Almost Tight Multi-User Security under Adaptive Corruptions from LWE in the Standard Model

Shuai Han[1,2], Shengli Liu[1,2](✉), Zhedong Wang[1,2], and Dawu Gu[1]

[1] School of Electronic Information and Electrical Engineering,
Shanghai Jiao Tong University, Shanghai 200240, China
{dalen17,slliu,wzdstill,dwgu}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China

Abstract. In this work, we construct the *first* digital signature (SIG) and public-key encryption (PKE) schemes with almost tight multi-user security under adaptive corruptions based on the learning-with-errors (LWE) assumption in the standard model. Our PKE scheme achieves almost tight IND-CCA security and our SIG scheme achieves almost tight strong EUF-CMA security, both in the multi-user setting with adaptive corruptions. The security loss is quadratic in the security parameter λ, and independent of the number of users, signatures or ciphertexts. Previously, such schemes were only known to exist under number-theoretic assumptions or in classical random oracle model, thus vulnerable to quantum adversaries.

To obtain our schemes from LWE, we propose new frameworks for constructing SIG and PKE with a core technical tool named *probabilistic* quasi-adaptive hash proof system (pr-QA-HPS). As a new variant of HPS, our pr-QA-HPS provides *probabilistic* public and private evaluation modes that may toss coins. This is in stark contrast to the traditional HPS [Cramer and Shoup, Eurocrypt 2002] and existing variants like approximate HPS [Katz and Vaikuntanathan, Asiacrypt 2009], whose public and private evaluations are deterministic in their inputs. Moreover, we formalize a new property called evaluation indistinguishability by requiring statistical indistinguishability of the two probabilistic evaluation modes, even in the presence of the secret key. The evaluation indistinguishability, as well as other nice properties resulting from the probabilistic features of pr-QA-HPS, are crucial for the multi-user security proof of our frameworks under adaptive corruptions.

As for instantiations, we construct pr-QA-HPS from the LWE assumption and prove its properties with almost tight reductions, which admit almost tightly secure LWE-based SIG and PKE schemes under our frameworks. Along the way, we also provide new almost-tight reductions from LWE to multi-secret LWE, which may be of independent interest.

1 Introduction

Tight Security. In modern cryptography, the security of cryptographic primitives like digital signatures (SIG) and public-key encryptions (PKE) is established by security reductions. Roughly speaking, a reduction turns an efficient adversary $\mathcal{A}$ breaking the security of the considered scheme with running time $t_{\mathcal{A}}$ and advantage $\epsilon_{\mathcal{A}}$ into an efficient algorithm $\mathcal{B}$ solving some computationally hard problem with running time $t_{\mathcal{B}}$ and advantage $\epsilon_{\mathcal{B}}$, and establishes a relation $\epsilon_{\mathcal{A}}/t_{\mathcal{A}} \leq \ell \cdot \epsilon_{\mathcal{B}}/t_{\mathcal{B}}$, where ℓ is called the *security loss* factor.

Usually, ℓ is a large polynomial in the number of users, signatures and/or ciphertexts in a deployed system. When instantiating the scheme in a theoretically sound manner, we have to compensate the security loss ℓ by increasing key lengths, group sizes or vector dimensions of the scheme. However, it might not be clear at the time of deployment that how many users will be involved and how many signatures or ciphertexts will be generated in the lifetime of the cryptographic system. If the estimation is too small, the provided security guarantee will not be backed by the security proof. Therefore, it is desirable that ℓ is a small constant or a small polynomial in the security parameter λ. Such a security reduction is called a *tight* one or an *almost tight* one. We do not distinguish tightness and almost tightness, but we will detail the security loss in the security theorems and scheme comparisons to reflect almost tightness.

Multi-User Security under Adaptive Corruptions (MU^{c}). The standard security notion for SIG is existential unforgeability under chosen-message attacks ($\mathsf{EUF\text{-}CMA}$) and that for PKE is indistinguishability under chosen-plaintext/ciphertext attacks ($\mathsf{IND\text{-}CPA/CCA}$). Both of the security notions are defined in a single-user setting. However, in practice, SIG and PKE are usually deployed in multi-user (and multi-challenge for PKE) settings, and leave more opportunities to adversaries implementing new attacks. An important attack is *user corruption* in that the adversary takes full control of some users and of course their secret keys. This happens since some adversary may snatch secrets from some user by system hacking or from key exposure due to the user's bad key management. Therefore, it is reasonable for us to consider $\mathsf{EUF\text{-}CMA}$ and $\mathsf{IND\text{-}CPA/CCA}$ securities in the multi-user (and multi-challenge) setting under adaptive corruptions [6, 33], denoted by $\mathsf{MU}^{\mathsf{c}}\text{-}\mathsf{CMA}$ and $\mathsf{MUMC}^{\mathsf{c}}\text{-}\mathsf{CPA/CCA}$, respectively. For ease of exposition, we also refer to them in a unified way as the MU^{c} security.

Apart from the motivations for the security itself, another important reason for considering MU^{c} security is that it captures the actual security requirements of many cryptosystems that use SIG and/or PKE as building blocks. A well-known example is authenticated key exchange (AKE) protocols which use SIG to authenticate protocol transcripts and use key encapsulation mechanism (KEM) or PKE to encapsulate elements contributing to session keys. Standard AKE security models, such as the Bellare-Rogaway [9] and the (extended) Canetti-Krawczyk [17, 32] models, are in multi-user settings and allow adversaries to corrupt secret keys of some users. In particular, Bader et al. [6] present the first tightly $\mathsf{MU}^{\mathsf{c}}\text{-}\mathsf{CMA}$ secure SIG and tightly $\mathsf{MUMC}^{\mathsf{c}}\text{-}\mathsf{CPA}$ secure KEM (and PKE), and use them to construct the first tightly secure AKE protocol. Another example is signcryption, which can be built from SIG and PKE in various ways like "Encrypt-then-Sign", "Sign-then-Encrypt" and "Encrypt-and-Sign" [3]. The insider security model, which is concluded by Badertscher et al. [8] as the standard for signcryption and followed up by Bellare and Stepanovs [10], is also in multi-user settings and allows adaptive corruptions. In such scenarios, $\mathsf{MU^c}$-CMA security for SIG and $\mathsf{MUMC^c}$-$\mathsf{CPA/CCA}$ security for PKE play central roles. Tight $\mathsf{MU^c}$ security of SIG and PKE would lead to tight security of the applied cryptosystems.

On Achieving Tight $\mathsf{MU^c}$ Security. Due to their importance, SIG and PKE with tight $\mathsf{MU^c}$ security have become an active area recently, including impossibility results [7, 39] and feasibility constructions [6, 25, 33, 26, 20, 40, 27].

On the one hand, it is quite challenging to construct SIG and PKE with tight $\mathsf{MU^c}$ security. In general, single-user security can only non-tightly imply $\mathsf{MU^c}$ security by a guessing strategy, which incurs a security loss linear in the number of users. As shown by Bader et al. [7], it is even impossible to achieve tight $\mathsf{MU^c}$-CMA and tight $\mathsf{MUMC^c}$-$\mathsf{CPA/CCA}$ securities if the relation between public key and secret key satisfies certain properties, which are satisfied by many existing SIG and PKE schemes. Alternatively, if the signing algorithm of SIG is deterministic, tight $\mathsf{MU^c}$-CMA security is also impossible to achieve [39].

