USENIX Security 2026 (2026), https://eprint.iacr.org/2025/1000

Supplementary Material

A Additional Preliminaries

A.1 Collision-Resistant Hash Functions

Definition 8 (Collision-Resistant Hash Functions). *A family of hash functions $\mathcal{H}$ is collision-resistant, if for any PPT adversary $\mathcal{A}$, it holds that*

$$\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{A}}(\lambda) := \Pr[\mathsf{H} \leftarrow_{\$} \mathcal{H}, (x, x') \leftarrow_{\$} \mathcal{A}(\mathsf{H}) : x \neq x' \ \wedge \ \mathsf{H}(x) = \mathsf{H}(x')] \leq \mathsf{negl}(\lambda).$$

A.2 Quasi-Adaptive Non-Interactive Zero-Knowledge

Quasi-Adaptive Non-Interactive Zero-Knowledge argument (QA-NIZK) was proposed by Jutla and Roy [31], where the common reference string (CRS) may depend on the specific language $\mathcal{L}_\rho$ for which proofs are generated. Below we formalize tag-based QA-NIZK according to [36] for gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \tilde{\mathcal{L}}_\rho)$, where $\mathcal{L}_\rho \subseteq \tilde{\mathcal{L}}_\rho$. Intuitively, completeness and zero-knowledge are guaranteed for instances in $\mathcal{L}_\rho$, while soundness is guaranteed for instances outside $\tilde{\mathcal{L}}_\rho$.

Definition 9 (QA-NIZK for Gap Language). *A tag-based QA-NIZK scheme $\Pi = (\mathsf{CRS}, \mathsf{Prv}, \mathsf{Vfy}, \mathsf{Sim})$ for gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \tilde{\mathcal{L}}_\rho)$ (where $\mathcal{L}_\rho \subseteq \tilde{\mathcal{L}}_\rho \subseteq \mathcal{X}$) with tag space $\mathcal{T}$ consists of a tuple of PPT algorithms:*[12]

- $\mathsf{crs} \leftarrow_{\$} \mathsf{CRS}(\rho)$*: Taking as input the language parameter ρ, the CRS generation algorithm outputs a common reference string* crs*.*
- $\pi \leftarrow_{\$} \mathsf{Prv}(\mathsf{crs}, x, \tau, w)$*: Taking as input* crs*, an instance $x \in \mathcal{L}_\rho$, a tag $\tau \in \mathcal{T}$ and witness w, the proof generation algorithm outputs a proof π.*
- $0/1 \leftarrow \mathsf{Vfy}(\mathsf{crs}, x, \tau, \pi)$*: Taking as input* crs*, $x \in \mathcal{X}$, a tag $\tau \in \mathcal{T}$ and a proof π, the deterministic verification algorithm outputs a bit indicating whether π is a valid proof.*
- $\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, x, \tau)$*: Taking as input* crs*, $x \in \mathcal{X}$ and a tag $\tau \in \mathcal{T}$, the simulation algorithm outputs a simulated proof π.*[13]

Moreover, Π should satisfy the following properties:

- ***Completeness.*** *For any $\mathsf{crs} \leftarrow_{\$} \mathsf{CRS}(\rho)$, $x \in \mathcal{L}_\rho$, $\tau \in \mathcal{T}$ and witness w, it holds that $\Pr[\pi \leftarrow_{\$} \mathsf{Prv}(\mathsf{crs}, x, \tau, w) : \mathsf{Vfy}(\mathsf{crs}, x, \tau, \pi) = 1] \geq 1 - \mathsf{negl}(\lambda)$.*
- ***Adaptive Soundness.*** *For any PPT adversary $\mathcal{A}$, it holds that*

$$\Pr\left[\mathsf{crs} \leftarrow_{\$} \mathsf{CRS}(\rho), (x, \tau, \pi) \leftarrow_{\$} \mathcal{A}(\mathsf{crs}) : \mathsf{Vfy}(\mathsf{crs}, x, \tau, \pi) = 1 \wedge x \notin \tilde{\mathcal{L}}_\rho\right] \leq \mathsf{negl}(\lambda).$$

[12] If the tag space $\mathcal{T} = \emptyset$, then it reduces to (non-tag-based) QA-NIZK. If $\mathcal{L}_\rho = \tilde{\mathcal{L}}_\rho$, then it reduces to QA-NIZK for (traditional non-gap) language $\mathcal{L}_\rho$. If CRS and crs do not depend on ρ, then it reduces to (traditional non-quasi-adaptive) NIZK.

[13] In the RO model, Sim can control the RO in order to generate simulated proofs.

- ***Special Zero-Knowledge.*** *For any PPT adversary $\mathcal{A}$, we have*

$$\mathsf{Adv}^{\mathsf{zk}}_{\Pi,\mathcal{A}} := \left|\Pr[\mathcal{A}^{\mathsf{Prv}(\mathsf{crs},\cdot,\cdot,\cdot)}(\mathsf{crs})=1] - \Pr[\mathcal{A}^{\mathcal{O}_{\mathrm{SIM}}(\mathsf{crs},\cdot,\cdot,\cdot)}(\mathsf{crs})=1]\right| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{crs} \leftarrow_{\$} \mathsf{CRS}(\rho)$. Here the oracle $\mathsf{Prv}(\mathsf{crs},x,\tau,w)$ outputs $\perp$ if w is not a witness for $x \in \mathcal{L}_\rho$ and outputs a valid proof $\pi \leftarrow_{\$} \mathsf{Prv}(\mathsf{crs},x,\tau,w)$ otherwise; the oracle $\mathcal{O}_{\mathrm{SIM}}(\mathsf{crs},x,\tau,w)$ outputs $\perp$ if w is not a witness for $x \in \mathcal{L}_\rho$ and outputs a simulated proof $\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs},x,\tau)$ otherwise.

Simulation-Soundness. Simulation soundness [50, 15] formalizes the inability of adversary to create a proof for a false statement x^* even after having seen simulated proofs for possibly false statements $\{x_i\}_i$ of its choice. *One-time simulation soundness* (OTSS) is a weaker notion that only allows the adversary to see one simulated proof, which is sufficient for our purpose.

Definition 10 (OTSS of Tag-based NIZK). *A tag-based QA-NIZK scheme Π for gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \tilde{\mathcal{L}}_\rho)$ has one-time simulation soundness (OTSS), if for any PPT $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{OTSS}}_{\Pi,\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{OTSS}}_{\Pi,\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$, where the experiment $\mathsf{Exp}^{\mathsf{OTSS}}_{\Pi,\mathcal{A}}$ is defined in Fig. 6*

$\underline{\mathsf{Exp}^{\mathsf{OTSS}}_{\Pi,\mathcal{A}}}$:
$\underline{\mathsf{crs} \leftarrow_{\$} \mathsf{CRS}(\rho)}$, $\mathcal{Q}_{\mathrm{SIM}} := \emptyset$
$(x^*,\tau^*,\pi^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\mathrm{SIM}}(\cdot,\cdot)}(\rho,\mathsf{crs})$
If $(x^* \notin \tilde{\mathcal{L}}_\rho) \wedge ((x^*,\tau^*,\pi^*) \notin \mathcal{Q}_{\mathrm{SIM}}) \wedge (\mathsf{Vfy}(\mathsf{crs},x^*,\tau^*,\pi^*)=1)$: Return 1; Else: Return 0

$\underline{\mathcal{O}_{\mathrm{SIM}}(x,\tau)}$: //can be queried only once
$\underline{\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs},x,\tau)}$, $\mathcal{Q}_{\mathrm{SIM}} := \mathcal{Q}_{\mathrm{SIM}} \cup \{(x,\tau,\pi)\}$
Return π

Fig. 6. The OTSS experiment $\mathsf{Exp}^{\mathsf{OTSS}}_{\Pi,\mathcal{A}}$ for tag-based QA-NIZK Π.

Simulation Extractability. Simulation extractability [15, 27] requires that a valid witness can be efficiently extracted from any accepting proof generated by an adversary, even after it has seen simulated proofs. We use the strong variant: extraction is required for every new statement–proof pair, including a new proof for a previously queried statement.

Definition 11 (Simulation Extractability of NIZK, adapted from [28, Def. 11]). *A (non-tag-based, non-QA) NIZK scheme $\Pi = (\mathsf{CRS},\mathsf{Prv},\mathsf{Vfy},\mathsf{Sim})$ for a (non-gap) language $\mathcal{L}$ associated with relation $\mathcal{R}$ is simulation-extractable, if there exists another PPT simulator $\mathsf{Sim} = (\mathsf{Sim.CRS},\mathsf{Sim.Prv},\mathsf{Sim.Ext})$ s.t. for any PPT $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{SE}}_{\Pi,\mathcal{A}}(\lambda) := |\Pr[\mathsf{Exp}^{\mathsf{SE\text{-}0}}_{\Pi,\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{Exp}^{\mathsf{SE\text{-}1}}_{\Pi,\mathcal{A}} \Rightarrow 1]| \leq \mathsf{negl}(\lambda)$, where the experiment $\mathsf{Exp}^{\mathsf{SE}\text{-}b}_{\Pi,\mathcal{A}}$ ($b \in \{0,1\}$) is defined in Fig. 7.*

B Useful Lemmas for Proof of Lemma 8

We recall some additional lattice background and useful lemmas necessary for the proof of Lemma 8.

$\underline{\mathsf{Exp}^{\mathsf{SE}\text{-}b}_{\Pi,\mathcal{A}}}$: // $b \in \{0,1\}$	
If $b=0$: $\mathsf{crs} \leftarrow_{\$} \mathsf{CRS}(1^\lambda)$	
If $b=1$: $(\mathsf{crs},\mathsf{st}) \leftarrow_{\$} \mathsf{Sim.CRS}(1^\lambda)$, $\mathcal{W} \leftarrow \emptyset$	
Return $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{PRV}}(\cdot,\cdot),\mathcal{O}_{\mathrm{VFY}}(\cdot,\cdot)}(\mathsf{crs})$	
$\underline{\mathcal{O}_{\mathrm{PRV}}(x,w)}$:	$\underline{\mathcal{O}_{\mathrm{VFY}}(x,\pi)}$:
If $(x,w) \notin \mathcal{R}$: Return $\perp$	If $\mathsf{Vfy}(\mathsf{crs},x,\pi)=0$: Return 0
If $b=0$: $\pi \leftarrow_{\$} \mathsf{Prv}(\mathsf{crs},x,w)$	If $b=0$: Return 1
If $b=1$:	If $(x,\pi) \in \mathcal{W}$: Return 1
$(\pi,\mathsf{st}) \leftarrow_{\$} \mathsf{Sim.Prv}(x,\mathsf{st})$	$(w,\mathsf{st}) \leftarrow_{\$} \mathsf{Sim.Ext}(x,\pi,\mathsf{st})$
$\mathcal{W} \leftarrow \mathcal{W} \cup \{(x,\pi)\}$	If $(x,w) \in \mathcal{R}$: Return 1 //no extraction failure
Return π	Return 0

Fig. 7. The simulation extractability experiment $\mathsf{Exp}^{\mathsf{SE}}_{\Pi,\mathcal{A}}$ for NIZK.

Let $\mathbf{B} \in \mathbb{R}^{n\times m}$ and let $\mathbf{\Sigma} \in \mathbb{R}^{m\times m}$ be a symmetric matrix. We say a symmetric matrix $\mathbf{\Sigma}$ is positive definite (resp. semi-definite) if and only if $\mathbf{x}^\top\mathbf{\Sigma}\mathbf{x} > 0$ (resp. ≥ 0) for all non-zero $\mathbf{x} \in \mathbb{R}^m$. The symmetric matrix $\mathbf{B}^\top\mathbf{B}$ is positive semi-definite, because $\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x} = \|\mathbf{B}\mathbf{x}\|^2 \geq 0$ for any non-zero $\mathbf{x} \in \mathbb{R}^m$, where the inequality is always strict if and only if $\mathbf{B}^\top\mathbf{B}$ is nonsingular. We say that $\mathbf{B}$ is a square root of $\mathbf{\Sigma}$, written by $\mathbf{B} := \sqrt{\mathbf{\Sigma}}$, if $\mathbf{B}^\top\mathbf{B} = \mathbf{\Sigma}$. Each positive semi-definite $\mathbf{\Sigma}$ has a square root, which can be computed efficiently, e.g., via the Cholesky decomposition. The (Moore-Penrose) pseudoinverse for $\mathbf{\Sigma}$, denoted by $\mathbf{\Sigma}^+$, is the unique matrix such that $(\mathbf{\Sigma}\mathbf{\Sigma}^+)\mathbf{\Sigma} = \mathbf{\Sigma}, \mathbf{\Sigma}^+(\mathbf{\Sigma}\mathbf{\Sigma}^+) = \mathbf{\Sigma}^+$ and both $\mathbf{\Sigma}\mathbf{\Sigma}^+$ and $\mathbf{\Sigma}^+\mathbf{\Sigma}$ are symmetric. Let $\lambda_{\min}(\mathbf{\Sigma}) := \min_{\mathbf{x}\in\mathbb{R}^m,\|\mathbf{x}\|=1} \mathbf{x}^\top\mathbf{\Sigma}\mathbf{x}$ (resp., $\lambda_{\max}(\mathbf{\Sigma})$) denotes the minimal (resp., maximum) eigenvalue of $\mathbf{\Sigma}$.

Below are some useful properties of ℓ_2-norm, spectral norm and eigenvalue.

Lemma 9 (Properties of ℓ_2-Norm, Spectral Norm and Eigenvalue). *Let $\mathbf{B} \in \mathbb{R}^{n\times m}, \mathbf{C} \in \mathbb{R}^{m\times k}$ and let $\mathbf{\Sigma}_k, \mathbf{\Sigma}_y \in \mathbb{R}^{n\times n}$ be symmetric matrices. We have the following properties:*

1. $s_1(\mathbf{B}) = \sqrt{\lambda_{\max}(\mathbf{B}^\top\mathbf{B})}$;
2. $s_1(\mathbf{B}) = \|\mathbf{B}\|_2$;
3. $s_1(\mathbf{B}\mathbf{C}) \leq s_1(\mathbf{B}) \cdot s_1(\mathbf{C})$;
4. *If $\mathbf{\Sigma}_k$ is positive definite, then* $\lambda_{\min}(\mathbf{\Sigma}_k^{-1}) = \frac{1}{\lambda_{\max}(\mathbf{\Sigma}_k)}$;
5. $\lambda_{\max}(\mathbf{\Sigma}_k + \mathbf{\Sigma}_y) \leq \lambda_{\max}(\mathbf{\Sigma}_k) + \lambda_{\max}(\mathbf{\Sigma}_y)$.

Proof of Lemma 9.

1. $s_1^2(\mathbf{B}) = \max_{\mathbf{x}\in\mathbb{R}^m,\|\mathbf{x}\|=1} \|\mathbf{B}\mathbf{x}\|^2 = \max_{\mathbf{x}\in\mathbb{R}^m,\|\mathbf{x}\|=1} \mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x} = \lambda_{\max}(\mathbf{B}^\top\mathbf{B})$.
2. By the definition, we have $\|\mathbf{B}\|_2^2 = \max_{\|\mathbf{x}\|_2=1} \|\mathbf{B}\mathbf{x}\|_2^2 = \max_{\|\mathbf{x}\|_2=1} \mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x} = \lambda_{\max}(\mathbf{B}^\top\mathbf{B}) = s_1(\mathbf{B})^2$. Then, $\|\mathbf{B}\|_2 = s_1(\mathbf{B})$.
3. Note that $s_1(\mathbf{B}\mathbf{C}) := \max_{\mathbf{x}\in\mathbb{R}^k,\|\mathbf{x}\|=1} \|\mathbf{B}\mathbf{C}\mathbf{x}\| \leq s_1(\mathbf{B}) \cdot \max_{\mathbf{x}\in\mathbb{R}^k,\|\mathbf{x}\|=1} \|\mathbf{C}\mathbf{x}\| = s_1(\mathbf{B}) \cdot s_1(\mathbf{C})$. That the inequality holds is because that $\|\mathbf{B}\mathbf{x}\| \leq s_1(\mathbf{B})\|\mathbf{x}\|$ holds for any non-zero $\mathbf{x} \in \mathbb{R}^m$.

4. Let $\lambda > 0$ be an eigenvalue of $\mathbf{\Sigma}_k$. Then $\frac{1}{\lambda}$ is an eigenvalue of $\mathbf{\Sigma}_k^{-1}$, because there exist $\mathbf{x} \in \mathbb{R}^m$ such that $\mathbf{\Sigma}_k\mathbf{x} = \lambda\mathbf{x} \implies \frac{1}{\lambda}\mathbf{x} = \mathbf{\Sigma}_k^{-1}\mathbf{x}$. Thus, if $\lambda_{\max}$ is the maximal eigenvalue of $\mathbf{\Sigma}$, then $\frac{1}{\lambda_{\max}(\mathbf{\Sigma})}$ is the minimal eigenvalue of $\mathbf{\Sigma}^{-1} = \lambda_{\min}(\mathbf{\Sigma}^{-1})$.
5. By definition, we have

$$\lambda_{\max}(\mathbf{\Sigma}_k + \mathbf{\Sigma}_y) = \max_{\mathbf{x}\in\mathbb{R}^n, \|\mathbf{x}\|=1} (\mathbf{x}^\top(\mathbf{\Sigma}_k + \mathbf{\Sigma}_y)\mathbf{x})$$

$$\leq \max_{\mathbf{x}_0\in\mathbb{R}^n, \|\mathbf{x}_0\|=1} (\mathbf{x}_0^\top\mathbf{\Sigma}_k\mathbf{x}_0) + \max_{\mathbf{x}_1\in\mathbb{R}^n, \|\mathbf{x}_1\|=1} (\mathbf{x}_1^\top\mathbf{\Sigma}_y\mathbf{x}_1) = \lambda_{\max}(\mathbf{\Sigma}_k) + \lambda_{\max}(\mathbf{\Sigma}_y). \square$$

The following lemma established in [23, 55] states that the conditional distribution of $\mathbf{k}$ given some hint $\mathbf{h} = \mathbf{R}\mathbf{k} + \mathbf{y}$ turns out to be a discrete Gaussian with center $\mathbf{c}$ and covariance parameter $\mathbf{\Sigma}$. Here $\mathbf{c}$ and $\mathbf{\Sigma}$ are dependent on $\mathbf{h}, \mathbf{R}$ and the covariance parameters of $\mathbf{k}$ and $\mathbf{y}$, but independent of $\mathbf{k}$.

Lemma 10 (Adapted from [55, Lemma F.2]). *Let $m, \ell \in \mathbb{N}$, $\mathbf{\Sigma}_0 \in \mathbb{R}^{md\times md}$ and $\mathbf{\Sigma}_y \in \mathbb{R}^{\ell d\times \ell d}$ be positive definite symmetric matrices, and $\mathbf{R} \in \mathbb{Z}^{\ell d\times md}$. Denote $\mathbf{\Sigma} := \left(\mathbf{\Sigma}_0^{-1} + \mathbf{R}^\top\mathbf{\Sigma}_y^{-1}\mathbf{R}\right)^{-1}$. Then the following distributions are identical:*

$$\left\{(\mathbf{k}, \mathbf{h}) \mid \mathbf{k} \leftarrow D_{\mathbb{Z}^{md}, \sqrt{\mathbf{\Sigma}_0}},\ \mathbf{y} \leftarrow D_{\mathbb{Z}^{\ell d}, \sqrt{\mathbf{\Sigma}_y}},\ \mathbf{h} = \mathbf{R}\mathbf{k} + \mathbf{y}\right\}$$

$$\equiv \left\{(\hat{\mathbf{k}}, \mathbf{h}) \;\middle|\; \begin{array}{c} \mathbf{k} \leftarrow D_{\mathbb{Z}^{md}, \sqrt{\mathbf{\Sigma}_0}},\ \mathbf{y} \leftarrow D_{\mathbb{Z}^{\ell d}, \sqrt{\mathbf{\Sigma}_y}},\ \mathbf{h} = \mathbf{R}\mathbf{k} + \mathbf{y}, \\ \mathbf{c} = \mathbf{\Sigma}\mathbf{R}^\top\mathbf{\Sigma}_y^{-1}\mathbf{h},\ \hat{\mathbf{k}} \leftarrow D_{\mathbb{Z}^{md}, \sqrt{\mathbf{\Sigma}}, \mathbf{c}} \end{array}\right\}.$$

Lemma 11 ([4, Lemma 2.5]). *Let $m > n$. For any $\epsilon > 0$, except with probability at most 2^{-d} over $\mathbf{B} \leftarrow\!\!\$\ \mathcal{R}_q^{n\times m}$, we have $\eta_\epsilon\left(\Lambda_q^\perp(\mathbf{B})\right) \leq \frac{8}{\sqrt{\pi}} q^{\frac{n}{m}} \sqrt{d\ln\left(2md + \frac{2md}{\epsilon}\right)}$, where $\Lambda_q^\perp(\mathbf{B}) := \{\mathbf{x} \in \mathcal{R}^m \mid \mathbf{B}\mathbf{x} = \mathbf{0} \bmod q\}$.*

Lemma 12 (Matrix-parameter variant of [26, Corollary 2.8]). *Let $\Lambda, \Lambda' \in \mathbb{R}^n$ be lattices with $\Lambda' \subseteq \Lambda$. Let $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ be a positive definite symmetric matrix. Then for any $\epsilon \in (0, \frac{1}{2})$ and $\mathbf{c} \in \mathbb{R}^n$, if $\sqrt{\mathbf{\Sigma}} \geq \eta_\epsilon(\Lambda')$, the distribution $D_{\Lambda, \sqrt{\mathbf{\Sigma}}, \mathbf{c}} \bmod \Lambda'$ is within statistical distance at most 2ϵ of the uniform distribution over $\Lambda \bmod \Lambda'$. Specifically, for $\forall \mathbf{v} \in \Lambda, \Pr[\mathbf{z} = \mathbf{v} \bmod \Lambda' \mid \mathbf{z} \leftarrow D_{\Lambda, \sqrt{\mathbf{\Sigma}}, \mathbf{c}}]$ is in the range $(1 \pm 4\epsilon)/|\Lambda/\Lambda'|$.*

Proof of Lemma 12. Under the invertible transformation $\mathbf{x} \mapsto \mathbf{\Sigma}^{-1/2}\mathbf{x}$, the distribution $D_{\Lambda, \sqrt{\mathbf{\Sigma}}, \mathbf{c}}$ is mapped exactly to $D_{\mathbf{\Sigma}^{-1/2}\Lambda, 1, \mathbf{\Sigma}^{-1/2}\mathbf{c}}$. Moreover, the assumption $\sqrt{\mathbf{\Sigma}} \geq \eta_\epsilon(\Lambda')$ is equivalent to $\eta_\epsilon(\mathbf{\Sigma}^{-1/2}\Lambda') \leq 1$. The result therefore follows from [26, Corollary 2.8], applied to $\mathbf{\Sigma}^{-1/2}\Lambda' \subseteq \mathbf{\Sigma}^{-1/2}\Lambda$. $\square$

C Postponed Proofs

C.1 Proof of Lemma 6

Lemma 6 *For uniformly sampled $\mathbf{A} \leftarrow\!\!\$\ \mathcal{R}_q^{n\times m}$ (w.l.o.g., assume that $m \geq n$), $\mathbf{A}$ has full rank with probability at least $1 - \frac{d}{q^{m-n}(q-1)}$.*

Proof. Recall that q is a prime satisfying $q \equiv 1 \bmod 2d$, and $R_q = \mathbb{Z}_q[X]/(X^d+1)$. Since $(\mathbb{Z}_q^\times, \cdot)$ is a cyclic group of order $q-1$ and $2d \mid (q-1)$, there exists an element $\zeta \in \mathbb{Z}_q^\times$ of order $2d$. For $\ell = 0, \ldots, d-1$, define $\alpha_\ell := \zeta^{2\ell+1}$. The elements $\alpha_0, \ldots, \alpha_{d-1}$ are pairwise distinct and satisfy $\alpha_\ell^d = -1$. It follows that $X^d + 1 = \prod_{\ell=0}^{d-1}(X - \alpha_\ell)$. Since these linear factors are pairwise coprime, the Chinese remainder theorem yields the following isomorphism

$$\mathcal{R}_q \longrightarrow \mathbb{Z}_q^d, \qquad f(X) \longmapsto \big(f(\alpha_0), \ldots, f(\alpha_{d-1})\big). \tag{15}$$

and hence $\mathcal{R}_q \cong \mathbb{Z}_q^d$.