On the other hand, there are very few SIG and PKE constructions in the literature proved to have tight $\mathsf{MU^c}$ security, even in the random oracle (RO) model. To the best of our knowledge, SIG schemes in [6, 25, 26, 20, 40, 27] and PKE schemes in [6, 33, 27] are the only ones with tight $\mathsf{MU^c}$ security. Almost all of them base their security on number-theoretic assumptions, such as the Diffie-Hellman assumptions in cyclic groups or ϕ-hiding assumptions, which lead to insecurity in the presence of powerful quantum adversaries. The only exception is the SIG scheme of Pan and Wagner [20], which can be instantiated under either the learning-with-errors (LWE) or isogeny-based assumptions. However, their tight $\mathsf{MU^c}$-CMA security proof is based on the classical RO model, and it is left as an open problem in [20] to extend their approach in the quantum RO model, or even in the standard model. As for PKE, there is currently no construction with tight $\mathsf{MUMC^c}$-CCA security based on post-quantum assumptions, no matter in the RO model or in the standard model. This raises the following question:

Can we construct SIG and PKE schemes with tight $\mathsf{MU^c}$ security based on post-quantum assumptions (such as LWE) in the standard model?

Our Contributions. In this work, we answer the above question affirmatively.

- We present the *first* SIG and PKE schemes whose $\mathsf{MU^c}$ security can be almost tightly reduced to the LWE assumptions in the standard model. The security loss is quadratic in the security parameter λ. Our PKE scheme achieves almost tight $\mathsf{MUMC^c}$-CCA security, and our SIG scheme achieves almost tight $\mathsf{MU^c}$-CMA security with *strong* existential unforgeability, denoted by *strong* $\mathsf{MU^c}$-CMA security, which even guarantees the hardness for adversary to forge a new signature for an already signed message.
- We obtain our schemes by proposing new frameworks for tightly $\mathsf{MU^c}$ secure SIG and PKE. The core technical tool in our frameworks is a new variant of hash proof system (HPS) named *probabilistic* quasi-adaptive HPS (pr-QA-HPS), with new properties resulting from its probabilistic features.

We instantiate pr-QA-HPS from the LWE assumption and prove its properties with almost tight reductions, which is crucial for the almost tight $\mathsf{MU^c}$ security of the resulting SIG and PKE schemes.

- Along the way, we also provide new almost-tight reductions from LWE to multi-secret LWE, which serves as pivots for the almost tight $\mathsf{MU^c}$ security of our SIG and PKE schemes.

Technical Overview. In a recent work, Han, Liu and Gu [27] provided nice solutions to almost tightly $\mathsf{MU^c}$ secure SIG and PKE in the standard model, with the help of quasi-adaptive HPS (QA-HPS). Here "quasi-adaptive" means that the projection key of HPS may depend on the language for which HPS hash values are generated. Note that their frameworks apply only when QA-HPS has exact correctness and their framework for SIG also requires QA-HPS to be publicly verifiable. For the LWE-based cases, however, their frameworks (named HLG frameworks) do not work any more, because of the following obstacles.

– **Obstacle 1: There is no LWE-based QA-HPS with exact correctness.** It is not an easy task to instantiate (traditional) HPS under LWE, as there are many subtleties regarding the correctness (aka projectiveness) of HPS, let alone QA-HPS. Loosely speaking, HPS has two evaluation modes for computing HPS hash values, a public mode Pub using a projection key and a private mode Priv using a secret key. The (exact) correctness requires that the two evaluation models result in the same value for element in the language. Due to the noise inherent in LWE, it is hard (and even seems impossible) to achieve exact correctness. Instead, there are several attempts in the literature [24, 31, 11, 46, 29] to instantiate HPS under LWE by relaxing the exact correctness to *approximate correctness*, i.e., requiring only that the two evaluation models result in sufficiently close values. We refer to such HPS as *approximate HPS*. This is sufficient for the purpose of [31, 11, 46, 29], but it is insufficient for the HLG framework [27] in proving $\mathsf{MU^c}$ security. Similar to the Cramer-Shoup argument [19], the computations of HPS hash value need to be switched from one mode (e.g., the real scheme uses the public mode) to the other mode (e.g., the security proof uses the private mode), without being noticed by the adversary. However, in the $\mathsf{MU^c}$ security proof, the adversary can first see the evaluated hash value, then ask to corrupt the user and obtain its secret key. With the secret key, the adversary is able to recompute the hash value in the private mode and compare it with the obtained hash value. Thus, any difference between the evaluated hash values in the two modes will be caught by the adversary.

– **Obstacle 2: There is no LWE-based QA-HPS with public verification.** In the HLG framework, in order to construct $\mathsf{MU^c}$-CMA secure SIG, the QA-HPS is required to support public verification of hash values given an extra verification key. Such QA-HPS is termed as *publicly-verifiable QA-HPS* (PV-QA-HPS) in [27]. PV-QA-HPS is necessary for the public verification of their SIG [27], but it only has instantiations over pairing groups, as it relies on the pairing operations to accomplish the public verifiability of hash values. In the LWE setting, there is no counterpart to pairing operations, so it is hard to obtain PV-QA-HPS and the HLG framework does not apply.

To circumvent the above obstacles, we propose the concept of *probabilistic QA-HPS* and new approaches to tight MU^{c} security with the help of pr-QA-HPS.

(1) Probabilistic QA-HPS (pr-QA-HPS) from LWE. Recall that QA-HPS $= (\alpha(\cdot), \mathsf{Pub}, \mathsf{Priv})$ for NP-language $\mathcal{L} \subseteq \mathcal{X}$ is associated with a subset membership problem (SMP) so that $\{c \leftarrow_{\$} \mathcal{L}\} \overset{c}{\approx} \{c \leftarrow_{\$} \mathcal{X}\}$. Its projection function $\alpha(\cdot)$ maps a secret key sk to a projection key $pk = \alpha(sk)$, its public evaluation algorithm $\mathsf{Pub}(pk, c, w)$ computes the hash value $\Lambda_{sk}(c)$ for $c \in \mathcal{L}$ with witness w, and its private evaluation algorithm $\mathsf{Priv}(sk, c)$ computes the hash value $\Lambda_{sk}(c)$ for $c \in \mathcal{X}$. The (exact) correctness asks that $\mathsf{Pub}(pk, c, w) = \mathsf{Priv}(sk, c) = \Lambda_{sk}(c)$ for all $c \in \mathcal{L}$ with witness w.

Now we consider the LWE case. All the LWE samples for matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ and error bound B constitute an NP-language

$$\mathcal{L}_{\mathbf{A}} := \{\mathbf{c} = \mathbf{A}^{\top}\mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}. \quad (1)$$

Then the LWE problem just serves as the SMP for $\mathcal{L}_{\mathbf{A}}$. Now we define $sk = \mathbf{k} \in \{0,1\}^m$, $pk = \mathbf{p} = \mathbf{Ak}$ and the hash value of instance $\mathbf{c} \in \mathbb{Z}_q^m$ is $\Lambda_{\mathbf{k}}(\mathbf{c}) := \mathbf{c}^{\top}\mathbf{k} \in \mathbb{Z}_q$. However, with $pk = \mathbf{p}$ and witness $(\mathbf{s}, \mathbf{e})$, public evaluation can only obtain a value like $\mathbf{s}^{\top}\mathbf{p} = \mathbf{s}^{\top}(\mathbf{Ak})$, which is hardly equal but close to $\Lambda_{\mathbf{k}}(\mathbf{c}) = \mathbf{c}^{\top}\mathbf{k} = (\mathbf{s}^{\top}\mathbf{A} + \mathbf{e}^{\top})\mathbf{k}$.

To circumvent the problem of lacking exact correctness, we put forward a new variant of QA-HPS, called *probabilistic QA-HPS* (pr-QA-HPS). In stark contrast to the traditional HPS [19] and variants like approximate HPS [31] or QA-HPS [28], whose public and private modes are deterministic in their inputs, our pr-QA-HPS has *probabilistic* public and private modes (denoted by prPub and prPriv, respectively), the outputs of which are probabilistic distributions over the hash value space. Instead of requiring exact correctness, we require the statistical indistinguishability of the two probabilistic evaluation modes, *even in the presence of the secret key*. We formalize this as the property of *evaluation indistinguishability*. See Definition 7 in Sect. 3 for the formal definition.

The property of evaluation indistinguishability enables the switch of evaluation mode from one to the other in a statistically indistinguishable way, even in the view of adversaries who can implement corruption attacks and obtain the secret key, thus serving well for our MU^{c} security proof, as shown later.

Below we give an overview of our LWE-based pr-QA-HPS. Let B and B' be error bounds satisfying $B' \geq mB \cdot 2^{\omega(\log \lambda)}$ with λ the security parameter.

- The secret key is $sk = \mathbf{k} \in \{0,1\}^m$, and for language $\mathcal{L}_{\mathbf{A}} = \{\mathbf{c} = \mathbf{A}^{\top}\mathbf{s} + \mathbf{e} \mid \mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m\}$, the projection key is $pk = \mathbf{p} = \mathbf{Ak} \in \mathbb{Z}_q^n$.
- The hash value of an instance $\mathbf{c} \in \mathbb{Z}_q^m$ is defined by $\Lambda_{\mathbf{k}}(\mathbf{c}) := \mathbf{c}^{\top}\mathbf{k} \in \mathbb{Z}_q$.
- For an instance $\mathbf{c} = \mathbf{A}^{\top}\mathbf{s} + \mathbf{e}$ in the language $\mathcal{L}_{\mathbf{A}}$, the *probabilistic* public evaluation mode prPub generates a hash value by first sampling a random

value $e' \leftarrow_{\$} [-B', B']$ uniformly, then computing $\mathbf{s}^\top\mathbf{p}+e'$ using the projection key $pk = \mathbf{p}$ and the witness $\mathbf{s}$ for $\mathbf{c} \in \mathcal{L}_{\mathbf{A}}$. Namely,

$$\mathbf{s}^\top\mathbf{p} + e' \leftarrow_{\$} \mathsf{prPub}(\mathbf{p}, \mathbf{c}, \mathbf{s}) \quad \text{with} \quad e' \leftarrow_{\$} [-B', B']. \tag{2}$$

- For an instance $\mathbf{c} \in \mathbb{Z}_q^m$ (no matter in $\mathcal{L}_{\mathbf{A}}$ or not), the *probabilistic* private evaluation mode prPriv generates a hash value by first sampling a random value $e' \leftarrow_{\$} [-B', B']$ uniformly, then computing $\mathbf{c}^\top\mathbf{k} + e'$ using the secret key $sk = \mathbf{k}$. Namely,

$$\mathbf{c}^\top\mathbf{k} + e' \leftarrow_{\$} \mathsf{prPriv}(\mathbf{k}, \mathbf{c}) \quad \text{with} \quad e' \leftarrow_{\$} [-B', B']. \tag{3}$$

That is to say, the HPS hash function $\Lambda_{\mathbf{k}}$ is still deterministic, while there are two probabilistic ways to evaluate it. Our LWE-based pr-QA-HPS has evaluation indistinguishability, since the bigger noise e' smudges the small error to make the statistical distance between the two probabilistic modes negligibly small:

$$\Delta(\mathbf{s}^\top\mathbf{p} + e', \mathbf{c}^\top\mathbf{k} + e') = \Delta(\cancel{\mathbf{s}^\top\mathbf{A}\mathbf{k}} + e', \cancel{\mathbf{s}^\top\mathbf{A}\mathbf{k}} + \mathbf{e}^\top\mathbf{k} + e') \leq mB/B' \leq 2^{-\omega(\log\lambda)}.$$

(2) New Framework for Constructing SIG with pr-QA-HPS (from LWE). In the HLG framework for SIG, QA-HPS is required to support public verification of hash values with an extra verification key (i.e., the so-called *publicly-verifiable* QA-HPS), since a QA-HPS hash value is part of the signature. However, in order to instantiate such QA-HPS, they rely on the pairing operations, which have no counterpart in the LWE setting.

In our case, it seems very hard to define an extra verification key vk for our aforementioned LWE-based pr-QA-HPS, so that the correctness of hash values in (2) or (3) can be publicly checked with vk.[1]

To circumvent the problem, we propose a new framework for SIG. Instead of requiring the public verifiability of hash values from QA-HPS, we resort to tag-based quasi-adaptive non-interactive zero-knowledge argument (QA-NIZK) [30] and augment the HPS hash value verification to QA-NIZK. Meanwhile, we also make use of dual-mode commitment, which has two computationally indistinguishable modes (i.e., a binding mode and a hiding mode), to bind the signing key and the verification key of SIG.

Below is our new framework for SIG from pr-QA-HPS = $(\alpha(\cdot), \mathsf{prPub}, \mathsf{prPriv})$, dual-mode commitment Com and QA-NIZK = $(\mathsf{Prove}, \mathsf{Vrfy})$, where QA-NIZK is for the language $\mathcal{L}_{\mathsf{QANIZK}} :=$

$$\Big\{(\mathbf{c}, vk, d) \ \Big| \ \exists(\mathbf{k}, r, e' \in [-B', B']), \text{s.t.}\, \mathbf{c} \in \mathcal{L}_{\mathbf{A}} \wedge vk = \mathsf{Com}(\mathbf{k}; r) \wedge d = \mathbf{c}^\top\mathbf{k} + e'\Big\}. \tag{4}$$

- The signing key $sigk = (\mathbf{k}, r)$ contains the secret key $\mathbf{k}$ of pr-QA-HPS and random coins r, and the verification key is the commitment $vk = \mathsf{Com}(\mathbf{k}; r)$.

[1] Of course, we cannot simply set sk to vk, since vk is public and the properties of (pr-)QA-HPS should not be harmed in the presence of vk.

– The signature for message m is given by $\sigma :=$

$$\big(\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}},\ d \leftarrow_{\$} \mathsf{prPriv}(\mathbf{k}, \mathbf{c}),\ \pi \leftarrow_{\$} \mathsf{Prove}(\text{tag} = m, (\mathbf{c}, vk, d), (\mathbf{k}, r, e'))\ \big).$$

– The verification of $(m, \sigma = (\mathbf{c}, d, \pi))$ is just the QA-NIZK verification.

Now we roughly sketch the proving idea for the strong $\mathsf{MU}^{\mathsf{c}}\text{-}\mathsf{CMA}$ security of our SIG. We aim to show that the fresh message-signature pair $(m^*, \sigma^* = (\mathbf{c}^*, d^*, \pi^*))$ forged by the adversary hardly passes the verification of QA-NIZK, even if the adversary can query messages for signatures via a signing oracle and corrupt the signing keys of some users.