Consequently, a uniformly random matrix $\mathbf{A} = (a_{ij})_{i,j} \leftarrow_\$ \mathcal{R}_q^{n\times m}$ corresponds to d independent uniformly random matrices $\mathbf{A}^{(0)}, \ldots, \mathbf{A}^{(d-1)} \leftarrow_\$ \mathbb{Z}_q^{n\times m}$, where $\mathbf{A}^{(\ell)} := \big(a_{ij}(\alpha_\ell)\big)_{i,j}$ for $\ell = 0, \ldots, d-1$. Moreover, $\mathbf{A}$ has full rank if and only if all $\mathbf{A}^{(0)}, \ldots, \mathbf{A}^{(d-1)} \in \mathbb{Z}_q^{n\times m}$ have full rank.

It remains to estimate the corresponding probability over $\mathbb{Z}_q$. Let $\mathbf{B} \leftarrow_\$ \mathbb{Z}_q^{n\times m}$. We analyze the rows of $\mathbf{B}$ sequentially. Let $\mathbf{b}_i^\top \in \mathbb{Z}_q^m$ denote the i-th row of $\mathbf{B}$. Conditioned on the first $i-1$ rows being linearly independent, their span contains exactly q^{i-1} vectors. Since $\mathbf{b}_i^\top$ is sampled uniformly from $\mathbb{Z}_q^m$, which contains q^m vectors, we have

$$\Pr\big[\mathbf{b}_i^\top \notin \mathrm{span}(\mathbf{b}_1^\top, \ldots, \mathbf{b}_{i-1}^\top) \,\big|\, \mathbf{b}_1, \ldots, \mathbf{b}_{i-1} \text{ are linearly independent}\big] = 1 - \tfrac{1}{q^{m-i+1}}.$$

Consequently, for a uniformly sampled matrix $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ with $n \le m$, the probability that $\mathbf{B}$ has full row rank is

$$\Pr[\mathrm{rank}(\mathbf{B}) = n] = \prod_{i=1}^{n}\big(1 - \tfrac{1}{q^{m-i+1}}\big) \ge 1 - \textstyle\sum_{i=1}^{n} \tfrac{1}{q^{m-i+1}} > 1 - \tfrac{1}{q^{m-n}(q-1)}.$$

Combining the above observations, we have

$$\Pr[\mathsf{rank}(\mathbf{A}) = n] = \Pr\Big[\textstyle\bigwedge_{i=0}^{d-1} \mathsf{rank}(\mathbf{A}^{(i)}) = n\Big] > (1 - \tfrac{1}{q^{m-n+1}})^d > 1 - \tfrac{d}{q^{m-n}(q-1)}. \qquad \square$$

C.2 Proof of Theorem 1

Theorem 1 (IND-CCA Security of TD_1) *Let $r, a \in \mathbb{N}$ such that $s := m - r - a \ge 2$, and let $\sigma_{k_0}, \sigma_{\mathsf{MLWE}} > 0$ and let $\epsilon, \epsilon' \in (0, \frac{1}{2})$. If*

- *Π has zero-knowledge and OTSS, and $\mathcal{H}$ is collision-resistant,*
- *the $\mathsf{SA\text{-}HMLWE}^{1,\sigma_1,\mathcal{S}}_{q,m,n,\sigma_0}$-assumption holds,[14] where $\mathcal{S} := \{\mathbf{k} \in \mathcal{R}^m : s_1(\mathsf{Rot}(\mathbf{k})) \le B_\mathcal{S}\}$ and $B_\mathcal{S} > 0$ is the bound of $s_1(\mathsf{Rot}(\mathbf{k}))$ for $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sqrt{C}\sigma_k}$.*
- *the s-$\mathsf{MLWE}_{q,r,a,\sigma_{\mathsf{MLWE}}}$ assumption holds,*

[14] By Lemma 1, the hardness of $\mathsf{SA\text{-}HMLWE}^{1,\sigma_1,\mathcal{S}}_{q,m,n,\sigma_0}$ can be reduced to $\mathsf{MLWE}_{\mathcal{R},q,m,n,\sigma'}$, where $\sigma' > 0$ satisfies $\frac{1}{\sigma_0^2} + \frac{B_\mathcal{S}^2}{\sigma_1^2} \le \frac{1}{\sigma'^2 + 2\eta_\epsilon^2(\mathcal{R}^m)}$ and $\sigma' > \sqrt{2}\cdot\eta_\epsilon(\mathcal{R}^m)$.

– $r = n + O(\log \lambda)$, $\sigma_k \geq \max(\sqrt{2} \cdot B_{\mathbf{D}} \cdot \sigma_{k_0}, \sqrt{2} \cdot \eta_{\epsilon'}(\mathcal{R}^m))$, $\sigma_{k_0} \geq \eta_{\epsilon'}(\mathcal{R}^s)$ *and* $\frac{1}{\sigma_{k_0}^2} + \frac{\ell d B'^2 B_{\mathbf{D}}^2}{\sigma_y^2} \leq \frac{1}{\sigma^2}$ *for* $\sigma \geq \frac{8}{\sqrt{\pi}} \cdot q^{\frac{1}{s}} \sqrt{d \ln(2sd(1+1/\epsilon))}$. *Here* $B_{\mathbf{D}} > 0$ *is the bound of* $s_1(\mathsf{Rot}(\mathbf{D}))$ *for* $\mathbf{D}^\top := [-\mathbf{R}^\top | \mathbf{I}_s | - \mathbf{E}^\top] \in \mathcal{R}^{s \times m}$, *where* $(\mathbf{R}, \mathbf{E}) \leftarrow D_{\mathcal{R},\sigma_{\mathsf{MLWE}}}^{a \times s} \times D_{\mathcal{R},\sigma_{\mathsf{MLWE}}}^{r \times s}$.

then the scheme TD_1 *in Fig. 4 is* IND-CCA *secure.*

More precisely, for any PPT adversary $\mathcal{A}$ *that makes at most* ℓ *queries to* $\mathcal{O}_{\mathrm{DEC}}$, *there exists PPT* $\mathcal{B}_i$ *for* $i \in [5]$ *s.t.*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{TD}_1,\mathcal{A}}(\lambda) \leq\ & \mathsf{Adv}^{\mathsf{zk}}_{\Pi,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{SA\text{-}HMLWE}}_{\mathsf{para}_1,\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{(m-r-a)\text{-}\mathsf{MLWE}}_{\mathsf{para}_2} \\ & + \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_4}(\lambda) + \ell \cdot \mathsf{Adv}^{\mathsf{OTSS}}_{\Pi,\mathcal{B}_5}(\lambda) + 8C \cdot \epsilon' + \tfrac{d}{q^{r-n}(q-1)} + 2\epsilon + 2^{-d} + dq^{-s},\end{aligned}$$

where $\mathsf{para}_1 := ((q, m, n, \sigma_0), (1, \sigma_1, \mathcal{S}))$ *and* $\mathsf{para}_2 := (q, r, a, \sigma_{\mathsf{MLWE}})$.

Proof of Theorem 1. Let $\mathcal{A}$ be an adversary against the IND-CCA security of TD_1. We use games G_0-G_{10} to show the advantage $\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{TD}_1,\mathcal{A}}(\lambda)$ is negligible, where G_0 is the original IND-CCA experiment and $\mathcal{A}$ has no advantage in G_{10}.

Game G_0: This is the original IND-CCA experiment $\mathsf{Exp}^{\mathsf{CCA}}_{\mathsf{TD}_1,\mathcal{A}}$ (cf. Fig. 2 with solid boxes). By definition, we have $\mathsf{Adv}^{\mathsf{CCA}}_{\mathsf{TD}_1,\mathcal{A}}(\lambda) = |\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}|$.

Let $\mathsf{pp} = (\mathbf{A}, \mathsf{crs}, \mathsf{H})$. The challenger invokes $(\mathbf{k}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_N)) \leftarrow_{\$} \mathsf{Share}(N, t, D_{\mathcal{R}^m,\sigma_k})$, sets $\mathsf{pk} := \mathbf{Ak}$ and returns $(\mathsf{pp}, \mathsf{pk})$ to $\mathcal{A}$. The challenger initializes $\mathsf{Cor} := \emptyset, \mathsf{Hon} := [N]$, $\mathsf{R}[\cdot,\cdot] := \emptyset$, $\mathcal{Q}_{\mathrm{DEC}} := \emptyset$. Then it answers $\mathcal{A}$'s $\mathcal{O}_{\mathrm{COR}}$, $\mathcal{O}_{\mathrm{ENC}}$, $\mathcal{O}_{\mathrm{DEC}}$ queries adaptively and generates the challenge ciphertext ct^* as follows.

– For corruption query $\mathcal{O}_{\mathrm{COR}}(i)$, the challenger adds i to Cor and returns sk_i and all ephemeral randomness previously used by user i, i.e. $\mathsf{R}[i,\cdot]$.
– For partial decryption query $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct} = (\mathbf{c}, d, \pi))$, the challenger adds i to $\mathcal{Q}_{\mathrm{DEC}}$ if $\mathsf{ct} = \mathsf{ct}^*$ and computes partial decryption as follows.
  - If $\mathsf{Vfy}(\mathsf{crs}, \mathbf{c}, \tau, \pi) = 0$ with $\tau := \mathsf{H}(\mathbf{c}, d)$, the challenger returns $\perp$.
  - For each $\mathbf{k}_I \in \mathsf{sk}_i$, the challenger computes $h_{i,I} = \mathbf{c}^\top \mathbf{k}_I + y_{i,I}$ with $y_{i,I} \leftarrow D_{\mathcal{R},\sigma_y}$ and updates $\mathsf{R}[i, \mathsf{ct}] := \mathsf{R}[i, \mathsf{ct}] \cup \{\{y_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i}\}$. Then it returns $h_i := \{h_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i}$ to $\mathcal{A}$.
– On receiving (μ_0, μ_1), the challenger chooses $\beta \leftarrow_{\$} \{0,1\}$ and generates the challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}^*, d^*, \pi^*)$ as an encryption of μ_β as follows:
  - Sample $\mathbf{s}^* \leftarrow D_{\mathcal{R}^n,\sigma_0}$, $\mathbf{e}^* \leftarrow D_{\mathcal{R}^m,\sigma_0}$ and compute ${\mathbf{c}^*}^\top := {\mathbf{s}^*}^\top \mathbf{A} + {\mathbf{e}^*}^\top$.
  - Sample $e'^* \leftarrow D_{\mathcal{R},\sigma_1}$ and compute $d^* := {\mathbf{s}^*}^\top \mathsf{pk} + e'^* + \lfloor \frac{q}{p} \rfloor \cdot \mu_\beta \in \mathcal{R}_q$.
  - Compute $\tau^* := \mathsf{H}(\mathbf{c}^*, d^*)$ and generate a proof $\pi^* \leftarrow_{\$} \mathsf{Prv}(\mathsf{crs}, \mathbf{c}^*, \tau^*, (\mathbf{s}^*, \mathbf{e}^*))$.

Recall that to avoid trivial attacks, we require $|\mathcal{Q}_{\mathrm{DEC}} \cup \mathsf{Cor}| \leq t - 1$.

Game G_1: It is the same as G_0 except that when generating the challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}^*, d^*, \pi^*)$, the challenger computes $d^* := {\mathbf{c}^*}^\top \mathbf{k} + e'^* - {\mathbf{e}^*}^\top \mathbf{k} + \lfloor \frac{q}{p} \rfloor \cdot \mu_\beta$ instead of $d^* := {\mathbf{s}^*}^\top \mathsf{pk} + e'^* + \lfloor \frac{q}{p} \rfloor \cdot \mu_\beta$. Since $\mathsf{pk} = \mathbf{Ak}$ and ${\mathbf{c}^*}^\top = {\mathbf{s}^*}^\top \mathbf{A} + {\mathbf{e}^*}^\top$, the

d^* in G_0 equals $\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}+e'^*+\lfloor\frac{q}{p}\rfloor\cdot\mu_\beta = (\mathbf{c}^{*\top}-\mathbf{e}^{*\top})\mathbf{k}+e'^*+\lfloor\frac{q}{p}\rfloor\cdot\mu_\beta$, which is just the d^* in G_1. So the change is conceptual and we have $\Pr[\mathsf{G}_0=1]=\Pr[\mathsf{G}_1=1]$.

Game G_2: It is the same as G_1 except that when generating the challenge ciphertext $\mathsf{ct}^*=(\mathbf{c}^*,d^*,\pi^*)$, it invokes $\pi^* \leftarrow_\$ \mathsf{Sim}(\mathsf{crs},\mathbf{c}^*,\tau^*)$ instead of $\pi^* \leftarrow_\$ \mathsf{Prv}(\mathsf{crs},\mathbf{c}^*,\tau^*,(\mathbf{s}^*,\mathbf{e}^*))$. By zero-knowledge of Π, $|\Pr[\mathsf{G}_1=1]-\Pr[\mathsf{G}_2=1]| \le \mathsf{Adv}^{\mathsf{zk}}_{\Pi,\mathcal{B}_1}(\lambda)$.

Game G_3: It is the same as G_2 except that when generating the challenge ciphertext $\mathsf{ct}^*=(\mathbf{c}^*,d^*,\pi^*)$, the challenger samples $\mathbf{c}^{*\top} \leftarrow_\$ \mathcal{R}_q^{1\times m}$ uniformly, instead of computing $\mathbf{c}^{*\top}:=\mathbf{s}^{*\top}\mathbf{A}+\mathbf{e}^{*\top}$.

By the SA-HMLWE assumption (cf. Def. 3) , the $\mathbf{c}^{*\top}=(\mathbf{s}^{*\top}\mathbf{A}+\mathbf{e}^{*\top})$ in G_2 is computationally indistinguishable from the $\mathbf{c}^{*\top} \leftarrow_\$ \mathcal{R}_q^{1\times m}$ in G_3, even conditioned on the $(e^{*\prime}-\mathbf{k}^\top\mathbf{e}^*)$ involved in the computation of d^*. Formally, we have the following claim.

Claim 1. $|\Pr[\mathsf{G}_2=1]-\Pr[\mathsf{G}_3=1]| \le \mathsf{Adv}^{\mathsf{SA\text{-}HMLWE}}_{\mathsf{para}_1,\mathcal{B}_2}(\lambda)+4C\cdot\epsilon'$, *where* $\mathsf{para}_1 := ((q,m,n,\ \sigma_0),(1,\sigma_1,\mathcal{S}))$.

Proof. To bound the difference between G_3 and G_2, we construct an adversary $\mathcal{B}_2$ against the $\mathsf{SA\text{-}HMLWE}^{1,\sigma_1,\mathcal{S}}_{q,m,n,\sigma_0}$ assumption (cf. Def. 3). $\mathcal{B}_2$ first invokes $(\mathbf{k},(\mathsf{sk}_1,\ldots,\mathsf{sk}_N)) \leftarrow_\$ \mathsf{RSS}\ (N,t,D_{\mathcal{R}^m,\sigma_k})$ and sets $\mathsf{pk}:=\mathbf{A}\mathbf{k}$. Let $\mathcal{I}=\{I \mid I\subseteq[N],|I|=N-t+1\}=\{I_1,\ldots,I_C\}$. For $j\in[C]$, $\mathbf{k}_{I_j} \leftarrow D_{\mathcal{R}^m,\sigma_k}$, so that the distribution of $\mathbf{k}:=\sum_{j\in[C]}\mathbf{k}_{I_j}$ is statistically close to $D_{\mathcal{R}^m,\sqrt{C}\sigma_k}$ by Lemma 5, as long as $\sigma_k \ge \sqrt{2}\cdot\eta_{\epsilon'}(\mathcal{R}^m)$. Consequently, $-\mathbf{k}\in\mathcal{S}$ holds with probability at least $1-4C\cdot\epsilon'=1-\mathsf{negl}(\lambda)$.

On receiving $\mathbf{A}$ from the challenger of SA-HMLWE, $\mathcal{B}_2$ returns $-\mathbf{k}$ to it. Then, $\mathcal{B}_2$ is given $(\mathbf{A},\mathbf{u}^\top,-\mathbf{k},h)$, and aims to decide whether $\mathbf{u}^\top=\mathbf{s}^\top\mathbf{A}+\mathbf{e}^\top$ or $\mathbf{u}\leftarrow_\$\mathcal{R}_q^m$, where $\mathbf{s}\leftarrow D_{\mathcal{R}^n,\sigma_0},\mathbf{e}\leftarrow D_{\mathcal{R}^m,\sigma_0}$ and $\mathbf{e}'\leftarrow D_{\mathcal{R},\sigma_1}$. $\mathcal{B}_2$ sends $(\mathsf{pp},\mathsf{pk})$ and simulates $\mathsf{G}_2/\mathsf{G}_3$ for $\mathcal{A}$ as follows:

- $\mathcal{B}_2$ can answer $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$ queries on its own with the help of pk and $\{\mathsf{sk}_i\}_{i\in[N]}$.
- On receiving (μ_0,μ_1), $\mathcal{B}_2$ randomly chooses $\beta\leftarrow_\$\{0,1\}$ and generates the challenge ciphertext $\mathsf{ct}^*=(\mathbf{c}^*,d^*,\pi^*)$ for $\mathcal{A}$ as follows:
  - Instead of computing $\mathbf{c}^{*\top}:=\mathbf{s}^{*\top}\mathbf{A}+\mathbf{e}^{*\top}$ with $\mathbf{s}^*\leftarrow D_{\mathcal{R}_q^n,\sigma_0}$ and $\mathbf{e}^*\leftarrow D_{\mathcal{R}^m,\sigma_0}$ as in G_2, $\mathcal{B}_2$ directly sets $\mathbf{c}^*:=\mathbf{u}$ using the $\mathbf{u}$ in $\mathcal{B}_2$'s input.
  - Instead of sampling $e'^*\leftarrow D_{\mathcal{R},\sigma_1}$ and using $e'^*-\mathbf{k}^\top\mathbf{e}^*$ to compute d^* as in G_2, $\mathcal{B}$ now computes $d^*:=\mathbf{c}^{*\top}\mathbf{k}+h+\lfloor\frac{q}{p}\rfloor\cdot\mu_\beta$ using h in $\mathcal{B}_2$'s input.
  - Compute $\tau^*:=\mathsf{H}(\mathbf{c}^*,d^*)$, and invoke $\pi^*\leftarrow_\$\mathsf{Sim}(\mathsf{crs},\mathbf{c}^*,\tau^*)$.
- Finally, $\mathcal{B}_2$ receives a bit β' from $\mathcal{A}$ and returns $\beta=\beta'$.

Overall, if $\mathbf{u}^\top=\mathbf{s}^{*\top}\mathbf{A}+\mathbf{e}^{*\top}$, then $\mathcal{B}_2$ implicitly sets $\mathbf{s}^*:=\mathbf{s},\mathbf{e}^*:=\mathbf{e},e'^*:=e'$, as h is expected to be $e'-\mathbf{k}^\top\mathbf{e}$. In this case, $\mathcal{B}_2$ simulates G_2 for $\mathcal{A}$. Otherwise if $\mathbf{u}\leftarrow_\$\mathcal{R}_q^m$, $\mathcal{B}_2$ simulates G_3 for $\mathcal{A}$. Thus, $|\Pr[\mathsf{G}_2=1]-\Pr[\mathsf{G}_3=1]| \le \mathsf{Adv}^{\mathsf{SA\text{-}HMLWE}}_{\mathsf{para}_1,\mathcal{B}_2}(\lambda)+4C\cdot\epsilon' = \mathsf{Adv}^{\mathsf{SA\text{-}HMLWE}}_{\mathsf{para}_1,\mathcal{B}_2}(\lambda)+\mathsf{negl}(\lambda)$. ∎

Game G_4: It is the same as G_3, except for the generation of $\mathbf{A} \in \mathcal{R}_q^{n\times m}$ in the Setup algorithm. More precisely, instead of sampling $\mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n\times m}$, the challenger samples $\mathbf{A}' \leftarrow_{\$} \mathcal{R}_q^{r\times(m-r)}$ and $\mathbf{U} \leftarrow_{\$} \mathcal{R}_q^{n\times r}$, and sets $\mathbf{A} \coloneqq \mathbf{U}\left[\mathbf{A}' \middle| \mathbf{I}_r\right]$.