- To generate signature $\sigma = (\mathbf{c}, d, \pi)$ for message m, the signing oracle invokes the simulator of QA-NIZK using a simulation trapdoor, instead of invoking algorithm Prove using the witness $(\mathbf{k}, r, e')$, to generate the proof π. This change is indistinguishable due to the zero-knowledge of QA-NIZK.
- To generate signature $\sigma = (\mathbf{c}, d, \pi)$ for message m, the signing oracle switches the language from $\mathcal{L}_{\mathbf{A}}$ to $\mathcal{L}_{\mathbf{A}_0}$, where $\mathbf{A}$ and $\mathbf{A}_0$ are uniformly and independently chosen. That is, it samples $\mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_0}$ instead of $\mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}}$. Note that $\mathcal{L}_{\mathbf{A}}$ is still used to determine the language $\mathcal{L}_{\mathsf{QANIZK}}$ in (4). By the LWE assumption, $(\mathbf{A}, \mathbf{A}^\top \mathbf{s} + \mathbf{e}) \overset{c}{\approx} (\mathbf{A}, \mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^m) \overset{c}{\approx} (\mathbf{A}, \mathbf{A}_0^\top \mathbf{s} + \mathbf{e})$, so this change is indistinguishable.
  Consequently, by the evaluation indistinguishability of pr-QA-HPS, the generation of $d \leftarrow_{\$} \mathsf{prPriv}(\mathbf{k}, \mathbf{c})$ can be changed to $d \leftarrow_{\$} \mathsf{prPub}(\mathbf{p}_0 = \mathbf{A}_0 \mathbf{k}, \mathbf{c}, \mathbf{s})$, where $\mathbf{s}$ is the witness for $\mathbf{c} = \mathbf{A}_0^\top \mathbf{s} + \mathbf{e} \in \mathcal{L}_{\mathbf{A}_0}$. This holds even if the adversary corrupts the user and obtains its signing key $\mathbf{k}$.
- The binding property of commitment makes sure that the unbounded simulation-soundness (USS) of QA-NIZK applies to the forged signature $\sigma^* = (\mathbf{c}^*, d^*, \pi^*)$. So a successful forgery for a target user must satisfy that $\mathbf{c}^* \in \mathcal{L}_{\mathbf{A}}$ and d^* lies close to $\mathbf{c}^{*\top} \mathbf{k} = (\mathbf{s}^{*\top} \mathbf{A} + \mathbf{e}^{*\top}) \mathbf{k}$, where $(\mathbf{s}^*, \mathbf{e}^*)$ is the witness for $\mathbf{c}^* = \mathbf{A}^\top \mathbf{s}^* + \mathbf{e}^* \in \mathcal{L}_{\mathbf{A}}$ and $\mathbf{k}$ is the signing key of the target user.
- The dual-mode commitment is switched to the hiding mode, then vk does not leak information about the secret key $\mathbf{k}$. Now all information about $\mathbf{k}$ learned by the adversary is bounded by $\mathbf{A}_0 \mathbf{k}$, if the adversary never corrupts the target user to obtain its signing key $\mathbf{k}$. When $m = 2n \log q + \omega(\log \lambda)$, there is still $n \log q + \omega(\log \lambda)$ bits of information left in $\mathbf{k}$. Taking $\mathbf{A}$ as an extractor, then $\mathbf{Ak}$ is statistically close to the uniform distribution (this is characterized as the $\langle \mathcal{L}_{\mathbf{A}_0}, \mathcal{L}_{\mathbf{A}} \rangle$*-one-time-extracting* property of pr-QA-HPS). As a result, the adversary can hardly forge a d^* such that d^* lies close to $\mathbf{c}^{*\top} \mathbf{k} = \mathbf{s}^{*\top} \mathbf{A} \mathbf{k} + \mathbf{e}^{*\top} \mathbf{k}$.[2] Then strong $\mathsf{MU}^{\mathsf{c}}\text{-}\mathsf{CMA}$ security follows.

Overall, the strong $\mathsf{MU}^{\mathsf{c}}\text{-}\mathsf{CMA}$ security proof is accomplished by the evaluation indistinguishability & $\langle \mathcal{L}_{\mathbf{A}_0}, \mathcal{L}_{\mathbf{A}} \rangle$-one-time-extracting property of pr-QA-HPS, SMP, zero-knowledge & USS of QA-NIZK, and indistinguishability of binding and hiding modes of commitment. Due to the nice properties of pr-QA-HPS,

[2] The bad case that $\mathbf{s}^* = \mathbf{0}$ has been excluded in the language $\mathcal{L}_{\mathbf{A}}$, see Footnote 6 for more details. We forgo making this explicit for the sake of simplicity.

we stress that all reduction algorithms can generate the signing keys of all users themselves, and hence can deal with adaptive corruptions by the adversary.

(3) Extending the HLG Framework for Constructing PKE with pr-QA-HPS (from LWE). The HLG framework for PKE needs the exact correctness of QA-HPS. To circumvent the obstacle in the LWE setting, we extend their framework by replacing QA-HPS with our pr-QA-HPS and augmenting error-correction code $\mathsf{ECC} = (\mathsf{Encode}, \mathsf{Decode})$ to deal with the LWE errors. Below is our extended framework.

- The secret key of PKE is just the secret key $sk = \mathbf{k}$ of pr-QA-HPS, and the public key is the projection key $pk = \mathbf{p} = \mathbf{A}\mathbf{k}$.
- The encryption of message m results in the ciphertext $ct :=$

  $$(\ \mathbf{c} \leftarrow_{\$} \mathcal{L}_{\mathbf{A}}, \ d \leftarrow_{\$} \mathsf{prPub}(\mathbf{p}, \mathbf{c}, \mathbf{s}) + \mathsf{Encode}(m), \ \pi \leftarrow_{\$} \mathsf{Prove}(\mathrm{tag}, \mathbf{c}, (\mathbf{s}, \mathbf{e})) \ ),$$

  where tag is a collision-resistant hashing of (pk, d) and QA-NIZK $= (\mathsf{Prove}, \mathsf{Vrfy})$ is for the language $\mathcal{L}_{\mathbf{A}}$ in (1).
- The decryption of $ct = (\mathbf{c}, d, \pi)$ needs a successful verification of π by Vrfy and then the computation of $m := \mathsf{Decode}(d - \mathsf{prPriv}(\mathbf{k}, \mathbf{c}))$.

Now we sketch the proving idea for the $\mathsf{MUMC^c}$-CCA security of our PKE. We aim to show that the multiple challenge ciphertexts (may under different public keys) $\{ct^* = (\mathbf{c}^*, d^*, \pi^*)\}$ for plaintexts $\{m_0\}$ are indistinguishable from those for $\{m_1\}$, even if the adversary has access to a decryption oracle and can corrupt the secret keys of some users (but not those for the challenge ciphertexts).

- To generate challenge ciphertexts $\{ct^* = (\mathbf{c}^*, d^*, \pi^*)\}$ for plaintexts $\{m_b\}$ with $b \in \{0, 1\}$, the encryption oracle switches public evaluation $\mathsf{prPub}(\mathbf{p}, \mathbf{c}^*, \mathbf{s}^*)$ to the private one $\mathsf{prPriv}(\mathbf{k}, \mathbf{c}^*)$ for the computation of d^*, so

  $$d^* \leftarrow_{\$} \mathsf{prPriv}(\mathbf{k}, \mathbf{c}^*) + \mathsf{Encode}(m_b) = \mathbf{c}^{*\top}\mathbf{k} + e' + \mathsf{Encode}(m_b). \qquad (5)$$

  Clearly pr-QA-HPS ensures the evaluation indistinguishability. Then the witness for $\mathbf{c}^* \in \mathcal{L}_{\mathbf{A}}$ is not needed any more, and the proof π^* can be computed by the simulator of QA-NIZK, instead of algorithm Prove. This change is indistinguishable due to the zero-knowledge of QA-NIZK.
- To generate challenge ciphertexts $\{ct^* = (\mathbf{c}^*, d^*, \pi^*)\}$, the encryption oracle switches the language from $\mathcal{L}_{\mathbf{A}}$ to $\mathcal{L}_{\mathbf{A}_0}$. That is, it samples $\mathbf{c}^* \leftarrow_{\$} \mathcal{L}_{\mathbf{A}_0}$ instead of $\mathbf{c}^* \leftarrow_{\$} \mathcal{L}_{\mathbf{A}}$. By the LWE assumption, $\{\mathbf{A}^\top \mathbf{s}^* + \mathbf{e}^*\} \overset{c}{\approx} \{\mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^m\} \overset{c}{\approx} \{\mathbf{A}_0^\top \mathbf{s}^* + \mathbf{e}^*\}$, so this change is indistinguishable.
  Consequently, for $\mathbf{c}^* = \mathbf{A}_0^\top \mathbf{s}^* + \mathbf{e}^* \in \mathcal{L}_{\mathbf{A}_0}$, (5) can be changed to

  $$d^* \leftarrow_{\$} \mathsf{prPub}(\mathbf{p}_0 = \mathbf{A}_0\mathbf{k}, \mathbf{c}^*, \mathbf{s}^*) + \mathsf{Encode}(m_b) = \mathbf{s}^{*\top}\mathbf{A}_0\mathbf{k} + e' + \mathsf{Encode}(m_b)$$

  due to the evaluation indistinguishability of pr-QA-HPS.
- To decrypt a ciphertext $ct = (\mathbf{c}, d, \pi)$, the decryption oracle rejects ct if $\mathbf{c} \notin \mathcal{L}_{\mathbf{A}}$. This change is indistinguishable, since π hardly passes the verification

of QA-NIZK when $\mathbf{c} \notin \mathcal{L}_{\mathbf{A}}$, thanks to the USS of QA-NIZK. Then due to the evaluation indistinguishability of pr-QA-HPS, the decryption of $ct = (\mathbf{c}, d, \pi)$ with $\mathbf{c} \in \mathcal{L}_{\mathbf{A}}$ can be done with prPub so that

$$m := \mathsf{Decode}(d - \mathsf{prPub}(\mathbf{p}, \mathbf{c}, \mathbf{s})) = \mathsf{Decode}(d - \mathbf{s}^\top \mathbf{A}\mathbf{k} - e'). \qquad (6)$$

- Now for any user i, let its secret key be $\mathbf{k}^{(i)}$. The public key and decryption oracle only leak $\mathbf{A}\mathbf{k}^{(i)}$ via $pk^{(i)} = \mathbf{p}^{(i)} = \mathbf{A}\mathbf{k}^{(i)}$ and (6). When $m = 2n \log q + \omega(\log \lambda)$, there is still $n \log q + \omega(\log \lambda)$ bits of information left in $\mathbf{k}^{(i)}$. Taking $\mathbf{A}_0$ as an extractor, then $\mathbf{A}_0\mathbf{k}^{(i)}$ is uniform (this is characterized by the $\langle \mathcal{L}_{\mathbf{A}}, \mathcal{L}_{\mathbf{A}_0} \rangle$*-key switching* property of pr-QA-HPS). So when computing ct^*, we have

$$(\mathbf{c}^{*\top} = \mathbf{s}^{*\top}\mathbf{A}_0 + \mathbf{e}^{*\top}, \mathbf{s}^{*\top}\mathbf{A}_0\mathbf{k}^{(i)} + e') \overset{s}{\approx} \mathbf{s}^{*\top}(\mathbf{A}_0 | \mathbf{a}^{(i)}) + (\mathbf{e}^{*\top} | e') \overset{c}{\approx} \mathbf{u}^{(i)},$$

  where $\mathbf{a}^{(i)} \leftarrow_{\$} \mathbb{Z}_q^n$, $\mathbf{u}^{(i)} \leftarrow_{\$} \mathbb{Z}_q^{m+1}$, and the last step is due to the LWE assumption. Therefore, we can use a random element, instead of $\mathsf{prPub}(\mathbf{p}_0, \mathbf{c}^*, \mathbf{s}^*)$, to perfectly hide m_b in d^* (this is characterized by the $\mathcal{L}_{\mathbf{A}_0}$*-multi-key multi-extracting* property of pr-QA-HPS), and the $\mathsf{MUMC^c\text{-}CCA}$ security follows.

Overall, the $\mathsf{MUMC^c\text{-}CCA}$ security proof is accomplished by the evaluation indistinguishability & $\langle \mathcal{L}_{\mathbf{A}}, \mathcal{L}_{\mathbf{A}_0} \rangle$-key switching & $\mathcal{L}_{\mathbf{A}_0}$-multi-key multi-extracting property of pr-QA-HPS, SMP, zero-knowledge & USS of QA-NIZK. Due to the nice properties of pr-QA-HPS, all reduction algorithms can generate the secret keys of all users themselves, and hence can deal with adaptive corruptions.

(4) Almost Tight $\mathsf{MU^c}$ Security from Reduction for Multi-Secret LWE. In the $\mathsf{MU^c}$ security model for SIG/PKE, there are multiple signing queries/multiple challenge ciphertexts. Therefore, we need multi-fold SMP requiring that

$$(\mathbf{A}, \mathbf{S}\mathbf{A} + \mathbf{E}) \overset{c}{\approx} (\mathbf{A}, \mathbf{U}) \qquad (7)$$

with $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n \times m}$, $\mathbf{S} \leftarrow_{\$} \mathbb{Z}_q^{Q \times n}$ and $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{Q \times m}$, which is in fact the *multi-secret LWE*. We show that the LWE assumption almost tightly implies multi-secret LWE, i.e., (7). The idea is inspired by [2]. Firstly, $\mathbf{A}$ can be divided into the first column $\mathbf{A}_1 \in \mathbb{Z}_q^n$ and the rest, which is denoted by $\mathbf{A}_2 \in \mathbb{Z}_q^{n \times (m-1)}$. Then $\mathbf{A}_2$ can be sampled with a lossy sampler $\mathbf{A}_2 := \mathbf{C}\mathbf{B} + \mathbf{F}$, where $\mathbf{C} \leftarrow_{\$} \mathbb{Z}_q^{n \times \ell}$, $\mathbf{B} \leftarrow_{\$} \mathbb{Z}_q^{\ell \times (m-1)}$, $\mathbf{F} \in \mathbb{Z}_q^{n \times (m-1)}$ follows the error distribution and $\ell < n$. This change is indistinguishable based on the LWE assumption, with a reduction loss n by a standard hybrid argument. With a lossy $\mathbf{A}_2$, $\mathbf{S}\mathbf{A}_2$ does not leak too much information about $\mathbf{S}$. Then the uniformly random $\mathbf{A}_1$ functions as an extractor so that $\mathbf{S}\mathbf{A}_1$ is uniformly distributed. Consequently, the first column of $\mathbf{S}\mathbf{A} + \mathbf{E}$ can be replaced with a uniform column. Column by column, $\mathbf{S}\mathbf{A} + \mathbf{E}$ can be replaced with a uniform matrix, and thus (7) follows. Overall, there are totally m steps and each step loses a factor n, so the overall loss factor is $O(mn)$. Thus it is an almost tight reduction from LWE to multi-secret LWE.