Claim 2. $|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_4 = 1]| \le \frac{d}{q^{r-n}(q-1)} = \mathsf{negl}(\lambda)$.

Proof. We can prove that the distribution of $\mathbf{A} \in \mathcal{R}_q^{n\times m}$ is identically distributed in G_3 and G_4 unless the following event $\mathsf{Bad}_{\mathsf{rk}}$ happens: $\mathbf{U}$ is not a full-rank matrix. For every full-rank matrix $\mathbf{U}$, the map $f_{\mathbf{U}}(\mathbf{x}) = \mathbf{U}\mathbf{x}$ $\mathcal{R}_q^r \longmapsto \mathcal{R}_q^n$ is a surjective homomorphism. It follows that, if $\mathbf{A}' \leftarrow_{\$} \mathcal{R}_q^{r\times(m-r)}$, then $\mathbf{U}\mathbf{A}'$ is also uniformly distributed over $\mathcal{R}_q^{n\times(m-r)}$ and independent on $\mathbf{U}$. Thus, $\mathbf{U}\left[\mathbf{A}' \middle| \mathbf{I}_r\right] = \left[\mathbf{U}\mathbf{A}' \middle| \mathbf{U}\right]$, has exactly the same distribution as $\mathbf{A}$ in G_3, conditioned on $\neg\mathsf{Bad}_{\mathrm{rk}}$. By Lemma 6, instantiated with $r = n + O(\log\lambda)$, $\Pr[\mathsf{Bad}_{\mathrm{rk}}] < \frac{d}{q^{r-n}(q-1)} = \mathsf{negl}(\lambda)$. Therefore, $\left|\Pr[\mathsf{G}_3 = 1] - \Pr[\mathsf{G}_4 = 1]\right| \le \Pr[\mathsf{Bad}_{\mathrm{rk}}] = \frac{d}{q^{r-n}(q-1)} = \mathsf{negl}(\lambda)$. ∎

Game G_5: It is the same as G_4, except for the generation of $\mathbf{A} = \mathbf{U}\left[\mathbf{A}' \middle| \mathbf{I}_r\right] \in \mathcal{R}_q^{n\times m}$ in the Setup algorithm. More precisely, instead of sampling $\mathbf{A}' \leftarrow_{\$} \mathcal{R}_q^{r\times(m-r)}$, the challenger samples $\mathbf{B} \leftarrow_{\$} \mathcal{R}_q^{r\times a}$, $\mathbf{R} \leftarrow D_{\mathcal{R}_q,\sigma_{\mathsf{MLWE}}}^{a\times(m-r-a)}$, $\mathbf{E} \leftarrow D_{\mathcal{R}_q,\sigma_{\mathsf{MLWE}}}^{r\times(m-r-a)}$ and sets $\mathbf{A}' \coloneqq \left[\mathbf{B} \middle| \mathbf{B}\mathbf{R} + \mathbf{E}\right]$. Then, the challenger samples $\mathbf{U} \leftarrow_{\$} \mathcal{R}_q^{n\times r}$ and sets $\mathbf{A} \coloneqq \mathbf{U}\left[\mathbf{A}' \middle| \mathbf{I}_r\right]$. The $\mathbf{A}' \leftarrow_{\$} \mathcal{R}_q^{r\times(m-r)}$ in G_5 is computationally indistinguishable from $\mathbf{A}$ in G_4 by Multi-Secret MLWE assumption (cf. Def. 1).

Claim 3. $|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_5 = 1]| \le \mathsf{Adv}^{(m-r-a)\text{-}\mathsf{MLWE}}_{q,r,a,\sigma_{\mathsf{MLWE}},\mathcal{B}_3}(\lambda)$.

Now we observe that the matrix $\mathbf{D} = \begin{bmatrix} -\mathbf{R} \\ \mathbf{I}_{m-r-a} \\ -\mathbf{E} \end{bmatrix}$ satisfies $\mathbf{A}\mathbf{D} = \mathbf{0}$.

Game G_6: It is the same as G_5, except that when replying to $\mathcal{A}$'s partial decryption queries $\mathcal{O}_{\mathrm{Dec}}(i, \mathsf{ct} = (\mathbf{c}, d, \pi))$, the challenger introduces an additional rejection rule: If $\mathsf{ct} \neq \mathsf{ct}^* \wedge (\mathbf{c}, \tau, \pi) = (\mathbf{c}^*, \tau^*, \pi^*)$, return $\perp$ directly.

Since $\mathsf{ct} = (\mathbf{c}, d, \pi) \neq (\mathbf{c}^*, d^*, \pi^*) = \mathsf{ct}^* \wedge (\mathbf{c}, \tau, \pi) = (\mathbf{c}^*, \tau^*, \pi^*)$ implies that $d \neq d^*$ but $\tau = \mathsf{H}(\mathbf{c}, d) = \mathsf{H}(\mathbf{c}^*, d^*) = \tau^*$, it yields a collision of H. So $\left|\Pr[\mathsf{G}_5 = 1] - \Pr[\mathsf{G}_6 = 1]\right| \le \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_4}(\lambda)$ by the collision-resistance of $\mathcal{H}$.

Game G_7: It is the same as G_6, except that when replying to $\mathcal{A}$'s partial decryption queries $\mathcal{O}_{\mathrm{Dec}}(i, \mathsf{ct} = (\mathbf{c}, d, \pi))$, the challenger introduces an additional rejection rule: If $\mathsf{ct} \neq \mathsf{ct}^* \wedge \mathbf{c} \notin \mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$, return $\perp$ directly.

Intuitively, The games G_6 and G_7 are identical unless $\mathcal{A}$ submits a partial decryption query $\mathcal{O}_{\mathrm{Dec}}(i, \mathsf{ct} = (\mathbf{c}, d, \pi))$ satisfies

$$\mathsf{ct} \neq \mathsf{ct}^* \wedge \mathbf{c} \notin \mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})} \wedge (\mathbf{c}, \tau, \pi) \neq (\mathbf{c}^*, \tau^*, \pi^*) \wedge \mathsf{Vfy}(\mathsf{crs}, \mathbf{c}, \tau, \pi) = 1.$$

By guessing the decryption query on which Bad occurs among the at most ℓ queries, one can construct a PPT adversary $\mathcal{B}_5$ against the OTSS security of Π. This yields the following Claim.

Claim 4. $\big|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]\big| \leq \ell \cdot \mathsf{Adv}^{\mathsf{OTSS}}_{\Pi,\mathcal{B}_5}(\lambda)$.

Proof. For each $j \in [\ell]$, let Bad_j denote the event that the j-th decryption query contains a ciphertext $\mathsf{ct}_j = (\mathbf{c}_j, d_j, \pi_j)$ with associated tag τ_j s.t.

$$\mathsf{ct}_j \neq \mathsf{ct}^* \ \wedge \ \mathbf{c}_j \notin \mathcal{L}^{(\mathsf{MLWE})}_{\mathbf{A}} \ \wedge \ (\mathbf{c}_j, \tau_j, \pi_j) \neq (\mathbf{c}^*, \tau^*, \pi^*) \ \wedge \ \mathsf{Vfy}(\mathsf{crs}, \mathbf{c}_j, \tau_j, \pi_j) = 1.$$

Let $\mathsf{Bad} := \bigvee_{j=1}^{\ell} \mathsf{Bad}_j$. The games G_6 and G_7 are the same unless Bad occurs. Therefore,$\big|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]\big| \leq \Pr[\mathsf{Bad}]$.

We next construct an adversary $\mathcal{B}_5$ against the OTSS security of Π. At the beginning of the experiment, $\mathcal{B}_5$ samples $j \leftarrow_{\$} [\ell]$ and runs $\mathcal{A}$ while simulating G_6. When generating $\mathsf{ct}^* = (\mathbf{c}^*, d, \pi^*)$, it queries $\mathcal{O}_{\mathrm{SIM}}(\mathbf{c}^*, \tau^*)$ to obtain π^*. Upon receiving the j-th decryption query, $\mathcal{B}_5$ outputs $(\mathbf{c}_j, \tau_j, \pi_j)$ to the OTSS challenger. If $\mathcal{A}$ makes less than j decryption queries, $\mathcal{B}_5$ returns a random tuple and aborts the game.

Hence, $\mathcal{B}_5$ wins the OTSS experiment when Bad_j occurs. Since j is chosen uniformly, we have

$$\mathsf{Adv}^{\mathsf{OTSS}}_{\Pi,\mathcal{B}_5}(\lambda) \geq \frac{\Pr[\mathsf{Bad}]}{\ell}.$$

Consequently, $\big|\Pr[\mathsf{G}_6 = 1] - \Pr[\mathsf{G}_7 = 1]\big| \leq \ell \cdot \mathsf{Adv}^{\mathsf{OTSS}}_{\Pi,\mathcal{B}_5}(\lambda)$ and Claim 4 follows. ∎

Game G_8: It is the same as G_7, except that when replying to $\mathcal{A}$'s partial decryption queries $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct} = (\mathbf{c}, d, \pi))$, the challenger checks all the additional rejection rules and returns $\perp$ if it occurs. For $\mathsf{ct} = \mathsf{ct}^*$, the challenger computes the partial decryption using sk_i. For $\mathsf{ct} \neq \mathsf{ct}^*$, the challenger computes the partial decryptions as follows: obtain $(\mathbf{s}, \mathbf{e})$ using brute force (since $\mathbf{c} \in \mathcal{L}^{(\mathsf{MLWE})}_{\mathbf{A}}$) and compute $h_i = \{h_{i,I} = (\mathbf{s}^\top \mathbf{A}\mathbf{k}_I + \mathbf{e}^\top \mathbf{k}_I + y_{i,I})\}_{\mathbf{k}_I \in \mathsf{sk}_i}$.

Since $\mathbf{c}^\top = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$, the h_i in G_7 equals $\{(\mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top)\mathbf{k}_I + y_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i} = \{(\mathbf{s}^\top \mathbf{A}\mathbf{k}_I + \mathbf{e}^\top \mathbf{k}_I + y_{i,I})\}_{\mathbf{k}_I \in \mathsf{sk}_i}$, which is just the h_i in G_8. So the change is conceptual and we have $\Pr[\mathsf{G}_7 = 1] = \Pr[\mathsf{G}_8 = 1]$.

Game G_9: It is the same as G_8, except the challenger now generates all the secret keys in an equivalent way as follows.

Let $\mathbf{\Gamma} := \mathsf{Rot}(\mathbf{D}) \in \mathbb{Z}^{md \times sd}$, $\mathbf{\Sigma} := \sigma_k^2 \mathbf{I}_{md} - \sigma_{k_0}^2 \mathbf{\Gamma}\mathbf{\Gamma}^\top$ and $\sigma_{k_0} > 0$. More precisely, for each $I \in \mathcal{I}$, the challenger no longer samples $\mathbf{k}_I \leftarrow D_{\mathcal{R}^m, \sigma_k}$. Instead, it samples $\mathbf{k}_I^{(0)} \leftarrow D_{\mathcal{R}^s, \sigma_{k_0}}$, $\mathbf{k}_I^{(1)} \leftarrow D_{\mathcal{R}^m, \sqrt{\mathbf{\Sigma}}}$ and compute $\mathbf{k}_I := \mathbf{D}\mathbf{k}_I^{(0)} + \mathbf{k}_I^{(1)}$. According to Lemma 7, if $\sigma_k > \sqrt{2} \cdot \sigma_{k_0} \cdot s_1(\mathbf{\Gamma})$ and $\sigma_{k_0} \geq \eta_{\epsilon'}(\mathbb{Z}^{sd})$, the distribution each $\mathbf{k}_I$ is statistically close to the distribution of $\mathbf{D}\mathbf{k}_I^{(0)} + \mathbf{k}_I^{(1)}$ with statistical distance at most $4\epsilon'$. By an union bound of $|\mathcal{I}| = C$, we have $|\Pr[\mathsf{G}_8 = 1] - \Pr[\mathsf{G}_9 = 1]| \leq C \cdot \frac{2\epsilon'}{1-\epsilon'} \leq 4C\epsilon'$.

Game G_{10}: It is the same as G_9 except that when generating the challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}^*, d^*, \pi^*)$, the challenger samples $d^* \leftarrow_{\$} \mathcal{R}_q$ uniformly, instead of computing $d^* := \mathbf{c}^{*\top}\mathbf{k} + e'^* - \mathbf{e}^{*\top}\mathbf{k} + \lfloor \frac{q}{p} \rfloor \cdot \mu_\beta$.

Claim 5. $|\Pr[\mathsf{G}_9 = 1] - \Pr[\mathsf{G}_{10} = 1]| \leq 2\epsilon + 2^{-d} + dq^{-s}$.

Proof. Let $\mathcal{I} := \{I \mid I \subseteq [N], |I| = N - t + 1\}$. Below we first analyze the information about $\mathsf{sk} = \mathbf{k}$ and its shares $\mathsf{sk}_i = \{\mathbf{k}_I \,|I \in \mathcal{I}, i \in I\}$ $(i \in [N])$ that might be leaked to $\mathcal{A}$ in G_9 and G_{10}.

- Firstly, $\mathcal{A}$ obtains $\mathsf{pk} = \mathbf{Ak}$.
- In the challenge phase, $\mathcal{A}$ obtains the challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}^*, d^*)$.
- Through $\mathcal{O}_{\mathrm{COR}}(i)$ queries, $\mathcal{A}$ obtains $\mathsf{sk}_i = \{\mathbf{k}_I \mid i \in I, I \in \mathcal{I}\}$ for $i \in \mathsf{Cor}$ and all the randomness $\{y_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i}$ used in $\mathcal{O}_{\mathrm{DEC}}(i, \cdot)$.
- Through $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct} = (\mathbf{c}, d, \pi))$ queries, $\mathcal{A}$ can learn partial decryption $h_i = \{h_{i,I} = (\mathbf{s}^\top \mathbf{Ak}_I + \mathbf{e}^\top \mathbf{k}_I + y_{i,I})\}_{\mathbf{k}_I \in \mathsf{sk}_i}$ of arbitrary ciphertext $\mathsf{ct} = (\mathbf{c}, d, \pi)$. There are two cases: (i) If $\mathsf{ct} = \mathsf{ct}^*$, then i is added to $\mathcal{Q}_{\mathrm{DEC}}$, and the information of $\mathbf{k}$ leaked by partial decryption h_i is overestimated by $\mathsf{sk}_i = \{\mathbf{k}_I \mid i \in I, I \in \mathcal{I}\}$. (ii) If $\mathsf{ct} \neq \mathsf{ct}^*$, by the changes in G_8, $\mathcal{A}$ learns $h_i = \{h_{i,I} = (\mathbf{s}^\top \mathbf{Ak}_I + \mathbf{e}^\top \mathbf{k}_I + y_{i,I})\}_{\mathbf{k}_I \in \mathsf{sk}_i}$ only when $\mathbf{c} = \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top \in \mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$. So, the information of $\mathbf{k}$ leaked by h_i is overestimated by $\{\mathbf{Ak}_I\}_{\mathbf{k}_I \in \mathsf{sk}_i}$ and linear relations of the form $\{\mathbf{e}^\top \mathbf{k}_I + y_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i}$, where $\mathbf{e}^\top$ in $\mathbf{c}$ is adaptively chosen by $\mathcal{A}$ depending on its view up to now. Note that $\{\mathbf{k}_I \mid \mathbf{k}_I \in \mathsf{sk}_i\} = \{\mathbf{k}_I \mid i \in I, I \in \mathcal{I}\}$.

Overall, $\mathcal{A}$ gets $\mathbf{Ak}$ and $\{\mathbf{Ak}_I\}_{I \in \mathcal{I}}$, might learn $\{\mathsf{sk}_i\}_{i \in \mathcal{Q}_{\mathrm{DEC}} \cup \mathsf{Cor}}$ (through $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$ in case (i)), and learn at most ℓ linear relations of the form $\{\mathbf{e}^\top \mathbf{k}_I + y_{i,I}\}_{i \in I}$ for each $\mathbf{k}_I$ ($\mathcal{O}_{\mathrm{DEC}}$ in case (ii)). Here ℓ denotes the number of $\mathcal{O}_{\mathrm{DEC}}$ queries made by $\mathcal{A}$. Given that $|\mathcal{Q}_{\mathrm{DEC}} \cup \mathsf{Cor}| \leq t - 1$ for $\mathcal{O}_{\mathrm{DEC}}$ in case (i) (to avoid trivial attacks), there must exist $\tilde{I} \in \mathcal{I}$ such that $\mathbf{k}_{\tilde{I}} \notin \bigcup_{i \in \mathcal{Q}_{\mathrm{DEC}} \cup \mathsf{Cor}} \mathsf{sk}_i$. The information about $\mathbf{k}_{\tilde{I}}$ learned by $\mathcal{A}$ is limited to

$$(\mathbf{Ak}_{\tilde{I}}, \{\mathbf{e}_j^\top \mathbf{k}_{\tilde{I}} + y_{j,\tilde{I}}\}_{j \in [\ell]}),$$

where $\mathbf{e}_j$ may depend on $(\mathbf{A}, \mathsf{ct}^* = (\mathbf{c}^*, d^*, \pi^*), \mathbf{Ak}_{\tilde{I}}, \{\mathbf{e}_j, \mathbf{e}_j^\top \mathbf{k}_{\tilde{I}} + y_{j,\tilde{I}}\}_{j \in [i-1]})$ and other variables independent of $\mathbf{k}_{\tilde{I}}$ with $\|\mathbf{e}_j\|_2 \leq \|(\mathbf{s}_j, \mathbf{e}_j)\|_2 \leq B'$ and $y_{j,\tilde{I}} \leftarrow D_{\mathcal{R},\sigma_y}$.

Recall that in G_9, we have that $\mathbf{k}_{\tilde{I}} = \mathbf{Dk}_{\tilde{I}}^{(0)} + \mathbf{k}_{\tilde{I}}^{(1)}$, where $\mathbf{k}_{\tilde{I}}^{(0)} \leftarrow D_{\mathcal{R}^s,\sigma_{k_0}}$ and $\mathbf{k}_{\tilde{I}}^{(1)} \leftarrow D_{\mathcal{R}^m,\sqrt{\mathbf{\Sigma}}}$. Due to the fact that $\mathbf{AD} = \mathbf{0}$ and the independency between $\mathbf{k}_{\tilde{I}}^{(1)}$ and $\mathbf{k}_{\tilde{I}}^{(0)}$, $\mathbf{Ak}_{\tilde{I}} = \mathbf{A}(\mathbf{Dk}_{\tilde{I}}^{(0)} + \mathbf{k}_{\tilde{I}}^{(1)}) = \mathbf{Ak}_{\tilde{I}}^{(1)}$ does not leak any information of $\mathbf{k}_{\tilde{I}}^{(0)}$. However, $\mathbf{e}_j^\top \mathbf{k}_{\tilde{I}} + y_{j,\tilde{I}} = \mathbf{e}_j^\top (\mathbf{Dk}_{\tilde{I}}^{(0)} + \mathbf{k}_{\tilde{I}}^{(1)}) + y_{j,\tilde{I}} = \mathbf{e}_j^\top \mathbf{Dk}_{\tilde{I}}^{(0)} + y_{j,\tilde{I}} + \mathbf{e}_j^\top \mathbf{k}_{\tilde{I}}^{(1)}$ might leak $\mathbf{e}_j^\top \mathbf{Dk}_{\tilde{I}}^{(0)} + y_{j,\tilde{I}}$ about $\mathbf{k}_{\tilde{I}}^{(0)}$, where $y_{j,\tilde{I}} \leftarrow D_{\mathcal{R},\sigma_y}$. The information about $\mathbf{k}_{\tilde{I}}^{(0)}$ learned by $\mathcal{A}$ is limited to

$$\{\mathbf{e}_j^\top \mathbf{Dk}_{\tilde{I}}^{(0)} + y_{j,\tilde{I}}\}_{j \in [\ell]},$$

where $\mathbf{e}_j$ may depend on $(\mathsf{ct}^* = (\mathbf{c}^*, d^*, \pi^*), \{\mathbf{e}_j, \mathbf{e}_j^\top \mathbf{k}_{\tilde{I}} + y_{j,\tilde{I}}\}_{j \in [i-1]})$ and other variables independent of $\mathbf{k}_{\tilde{I}}^{(0)}$ including $\mathbf{A}, \mathsf{crs}, \mathsf{H}, \{\mathbf{Ak}_I\}_{I \in \mathcal{I}, I \neq \tilde{I}}, \mathbf{Ak}_{\tilde{I}}^{(1)}, (\{\mathbf{k}_I\}_{I \neq \tilde{I}},$

$\{y_{i,I}\}_{I\neq\tilde{I}}$). To bound $s_1(\mathsf{Rot}(\mathbf{e}_j^\top\mathbf{D}))$, we note that $s_1(\mathsf{Rot}(\mathbf{e}_j^\top\cdot\mathbf{D}))\leq s_1(\mathsf{Rot}(\mathbf{e}_j^\top)\cdot\mathsf{Rot}(\mathbf{D}))\leq s_1(\mathsf{Rot}(\mathbf{e}_j^\top))\cdot s_1(\mathsf{Rot}(\mathbf{D}))\leq s_1(\mathsf{Rot}(\mathbf{e}_j^\top))\cdot B_{\mathbf{D}}$, where $s_1(\mathsf{Rot}(\mathbf{e}_j^\top))\leq\sqrt{d}\|\mathbf{e}_j\|_2\leq\sqrt{d}B'$. Therefore, we have $s_1(\mathsf{Rot}(\mathbf{e}_j^\top\mathbf{D}))\leq\sqrt{d}B'\cdot B_{\mathbf{D}}$. Recall that in G_9, $\mathsf{ct}^*=(\mathbf{c}^*,d^*,\pi^*)$, where $\mathbf{c}^*\leftarrow_{\$}\mathcal{R}_q^m$ and

$$\begin{aligned}d^*=&\mathbf{c}^{*\top}\mathbf{k}+e'^*-\mathbf{e}^{*\top}\mathbf{k}+\lfloor\tfrac{q}{p}\rfloor\cdot\mu_\beta=(\mathbf{c}^*-\mathbf{e}^*)^\top\textstyle\sum_{I\in\mathcal{I}}\mathbf{k}_I+e'^*+\lfloor\tfrac{q}{p}\rfloor\cdot\mu_\beta\\&=(\mathbf{c}^*-\mathbf{e}^*)^\top\mathbf{k}_{\tilde{I}}+\textstyle\sum_{I\in\mathcal{I},I\neq\tilde{I}}(\mathbf{c}^*-\mathbf{e}^*)^\top\mathbf{k}_I+e'^*+\lfloor\tfrac{q}{p}\rfloor\cdot\mu_\beta\\&=(\mathbf{c}^*-\mathbf{e}^*)^\top\mathbf{D}\mathbf{k}_{\tilde{I}}^{(0)}+(\mathbf{c}^*-\mathbf{e}^*)^\top\mathbf{k}_{\tilde{I}}^{(1)}+\textstyle\sum_{I\in\mathcal{I},I\neq\tilde{I}}(\mathbf{c}^*-\mathbf{e}^*)^\top\mathbf{k}_I+e'^*+\lfloor\tfrac{q}{p}\rfloor\cdot\mu_\beta.\end{aligned}\tag{16}$$

Define $\mathbf{u}^\top:=(\mathbf{c}^*-\mathbf{e}^*)^\top\mathbf{D}$. Given that $\mathbf{c}^*$ is uniformly chosen and $\mathbf{c}^*$ is independent of $\mathbf{e}^*$, we have $\mathbf{u}$ is uniform as well since $\mathbf{D}$ has full-rank. Thus,

$$(\mathbf{u},\mathbf{c}^*,\mathbf{e}^*,\mathbf{D})\equiv(\tilde{\mathbf{u}},\tilde{\mathbf{c}},\mathbf{e}^*,\mathbf{D}),\tag{17}$$

where $\tilde{\mathbf{u}}\leftarrow_{\$}\mathcal{R}_q^s$ and $\tilde{\mathbf{c}}\leftarrow_{\$}\mathcal{C}_{\tilde{\mathbf{u}}}:=\{\mathbf{x}\in\mathcal{R}_q^m\mid(\mathbf{x}-\mathbf{e}^*)^\top\mathbf{D}=\tilde{\mathbf{u}}^\top\}$. By (17), we can replace $(\mathbf{u},\mathbf{c}^*)$ with $(\tilde{\mathbf{u}},\tilde{\mathbf{c}})$ in our following analysis. Accordingly,

$$d^*=\tilde{\mathbf{u}}^\top\mathbf{k}_{\tilde{I}}^{(0)}+\underbrace{(\tilde{\mathbf{c}}-\mathbf{e}^*)^\top\mathbf{k}_{\tilde{I}}^{(1)}+\textstyle\sum_{I\in\mathcal{I},I\neq\tilde{I}}(\tilde{\mathbf{c}}-\mathbf{e}^*)^\top\mathbf{k}_I+e'^*+\lfloor\tfrac{q}{p}\rfloor\cdot\mu_\beta}_{:=\zeta}.\tag{18}$$

Let Tr record the complete transcripts between the challenger and $\mathcal{A}$, including the public parameters and public key $\mathbf{Ak}$, all queries and answers, the challenge message tuple (m_0,m_1) and the ciphertext $(\mathbf{c}^*,d^*,\pi^*)$, and the final output β'. Each Tr can uniquely determine an uncorrupted index $\tilde{I}$, as it records the corruption set CS and challenge ciphertext decryption set $\mathcal{Q}_{\text{DEC}}$.[15]

Define the record distribution $\mathsf{Rcd}:=(\beta,\mathsf{Tr},\mathsf{Aux})$, where Aux records:

- The setup variables, adversary's random tape, and the independent randomness used for proof simulation;
- $\mathbf{e}^*,e'^*,\tilde{\mathbf{u}},\tilde{\mathbf{c}},\{\mathbf{k}_I^{(1)}\}_{I\in\mathcal{I}}$ and $\{\mathbf{k}_I^{(0)}\}_{I\in\mathcal{I},I\neq\tilde{I}}$;
- All decryption noises $\{y_{j,I}\}$ for $I\neq\tilde{I}$, and the values $\{h_j:=\mathbf{e}_j^\top\mathbf{D}\mathbf{k}_{\tilde{I}}^{(0)}+y_{j,\tilde{I}}\}$.

Denote by $\mathsf{Rcd}_{\mathsf{G}_9}$ and $\mathsf{Rcd}_{\mathsf{G}_{10}}$ the distributions of Rcd in G_9 and G_{10}, respectively. Note that (β,β') is contained in Rcd, it holds that

$$\begin{aligned}&|\Pr[\mathsf{G}_9=1]-\Pr[\mathsf{G}_{10}=1]|\leq\Delta(\mathsf{Rcd}_{\mathsf{G}_9},\mathsf{Rcd}_{\mathsf{G}_{10}})\\=&\tfrac{1}{2}\cdot\textstyle\sum_{\widehat{\mathsf{Rcd}}}\cdot|\Pr[\mathsf{Rcd}_{\mathsf{G}_9}=\widehat{\mathsf{Rcd}}]-\Pr[\mathsf{Rcd}_{\mathsf{G}_{10}}=\widehat{\mathsf{Rcd}}]|\\=&\tfrac{1}{2}\cdot\textstyle\sum_{\widehat{\mathsf{Rcd}}}\Pr[\mathsf{Rcd}_{\mathsf{G}_{10}}=\widehat{\mathsf{Rcd}}]\cdot\underbrace{\left|\frac{\Pr[\mathsf{Rcd}_{\mathsf{G}_9}=\widehat{\mathsf{Rcd}}]}{\Pr[\mathsf{Rcd}_{\mathsf{G}_{10}}=\widehat{\mathsf{Rcd}}]}-1\right|}_{:=\gamma_{\widehat{\mathsf{Rcd}}}},\end{aligned}\tag{19}$$

where $\widehat{\mathsf{Rcd}}$ denotes a constant in the corresponding range of Rcd.

[15] Nevertheless, if there multiple uncorrupted $\{\mathbf{k}_I\}$, we can choose one $\mathbf{k}_{\tilde{I}}$ in a deterministic way.

Consequently, it suffices to bound $\gamma_{\widehat{\mathsf{Rcd}}}$ for each possible $\widehat{\mathsf{Rcd}}$, for which we aim to leverage our Adaptive LHL (Lemma 8). For $j \in [\ell]$, let $\hat{\mathbf{e}}_j$ and $\hat{h}_j$ be the corresponding values taken by $\mathbf{e}_j$ and $h_j = \mathbf{e}_j^\top \mathbf{D}\mathbf{k}_{\tilde{I}}^{(0)} + y_{j,\tilde{I}}$ in $\widehat{\mathsf{Rcd}}$. Let $\hat{\mathbf{E}} := (\hat{\mathbf{e}}_1^\top, \ldots, \hat{\mathbf{e}}_\ell^\top) \in \mathcal{R}^{\ell \times m}$ and $\hat{\mathbf{h}} := (\hat{h}_1, \ldots, \hat{h}_\ell) \in \mathcal{R}^\ell$. The only difference of the probability $\Pr[\mathsf{Rcd}_{\mathsf{G}_9} = \widehat{\mathsf{Rcd}}]$ and $\Pr[\mathsf{Rcd}_{\mathsf{G}_{10}} = \widehat{\mathsf{Rcd}}]$ arises from that, d^* is computed following (16) in G_9 and $d^* \leftarrow\!\!\$\ \mathcal{R}_q$ in G_{10}.

Note that the dependence on $\mathbf{k}_{\tilde{I}}^{(0)}$ consists exactly of the noisy relation $\hat{\mathbf{h}} = \hat{\mathbf{E}}\mathbf{D}\mathbf{k}_{\tilde{I}}^{(0)} + \mathbf{y}$, where $\mathbf{y} := (y_{\tilde{I},1}, \ldots y_{\tilde{I},\ell})$ for each $y_{\tilde{I},\ell} \leftarrow D_{\mathcal{R},\sigma_y}$, and the computation of d^* (16) in G_9.

Let $\hat{d}^*$, $\hat{\tilde{\mathbf{u}}}$, $\hat{\mathbf{D}}$ and $\hat{\zeta}$ (18) be the concrete values taken by the random variables $d^*_{\mathsf{G}_9}/d^*_{\mathsf{G}_{10}}$, $\tilde{\mathbf{u}}$, $\mathbf{D}$ and ζ in $\widehat{\mathsf{Rcd}}$, respectively. In the following calculation, all quantities fixed by $\widehat{\mathsf{Rcd}}$, including $\hat{\zeta}$, are treated as constants. The probability is taken over the original independent distributions of $\mathbf{k}_{\tilde{I}}^{(0)}$ and $\mathbf{y}$. It holds that

$$\begin{aligned}
&\Pr[d^*_{\mathsf{G}_9} = \hat{d}^* \mid \tilde{\mathbf{u}} = \hat{\tilde{\mathbf{u}}}, \hat{\mathbf{h}} = \hat{\mathbf{E}}\hat{\mathbf{D}}\mathbf{k}_{\tilde{I}}^{(0)} + \mathbf{y}] \\
=&\Pr[\tilde{\mathbf{u}}^\top \mathbf{k}_{\tilde{I}}^{(0)} = \hat{d}^* - \hat{\zeta} \mid \tilde{\mathbf{u}} = \hat{\tilde{\mathbf{u}}}, \hat{\mathbf{h}} = \hat{\mathbf{E}}\hat{\mathbf{D}}\mathbf{k}_{\tilde{I}}^{(0)} + \mathbf{y}] \\
=&\Pr[\hat{\tilde{\mathbf{u}}}^\top \mathbf{k}_{\tilde{I}}^{(0)} = \hat{d}^* - \hat{\zeta} \mid \hat{\mathbf{h}} = \hat{\mathbf{E}}\hat{\mathbf{D}}\mathbf{k}_{\tilde{I}}^{(0)} + \mathbf{y}],
\end{aligned} \tag{20}$$

where the probability is taken over $\mathbf{k}_{\tilde{I}}^{(0)} \leftarrow D_{\mathcal{R}^s,\sigma_{k_0}}$ and $\mathbf{y} \leftarrow D_{\mathcal{R}^\ell,\sigma_y}$.

By appropriately setting σ_{k_0} and σ_y s.t.

$$\tfrac{1}{\sigma_{k_0}^2} + \tfrac{\ell d B'^2 B_{\mathbf{D}}^2}{\sigma_y^2} \le \tfrac{1}{\sigma^2} \text{ with } \sigma \ge \tfrac{8}{\sqrt{\pi}} \cdot q^{\frac{1}{s}} \sqrt{d \ln(2sd(1+1/\epsilon))},$$

by Lemma 8, except with a fraction $2^{-d} + dq^{-s}$ of all $\hat{\tilde{\mathbf{u}}} \in \mathcal{R}_q^s$, it holds that

$$\Pr[\hat{\tilde{\mathbf{u}}}^\top \mathbf{k}_{\tilde{I}}^{(0)} = \hat{d}^* - \hat{\zeta} \mid \hat{\mathbf{h}} = \hat{\mathbf{E}}\hat{\mathbf{D}}\mathbf{k}_{\tilde{I}}^{(0)} + \mathbf{y}] \in (1 \pm 4\epsilon) \cdot q^{-d}. \tag{21}$$

In G_{10}, as $d^*_{\mathsf{G}_{10}} \leftarrow\!\!\$\ \mathcal{R}_q$, it holds that

$$\Pr[d^*_{\mathsf{G}_{10}} = \hat{d}^* \mid \tilde{\mathbf{u}} = \hat{\tilde{\mathbf{u}}}, \hat{\mathbf{h}} = \hat{\mathbf{E}}\hat{\mathbf{D}}\mathbf{k}_{\tilde{I}}^{(0)} + \mathbf{y}] = \Pr[d^*_{\mathsf{G}_{10}} = \hat{d}^*] = q^{-d}. \tag{22}$$

Below we only consider a "good" $\hat{\tilde{\mathbf{u}}}$ s.t. (21) holds. Following the analysis in the proof of Lemma 8, for fixed $\widehat{\mathsf{Rcd}}$, the adversary's choices, the simulated proof, the all variables and responses involving other components contribute identical factors in both games. Combining our above analysis together with (20), (21) and (22), it holds that

$$\begin{aligned}
\gamma_{\widehat{\mathsf{Rcd}}} &= \left| \frac{\Pr[\mathsf{Rcd}_{\mathsf{G}_9} = \widehat{\mathsf{Rcd}}]}{\Pr[\mathsf{Rcd}_{\mathsf{G}_{10}} = \widehat{\mathsf{Rcd}}]} - 1 \right| \\
&= \left| \underbrace{\frac{\overbrace{\Pr[d^*_{\mathsf{G}_9} = \hat{d}^* \mid \tilde{\mathbf{u}} = \hat{\tilde{\mathbf{u}}}, \hat{\mathbf{h}} = \hat{\mathbf{E}}\hat{\mathbf{D}}\mathbf{k}_{\tilde{I}}^{(0)} + \mathbf{y}]}^{\in (1\pm 4\epsilon)\cdot q^{-d}}}{\underbrace{\Pr[d^*_{\mathsf{G}_{10}} = \hat{d}^* \mid \tilde{\mathbf{u}} = \hat{\tilde{\mathbf{u}}}, \hat{\mathbf{h}} = \hat{\mathbf{E}}\hat{\mathbf{D}}\mathbf{k}_{\tilde{I}}^{(0)} + \mathbf{y}]}_{=q^{-d}}} - 1 \right| \le 4\epsilon.
\end{aligned} \tag{23}$$

In the case of $\hat{\mathbf{u}}$ is "good", plugging (23) to (19) yielding

$$\sum_{\widehat{\mathsf{Rcd}}} \Pr[\mathsf{Rcd}_{\mathsf{G}_{10}} = \widehat{\mathsf{Rcd}}] \cdot \gamma_{\widehat{\mathsf{Rcd}}} \leq \sum_{\widehat{\mathsf{Rcd}}} \Pr[\mathsf{Rcd}_{\mathsf{G}_{10}} = \widehat{\mathsf{Rcd}}] \cdot 4\epsilon \leq 4\epsilon. \tag{24}$$

Taking the probability that $\hat{\mathbf{u}}$ is not "good" into account, it affects the probability by at most $2^{-d} + dq^{-s}$. This completes the proof of Claim 5. ▮

Finally in G_{10}, $d^* \leftarrow_{\$} \mathcal{R}_q$, so the challenge bit β is totally hidden from $\mathcal{A}$ and it holds that $\Pr[\mathsf{G}_{10} = 1] = \frac{1}{2}$. Taking all things together, Theorem 1 follows. □

C.3 Proof of Theorem 2

Theorem 2 *Assume that (i) Π has adaptive soundness, (ii) $\overline{\Pi}$ has adaptive soundness, (iii) $2C \cdot (\sqrt{md}B'\tilde{B}_k + \tilde{B}_y) \leq \lfloor \frac{q}{p} \rceil/2$, then the scheme $\mathsf{TD}_1^{\mathsf{rob}}$ in Fig. 4 (with gray boxes) has robustness.*

Proof of Theorem 2. Let $\mathbf{A} \in \mathcal{R}_q^{n \times m}$ and $\mathsf{pk} = (\mathbf{Ak}, \{\mathsf{pk}_i\}_{i \in [N]})$. Let $(\mathsf{ct} = (\mathbf{c}, d, \pi), S, \{h_i\}_{i \in S}, S', \{h_i\}_{i \in S'})$ be the output of $\mathcal{A}$, where each $h_i = \{h_{i,I}, \pi_{i,I}\}_{\mathbf{k}_I \in \mathsf{sk}_i}$. Let μ, μ' be the outputs of Comb w.r.t. $\mathcal{A}$'s output, i.e., $\mu \leftarrow \mathsf{Comb}(\mathsf{pk}, \mathsf{ct}, S, \{h_i\}_{i \in S})$ and $\mu' \leftarrow \mathsf{Comb}(\mathsf{pk}, \mathsf{ct}, S', \{h_i\}_{i \in S'})$. $\mathcal{A}$ wins the game if $\mu, \mu' \neq \perp$ and $\mu \neq \mu'$. In our scheme, when combining, we will invoke $\mathsf{Vfy}(\mathsf{crs}, \mathbf{c}, \tau, \pi)$ for $\tau = \mathsf{H}(\mathbf{c}, d)$ to check the validity of ct, and invoke $\mathsf{PVer}(\mathsf{pk}_i, \mathsf{ct}, h_{i,I}, \pi_{i,I})$ to check the correctness of each $h_{i,I}$.

We claim that $\mathcal{A}$ wins the game only in the case of $\mathbf{c} \in \mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$, i.e., $\mathbf{c}^\top = \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top$ for some $\mathbf{s} \in \mathcal{R}_q^n$ and $\|\mathbf{e}\|_2 \leq B'$. If $\mathbf{c} \notin \mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$ but $\mathsf{Vfy}(\mathsf{crs}, \mathbf{c}, \tau, \pi) = 1$, then we can construct an algorithm $\mathcal{B}_1$ to break the adaptive soundness of Π. In the case of $\mathbf{c} \in \mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$, we claim that $\mathcal{A}$ wins the game only in the case of $(\mathsf{pk}_{i,I}, h_{i,I}, \mathbf{c}) \in \tilde{\mathcal{L}}_{\mathbf{A}}^{(\mathsf{lin})}$ for each sk_i and $\mathbf{k}_I \in \mathsf{sk}_i$. If $(\mathsf{pk}_{i,I}, h_{i,I}, \mathbf{c}) \notin \tilde{\mathcal{L}}_{\mathbf{A}}^{(\mathsf{lin})}$ but $\overline{\mathsf{Vfy}}(\overline{\mathsf{crs}}, (\mathsf{pk}_{i,I}, h_{i,I}, \mathbf{c}), \pi_{i,I}) = 1$, then we can construct an algorithm $\mathcal{B}_2$ to break the adaptive soundness of $\overline{\Pi}$.

In the case of $\mathbf{c} \in \mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$ and $(\mathsf{pk}_{i,I}, h_{i,I}, \mathbf{c}) \in \tilde{\mathcal{L}}_{\mathbf{A}}^{(\mathsf{lin})}$, for each sk_i and $\mathbf{k}_I \in \mathsf{sk}_i$, it holds that $h_{i,I} = \mathbf{c}^\top \mathsf{sk}_{i,I} + y_{i,I}$ for $\mathsf{sk}_{i,I} = \mathbf{k}_I \in \mathcal{R}^m$ and $y_{i,I} \in \mathcal{R}$ such that $\|\mathsf{sk}_{i,I}\|_\infty \leq \tilde{B}_k$ and $\|y_{i,I}\|_\infty \leq \tilde{B}_y$. Let $h := \sum_{i \in S} \sum_{i \in w_i} h_{i,I}$ and $h' := \sum_{i \in S'} \sum_{i \in w_i'} h'_{i,I}$. By the correctness of RSS and soundness of $\overline{\Pi}$, it holds that $\mathbf{A}(\sum_{i \in S} \sum_{i \in w_i} \mathbf{k}_I) = \sum_{i \in S} \sum_{i \in w_i} \mathbf{Ak}_I = \sum_{i \in S} \sum_{i \in w_i} \mathsf{pk}_{i,I} = \mathsf{pk} = \sum_{i \in S'} \sum_{i \in w_i'} \mathsf{pk}_{i,I} = \mathbf{A}(\sum_{i \in S'} \sum_{i \in w_i'} \mathbf{k}_I)$. Consequently, we have

$$\begin{aligned} h - h' &= (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top) \sum_{i \in S} \sum_{i \in w_i} \mathbf{k}_I + \sum_{i \in S} \sum_{i \in w_i} y_{i,I} - (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top) \sum_{i \in S'} \sum_{i \in w_i'} \mathbf{k}_I - \sum_{i \in S'} \sum_{i \in w_i'} y'_{i,I} \\ &= \mathbf{e}^\top \sum_{i \in S} \sum_{i \in w_i} \mathbf{k}_I - \mathbf{e}^\top \sum_{i \in S'} \sum_{i \in w_i'} \mathbf{k}_I + \sum_{i \in S} \sum_{i \in w_i} y_{i,I} - \sum_{i \in S'} \sum_{i \in w_i'} y'_{i,I} \\ &\leq 2C(B' \cdot \sqrt{md}\tilde{B}_k + \tilde{B}_y) \leq \lfloor \frac{q}{p} \rceil/2. \end{aligned}$$

According to the combination algorithm and let $\gamma := \lfloor \frac{q}{p} \rfloor$, we have $|d-h-\gamma\mu| < \gamma/4$ and $|d-h'-\gamma\mu'| < \gamma/4$. Thus, we have

$$|\gamma\mu-\gamma\mu'| \le |\gamma\mu-(d-h)|+|h'-h|+|(d-h')-\gamma\mu| < \gamma/4+\gamma/2+\gamma/4 = \gamma. \quad (25)$$

If $\mu \neq \mu'$, we have $|\gamma\mu - \gamma\mu'| \ge \gamma$, which contradicts (25). Taking all things together, Theorem 2 follows. □

C.4 Proof of Theorem 3

Theorem 3 (smCPA **Security of** $\mathsf{TD}_2^{\mathsf{smCPA}}$) *Let* $\mathsf{H}_{\mathsf{mask}}, \mathsf{H}_{\mathsf{rand}}$ *be RO. If*

- *the* $\mathsf{MSIS}_{q,n,m+n,2B'_{\mathsf{E}}}$ *assumption and* $\mathsf{MLWE}_{q,m+1,n,\sigma_0}$ *assumption hold,*
- *the* $\mathsf{AHMLWE}^{\ell,\mathcal{V},\sigma_y}_{q,m,n,\sigma_k}$ *assumption holds, where* $\mathcal{V} := \{\mathbf{v} = (v_i)_{i\in[n+m]} \in \mathcal{R}^{n+m} : \sum_{i\in[n+m]} \|v_i\|_1^2 \le B_{\mathsf{hint}}\}$ *for* $B_{\mathsf{hint}} := d \cdot B'^2_{\mathsf{E}}$,[16]

then the scheme $\mathsf{TD}_2^{\mathsf{smCPA}}$ *in Fig. 5 is* $\mathsf{IND\text{-}smCPA}$ *secure. More precisely, for any PPT adversary* $\mathcal{A}$ *that makes at most* ℓ *times of* $\mathcal{O}_{\mathrm{DEC}}$*, there exist PPT algorithms* $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ *s.t.*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{smCPA}}_{\mathsf{TD}_2^{\mathsf{smCPA}},\mathcal{A}}(\lambda) \le\ & \mathsf{Adv}^{\mathsf{MSIS}}_{q,n,m+n,2B_0,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{AHMLWE}}_{(q,m,n,\sigma_k),(\ell,\sigma_y,\mathcal{V}),\mathcal{B}_2}(\lambda) \\ & + \mathsf{Adv}^{\mathsf{MLWE}}_{q,m+1,n,\sigma_0,\mathcal{B}_3}(\lambda) + Q_{\mathsf{Enc}} \cdot 2^{-(n+m+1)d} + \frac{Q_{\mathsf{mask}}}{2^{\lambda-1}},\end{aligned}$$

where $\mathcal{V} := \{\mathbf{v} = (v_i)_{i\in[n+m]} \in \mathcal{R}_q^{n+m} : \sum_{i\in[n+m]} \|v_i\|_1^2 \le B_{\mathsf{hint}}\}$.