In Sect. 5, we give a fine-grained almost tight reduction, with loss factor further decreased to $O(cn)$ ($c < m$), which can be as small as $O(\lambda^2)$.

Instantiation of Our Frameworks. In addition to our LWE-based pr-QA-HPS described earlier, we also need tightly secure dual-mode commitment and QA-NIZK from LWE to obtain tightly MU^{c} secure SIG and PKE schemes via our frameworks. For the dual-mode commitment scheme, we instantiate it by adapting the Regev's PKE scheme [45]. As for QA-NIZK, we instantiate it based on the recent advances in LWE-based NIZK in the standard model [16, 43, 34]. In particular, we follow one of the most efficient paradigms for LWE-based NIZK to date, which is due to Libert et al. [34], and construct tightly-secure QA-NIZK based on LWE directly for the languages defined in (4) for SIG and (1) for PKE respectively, bypassing a heavy reduction to an NP-complete problem [16, 43]. To this end, we first construct trapdoor Σ-protocols based on LWE, then compile them via the tightness-preserving transformation proposed by Libert et al. [34] to obtain tightly-secure QA-NIZKs. See Subsect. 6.4 for more details.

To deal with the LWE errors, all the building blocks pr-QA-HPS, dual-mode commitment and QA-NIZK must support *gap language* (i.e., a pair of languages $\mathcal{L} \subseteq \widetilde{\mathcal{L}}$). For simplicity, we do not make this explicit in our overview and refer to the main body for more details.

On Efficiency of Our Schemes. Finally, we discuss the efficiency of our LWE-based SIG and PKE schemes with tight MU^{c} security. For our SIG, the verification key is a single matrix[3], the secret key consists of a bit-string plus a matrix, and the signature is made up of a single vector and a QA-NIZK proof. For our PKE, the public key is a single vector, the secret key is a single bit-string, and the ciphertext is made up of a single vector and a QA-NIZK proof.

Although we instantiate LWE-based QA-NIZK following one of the most efficient paradigm to date by Libert et al. [34], it is not quite practical at the moment. Consequently, our tightly MU^{c} secure SIG and PKE schemes may not be as efficient as the existing LWE-based SIG (e.g., [14, 38, 12, 22]) and PKE schemes (e.g., [44, 41, 38]) in the standard model, almost all of which do not have tight reductions even in the single-user setting. However, we stress that the main purpose of this work is taking the first theoretical step to study whether tightly MU^{c} security from LWE in the standard model is possible and how to achieve it. We believe that our ideas may open the door to further improvements, e.g., by improving the efficiency of LWE-based QA-NIZK.

Furthermore, similar to [27], we note that we can obtain more cryptographic primitives with tight MU^{c} security from our SIG and PKE schemes, including signcryption (SC), message authentication code (MAC) and authenticated encryption (AE) schemes.

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

[3] Here we do not count the public parameters in the verification key, as it can be shared among all users. The same applies to the public key of PKE.

input. Let $\emptyset$ denote the empty set. If x is defined by y or the value of y is assigned to x, we write $x := y$. For $i \in \mathbb{N}$, define $[i] := \{1, 2, ..., i\}$. For a set $\mathcal{X}$, denote by $x \leftarrow_\$ \mathcal{X}$ the procedure of sampling x from $\mathcal{X}$ uniformly at random. If $\mathcal{X}$ is distribution, $x \leftarrow_\$ \mathcal{X}$ means that x is sampled according to $\mathcal{X}$. We use $y \leftarrow_\$ \mathcal{A}(x)$ to define the random variable y obtained by executing algorithm $\mathcal{A}$ on input x. We use $y \in \mathcal{A}(x)$ to indicate that y lies in the support of $\mathcal{A}(x)$. If $\mathcal{A}$ is deterministic we write $y \leftarrow \mathcal{A}(x)$. We also use $y \leftarrow \mathcal{A}(x; r)$ to make explicit the random coins r used in the probabilistic computation. Denote by $\mathbf{T}(\mathcal{A})$ the running time of $\mathcal{A}$. "PPT" abbreviates probabilistic polynomial-time. Denote by poly some polynomial function and negl some negligible function.

For distributions X, Y, Z, let $\varDelta(X, Y) := \frac{1}{2} \cdot \sum_x |\Pr[X = x] - \Pr[Y = x]|$ denote the statistical distance between X and Y, $\varDelta(X, Y \,|\, Z)$ a shorthand for $\varDelta((X, Z), (Y, Z))$, and $\widetilde{\mathbf{H}}_\infty(X \,|\, Y) := -\log\big(\mathbb{E}_{y \leftarrow_\$ Y}\big[\max_x \Pr[X = x \,|\, Y = y]\big]\big)$ the average min-entropy of X conditioned on Y. If $\varDelta(X, Y) \leq \mathsf{negl}(\lambda)$, we say that X and Y are statistically indistinguishable (close), and denote it by $X \overset{s}{\approx} Y$. If $|\Pr[\mathcal{D}(X) = 1] - \Pr[\mathcal{D}(Y) = 1]| \leq \mathsf{negl}(\lambda)$ for all PPT distinguishers $\mathcal{D}$, we say that X and Y are computationally indistinguishable, and denote it by $X \overset{c}{\approx} Y$. For a metric space $\mathcal{M}$ with metric dist, we use $\mathsf{Ball}_{\boldsymbol{\varepsilon}}(m) := \{m' \in \mathcal{M} \mid \mathsf{dist}(m, m') \leq \boldsymbol{\varepsilon}\}$ to denote the ball centered at $m \in \mathcal{M}$ of radius $\boldsymbol{\varepsilon} > 0$. We use lower-case bold letters (like $\mathbf{v}$) to denote column vectors and upper-case bold letters (like $\mathbf{A}$) to denote matrices. For a vector $\mathbf{v}$, we let $\|\mathbf{v}\|$ (resp., $\|\mathbf{v}\|_\infty$) denote its ℓ_2 (resp., infinity) norm. For a matrix $\mathbf{A}$, we define $\|\mathbf{A}\|$ (resp., $\|\mathbf{A}\|_\infty$) as the largest ℓ_2 (resp., infinity) norm of $\mathbf{A}$'s rows. A distribution χ is B-bounded if its support is limited to $[-B, B]$. Let $\mathbb{Z}_q$ be the ring of integers modulo q, and its elements are represented by the integers in $(-q/2, q/2]$.

Lemma 1 ([21]). *Let X, Y, Z be three (possibly correlated) random variables. If Z has at most 2^λ possible values, then $\widetilde{\mathbf{H}}_\infty(X \,|\, (Y, Z)) \geq \widetilde{\mathbf{H}}_\infty(X \,|\, Y) - \lambda$.*

In Appendix A, we present additional preliminaries. More precisely, we present the syntax of digital signature (SIG) and its strong $\mathsf{MU^c\text{-}CMA}$ security in Appendix A.1, the syntax of public-key encryption (PKE) and its $\mathsf{MUMC^c\text{-}CCA}$ security in Appendix A.2, the syntax of tag-based quasi-adaptive non-interactive zero-knowledge argument (QA-NIZK) for gap language and its zero-knowledge and unbounded simulation-soundness (USS) in Appendix A.3, the definition of collision-resistant hash functions in Appendix A.4, and the definition of error-correcting codes in Appendix A.5.