Proof of Theorem 3. Let $\mathcal{A}$ be an adversary against the $\mathsf{IND\text{-}smCPA}$ security of $\mathsf{TD}_2^{\mathsf{smCPA}}$. We use games G_0-G_{10} to show the advantage $\mathsf{Adv}^{\mathsf{smCPA}}_{\mathsf{TD}_2^{\mathsf{smCPA}},\mathcal{A}}(\lambda)$ is negligible, where G_0 is the original $\mathsf{IND\text{-}smCPA}$ experiment and $\mathcal{A}$ has no advantage in G_{10}.

Game G_0: This is the $\mathsf{IND\text{-}smCPA}$ experiment $\mathsf{Exp}^{\mathsf{smCPA}}_{\mathsf{TD}_2^{\mathsf{smCPA}},\mathcal{A}}$ in the synchronous setting (cf. Fig. 2 with gray boxes and dashed boxes). By definition, we have $\mathsf{Adv}^{\mathsf{smCPA}}_{\mathsf{TD}_2^{\mathsf{smCPA}},\mathcal{A}}(\lambda) = |\Pr[\mathsf{G}_0 = 1] - \frac{1}{2}|$.

Let $\mathsf{pp} = \mathbf{A} \leftarrow_{\$} \mathcal{R}_q^{n\times m}$. The challenger samples $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}, \mathbf{f} \leftarrow D_{\mathcal{R}^n,\sigma_k}$, invokes $(\mathbf{k}_1,\ldots,\mathbf{k}_N) \leftarrow_{\$} \mathsf{SShare}(N,t,\mathbf{k})$, and sets $\mathsf{pk} := \mathbf{A}\mathbf{k}+\mathbf{f} \in \mathcal{R}_q^n$, $\mathsf{sk}_i := (\mathbf{k}_i, \{\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i}\}_{j\in[N]})$ for each $i \in [N]$. The challenger sends $(\mathsf{pp},\mathsf{pk})$ to $\mathcal{A}$ and initializes $\mathsf{Cor} := \emptyset$, $\mathsf{Hon} := [N]$, $\mathcal{Q}_{\mathrm{DEC}}[\cdot] := \emptyset$. Then it answers $\mathcal{A}$'s $\mathsf{H}_{\mathsf{mask}}, \mathsf{H}_{\mathsf{rand}}$, $\mathcal{O}_{\mathrm{COR}}$, $\mathcal{O}_{\mathrm{ENC}}$, $\mathcal{O}_{\mathrm{DEC}}$ queries adaptively and generates a challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}^*, d^*)$ for $\mathcal{A}$ following the scheme in Fig. 5. Note that, upon corruption of user i, the adversary can reconstruct the randomness used in all previous partial decryption queries of user i from corresponding (ct, S) and $\{\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i}\}_{j\in[N]}$ contained in sk_i. Thus, the randomness y_i need not be revealed separately.

Game G_1: In this game (and the subsequent games), for $S \subseteq [N]$, we denote the honest (resp. corrupted) users in S by $\mathsf{HS} := S \cap \mathsf{Hon}$ (resp. $\mathsf{CS} := S \cap \mathsf{Cor}$). Moreover, in this game, the challenger introduces a set QDec and three tables $\mathsf{UnDecHS}$, Mask and Rnd as follows:

[16] By Lemma 2, under appropriate parameters, AHMLWE can be reduced to MLWE.

– QDec records the tuples (ct, S) for which $\mathcal{A}$ has queried $\mathcal{O}_{\mathrm{DEC}}(\cdot, \mathsf{ct}, S)$.
– $\mathsf{UnDecHS}[(\mathsf{ct}, S)]$ for $(\mathsf{ct}, S) \in \mathsf{QDec}$ stores the users $i \in \mathsf{HS}$ that $\mathcal{A}$ does not issue $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}, S)$. We also denote $\widetilde{\mathsf{HS}} := \mathsf{UnDecHS}[(\mathsf{ct}, S)]$ when (ct, S) is clear from the context.
– $\mathsf{Mask}[(\mathsf{ct}, S), i]$ stores the mask $\Delta_{i,S} := \sum_{j \in S \setminus \{i\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j}, (\mathsf{ct}, S)) - \sum_{j \in S \setminus \{i\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,i}, (\mathsf{ct}, S))$. In addition, when answering $\mathcal{O}_{\mathrm{COR}}(i)$, the challenger precomputes $\Delta_{i,S}$ for each $(\mathsf{ct}, S) \in \mathsf{QDec}$ such that $i \in S$, and stores it in $\mathsf{Mask}[(\mathsf{ct}, S), i]$.
– $\mathsf{Rnd}[\mathsf{ct}]$ stores the randomness $(\mathbf{s}, \mathbf{e}, e')$ that $\mathcal{A}$ has submitted to $\mathcal{O}_{\mathrm{ENC}}$, where the index ct is the output of $\mathcal{O}_{\mathrm{ENC}}$, i.e., the ciphertext w.r.t. $(\mathbf{s}, \mathbf{e}, e')$.

Now the challenger answers $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$ queries according to Fig. 8, where we highlight the changes in gray boxes. Note that the change of G_1 does not affect the view of $\mathcal{A}$, so we have $\Pr[\mathsf{G}_0 = 1] = \Pr[\mathsf{G}_1 = 1]$.

$\underline{\mathsf{G}_1}$:

$\underline{\mathcal{O}_{\mathrm{ENC}}(\mu, \mathsf{rnd} = (\mathbf{s}, \mathbf{e}, e'))}$
If $\mathsf{rnd} \notin \mathcal{RD}$: Return $\perp$
$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mu; \mathsf{rnd})$
$\mathcal{Q}_{\mathrm{ENC}} := \mathcal{Q}_{\mathrm{ENC}} \cup \{\mathsf{ct}\}$
$\mathsf{Rnd}[\mathsf{ct}] := \mathsf{Rnd}[\mathsf{ct}] \cup (\mathbf{s}, \mathbf{e}, e')$
Return ct

$\underline{\mathcal{O}_{\mathrm{COR}}(i)}$:
If $\exists\ S$ s.t. $|\mathcal{Q}_{\mathrm{DEC}}[S] \cup \mathsf{Cor}| \geq t - 1$: Return $\perp$
For $(\mathsf{ct}, S) \in \mathsf{QDec}$ s.t. $(i \in S) \wedge (\mathsf{Mask}[(\mathsf{ct}, S), i] = \perp)$:
  Parse $\mathsf{sk}_i = (\mathbf{k}_i, \{\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i}\}_{j \in [N]})$
  $\Delta_{i,S} := \sum_{j \in S \setminus \{i\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j}, (\mathsf{ct}, S)) - \sum_{j \in S \setminus \{i\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,i}, (\mathsf{ct}, S)) \in \mathcal{R}_q$
  $\mathsf{Mask}[(\mathsf{ct}, S), i] := \Delta_{i,S}$
  $\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{UnDecHS}[(\mathsf{ct}, S)] \setminus \{i\}$
$\mathsf{Cor} := \mathsf{Cor} \cup \{i\}$, $\mathsf{Hon} := \mathsf{Hon} \setminus \{i\}$
Return sk_i

$\underline{\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct} = (\mathbf{c}, d), S)}$:
If $\mathsf{ct} \neq \mathsf{ct}^*$ and $\mathsf{ct} \notin \mathcal{Q}_{\mathrm{ENC}}$: Return $\perp$
If $(\mathsf{ct}, S) \notin \mathsf{QDec}$:
  $\mathsf{QDec} := \mathsf{QDec} \cup \{(\mathsf{ct}, S)\}$
  $\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{HS}$
Parse $\mathsf{sk}_i = (\mathbf{k}_i, \{\mathsf{seed}_{i,j}, \mathsf{seed}_{j,i}\}_{j \in [N]})$
$\Delta_{i,S} := \sum_{j \in S \setminus \{i\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j}, (\mathsf{ct}, S)) - \sum_{j \in S \setminus \{i\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,i}, (\mathsf{ct}, S)) \in \mathcal{R}_q$
$y_{i,S} = \mathsf{H}_{\mathsf{rand}}(\mathsf{seed}_{i,i}, \mathsf{ct}, S)$
$h_{i,S} := \mathbf{c}^\top(\lambda_{i,S} \cdot \mathbf{k}_i) + y_{i,S} + \Delta_{i,S}$
$\mathsf{Mask}[(\mathsf{ct}, S), i] := \Delta_{i,S}$
$\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{UnDecHS}[(\mathsf{ct}, S)] \setminus \{i\}$
If $\mathsf{ct} = \mathsf{ct}^*$: $\mathcal{Q}_{\mathrm{DEC}}[S] := \mathcal{Q}_{\mathrm{DEC}} \cup \{i\}$
Return $h_{i,S}$

Fig. 8. The game G_1 with changes highlighted in gray boxes.

$\underline{\text{Game } \mathsf{G}_2\text{:}}$ It is the same as G_1, except when generating a ciphertext ct in $\mathcal{O}_{\mathrm{ENC}}$ and when generating the challenge ciphertext ct^*, if $\mathsf{ct} \in \mathcal{Q}_{\mathrm{ENC}} \cup \{\mathsf{ct}^*\}$ or $\mathsf{ct}^* \in \mathcal{Q}_{\mathrm{ENC}}$, the challenger aborts the game directly returns a random bit.

Claim 6. $|\Pr[\mathsf{G}_1 = 1] - \Pr[\mathsf{G}_2 = 1]| \leq \mathsf{Adv}^{\mathsf{MSIS}}_{q, m+n, m, 2B'_{\mathsf{E}}, \mathcal{B}_1}(\lambda) + Q_{\mathsf{Enc}} \cdot 2^{-(n+m+1)d}$.

Proof. We prove Claim 6 by contradiction. Let $\mathsf{Enc}(\mathsf{pk}, \mu_0; (\mathbf{s}_0, \mathbf{e}_0, e'_0)) = (\mathbf{c}, d) = \mathsf{Enc}(\mathsf{pk}, \mu_1; (\mathbf{s}_1, \mathbf{e}_1, e'_1))$ be a collision of encryption with $(\mu_0, (\mathbf{s}_0, \mathbf{e}_0, e'_0)) \neq (\mu_1, (\mathbf{s}_1, \mathbf{e}_1, e'_1))$. We consider the following two cases:

– If $(\mathbf{s}_0, \mathbf{e}_0) = (\mathbf{s}_1, \mathbf{e}_1)$, then $\mathbf{s}_0^\top \mathsf{pk} + e'_0 + \lfloor \frac{q}{p} \rfloor \cdot \mu_0 = d = \mathbf{s}_1^\top \mathsf{pk} + e'_1 + \lfloor \frac{q}{p} \rfloor \cdot \mu_1$ implies $e'_0 - e'_1 = \lfloor \frac{q}{p} \rfloor \cdot (\mu_0 - \mu_1)$. Since $|e'_0 - e'_1| \leq 2B_0$ and $2B_0 \ll \lfloor \frac{q}{p} \rfloor \leq |\lfloor \frac{q}{p} \rfloor \cdot (\mu_0 - \mu_1)|$, it must hold $(e'_0, \mu_0) = (e'_1, \mu_1)$, which contradicts $(\mu_0, (\mathbf{s}_0, \mathbf{e}_0, e'_0)) \neq (\mu_1, (\mathbf{s}_1, \mathbf{e}_1, e'_1))$.
– If $(\mathbf{s}_0, \mathbf{e}_0) \neq (\mathbf{s}_1, \mathbf{e}_1)$, then we have $\mathbf{s}_0^\top \mathbf{A} + \mathbf{e}_0^\top = \mathbf{s}_1^\top \mathbf{A} + \mathbf{e}_1^\top$, so that $\mathbf{x} := (\mathbf{s}_0 - \mathbf{s}_1, \mathbf{e}_0 - \mathbf{e}_1) \in \mathcal{R}^{n+m}$ forms MSIS solution (cf. Def. 2) w.r.t. $[\mathbf{A}^\top \mid \mathbf{I}_m]$ with $\|\mathbf{x}\| \leq 2B'_{\mathsf{E}}$. Consequently, there exists PPT $\mathcal{B}_1$ s.t. $(\mathbf{s}_0, \mathbf{e}_0) \neq (\mathbf{s}_1, \mathbf{e}_1)$ happens with probability less than $\mathsf{Adv}^{\mathsf{MSIS}}_{q,m+n,m,2B'_{\mathsf{E}},\mathcal{B}_1}(\lambda)$.

The remaining part is to exclude the case that the randomness $\mathsf{rnd}^* := (\mathbf{s}^*, \mathbf{e}^*, e'^*)$ used in challenge ciphertext ct^* has been used in the query of $\mathcal{O}_{\mathrm{ENC}}(\cdot, \mathsf{rnd}^*)$. Since $\max_{\mathsf{rnd}} D_{\mathcal{R}^{n+m+1},\sigma_0}(\mathsf{rnd}) = 1/\rho_\sigma(\mathcal{R}^{n+m+1}) \leq \sigma_0^{-(n+m+1)d} \leq 2^{-(n+m+1)d}$ by Poisson summation, a union bound shows that rnd^* collides with any of the Q_{Enc} adversarially chosen rnd with probability at most $Q_{\mathsf{Enc}} \cdot 2^{-(n+m+1)d} = \mathsf{negl}(\lambda)$.

Game G_3: It is the same as G_2, except for the reply of $\mathsf{H}_{\mathsf{mask}}$ and the computation of $\Delta_{i,S}$ in $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$. More precisely, the challenger answers $\mathsf{H}_{\mathsf{mask}}$, $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$ queries according to Fig. 9.

G_3:
$\mathsf{H}_{\mathsf{mask}}(\mathsf{input})$: //for $\mathcal{A}$'s queries
Parse $\mathsf{input} = (\mathsf{seed}, (\mathsf{ct}, S))$
Parse $\mathsf{seed} = (i, j, \mathsf{rand})$
If $i, j \in \mathsf{Hon} \wedge \mathsf{rand} = \mathsf{rand}_{i,j}$: Return $\bot$
If $\mathcal{Q}_{\mathrm{MASK}}[\mathsf{input}] = \bot$:
  $\mathbf{m} \leftarrow\!\!\$\ \mathcal{R}_q$, $\mathcal{Q}_{\mathrm{MASK}}[\mathsf{input}] := \mathbf{m}$
Return $\mathcal{Q}_{\mathrm{MASK}}[\mathsf{input}]$

$\mathcal{O}_{\mathrm{COR}}(i)$ and $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}, S)$: //replace any //computation of $\Delta_{i,S}$ with the following:
For $j \in \mathsf{CS}$:
  $\mathbf{m}_{i,j} := \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j}, (\mathsf{ct}, S))$
  $\mathbf{m}_{j,i} := \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,i}, (\mathsf{ct}, S))$
//note that $(\mathsf{ct}, S) \in \mathsf{QDec}$ when computing $\Delta_{i,S}$
$\widetilde{\mathsf{HS}} := \mathsf{UnDecHS}[(\mathsf{ct}, S)]$
For $j \in \mathsf{HS} \setminus \widetilde{\mathsf{HS}}$:
  $\mathbf{m}_{i,j} := \mathcal{Q}_{\mathrm{MASK}}[\mathsf{seed}_{i,j}, (\mathsf{ct}, S)]$
  $\mathbf{m}_{j,i} := \mathcal{Q}_{\mathrm{MASK}}[\mathsf{seed}_{j,i}, (\mathsf{ct}, S)]$
For $j \in \widetilde{\mathsf{HS}} \setminus \{i\}$:
  $\mathbf{m}_{i,j} \leftarrow\!\!\$\ \mathcal{R}_q$, $\mathcal{Q}_{\mathrm{MASK}}[\mathsf{seed}_{i,j}, (\mathsf{ct}, S)] := \mathbf{m}_{i,j}$
  $\mathbf{m}_{j,i} \leftarrow\!\!\$\ \mathcal{R}_q$, $\mathcal{Q}_{\mathrm{MASK}}[\mathsf{seed}_{j,i}, (\mathsf{ct}, S)] := \mathbf{m}_{j,i}$
$\Delta_{i,S} := \sum_{j \in S \setminus \{i\}} (\mathbf{m}_{i,j} - \mathbf{m}_{j,i}) \in \mathcal{R}_q$
$\mathsf{Mask}[(\mathsf{ct}, S), i] := \Delta_{i,S}$

Fig. 9. The game G_3 with changes highlighted in gray boxes.

Claim 7. $|\Pr[\mathsf{G}_2 = 1] - \Pr[\mathsf{G}_3 = 1]| \leq \frac{Q_{\mathsf{mask}}}{2^\lambda}$.

Proof. Let Abort denote the event that the hash oracle $\mathsf{H}_{\mathsf{mask}}(\cdot)$ outputs $\bot$. We first analyze the probability $\Pr[\mathsf{Abort}]$. For each $\mathsf{H}_{\mathsf{mask}}$ query made by $\mathcal{A}$, if $i, j \in \mathsf{Hon}$, the value of $\mathsf{rand}_{i,j} \in \{0,1\}^\lambda$ is hidden from $\mathcal{A}$, so $\Pr[\mathsf{rand} = \mathsf{rand}_{i,j}] = \frac{1}{2^\lambda}$. By the union bound, we have $\Pr[\mathsf{Abort}] \leq \frac{Q_{\mathsf{mask}}}{2^\lambda} = \mathsf{negl}(\lambda)$, where Q_{mask} is the number of $\mathsf{H}_{\mathsf{mask}}$ queries made by $\mathcal{A}$.

We further argue that for any $j \in \mathsf{HS} \setminus \widetilde{\mathsf{HS}}$, the hash values $\mathcal{Q}_{\mathrm{MASK}}[\mathsf{seed}_{i,j}, (\mathsf{ct}, S)]$ and $\mathcal{Q}_{\mathrm{MASK}}[\mathsf{seed}_{j,i}, (\mathsf{ct}, S)]$ have already been defined, whereas for any $j \in \widetilde{\mathsf{HS}} \setminus \{i\}$, the corresponding hash values remain undefined.

– For any $j \in \mathsf{HS} \setminus \widetilde{\mathsf{HS}}$, by the definition of HS and $\widetilde{\mathsf{HS}}$, $\mathcal{A}$ has previously queried $\mathcal{O}_{\mathrm{DEC}}(j, \mathsf{ct}, S)$, during which the values $\mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j}, (\mathsf{ct}, S))$ and $\mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,i}, (\mathsf{ct}, S))$ have already been defined.
– For any $j \in \widetilde{\mathsf{HS}} \setminus \{i\}$, by the definition of $\widetilde{\mathsf{HS}}$, $\mathcal{A}$ has not queried $\mathcal{O}_{\mathrm{DEC}}(j, \mathsf{ct}, S)$ so far. Moreover, since $i, j \in \mathsf{Hon}$ at this point, $\mathcal{A}$ can hardly query $\mathsf{H}_{\mathsf{mask}}$ with honest $\mathsf{seed}_{i,j}$ or $\mathsf{seed}_{j,i}$. Hence, the values $\mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j}, (\mathsf{ct}, S))$ and $\mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,i}, (\mathsf{ct}, S))$ have not been defined prior to this point.

Putting these observations together, Claim 7 follows. ▌

Game G_4: It is the same as G_3, except that the challenger randomly samples all masks except for the last honest user, and the last mask is computed deterministically by other masks in $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$. Additionally, the oracle $\mathsf{H}_{\mathsf{mask}}$ is not programmed accordingly until the corresponding user is corrupted. More precisely, the challenger answers oracles $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$ according to Fig. 10.

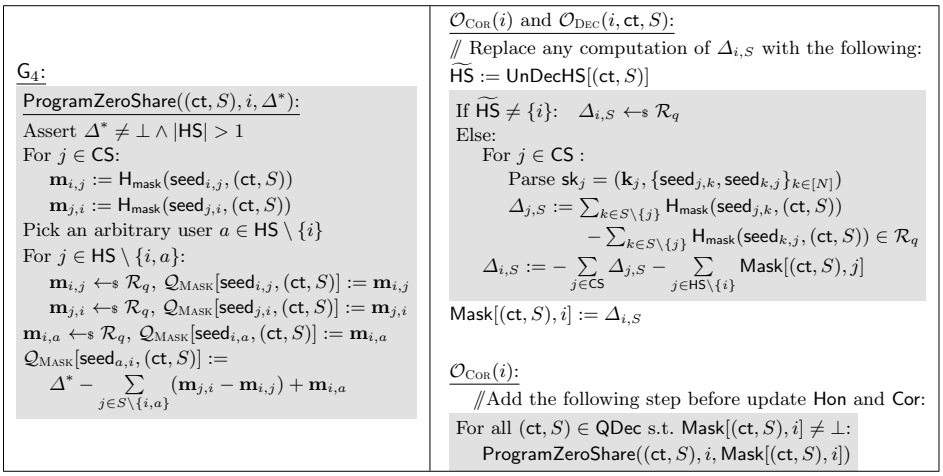

Fig. 10. The game G_4 with changes highlighted in gray boxes.