2.1 Gap Language Distribution

In this work, we consider *gap* languages (i.e., a pair of NP-languages $\mathcal{L} \subseteq \widetilde{\mathcal{L}}$) and formalize a collection of gap languages as a gap language distribution.

Definition 1 (Gap Language Distribution). *A gap language distribution $\mathscr{L}$ is a probability distribution that outputs a language parameter ρ as well as* *a trapdoor td_ρ in polynomial time. The language parameter ρ publicly defines a gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ satisfying $\mathcal{L}_\rho \subseteq \widetilde{\mathcal{L}}_\rho \subseteq \mathcal{X}$, with $\mathcal{X}$ the universe.*

Moreover, $\mathscr{L}$ is associated with three PPT algorithms $(\mathsf{Sample}_{\mathcal{L}}, \mathsf{Sample}_{\mathcal{X}}, \mathsf{Check}_{\widetilde{\mathcal{L}}})$: $\mathsf{Sample}_{\mathcal{L}}(\rho)$ samples an instance x from $\mathcal{L}_\rho$ together with a witness w; $\mathsf{Sample}_{\mathcal{X}}$ samples an instance x from $\mathcal{X}$; $\mathsf{Check}_{\widetilde{\mathcal{L}}}(\rho, td_\rho, x)$ is a deterministic algorithm that outputs a decision bit about whether x is in $\widetilde{\mathcal{L}}_\rho$, with the help of td_ρ. We require that for all $(\rho, td_\rho) \in \mathscr{L}$ and $x \in \mathcal{X}$, $\mathsf{Check}_{\widetilde{\mathcal{L}}}(\rho, td_\rho, x) = 1$ holds if and only if $x \in \widetilde{\mathcal{L}}_\rho$. For simplicity, we will slightly abuse notations "$x \leftarrow_{\$} \mathcal{L}_\rho$" and "$x \leftarrow_{\$} \mathcal{X}$" to denote sampling x according to $\mathsf{Sample}_{\mathcal{L}}(\rho)$ and $\mathsf{Sample}_{\mathcal{X}}$, respectively.

A gap language distribution $\mathscr{L}$ is associated with a subset membership problem (SMP), which asks whether an element is randomly chosen from $\mathcal{L}_\rho$ or $\mathcal{X}$. SMP can be extended to multi-fold SMP by considering multiple elements.

Definition 2 (SMP). *The subset membership problem (SMP) related to $\mathscr{L}$ is hard, if for any PPT adversary $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{smp}}_{\mathscr{L},\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}(\rho, x) = 1] - \Pr[\mathcal{A}(\rho, x') = 1]| \leq \mathsf{negl}(\lambda)$, where $(\rho, td_\rho) \leftarrow_{\$} \mathscr{L}$, $x \leftarrow_{\$} \mathcal{L}_\rho$ and $x' \leftarrow_{\$} \mathcal{X}$.*

Definition 3 (Multi-fold SMP). *The multi-fold SMP related to $\mathscr{L}$ is hard, if for any PPT adversary $\mathcal{A}$ and any polynomial $Q = \mathsf{poly}(\lambda)$, it holds that $\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{A},Q}(\lambda) := |\Pr[\mathcal{A}(\rho, \{x_j\}_{j\in[Q]}) = 1] - \Pr[\mathcal{A}(\rho, \{x'_j\}_{j\in[Q]}) = 1]| \leq \mathsf{negl}(\lambda)$, where $(\rho, td_\rho) \leftarrow_{\$} \mathscr{L}$, $x_1, ..., x_Q \leftarrow_{\$} \mathcal{L}_\rho$ and $x'_1, ..., x'_Q \leftarrow_{\$} \mathcal{X}$.*

Multi-fold SMP can generally be reduced to SMP with a security loss of the number of folds. In this work, we will instantiate gap language distributions based on LWE and show an almost tight reduction from SMP to multi-fold SMP.

2.2 Commitment Scheme

A dual-mode commitment scheme has two indistinguishable parameter generation modes, i.e., a binding mode and a hiding mode. Below we propose a new variant called dual-mode *gap* commitment scheme, by requiring the hiding property hold for messages in a message space $\mathcal{M}$ but the binding property hold for messages in a possibly larger message space $\widetilde{\mathcal{M}}$.

Definition 4 (Dual-Mode Gap Commitment Scheme). *A dual-mode gap commitment scheme $\mathsf{CMT} = (\mathsf{BSetup}, \mathsf{HSetup}, \mathsf{Com})$ consists of PPT algorithms:*

- $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{BSetup}/\mathsf{HSetup}$*: The binding-mode/hiding-mode setup algorithm outputs a public parameter $\mathsf{pp}_{\mathsf{CMT}}$, which implicitly defines two message spaces $\mathcal{M} \subseteq \widetilde{\mathcal{M}}$ and two randomness spaces $\mathcal{R} \subseteq \widetilde{\mathcal{R}}$.*
- $\mathsf{com} \leftarrow \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, m; r)$*: Taking as input $\mathsf{pp}_{\mathsf{CMT}}$, a message $m \in \widetilde{\mathcal{M}}$ and a randomness $r \in \widetilde{\mathcal{R}}$, the committing algorithm outputs a commitment* com.

Moreover, there exist negligible functions $\varepsilon_{\mathsf{binding}}$ and $\varepsilon_{\mathsf{hiding}}$ (in λ), such that the following properties hold:

- **Parameter Indistinguishability:** *For any PPT $\mathcal{A}$, it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{para\text{-}ind}}_{\mathsf{CMT},\mathcal{A}}(\lambda) := \big|\Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{CMT}}) = 1 \mid \mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{BSetup}]\\ - \Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{CMT}}) = 1 \mid \mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{HSetup}]\big| \le \mathsf{negl}(\lambda).\end{aligned}$$

- **$\varepsilon_{\mathsf{binding}}$-Statistical Binding for $\widetilde{\mathcal{M}}$ under BSetup:** *It holds that*

$$\Pr\left[\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{BSetup} \ : \begin{array}{c}\exists\, m \ne m' \in \widetilde{\mathcal{M}}, r, r' \in \widetilde{\mathcal{R}},\\ s.t.\ \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, m; r) = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, m'; r')\end{array}\right] \le \varepsilon_{\mathsf{binding}}.$$

- **$\varepsilon_{\mathsf{hiding}}$-Statistical Hiding for $\mathcal{M}$ under HSetup:** *It holds that*

$$\max_{m_0, m_1 \in \mathcal{M}} \Delta\big((\mathsf{pp}_{\mathsf{CMT}}, \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, m_0; r)), (\mathsf{pp}_{\mathsf{CMT}}, \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, m_1; r))\big) \le \varepsilon_{\mathsf{hiding}},$$

where the probability is over $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{HSetup}$ *and* $r \leftarrow_{\$} \mathcal{R}$.

2.3 Lattice Backgrounds

For $\sigma > 0$ and $\mathbf{c} \in \mathbb{R}^n$, we define the Gaussian function on $\mathbb{R}^n$ centered at $\mathbf{c}$ with parameter σ by $\rho_{\sigma,\mathbf{c}}(\mathbf{x}) := e^{-\pi\|\mathbf{x}-\mathbf{c}\|^2/\sigma^2}$. The *discrete* Gaussian distribution $D_{\Lambda,\sigma,\mathbf{c}}$ over an n-dimensional lattice $\Lambda \subseteq \mathbb{R}^n$ is defined by $D_{\Lambda,\sigma,\mathbf{c}}(\mathbf{x}) := \rho_{\sigma,\mathbf{c}}(\mathbf{x})/\rho_{\sigma,\mathbf{c}}(\Lambda)$ for any lattice vector $\mathbf{x} \in \Lambda$, where $\rho_{\sigma,\mathbf{c}}(\Lambda) := \sum_{\mathbf{z}\in\Lambda} \rho_{\sigma,\mathbf{c}}(\mathbf{z})$. The subscript $\mathbf{c}$ is taken to be $\mathbf{0}$ when omitted.