Claim 8. $\Pr[\mathsf{G}_3 = 1] = \Pr[\mathsf{G}_4 = 1]$.

Proof. We need to show that the views of $\mathcal{A}$ in G_3 and G_4 are identical. Firstly, we prove that the values of $\Delta_{i,S}$ are identically distributed in both games.

– If i is not the last honest user, i.e., $\widetilde{\mathsf{HS}} \neq \{i\}$ when querying $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}, S)$, the values $\mathbf{m}_{i,j}$ and $\mathbf{m}_{j,i}$ for $j \in \widetilde{\mathsf{HS}} \setminus \{i\}$ are sampled at random. Thus, $\Delta_{i,S} := \sum_{j \in S \setminus \{i\}} (\mathbf{m}_{i,j} - \mathbf{m}_{j,i})$ in G_3 is uniformly random, as it is in G_4.
– If $\widetilde{\mathsf{HS}} = \{i\}$ when querying $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}, S)$, the values $(\mathbf{m}_{i,j}, \mathbf{m}_{j,i})$ for $j \in \mathsf{HS} \setminus \{i\}$ have already been stored in $\mathcal{Q}_{\mathrm{MASK}}$ in G_3, and hence $\mathsf{Mask}[(\mathsf{ct}, S), j]$ is determined. Note that $\Delta_{i,S}$ should satisfy the equation $\sum_{j \in S} \Delta_{j,S} = 0$ in both games (cf. (15)). We have the following equation in both G_3 and G_4.

$$\Delta_{i,S} = - \sum_{j \in \mathsf{CS}} \Delta_{j,S} - \sum_{j \in \mathsf{HS} \setminus \{i\}} \Delta_{j,S} = - \sum_{j \in \mathsf{CS}} \Delta_{j,S} - \sum_{j \in \mathsf{HS} \setminus \{i\}} \mathsf{Mask}[(\mathsf{ct}, S), j].$$

Secondly, we argue that the output of $\mathsf{H}_{\mathsf{mask}}$ in G_4 and G_3 are identically distributed.

- If $i, j \in \mathsf{Hon} \wedge \mathsf{seed} = \mathsf{seed}_{i,j}$, the challenger aborts in $\mathsf{H}_{\mathsf{mask}}$ in both games.
- In G_3 $\Delta_{i,S}$ is generated via $\mathsf{H}_{\mathsf{mask}}$. In contrast, in G_4, $\Delta_{i,S}$ can also be viewed as $\Delta_i := \sum_{j \in S \setminus \{i\}} (\mathbf{m}_{i,j} - \mathbf{m}_{j,i}) \in \mathcal{R}_q$ via the algorithm $\mathsf{ProgramZeroShare}$, since we reprogram $\mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{i,j}, (\mathsf{ct}, S))$ only in the case of $i, j \in \mathsf{Hon}$.

The views of $\mathcal{A}$ in both games are identical. Thus, $\Pr[\mathsf{G}_3 = 1] = \Pr[\mathsf{G}_4 = 1]$. ▌

Game G_5: It is the same as G_4, except that the challenger randomly samples all partial decryption except for the last honest user, and the last partial decryption is computed from previous partial decryptions and a partial information about the secret key $\mathbf{k}$ (i.e., $\mathbf{c}^\top \mathbf{k} + y_{i,S}$ in Fig. 11 below). Additionally, the computation of masks in $\mathcal{O}_{\mathrm{COR}}$ is changed accordingly. Besides, in this game, the challenger introduces two tables RandY, $\mathsf{MaskedDec}$ as defined below:

- $\mathsf{RandY}[(\mathsf{ct}, S), i]$ records the noise term $y_{i,S}$ when answering $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}, S)$.
- $\mathsf{MaskedDec}[(\mathsf{ct}, S), i]$ records the partial decryption results $h_{i,S}$ when answering $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}, S)$.

More precisely, the challenger answers $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$ queries according to Fig. 11.

Claim 9. $|\Pr[\mathsf{G}_4 = 1] - \Pr[\mathsf{G}_5 = 1]| \leq \frac{Q_{\mathsf{mask}}}{2^\lambda}$.

Proof. We introduce an intermediate game $\mathsf{G}_{4.5}$ that is identical to G_4 except that now the challenger samples $\Delta^*_{i,S} \leftarrow_{\$} \mathcal{R}_q$ and sets $\Delta_{i,S} := \Delta^*_{i,S} - \big(\mathbf{c}^\top(\lambda_{i,S}\mathbf{k}_i) + y_{i,S}\big)$ when computing $\Delta_{i,S}$, and store $\mathsf{RandY}[(\mathsf{ct}, S), i] = y_{i,S}$. Since both $\Delta_{i,S}$ and $\Delta^*_{i,S}$ are uniformly distributed over $\mathcal{R}_q$, $\mathsf{G}_{4.5}$ is identically distributed to G_4. Besides, observe that in $\mathsf{G}_{4.5}$, when $\widetilde{\mathsf{HS}} \neq \{i\}$, due to $h_{i,S} := \mathbf{c}^\top(\lambda_{i,S}\mathbf{k}_i) + y_{i,S} + \Delta_{i,S}$, we have $h_{i,S} = \Delta^*_{i,S}$, which will be stored in $\mathsf{MaskedDec}[(\mathsf{ct}, S), i]$. When $\widetilde{\mathsf{HS}} = \{i\}$, we have $\Delta_{i,S} := -\sum_{j \in \mathsf{CS}} \Delta_{k,S} - \sum_{j \in \mathsf{HS} \setminus \{i\}} \Delta_{j,S}$ in $\mathsf{G}_{4.5}$ and thus

$$\begin{aligned} h_{i,S} &:= \mathbf{c}^\top(\lambda_{i,S}\mathbf{k}_i) + y_{i,S} - \sum_{j \in \mathsf{CS}} \Delta_{j,S} - \sum_{j \in \mathsf{HS} \setminus \{i\}} \Delta_{j,S} \\ &= \mathbf{c}^\top(\lambda_{i,S}\mathbf{k}_i) + y_{i,S} - \sum_{j \in \mathsf{CS}} \Delta_{j,S} - \sum_{j \in \mathsf{HS} \setminus \{i\}} \Delta^*_{j,S} + \sum_{j \in \mathsf{HS} \setminus \{i\}} (\mathbf{c}^\top(\lambda_{j,S}\mathbf{k}_j) + y_{j,S}) \\ &= \mathbf{c}^\top \mathbf{k} - \sum_{j \in \mathsf{CS}} \mathbf{c}^\top(\lambda_{j,S}\mathbf{k}_j) + y_{i,S} + \sum_{j \in \mathsf{HS} \setminus \{i\}} y_{j,S} - \sum_{j \in \mathsf{HS} \setminus \{i\}} h_{j,S} - \sum_{j \in \mathsf{CS}} \Delta_{j,S}. \end{aligned} \tag{26}$$

By substituting $y_{j,S}$ by $\mathsf{RandY}[(\mathsf{ct}, S), j]$ and $h_{j,S}$ by $\mathsf{MaskedDec}[(\mathsf{ct}, S), j]$ respectively, (26) is exactly the simulation of $h_{i,S}$ in G_5 as shown in Fig. 11.

On the other hand, we need to check the consistency of the simulation of $\Delta_{i,S}$ in $\mathcal{O}_{\mathrm{COR}}(i)$ if $\widetilde{\mathsf{HS}} = \{i\}$. In $\mathsf{G}_{4.5}$, we have $\mathsf{Mask}[(\mathsf{ct}, S), j] = \Delta^*_{j,S} - \mathbf{c}^\top(\lambda_{j,S}\mathbf{k}_j) - y_{j,S}$

$\underline{\mathsf{G}_5}$:

$\underline{\mathcal{O}_{\text{COR}}(i)}$:

If $\exists\ S \subseteq [N]$ s.t. $|\mathsf{Cor} \cup \mathcal{Q}_{\text{DEC}}[S]| \geq t-1$: Return $\bot$

For all $(\mathsf{ct} = (\mathbf{c}, d), S) \in \mathsf{QDec}$ s.t. $(i \in S) \wedge (\mathsf{Mask}[(\mathsf{ct}, S)] = \bot)$:

If $\mathsf{MaskedDec}[(\mathsf{ct}, S), i] \neq \bot$:

$\mathsf{Mask}[(\mathsf{ct}, S), i] := \mathsf{MaskedDec}[(\mathsf{ct}, S), i] - \mathbf{c}^\top \lambda_{i,S} \mathbf{k}_i - \mathsf{RandY}[(\mathsf{ct}, S), i]$

$\mathsf{H}_{\mathsf{rand}}(\mathsf{seed}_{i,i}, (\mathsf{ct}, S)) := \mathsf{RandY}[(\mathsf{ct}, S), i]$

Else: $\widetilde{\mathsf{HS}} := \mathsf{UnDecHS}[(\mathsf{ct}, S)]$

If $\widetilde{\mathsf{HS}} \neq \{i\}$: $\mathsf{Mask}[(\mathsf{ct}, S), i] \leftarrow\!\!\$\ \mathcal{R}_q$

Else:

For $j \in \mathsf{CS}$: Parse $\mathsf{sk}_j = (\mathbf{k}_j, \{\mathsf{seed}_{j,k}, \mathsf{seed}_{k,j}\}_{k\in[N]})$

$\Delta_{j,S} := \sum_{k\in S\setminus\{j\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,k}, (\mathsf{ct}, S)) - \sum_{k\in S\setminus\{j\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{k,j}, (\mathsf{ct}, S)) \in \mathcal{R}_q$

$\Delta_{i,S} := \mathbf{c}^\top \mathbf{k} - \sum_{j\in \mathsf{CS}\cup\{i\}} \mathbf{c}^\top \cdot \lambda_{j,S} \cdot \mathbf{k}_j + \sum_{j\in\mathsf{HS}\setminus\{i\}} \mathsf{RandY}[(\mathsf{ct}, S), j]$
$- \sum_{j\in\mathsf{HS}\setminus\{i\}} \mathsf{MaskedDec}[(\mathsf{ct}, S), j] - \sum_{j\in\mathsf{CS}} \Delta_{j,S}.$

$\mathsf{Mask}[(\mathsf{ct}, S), i] := \Delta_{i,S}$

$\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{UnDecHS}[(\mathsf{ct}, S)]\setminus\{i\}$

For all $(\mathsf{ct}, S) \in \mathsf{QDec}$ s.t. $\mathsf{Mask}[(\mathsf{ct}, S), i] \neq \bot$

$\mathsf{ProgramZeroShare}((\mathsf{ct}, S), i, \mathsf{Mask}[(\mathsf{ct}, S), i])$

$\mathsf{Cor} := \mathsf{Cor} \cup \{i\}$, $\mathsf{Hon} := \mathsf{Hon}\setminus\{i\}$

Return sk_i

$\underline{\mathcal{O}_{\text{DEC}}(i, \mathsf{ct}, S)}$:

If $\mathsf{ct} \neq \mathsf{ct}^*$ and $\mathsf{ct} \notin \mathcal{Q}_{\text{ENC}}$: Return $\bot$

If $(\mathsf{ct}, S) \notin \mathsf{QDec}$:

$\mathsf{QDec} := \mathsf{QDec} \cup \{(\mathsf{ct}, S)\}$

$\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{HS}$, $\widetilde{\mathsf{HS}} := \mathsf{HS}$

If $(\mathsf{ct}, S) \in \mathsf{QDec}$: $\widetilde{\mathsf{HS}} := \mathsf{UnDecHS}[(\mathsf{ct}, S)]$

If $\widetilde{\mathsf{HS}} \neq \{i\}$: $h_{i,S} \leftarrow\!\!\$\ \mathcal{R}_q$,
$y_{i,S} \leftarrow\!\!\$\ D_{\mathcal{R},\sigma_y}$, $\mathsf{RandY}[(\mathsf{ct}, S), i] := y_{i,S}$

Else:

Parse $\mathsf{ct} = (\mathbf{c}, d)$

For $j \in \mathsf{CS}$: Parse $\mathsf{sk}_j = (\mathbf{k}_j, \{\mathsf{seed}_{j,k}, \mathsf{seed}_{k,j}\}_{k\in[N]})$

$\Delta_{j,S} := \sum_{k\in S\setminus\{j\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,k}, (\mathsf{ct}, S)) - \sum_{k\in S\setminus\{j\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{k,j}, (\mathsf{ct}, S)) \in \mathcal{R}_q$

$y_{i,S} \leftarrow\!\!\$\ D_{\mathcal{R},\sigma_y}$

$h_{i,S} := \mathbf{c}^\top \mathbf{k} + y_{i,S} + \sum_{j\in\mathsf{HS}\setminus\{i\}} \mathsf{RandY}[(\mathsf{ct}, S), j]$
$- \sum_{j\in\mathsf{HS}\setminus\{i\}} \mathsf{MaskedDec}[(\mathsf{ct}, S), j] - \sum_{j\in\mathsf{CS}} \lambda_{j,S} \mathbf{c}^\top \mathbf{k}_j - \sum_{j\in\mathsf{CS}} \Delta_{j,S}$

$\mathsf{MaskedDec}[(\mathsf{ct}, S), i] := h_{i,S}$

$\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{UnDecHS}[(\mathsf{ct}, S)]\setminus\{i\}$

If $\mathsf{ct} = \mathsf{ct}^*$: $\mathcal{Q}_{\text{DEC}}[S] := \mathcal{Q}_{\text{DEC}}[S] \cup \{i\}$

Return $h_{i,S}$

Fig. 11. The game G_5 with changes highlighted in gray boxes.

and hence

$$\begin{aligned}\Delta_{i,S} &:= -\sum_{j\in\mathsf{HS}\setminus\{i\}}\Delta_{j,S}-\sum_{j\in\mathsf{CS}}\Delta_{j,S}=-\sum_{j\in\mathsf{HS}\setminus\{i\}}\left(\Delta^*_{j,S}-\mathbf{c}^\top(\lambda_{j,S}\mathbf{k}_j)-y_{j,S}\right)-\sum_{j\in\mathsf{CS}}\Delta_{j,S}\\ &=\mathbf{c}^\top\mathbf{k}-\sum_{j\in\mathsf{CS}\cup\{i\}}\mathbf{c}^\top(\lambda_{j,S}\mathbf{k}_j)+\sum_{j\in\mathsf{HS}\setminus\{i\}}y_{j,S}-\sum_{j\in\mathsf{HS}\setminus\{i\}}h_{j,S}-\sum_{j\in\mathsf{CS}}\Delta_{j,S}. \qquad (27)\end{aligned}$$

By substituting $y_{j,S}$ by $\mathsf{RandY}[(\mathsf{ct},S),j]$ and $h_{j,S}$ by $\mathsf{MaskedDec}[(\mathsf{ct},S),j]$ respectively, (27) is exactly the simulation of $\Delta_{i,S}$ in G_5 as shown in Fig. 11. Therefore, the simulation of $\Delta_{i,S}$ in $\mathsf{G}_{4.5}$ is identical to that in G_5.

Overall, we conclude that the views of $\mathcal{A}$ among G_4, $\mathsf{G}_{4.5}$ and G_5 are identical except it correctly guess $\mathsf{seed}_{i,i}$ for some $i\in\mathsf{Hon}$, which happens with probability less than $Q_{\mathsf{mask}}/2^\lambda$. Thus, $|\Pr[\mathsf{G}_4=1]-\Pr[\mathsf{G}_5=1]|\le\frac{Q_{\mathsf{mask}}}{2^\lambda}=\mathsf{negl}(\lambda)$. ▌

Game G_6: It is the same as G_5 except the computation $h_{i,S}$ for the last honest user, and $\mathsf{H}_{\mathsf{rand}}(\mathsf{seed}_{i,i},(\mathsf{ct},S))$ is not programmed until user i is corrupted. In G_6, when replying to $\mathcal{O}_{\mathrm{DEC}}(i,\mathsf{ct}=(\mathbf{c},d))$ and i is the last honest user, it first retrieves $(\mathbf{s},\mathbf{e},e'):=\mathsf{Rnd}[\mathsf{ct}]$ then computes

$$h_{i,S}:=\mathbf{s}^\top\mathsf{pk}+\mathbf{e}^\top\mathbf{k}-\mathbf{s}^\top\mathbf{f}+y_{i,S}+\textstyle\sum_{j\in\mathsf{HS}\setminus\{i\}}(y_{j,S}-h_{j,S})-\sum_{j\in\mathsf{CS}}(\Delta_{j,S}+\lambda_{j,S}\cdot\mathbf{c}^\top\mathbf{k}_j). \qquad (28)$$

Additionally, the computation of masks in $\mathcal{O}_{\mathrm{COR}}$ is changed accordingly.

Note that $\mathbf{s}^\top\mathsf{pk}-\mathbf{s}^\top\mathbf{f}+\mathbf{e}^\top\mathbf{k}=\mathbf{s}^\top(\mathbf{A}\mathbf{k}+\mathbf{f})-\mathbf{s}^\top\mathbf{f}+\mathbf{e}^\top\mathbf{k}=(\mathbf{s}^\top\mathbf{A}+\mathbf{e}^\top)\mathbf{k}=\mathbf{c}^\top\mathbf{k}$ is merely a rearrange. We have $\Pr[\mathsf{G}_6=1]=\Pr[\mathsf{G}_5=1]$.

Game G_7: It is the same as G_6, except the responses to $\mathcal{A}$'s $\mathcal{O}_{\mathrm{DEC}}$ and $\mathcal{O}_{\mathrm{COR}}$ queries. In G_7, when replying to $\mathcal{O}_{\mathrm{DEC}}(i,\mathsf{ct}=(\mathbf{c},d))$ and i is not the last honest user, the challenger only samples $h_{i,S}\leftarrow\!\!\$\ \mathcal{R}_q$ and returns it. If i is the last honest user, it samples $y_{k,S}\leftarrow D_{\mathcal{R},\sigma_y}$ for $k\in[|\mathsf{HS}|-1]$ and stores all the $\{y_{k,S}\}_{k\in[|\mathsf{HS}|-1]}$ to a table $\mathsf{UnuseY}[(\mathsf{ct},S)]$. Moreover, it samples a single $y\leftarrow D_{\mathcal{R},\sigma_y}$. Then it retrieves $(\mathbf{s},\mathbf{e},e'):=\mathsf{Rnd}[\mathsf{ct}]$ and computes

$$\begin{aligned}h_{i,S}:=&\mathbf{s}^\top\mathsf{pk}+\mathbf{e}^\top\mathbf{k}-\mathbf{s}^\top\mathbf{f}+y\\ &+\textstyle\sum_{k\in[|\mathsf{HS}|-1]}y_{k,S}-\sum_{j\in\mathsf{HS}\setminus\{i\}}h_{j,S}-\sum_{j\in\mathsf{CS}}(\Delta_{j,S}+\lambda_{j,S}\cdot\mathbf{c}^\top\mathbf{k}_j).\end{aligned} \qquad (29)$$

Additionally, the computation of masks in $\mathcal{O}_{\mathrm{COR}}$ is changed accordingly.

When replying to $\mathcal{A}$'s $\mathcal{O}_{\mathrm{COR}}(i)$ query, for $(\mathsf{ct},S)\in\mathsf{QDec}$ s.t. $i\in S$ and $\mathsf{UnDecHS}[(\mathsf{ct},S)]=\emptyset$, the challenger randomly picks $y\in\mathsf{UnuseY}[\mathsf{ct},S]$, sets $\mathsf{UnuseY}[\mathsf{ct},S]:=\mathsf{UnuseY}[\mathsf{ct},S]\setminus\{y\}$ and programs $\mathsf{H}_{\mathsf{rand}}(\mathsf{seed}_{i,i},(\mathsf{ct},S)):=y$.

More precisely, the challenger answers $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{DEC}}$ queries according to Fig. 12.

Note that the computation of $h_{i,S}$ in (29) is just a rename of that in (28), since (G_6) $\{y_{i,S}\}_{i\in\mathsf{HS}}\equiv\{y,y_{1,S},\ldots,y_{|\mathsf{HS}|-1,S}\}$ (G_7). And in G_6, there exists at least one $y_{i,S}$ for some $i\in\mathsf{HS}$ that is always hidden from $\mathcal{A}$, as it can corrupt at most $t-1$ users. This hidden $y_{i,S}$ is implicitly named by y in G_7. Consequently, we have $\Pr[\mathsf{G}_7=1]=\Pr[\mathsf{G}_6=1]$.

Game G_8: It is the same as G_7 except that now the secret key $\mathbf{k}_i$ is sampled uniformly over $\mathcal{R}_q^m$ only when the user i is corrupted.

$\underline{\mathsf{G}_7}$:

$\underline{\mathcal{O}_{\text{COR}}(i)}$:

If $\exists\ S \subseteq [N]$ s.t. $|\mathsf{Cor} \cup \mathcal{Q}_{\text{DEC}}[S]| \geq t-1$: Return $\perp$

For all $(\mathsf{ct} = (\mathbf{c}, d), S) \in \mathsf{QDec}$ s.t. $(i \in S) \wedge (\mathsf{Mask}[(\mathsf{ct}, S)] = \perp)$:

If $\mathsf{MaskedDec}[(\mathsf{ct}, S), i] \neq \perp$:

If $\mathsf{UnuseY}[(\mathsf{ct}, S)] \neq \perp$:

Pick $y \in \mathsf{UnuseY}[(\mathsf{ct}, S)]$, $\mathsf{UnuseY}[(\mathsf{ct}, S)] := \mathsf{UnuseY}[(\mathsf{ct}, S)] \setminus \{y\}$

$\mathsf{H}_{\mathsf{rand}}(\mathsf{seed}_{i,i}, (\mathsf{ct}, S)) := y$

Else: $y := \mathsf{H}_{\mathsf{rand}}(\mathsf{seed}_{i,i}, (\mathsf{ct}, S))$

$\mathsf{Mask}[(\mathsf{ct}, S), i] := \mathsf{MaskedDec}[(\mathsf{ct}, S), i] - \mathbf{c}^\top \lambda_{i,S} \mathbf{k}_i - y$

Else: $\widetilde{\mathsf{HS}} := \mathsf{UnDecHS}[(\mathsf{ct}, S)]$

If $\widetilde{\mathsf{HS}} \neq \{i\}$: $\mathsf{Mask}[(\mathsf{ct}, S), i] \leftarrow\!\!\$\ \mathcal{R}_q$

Else:

For $j \in \mathsf{CS}$: Parse $\mathsf{sk}_j = (\mathbf{k}_j, \{\mathsf{seed}_{j,k}, \mathsf{seed}_{k,j}\}_{k\in[N]})$

$\Delta_{j,S} := \sum_{k\in S\setminus\{j\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,k}, (\mathsf{ct}, S)) - \sum_{k\in S\setminus\{j\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{k,j}, (\mathsf{ct}, S)) \in \mathcal{R}_q$

For $k \in [|\mathsf{HS}| - 2]$: $y_{k,S} \leftarrow\!\!\$\ D_{\mathcal{R},\sigma_y}$ // $|\mathsf{HS}| \geq 2$

$\mathsf{UnuseY}[(\mathsf{ct}, S)] := \{y_{1,S}, \ldots, y_{|\mathsf{HS}|-2,S}\}$

$y \leftarrow\!\!\$\ D_{\mathcal{R},\sigma_y}$, $(\mathbf{s}, \mathbf{e}, e') := \mathsf{Rnd}[\mathsf{ct}]$

$\Delta_{i,S} := \mathbf{s}^\top \mathsf{pk} + \mathbf{e}^\top \mathbf{k} - \mathbf{s}^\top \mathbf{f} + y - \sum_{j\in\mathsf{CS}\cup\{i\}} \mathbf{c}^\top \cdot \lambda_{j,S} \cdot \mathbf{k}_j + \sum_{k\in[|\mathsf{HS}|-2]} y_{k,S}$
$- \sum_{j\in\mathsf{HS}\setminus\{i\}} \mathsf{MaskedDec}[(\mathsf{ct}, S), j] - \sum_{j\in\mathsf{CS}} \Delta_{j,S}.$

$\mathsf{Mask}[(\mathsf{ct}, S), i] := \Delta_{i,S}$

$\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{UnDecHS}[(\mathsf{ct}, S)] \setminus \{i\}$

For all $(\mathsf{ct}, S) \in \mathsf{QDec}$ s.t. $\mathsf{Mask}[(\mathsf{ct}, S), i] \neq \perp$: $\mathsf{ProgramZeroShare}((\mathsf{ct}, S), i, \mathsf{Mask}[(\mathsf{ct}, S), i])$

$\mathsf{Cor} := \mathsf{Cor} \cup \{i\}$, $\mathsf{Hon} := \mathsf{Hon} \setminus \{i\}$

Return sk_i

$\underline{\mathcal{O}_{\text{DEC}}(i, \mathsf{ct} = (\mathbf{c}, d), S)}$:

If $\mathsf{ct} \neq \mathsf{ct}^*$ and $\mathsf{ct} \notin \mathcal{Q}_{\text{ENC}}$: Return $\perp$

If $(\mathsf{ct}, S) \notin \mathsf{QDec}$:

$\mathsf{QDec} := \mathsf{QDec} \cup \{(\mathsf{ct}, S)\}$

$\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{HS}$, $\widetilde{\mathsf{HS}} := \mathsf{HS}$

If $(\mathsf{ct}, S) \in \mathsf{QDec}$: $\widetilde{\mathsf{HS}} := \mathsf{UnDecHS}[(\mathsf{ct}, S)]$

If $\widetilde{\mathsf{HS}} \neq \{i\}$: $h_{i,S} \leftarrow\!\!\$\ \mathcal{R}_q$

Else:

For $j \in \mathsf{CS}$: Parse $\mathsf{sk}_j = (\mathbf{k}_j, \{\mathsf{seed}_{j,k}, \mathsf{seed}_{k,j}\}_{k\in[N]})$

$\Delta_{j,S} := \sum_{k\in S\setminus\{j\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{j,k}, (\mathsf{ct}, S)) - \sum_{k\in S\setminus\{j\}} \mathsf{H}_{\mathsf{mask}}(\mathsf{seed}_{k,j}, (\mathsf{ct}, S)) \in \mathcal{R}_q$

For $k \in [|\mathsf{HS} - 1|]$: $y_{k,S} \leftarrow\!\!\$\ D_{\mathcal{R},\sigma_y}$ // $|\mathsf{HS}| \geq 1$

$\mathsf{UnuseY}[(\mathsf{ct}, S)] := \{y_{1,S}, \ldots, y_{|\mathsf{HS}-1|,S}\}$

$y \leftarrow\!\!\$\ D_{\mathcal{R},\sigma_y}$, $(\mathbf{s}, \mathbf{e}, e') := \mathsf{Rnd}[\mathsf{ct}]$

$h_{i,S} := \mathbf{s}^\top \mathsf{pk} + \mathbf{e}^\top \mathbf{k} - \mathbf{s}^\top \mathbf{f} + y + \sum_{k\in[|\mathsf{HS}|-1]} y_{k,S}$
$- \sum_{j\in\mathsf{HS}\setminus\{i\}} \mathsf{MaskedDec}[(\mathsf{ct}, S), j] - \sum_{j\in\mathsf{CS}} \lambda_{j,S} \mathbf{c}^\top \mathbf{k}_j - \sum_{j\in\mathsf{CS}} \Delta_{j,S}$

$\mathsf{MaskedDec}[(\mathsf{ct}, S), i] := h_{i,S}$, $\mathsf{UnDecHS}[(\mathsf{ct}, S)] := \mathsf{UnDecHS}[(\mathsf{ct}, S)] \setminus \{i\}$

If $\mathsf{ct} = \mathsf{ct}^*$: $\mathcal{Q}_{\text{DEC}}[S] := \mathcal{Q}_{\text{DEC}}[S] \cup \{i\}$

Return $h_{i,S}$

Fig. 12. The game G_7 with changes highlighted in gray boxes.

Note that $\mathcal{A}$ can obtain at most $t-1$ $\mathbf{k}_i$ from corruption queries. By Langrage interpolation, the map from t evaluations to a polynomial is a bijection, we have $\{\mathbf{k}_i\}_{i\in\mathsf{Cor}} \equiv \{\mathbf{k}'_i\}_{i\in\mathsf{Cor}}$, where $\mathbf{k}_i$ is the evaluation $f(i)$ of a random $t-1$-degree polynomial $f \in \mathcal{R}_q^m[Y]$ (G_7) and $\mathbf{k}'_i \leftarrow_{\$} \mathcal{R}_q^m$ (G_8). Thus, we have $\Pr[\mathsf{G}_8 = 1] = \Pr[\mathsf{G}_7 = 1]$.

Game G_9: It is the same as G_8 except that now we directly sample $\mathsf{pk} \leftarrow_{\$} \mathcal{R}_q^n$, instead of computing it by $\mathsf{pk} := \mathbf{A}\mathbf{k} + \mathbf{f}$ with $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}$ and $\mathbf{f} \leftarrow D_{\mathcal{R}^n,\sigma_k}$.

Claim 10. Assume $\mathcal{A}$ queries at most ℓ times of $\mathcal{O}_{\mathrm{DEC}}$, then $|\Pr[\mathsf{G}_9 = 1] - \Pr[\mathsf{G}_8 = 1]| \le \mathsf{Adv}^{\mathsf{AHMLWE}}_{(q,m,n,\sigma_k),(\ell,\sigma_y,\mathcal{V}),\mathcal{B}_2}(\lambda)$.

Proof. We prove Claim 10 by constructing a PPT algorithm $\mathcal{B}_2$ that against the $\mathsf{AHMLWE}^{\ell,\sigma_y,\mathcal{V}}_{q,m,n,\sigma_k}$ by simulating $\mathsf{G}_8/\mathsf{G}_9$ with $\mathcal{A}$.

Algorithm $\mathcal{B}_2$. $\mathcal{B}_2$ receives $\mathbf{A}, \mathbf{b} \in \mathcal{R}_q^n$ from the challenger of AHMLWE and wish to distinguish whether $\mathbf{b} = \mathbf{A}\mathbf{k} + \mathbf{f}$ with $\mathbf{k} \leftarrow D_{\mathcal{R}^m,\sigma_k}$ and $\mathbf{f} \leftarrow D_{\mathcal{R}^n,\sigma_k}$ or $\mathbf{b} \leftarrow_{\$} \mathcal{R}_q^n$. $\mathcal{B}_2$ sets $\mathsf{pp} := \mathbf{A}$ and $\mathsf{pk} := \mathbf{b}$. $\mathcal{B}_2$ simulates the game with $\mathcal{A}$ just as G_8, except the responses to $\mathcal{A}$'s $\mathcal{O}_{\mathrm{DEC}}, \mathcal{O}_{\mathrm{COR}}$ queries.

- When replying to $\mathcal{A}$'s $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}, S)$ query, $\mathcal{B}_2$ first checks if i is the last honest user in $\widetilde{\mathsf{HS}} := \mathsf{UnDecHS}[(\mathsf{ct}, S)]$. If i is not the honest user, it directly samples $h_{i,S} \leftarrow_{\$} \mathcal{R}_q$ and replies $h_{i,S}$ to $\mathcal{A}$. Otherwise, $\mathcal{B}_2$ first samples $y_{k,S} \leftarrow D_{\mathcal{R},\sigma_y}$ for $k \in [|\mathsf{HS}|-1]$ and stores all the $\{y_{k,S}\}_{k\in[|\mathsf{HS}|-1]}$ to a table $\mathsf{UnuseY}[(\mathsf{ct}, S)]$. Then $\mathcal{B}_2$ retrieves $(\mathbf{s}, \mathbf{e}, e') := \mathsf{Rnd}[\mathsf{ct}]$ and computes

$$h_{i,S} := \mathbf{s}^\top \mathsf{pk} + h + \textstyle\sum_{k\in[|\mathsf{HS}|-1]} y_{k,S} - \sum_{j\in\mathsf{HS}\setminus\{i\}} h_{j,S} - \sum_{j\in\mathsf{CS}} (\Delta_{j,S} + \lambda_{j,S} \cdot \mathbf{c}^\top \mathbf{k}_j), \tag{30}$$

  where the hint h is obtained by querying $\mathcal{O}_{\mathrm{HINT}}(-\mathbf{s}, \mathbf{e})$ and is excepted to be $\mathbf{e}^\top\mathbf{k} - \mathbf{s}^\top\mathbf{f} + y$ with $y \leftarrow D_{\mathcal{R},\sigma_y}$.
- When replying to $\mathcal{A}$'s $\mathcal{O}_{\mathrm{COR}}(i)$ query, $\mathcal{B}_2$ first samples $\mathsf{sk}_i \leftarrow_{\$} \mathcal{R}_q^m$. For $(\mathsf{ct}, S) \in \mathsf{QDec}$ s.t. $i \in S$ and $\mathsf{UnDecHS}[(\mathsf{ct}, S)] = \{i\}$, $\mathcal{B}_2$ first samples $y_{k,S} \leftarrow D_{\mathcal{R},\sigma_y}$ for $k \in [|\mathsf{HS}|-2]$ and stores all the $\{y_{k,S}\}_{k\in[|\mathsf{HS}|-2]}$ to a table $\mathsf{UnuseY}[(\mathsf{ct}, S)]$. Then $\mathcal{B}_2$ retrieves $(\mathbf{s}, \mathbf{e}, e') := \mathsf{Rnd}[\mathsf{ct}]$ and programs $\mathsf{H}_{\mathsf{mask}}$ s.t.

$$\Delta_{i,S} := \mathbf{s}^\top \mathsf{pk} + h + \textstyle\sum_{k\in[|\mathsf{HS}|-2]} y_{k,S} - \sum_{j\in\mathsf{HS}\setminus\{i\}} h_{j,S} - \sum_{j\in\mathsf{CS}} \Delta_{j,S} - \sum_{j\in\mathsf{CS}\cup\{i\}} \lambda_{j,S} \cdot \mathbf{c}^\top \mathbf{k}_j,$$

  where the hint h is obtained by querying $\mathcal{O}_{\mathrm{HINT}}(-\mathbf{s}, \mathbf{e})$ and is excepted to be $\mathbf{e}^\top\mathbf{k} - \mathbf{s}^\top\mathbf{f} + y$ with $y \leftarrow D_{\mathcal{R},\sigma_y}$. By the cauchy-schawaz bound, we have $\sum_{i\in[n+m]} \|(-\mathbf{s}, \mathbf{e})_i\|_1^2 \le d \cdot \|(-\mathbf{s}, \mathbf{e})\|_2^2 = d \cdot B_0'^2 = B_{\mathsf{hint}}$.
  For $(\mathsf{ct}, S) \in \mathsf{QDec}$ s.t. $i \in S$ and $\mathsf{UnDecHS}[(\mathsf{ct}, S)] = \emptyset$, $\mathcal{B}_2$ randomly picks $y \in \mathsf{UnuseY}[\mathsf{ct}, S]$, sets $\mathsf{UnuseY}[\mathsf{ct}, S] := \mathsf{UnuseY}[\mathsf{ct}, S] \setminus \{y\}$, programs $\mathsf{H}_{\mathsf{rand}}(\mathsf{seed}_{i,i}, (\mathsf{ct}, S)) := y$ and programs $\mathsf{H}_{\mathsf{mask}}$ s.t. $\Delta_{i,S} := h_{i,S} - y - \lambda_{i,S} \cdot \mathbf{c}^\top \mathbf{k}_i$. Finally, it updates $\mathsf{Hon} := \mathsf{Hon} \setminus \{i\}$ and $\mathsf{Cor} := \mathsf{Cor} \cup \{i\}$, and returns sk_i.

Since $h = \mathbf{e}^\top\mathbf{k} - \mathbf{s}^\top\mathbf{f} + y$, it holds that $h_{i,S}$ is identically distributed in both (29) and (30). The similar analysis applies to the computation of masks when replying $\mathcal{A}$'s corruption query $\mathcal{O}_{\mathrm{COR}}(i)$. Consequently, if $\mathbf{b} = \mathbf{A}\mathbf{k} + \mathbf{f}$, then $\mathcal{B}_2$

simulates G_8 perfectly. If $\mathbf{b} \leftarrow\!\!\$\ \mathcal{R}_q^n$, then $\mathcal{B}_2$ simulates G_9 perfectly. Thus, we have $|\Pr[\mathsf{G}_9 = 1] - \Pr[\mathsf{G}_8 = 1]| \leq \mathsf{Adv}^{\mathsf{AHMLWE}}_{(q,m,n,\sigma_k),(\ell,\sigma_y,\mathcal{V}),\mathcal{B}_2}(\lambda)$. ▮

Game G_{10}: It is the same as G_9 except that when generating the challenge ciphertext $\mathsf{ct}^* = (\mathbf{c}^*, d^*)$, the challenger samples $(\mathbf{c}^*, d^*) \leftarrow\!\!\$\ \mathcal{R}_q^{m+1}$ uniformly, instead of computing $\mathbf{c}^* := \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top, d^* := \mathbf{s}^\top \mathsf{pk} + e' + \lfloor \frac{q}{p} \rfloor \cdot \mu_\beta$, where $(\mathbf{s}, \mathbf{e}, e') \leftarrow D_{\mathcal{R}^n,\sigma_0} \times D_{\mathcal{R}^m,\sigma_0} \times D_{\mathcal{R},\sigma_0}$.

Note that in G_9, the public key $\mathsf{pk} \leftarrow\!\!\$\ \mathcal{R}_q$ is uniformly random over $\mathcal{R}_q$. Moreover, the randomness $(\mathbf{s}, \mathbf{e}, e')$ will not be used in G_{10}. This is because the randomness will be used only in the responses to $\mathcal{A}$'s $\mathcal{O}_{\mathrm{DEC}}(i, \mathsf{ct}^*, S)$ and $\mathcal{O}_{\mathrm{COR}}(i)$ and i is the last honest user in $\mathsf{UnDecHS}[(\mathsf{ct}^*, S)]$. However, this means $\mathcal{A}$ can decrypt ct^* and lead to a trivial attack, so that the challenger will directly return $\perp$ in such queries. Consequently, we can replace $\mathbf{s}^\top[\mathbf{A} \mid \mathsf{pk}] + [\mathbf{e}^\top \mid e']$, i.e., an MLWE instance w.r.t. $[\mathbf{A} \mid \mathsf{pk}]$ to a uniformly random vector in $\mathcal{R}_q^{m+1}$ as in G_{10}. It follows that $|\Pr[\mathsf{G}_9 = 1] - \Pr[\mathsf{G}_{10} = 1]| \leq \mathsf{Adv}^{\mathsf{MLWE}}_{q,m+1,n,\sigma_0,\mathcal{B}_3}(\lambda)$.

In G_{10}, we have $d^* \leftarrow\!\!\$\ \mathcal{R}_q$, so the challenge bit β is totally hidden from $\mathcal{A}$ and it holds $\Pr[\mathsf{G}_{10} = 1] = \frac{1}{2}$. Putting all things together, Theorem 3 follows. □

D Instantiations of NIZK from Lattices

In this section, we provide the NIZK constructions used in our TD_1, TD_2 schemes. We first recall the buliding blocks in Appendix D.1. We then instantiate the tag-based OTSS QA-NIZK needed in our TD_1 in Appendix D.2 and the simulation-extractable NIZK needed in our TD_2 in Appendix D.3.

D.1 Building Blocks

Definition 12 (Weakly Simulation-Sound for Tag-based QA-NIZK). *Let $\Pi = (\mathsf{CRS}, \mathsf{Prv}, \mathsf{Vfy}, \mathsf{Sim})$ be a tag-based QA-NIZK for $\mathcal{L}_\rho$ with tag space $\mathcal{T}$. We say that Π has weak simulation-soundness, if for any PPT $\mathcal{A}$, it holds $\mathsf{Adv}^{\mathsf{WSS}}_{\Pi,\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{WSS}}_{\Pi,\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$, where $\mathsf{Exp}^{\mathsf{WSS}}_{\Pi,\mathcal{A}}$ is defined in Fig. 13.*

$\underline{\mathsf{Exp}^{\mathsf{WSS}}_{\Pi,\mathcal{A}}(\lambda)}$	
$(\mathsf{crs}, \mathsf{st}) \leftarrow\!\!\$\ \mathsf{CRS}(1^\lambda)$	$\underline{\mathcal{O}_{\mathrm{SIM}}(x, \tau)}$
$\mathcal{Q} := \emptyset$	$(\pi, \mathsf{st}) \leftarrow\!\!\$\ \mathsf{Sim}(x, \tau, \mathsf{st})$
$(x^*, \tau^*, \pi^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{SIM}}}(\mathsf{crs})$	$\mathcal{Q} := \mathcal{Q} \cup \{(x, \tau)\}$
Return $\mathsf{Vfy}(\mathsf{crs}, x^*, \tau^*, \pi^*) = 1$	Return π
$\wedge (x^*, \tau^*) \notin \mathcal{Q} \wedge x^* \notin \mathcal{L}_\rho$	

Fig. 13. The weak simulation-soundness experiment $\mathsf{Exp}^{\mathsf{WSS}}_{\Pi,\mathcal{A}}$ for NIZK.

Definition 13 (One-Time Signature). *A one-time signature scheme $\mathsf{OTS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vfy})$ for message space $\{0,1\}^*$ is defined by three PPT algorithms:*

– $(\mathsf{vk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)$*: The key generation algorithm outputs a verification key* vk *and a signing key* sk*.*
– $\sigma \leftarrow_{\$} \mathsf{Sign}(\mathsf{sk}, m)$*: Given a signing key* sk *and a message* $m \in \{0,1\}^*$*, the signing algorithm outputs a signature* σ*.*
– $b \leftarrow \mathsf{Vfy}(\mathsf{vk}, m \in \{0,1\}^*, \sigma)$*: Given a verification key* vk*, a message* m *and a signature* σ*, the verification algorithm outputs* $b \in \{0,1\}$*.*

It is correct if for all $(\mathsf{vk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)$ *and all* $m \in \{0,1\}^*$*,*

$$\mathsf{Vfy}\big(\mathsf{vk}, m, \mathsf{Sign}(\mathsf{sk}, m)\big) = 1.$$

Definition 14 (Strong Unforgeability for One-Time Signature). *A one-time signature* $\mathsf{OTS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vfy})$ *is strongly unforgeable under a one-time chosen-message attack, if for any PPT* $\mathcal{A}$*, it holds that* $\mathsf{Adv}^{\mathsf{sEUF\text{-}CMA}}_{\mathsf{OTS},\mathcal{A}} := \Pr[\mathsf{Exp}^{\mathsf{sEUF\text{-}CMA}}_{\mathsf{OTS},\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*, where* $\mathsf{Exp}^{\mathsf{sEUF\text{-}CMA}}_{\mathsf{OTS},\mathcal{A}}$ *is defined in Fig. 14.*

$\underline{\mathsf{Exp}^{\mathsf{sEUF\text{-}CMA}}_{\mathsf{OTS},\mathcal{A}}(\lambda)}$	
$(\mathsf{vk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)$	$\underline{\mathcal{O}_{\text{SIGN}}(m)}$ //can be queried only once
$\mathcal{Q} := \emptyset$	$\sigma \leftarrow_{\$} \mathsf{Sign}(\mathsf{sk}, m)$
$(m^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\text{SIGN}}}(\mathsf{vk})$	$\mathcal{Q} := \mathcal{Q} \cup \{(m, \sigma)\}$
Return $\mathsf{Vfy}(\mathsf{vk}, m^*, \sigma^*) = 1$	*Return* σ
$\wedge\ (m^*, \sigma^*) \notin \mathcal{Q}$	

Fig. 14. The strong unforgeability experiment $\mathsf{Exp}^{\mathsf{sEUF\text{-}CMA}}_{\mathsf{OTS},\mathcal{A}}$ for One-Time Signature.

Definition 15 (Simulation-Secure PKE with Uniform Keys). *A PKE scheme* $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ *with public-key space* $\mathcal{PK}$ *and randomness space* $\mathcal{RD}$ *is simulation-secure with uniform keys if* PKE *has the following properties:*

– *Key Simulation.* $\{\mathsf{pk} : (\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{KGen}(1^\lambda)\} \approx_c \{\mathsf{pk} : \mathsf{pk} \leftarrow_{\$} \mathcal{PK}\}$.
– *Ciphertext simulation. There exists a PPT ciphertext simulator* $\mathsf{Sim.Enc}$ *such that, for* $\mathsf{pk} \leftarrow_{\$} \mathcal{PK}$*, no PPT adversary given* pk *can distinguish an oracle that, on each adaptively chosen message* m*, returns* $\mathsf{PKE.Enc}(\mathsf{pk}, m; \mathsf{rnd})$ *from one that returns* $\mathsf{PKE.Sim.Enc}(\mathsf{pk})$*.*

D.2 OTSS Tag-based QA-NIZK for $\mathcal{L}^{(\mathsf{MLWE})}_{\mathbf{A}}$

In this subsection, we construct a tag-based QA-NIZK for the language $\mathcal{L}^{(\mathsf{MLWE})}_{\mathbf{A}}$ that achieves one-time simulation-soundness (OTSS) in the random oracle model. Our construction builds on the lattice-based NIZK Lantern [41] for proving linear relations, which is proven to be weak simulation soundness by Blömer et al. [1]. To achieve OTSS, we provide a generic construction that transforms any tag-based weakly simulation-sound NIZK into a tag-based OTSS NIZK using a strongly unforgeable one-time signature. We first construct and prove this generic transformation, and then instantiate it with Lantern [41] and ML-DSA [21] to obtain the desired QA-NIZK for $\mathcal{L}^{(\mathsf{MLWE})}_{\mathbf{A}}$.

Theorem 5. *Suppose that* $\overline{\Pi} = \left(\overline{\Pi}.\mathsf{CRS}, \overline{\Pi}.\mathsf{Prv}, \overline{\Pi}.\mathsf{Vfy}, \overline{\Pi}.\mathsf{Sim}\right)$ *is a tag-based weakly simulation-sound NIZK for* $\mathcal{L}_\rho$ *and that signature* $\mathsf{OTS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vfy})$ *is strongly unforgeable under a one-time chosen-message attack. Then,* Π *shown in Fig. 15 is a tag-based one-time simulation-sound NIZK for* $\mathcal{L}_\rho$*. More precisely, for every PPT adversary* $\mathcal{A}$*, there exist PPT adversaries* $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *such that*

$$\mathsf{Adv}^{\mathsf{OTSS}}_{\Pi,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{WSS}}_{\overline{\Pi},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{sEUF\text{-}CMA}}_{\mathsf{OTS},\mathcal{B}_2}(\lambda).$$

$\underline{\mathsf{CRS}(\rho)}$:	$\underline{\mathsf{Vfy}(\mathsf{crs}, x, \tau, \pi)}$:
$\overline{\mathsf{crs}} \leftarrow\!\!\$\ \overline{\Pi}.\mathsf{CRS}(\rho)$	Parse $\pi = (\mathsf{vk}, \overline{\pi}, \sigma)$
Return $\mathsf{crs} := \overline{\mathsf{crs}}$	$\overline{\tau} := (\tau, \mathsf{vk})$
	Return $\overline{\Pi}.\mathsf{Vfy}(\mathsf{crs}, x, \overline{\tau}, \overline{\pi}) \wedge \mathsf{OTS}.\mathsf{Vfy}(\mathsf{vk}, (x, \overline{\pi}), \sigma)$
$\underline{\mathsf{Prv}(\mathsf{crs}, x, \tau, w)}$:	
$(\mathsf{vk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{OTS}.\mathsf{KGen}(1^\lambda)$	$\underline{\mathsf{Sim}(\mathsf{crs}, x, \tau)}$:
$\overline{\tau} := (\tau, \mathsf{vk})$	$(\mathsf{vk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{OTS}.\mathsf{KGen}(1^\lambda)$
$\overline{\pi} \leftarrow\!\!\$\ \overline{\Pi}.\mathsf{Prv}(\mathsf{crs}, x, \overline{\tau}, w)$	$\overline{\tau} := (\tau, \mathsf{vk})$, $\overline{\pi} \leftarrow\!\!\$\ \overline{\Pi}.\mathsf{Sim}(x, \overline{\tau})$
$\sigma \leftarrow\!\!\$\ \mathsf{OTS}.\mathsf{Sign}(\mathsf{sk}, (x, \overline{\pi}))$	$\sigma \leftarrow\!\!\$\ \mathsf{OTS}.\mathsf{Sign}(\mathsf{sk}, (x, \overline{\pi}))$
Return $\pi := (\mathsf{vk}, \overline{\pi}, \sigma)$	Return $\pi := (\mathsf{vk}, \overline{\pi}, \sigma)$

Fig. 15. Construction of a tag-based OTSS QA-NIZK Π from a tag-based QA-NIZK $\overline{\Pi}$ and a one-time signature OTS.

Proof of Theorem 5. Completeness of Π follows from the completeness of $\overline{\Pi}$ and correctness of OTS. Zero-knowledge follows directly from that of $\overline{\Pi}$.

It remains to prove one-time simulation soundness. When the adversary $\mathcal{A}$ issues a query of $\mathcal{O}_{\text{SIM}}(x, \tau)$, the challenger invokes $\pi = (\mathsf{vk}, \overline{\pi}, \sigma) \leftarrow\!\!\$\ \mathsf{Sim}(\mathsf{crs}, x, \tau)$ and returns π to $\mathcal{A}$ (note that the Sim algorithm can always generate valid proof π for any x). Then, it adds (x, τ, π) to $\mathcal{Q}_{\text{SIM}}$. Assume that at some moment, $\mathcal{A}$ outputs (x^*, τ^*, π^*) with $\pi^* = (\mathsf{vk}^*, \overline{\pi}^*, \sigma^*)$ such that

$$(x^* \notin \mathcal{L}_\rho) \wedge ((x^*, \tau^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}}) \wedge (\mathsf{Vfy}(\mathsf{crs}, x^*, \tau^*, \pi^*) = 1).$$

Define $\overline{\tau} := (\tau, \mathsf{vk})$ and $\overline{\tau}^* := (\tau^*, \mathsf{vk}^*)$ We consider the following two cases:

- $(x^*, \overline{\tau}^*) \neq (x, \overline{\tau})$. In this case, we can construct an adversary $\mathcal{B}_1$ to break the weak simulation soundness of $\overline{\Pi}$. To answer $\mathcal{A}$'s unique query (x, τ), it generates an OTS key pair $(\mathsf{vk}, \mathsf{sk})$ on its own, computes $\overline{\tau} = (\tau, \mathsf{vk})$, obtains $\overline{\pi}$ from its own $\mathcal{O}_{\text{SIM}}$ oracle on $(x, \overline{\tau})$, computes the signature σ using sk and returns $\pi := (\mathsf{vk}, \overline{\pi}, \sigma)$. Finally, it outputs $(x^*, \overline{\tau}^*, \overline{\pi}^*)$, where $(x^*, \overline{\tau}^*) \neq (x, \overline{\tau})$ and $\mathbf{x}^* \notin \mathcal{L}_\rho$. By the weak simulation soundness of $\overline{\Pi}$, $\mathcal{B}_1$ wins with probability at most $\mathsf{Adv}^{\mathsf{WSS}}_{\overline{\Pi},\mathcal{B}_1}(\lambda)$.
- $(x^*, \overline{\tau}^*) = (x, \overline{\tau})$. In this case, it holds that $x^* = x$, $\tau^* = \tau$, $\mathsf{vk}^* = \mathsf{vk}$ and $(\overline{\pi}^*, \sigma^*) \neq (\overline{\pi}, \sigma)$. It follows that $((x^*, \overline{\pi}^*), \sigma^*) \neq ((x, \overline{\pi}), \sigma)$, which can directly be reduced to the strong unforgeability of OTS.

In conclusion, $\mathcal{A}$ breaks the one-time simulation soundness game with only advantage less than $\mathsf{Adv}^{\mathsf{WSS}}_{\overline{\Pi},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{sEUF\text{-}CMA}}_{\mathsf{OTS},\mathcal{B}_2}(\lambda)$ and Theorem 5 follows.

Instantiation of One-Time Simulation Sound Tag-based QA-NIZK for $\mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$. We instantiate it with Lantern serving the tag-based weakly simulation-sound NIZK, and ML-DSA serving as the strong unforgeable signature scheme.

D.3 Simulation-Extractable NIZK for relation $\mathcal{R}^{(\mathsf{ct})}$

In this subsection, for completeness, we recall the generic construction of simulation-extractable NIZK in Fig. 16, the simulation-extractability of the generic construction in Theorem 6, and its instantiation from Hwang et al. [28].

$\underline{\mathsf{CRS}(1^\lambda)}$:	$\underline{\mathsf{Prv}(\mathsf{crs}, x, w)}$:
$\overline{\mathsf{crs}} \leftarrow\!\!\$\ \overline{\Pi}.\mathsf{CRS}(1^\lambda)$	$\mathsf{pk} := \mathsf{H}(\mathsf{seed})$
$\mathsf{seed} \leftarrow\!\!\$\ \{0,1\}^\lambda$	$(\mathsf{vk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{OTS.KGen}(1^\lambda)$
Return $\mathsf{crs} := (\overline{\mathsf{crs}}, \mathsf{seed})$	$(c, \mathsf{rnd}) \leftarrow\!\!\$\ \mathsf{PKE.Enc}(\mathsf{pk}, w)$
	$\overline{\tau} := \mathsf{vk}$
$\underline{\mathsf{Vfy}(\mathsf{crs}, x, \pi)}$:	$\overline{\pi} \leftarrow\!\!\$\ \overline{\Pi}.\mathsf{Prv}(\overline{\mathsf{crs}}, (x, \mathsf{pk}, c), \overline{\tau}, (w, \mathsf{rnd}))$
Parse $\mathsf{crs} = (\overline{\mathsf{crs}}, \mathsf{seed})$	$\sigma \leftarrow\!\!\$\ \mathsf{OTS.Sign}(\mathsf{sk}, (x, c, \overline{\pi}))$
Parse $\pi = (\mathsf{vk}, c, \overline{\pi}, \sigma)$	Return $\pi := (\mathsf{vk}, c, \overline{\pi}, \sigma)$
$\mathsf{pk} := \mathsf{H}(\mathsf{seed})$	
$b_1 \leftarrow \mathsf{OTS.Vfy}(\mathsf{vk}, (x, c, \overline{\pi}), \sigma)$	
$b_2 \leftarrow \overline{\Pi}.\mathsf{Vfy}(\mathsf{crs}, (x, \mathsf{pk}, c), \mathsf{vk}, \overline{\pi})$	
Return $b_1 \wedge b_2$	

Fig. 16. Construction of a simulation-extractable NIZK Π for relation $\mathcal{R}$ from a tag-based weakly simulation-sound NIZK $\overline{\Pi}$ relation $\overline{\mathcal{R}}$, a strongly unforgeable one-time signature OTS and a simulation-secure PKE with uniform keys.

Theorem 6 (Adapted from [28, Theorem 4]). *Suppose that $\overline{\Pi} = (\mathsf{CRS}, \mathsf{Prv}, \mathsf{Vfy})$ is a tag-based weakly simulation-sound NIZK for relation $\mathcal{R}'$ defined as follows:*

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

where $\mathcal{RD}$ is the encryption randomness space of PKE, the one-time signature $\mathsf{OTS} = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vfy})$ is strongly unforgeable under a one-time chosen-message attack, and public key encryption $\mathsf{PKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ is simulation-secure with uniform keys. Then, the NIZK $\Pi = (\mathsf{CRS}, \mathsf{Prv}, \mathsf{Vfy})$ constructed in Fig. 16 is simulation-extractable for relation $\mathcal{R}$.

Instantiation of Simulation-extractable NIZK for $\mathcal{R}^{(\mathsf{ct})}$. We instantiate it with Lantern, Regev public key encryption [49] and ML-DSA, with Regev public key encryption serving as the simulation-secure PKE scheme with uniform keys.

E Concrete Efficiency Estimation

E.1 Suggested Parameters and Efficiency Estimation of TD_1

Suggested Parameters. In Table 2, we present concrete parameter choices for different (N, t) tuples and different upper bounds on the number of oracle queries

ℓ. We also report the estimated efficiency of our TD_1, including the ciphertext size and the partial decryption size per party.

Table 2. Concrete parameters and estimated efficiency of TD_1 for 128-bit security and message space $\{0,1\}^{512}$. "P.D." abbreviates "Partial Decryption".

Threshold		Parameters								Ciphertext			P.D.
N	t	ℓ	n	m	$\log_2 q$	σ_k	σ_0	σ_1	σ_y	$\lvert(\mathbf{c},d)\rvert$	$\lvert\pi\rvert$	Total	$\lvert h_i\rvert$
8	4	2^{30}	9	31	62	2^{24}	2^6	2^{40}	2^{54}	62 KB	63 KB	125 KB	68 KB
16	2	2^{30}	9	29	62	2^{25}	2^6	2^{40}	2^{55}	58 KB	61 KB	119 KB	29 KB
16	8	2^{30}	10	32	67	2^{25}	2^6	2^{45}	2^{55}	69 KB	66 KB	135 KB	13.1 MB
8	4	2^{60}	12	38	78	2^{26}	2^6	2^{42}	2^{70}	96 KB	74 KB	170 KB	86 KB
16	2	2^{60}	12	33	82	2^{30}	2^6	2^{45}	2^{74}	87 KB	71 KB	158 KB	38 KB
16	8	2^{60}	13	38	84	2^{27}	2^6	2^{47}	2^{72}	102 KB	76 KB	178 KB	16.4 MB

The parameters in Table 2 satisfy all the requirements of the correctness and security (i.e., Theorem 1) of TD_1. Concretely, the parameters as chosen as follows:

- We set $\lambda = 128, d = 256, p = 4, \epsilon := 2^{-129}, \epsilon' = \epsilon/C, s \geq 2, C = \binom{N}{t-1}$, and choose prime $q \equiv 1 \bmod 2d$.
- We set $r = n + O(\log \lambda)$ by Lemma 6, s.t. $\mathbf{U} \in \mathcal{R}_q^{n \times r}$ has full rank with probability at least $1 - 2^{-\lambda}$ and set $a = n$.
- The bounds $B_{\mathbf{D}}$ and $B_{\mathcal{S}}$ are usually determined experimentally. Following analysis in [42, Section 5.2] and [30, Subsect. 4.4], we set $B_{\mathbf{D}} = (\sqrt{m} + \sqrt{a} + 2\sqrt{s})\sqrt{d}\sigma_{\mathsf{MLWE}}$ and $B_{\mathcal{S}} = (\sqrt{1} + \sqrt{m})\sqrt{d} \cdot \sqrt{C}\sigma_k$.
- By Lemma 7 and Lemma 8, we set $\sigma_k \geq \max(\sqrt{2}\sigma_{k_0} B_{\mathbf{D}}, \sqrt{2}\eta_{\epsilon'}(\mathcal{R}^m))$, $\sigma_{k_0} \geq \eta_{\epsilon'}(\mathcal{R}^s)$ and $\frac{1}{\sigma_{k_0}^2} + \frac{\ell d B'^2 B_{\mathbf{D}}^2}{\sigma_y^2} \leq \frac{1}{\sigma^2}$ with $\sigma \geq \frac{8}{\sqrt{\pi}} \cdot q^{\frac{1}{s}} \sqrt{d \ln(2sd(1+1/\epsilon))}$ and $\sigma_{k_0} \geq \eta_{\epsilon'}(\mathcal{R}^m)$.
- According to Lemma 1, the hardness of $\mathsf{SA\text{-}HMLWE}^{1,\sigma_1,\mathcal{S}}_{q,m,n,\sigma_0}$ can be reduced to $\mathsf{MLWE}_{\mathcal{R},q,m,n,\sigma'}$, where $\sigma' > 0$ satisfies $\frac{1}{\sigma_0^2} + \frac{B_{\mathcal{S}}^2}{\sigma_1^2} \leq \frac{1}{\sigma'^2 + 2\eta_\epsilon^2(\mathcal{R}^m)}$ and $\sigma' > \sqrt{2} \cdot \eta_\epsilon(\mathcal{R}^m)$.
- By Lemma 3, we set $B' \geq \beta\sqrt{(m+n)d} \cdot \sigma_0, B'_k \geq \beta\sqrt{md} \cdot \sigma_k, B_k \geq \alpha\sigma_k, B_1 \geq \alpha\sigma_1, B_y \geq \alpha\sigma_y$ for $\alpha = 6$ and $\beta = \frac{e^{1/5}}{\sqrt{2\pi}}$ to ensure that each bound holds with probability at least $1 - 2^{-\lambda}$.
- For correctness, we set $B_1 + \sqrt{C}(B'B'_k + B_y) \leq \lfloor \frac{q}{p} \rfloor / 2$.
- We find (q, m, n, σ') and $(q, r, a, \sigma_{\mathsf{MLWE}})$ that can achieve concrete 128-bits security of MLWE using the state-of-art lattice estimator.[17]

[17] `https://github.com/malb/lattice-estimator`

E.2 Suggested Parameters and Efficiency Estimation of TD_2

Suggested Parameters. In Table 3, we present concrete parameter choices for $N = 1024$, $t \leq 1024$ and $\ell = 2^{60}$. We also report the estimated efficiency of our TD_2, including the ciphertext size and the partial decryption size per party.

Table 3. Concrete parameters and estimated efficiency of TD_2 for 128-bit security and message space $\{0,1\}^{1536}$. "P.D." abbreviates "Partial Decryption".

Threshold		Parameters							Ciphertext			P.D.
N	t	ℓ	n	m	$\log_2 q$	σ_k	σ_0	σ_y	$\lvert(\mathbf{c}, d)\rvert$	$\lvert\pi\rvert$	total	$\lvert h_i\rvert$
1024	≤ 1024	2^{60}	9	9	60	2^5	2^2	2^{45}	19 KB	82 KB	101 KB	1.87 KB

The parameters in Table 3 satisfy all the requirements of the correctness and security (i.e.,Theorem 3 and Theorem 4) of TD_2. Concretely, the parameters as chosen as follows:

- We set $\lambda = 128, d = 256$, $p = 2^6$, and choose prime q.
- By Lemma 3, we set $B_0 \geq \alpha\sigma_0$, $B_y \geq \alpha\sigma_y$, $B'_0 \geq \beta\sqrt{(m+n)d}\sigma_0$, $B'_k \geq \beta\sqrt{(m+n)d}\sigma_k$, $B'_{\mathsf{E}} \geq \beta\sqrt{(m+n+1)d}\sigma_0$ with $\alpha = 6$ and $\beta = \frac{e^{1/5}}{\sqrt{2\pi}}$, to ensure that each bound holds with probability at least $1 - 2^{-\lambda}$.
- For correctness, we set $B'_0 B'_k + B_0 + \sqrt{N} B_y \leq \lfloor \frac{q}{p} \rfloor / 2$.
- According to Lemma 2, the hardness of $\mathsf{AHMLWE}^{\ell,\mathcal{V},\sigma_y}_{q,m,n,\sigma_k}$ assumption can be reduced to $\mathsf{MLWE}_{q,n,m,\sigma}$, where $\mathcal{V} := \{\mathbf{v} = (v_i)_{i\in[n+m]} \in \mathcal{R}^{n+m} : \sum_{i\in[n+m]} \|v_i\|_1^2 \leq B_{\mathsf{hint}}\}$ for $B_{\mathsf{hint}} := d \cdot B'^2_{\mathsf{E}}$ and $\sigma, \kappa > 0$ satisfies
  - $\sigma_y \geq \sqrt{4B_{\mathsf{hint}}} \cdot \eta_\epsilon(\mathbb{Z}^d)$, $\sigma \geq \sqrt{2} \cdot \eta_\epsilon(\mathbb{Z}^d)$,$\epsilon \in (0, \frac{1}{2})$;
  - $\frac{1}{\sigma^2} = 2 \cdot (\frac{1}{\sigma_k^2} + \frac{\kappa^2 \ell}{\sigma_y^2})$ for $\kappa := \sqrt{\frac{1 - \sqrt{1 - 4B_{\mathsf{hint}}\eta_\epsilon^2(\mathbb{Z}^d)/\sigma_y^2}}{2\eta_\epsilon^2(\mathbb{Z}^d)/\sigma_y^2}}$.
- We choose the parameters of the $\mathsf{MSIS}_{q,n,m+n,2B'_{\mathsf{E}}}$, $\mathsf{MLWE}_{q,m+1,n,\sigma_0}$ and $\mathsf{MLWE}_{q,n,m,\sigma}$ assumptions to ensure at least 128-bit security.

Table of Contents

Adaptively CCA-Secure Threshold Decryption from Lattices with Polynomial Modulus 1
Yuxin Zhang, Yunxiao Zhou, Shuai Han(✉), Shengli Liu(✉), and Xinyi Huang
1 Introduction 1
2 Preliminaries 12
2.1 Lattice Backgrounds 12
2.2 Secret Sharing 14
2.3 Definition of Threshold Decryption 15
3 Adaptive LHL with Noisy Linear Leakage 18
4 Adaptively CCA-Secure Asynchronous TD Scheme 22
4.1 TD_1: Adaptively CCA-Secure TD Scheme for Small N 22
4.2 $\mathsf{TD}_1^{\mathsf{rob}}$: Robustness Extension 24
5 Adaptively CCA-Secure Synchronous TD Scheme 25
5.1 $\mathsf{TD}_2^{\mathsf{smCPA}}$: Adaptively smCPA-Secure Scheme for Polynomial N . . 25
5.2 TD_2: Achieving Adaptive CCA-Security via BKW Transform . . . 27
A Additional Preliminaries 33
A.1 Collision-Resistant Hash Functions 33
A.2 Quasi-Adaptive Non-Interactive Zero-Knowledge 33
B Useful Lemmas for Proof of Lemma 8 34
C Postponed Proofs 36
C.1 Proof of Lemma 6 36
C.2 Proof of Theorem 1 37
C.3 Proof of Theorem 2 45
C.4 Proof of Theorem 3 46
D Instantiations of NIZK from Lattices 55
D.1 Building Blocks 55
D.2 OTSS Tag-based QA-NIZK for $\mathcal{L}_{\mathbf{A}}^{(\mathsf{MLWE})}$ 56
D.3 Simulation-Extractable NIZK for relation $\mathcal{R}^{(\mathsf{ct})}$ 58
E Concrete Efficiency Estimation 58
E.1 Suggested Parameters and Efficiency Estimation of TD_1 58
E.2 Suggested Parameters and Efficiency Estimation of TD_2 60