We will use the following variant of the leftover hash lemma.

Lemma 2 (Particular case of [37, Lemma 2.3]). *Let $n, m, q \in \mathbb{N}$ be integers and $\epsilon \in (0, 1)$. Suppose $\mathbf{s}$ is chosen from some distribution over $\mathbb{Z}_q^m$ and $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^n$ are chosen independently of $\mathbf{s}$ from uniform distribution. Furthermore let Y be a random-variable (possibly) correlated with $\mathbf{s}$.*

- *If q is a prime, and $\widetilde{\mathbf{H}}_\infty(\mathbf{s} \bmod q | Y) \ge n\log q + 2\log\left(\frac{1}{\epsilon}\right)$. Then we have: $\Delta\big((\mathbf{A}, \mathbf{A}\mathbf{s}), (\mathbf{A}, \mathbf{u}) | Y\big) \le \epsilon$.*
- *If q is a composite number, and $\widetilde{\mathbf{H}}_\infty(\mathbf{s} \bmod p | Y) \ge 2n\log q + 2\log\left(\frac{1}{\epsilon}\right)$ for any q's prime factor p. Then we have: $\Delta\big((\mathbf{A}, \mathbf{A}\mathbf{s}), (\mathbf{A}, \mathbf{u}) | Y\big) \le \epsilon$.*

Definition 5 (LWE Assumption [45]). *Let $n, m, q \in \mathbb{N}$, and χ be a distribution over $\mathbb{Z}_q$. The $\mathsf{LWE}_{n,q,\chi,m}$-assumption holds, if for any PPT adversary $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,m],\mathcal{A}}(\lambda) := \big|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{u}^\top) = 1]\big| \le \mathsf{negl}(\lambda)$, where $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow_{\$} \chi^m$ and $\mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^m$.*

Definition 6 (Multi-secret LWE Assumption). *Let $n, m, q, Q \in \mathbb{N}$, and χ be a distribution over $\mathbb{Z}_q$. The Q-$\mathsf{LWE}_{n,q,\chi,m}$-assumption holds, if for any PPT $\mathcal{A}$ it holds that $\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,\chi,m],\mathcal{A}}(\lambda) := \big|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{S}\mathbf{A} + \mathbf{E}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{U}) = 1]\big| \le \mathsf{negl}(\lambda)$, where $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{S} \leftarrow_{\$} \mathbb{Z}_q^{Q\times n}$, $\mathbf{E} \leftarrow_{\$} \chi^{Q\times m}$ and $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{Q\times m}$.*

A simple hybrid argument can show that $\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,\chi,m]}(\lambda) \leq Q \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,m]}(\lambda)$. However, the security loss factor depends on the number of the secrets. In this paper, we will show an almost tight security reduction from LWE to multi-secret Q-LWE (see Theorem 3 in Sect. 5).

In [1, 38], an algorithm named TrapGen is proposed to sample a "nearly" uniform random matrix $\mathbf{A}$ along with a low-norm trapdoor matrix $\mathbf{T}_{\mathbf{A}}$ such that $\mathbf{A} \cdot \mathbf{T}_{\mathbf{A}} = \mathbf{0}$ (cf. Lemma 3). Meanwhile, another algorithm called Invert is proposed to make use of $\mathbf{T}_{\mathbf{A}}$ to invert an LWE sample $(\mathbf{A}, \mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top)$ to obtain $\mathbf{s}$ and $\mathbf{e}$ (cf. Lemma 4).

Lemma 3 ([1, 38]). *There exists a PPT algorithm* TrapGen *that takes as input positive integers n, q $(q \geq 2)$ and a sufficiently large $m = O(n \log q)$, outputs a matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and a trapdoor matrix $\mathbf{T}_{\mathbf{A}} \in \mathbb{Z}_q^{m\times m}$ such that $\mathbf{A}$ is statistically close to the uniform distribution, $\mathbf{A} \cdot \mathbf{T}_{\mathbf{A}} = \mathbf{0}$, and $\|\mathbf{T}_{\mathbf{A}}\| = O(\sqrt{n \log q})$.*

Lemma 4 ([38, Theorem 5.4]). *There exists a deterministic polynomial-time algorithm* Invert *that takes as inputs the trapdoor information $\mathbf{T}_{\mathbf{A}}$[4] and a vector $\mathbf{s}^\top \mathbf{A} + \mathbf{e}^\top$ with $\mathbf{s} \in \mathbb{Z}_q^n$ and $\|\mathbf{e}\| \leq q/(10\sqrt{m})$, and outputs $\mathbf{s}$ and $\mathbf{e}$.*

We recall the tail bound about the discrete Gaussian distributions over $\mathbb{Z}^m$.

Lemma 5 (Tail Bound [36]). *For any $t > 0$, we have $\Pr_{x \leftarrow_\$ D_{\mathbb{Z},\sigma}}\left[|x| \geq t \cdot \sigma\right] \leq 2e^{-\frac{t^2}{2}}$ and $\Pr_{\mathbf{x} \leftarrow_\$ D_{\mathbb{Z}^m,\sigma}}\left[\|\mathbf{x}\| \geq \|\mathbf{x}\|_\infty \geq t \cdot \sigma\sqrt{m}\right] \leq t^m \cdot e^{\frac{m}{2}(1-t^2)}$.*

In particular, for $t \geq \omega(\sqrt{\log \lambda})$, the probability that $|x| \geq t \cdot \sigma$ and $\|\mathbf{x}\| \geq \|\mathbf{x}\|_\infty \geq t \cdot \sigma\sqrt{m}$ is negligible.

The next smudging lemma shows that a uniform distribution over a sufficiently large interval $[-B', B']$ can swallow any distribution over a small interval $[-B, B]$ and yield a nearly uniform distribution over $[-B', B']$.

Lemma 6 (Smudging Lemma, [5, Lemma 1]). *Let B, B' be positive integers, and $e \in [-B, B]$ a fixed integer. Then for a uniformly chosen $e' \leftarrow_\$ [-B', B']$, it holds that $\Delta(e + e', e') = B/B'$.*

3 Probabilistic QA-HPS

Hash proof system (HPS) was proposed by Cramer and Shoup [19], and turned out to be a powerful tool in a wide range of applications. Han et al. [28, 27] generalized HPS in a quasi-adaptive setting, termed as *Quasi-Adaptive HPS* (QA-HPS), by allowing the projection key to depend on the specific language $\mathcal{L}_\rho$ for which hash values are computed. (For completeness, the formal definition of QA-HPS is recalled in Appendix A.6.)

In this section, we propose a new primitive called *Probabilistic QA-HPS* (pr-QA-HPS), by further generalizing QA-HPS in two aspects. Firstly, pr-QA-HPS

[4] More precisely, the trapdoor information is not $\mathbf{T}_{\mathbf{A}}$ itself, but some sensitive information used to generate $\mathbf{T}_{\mathbf{A}}$. Here we abuse them for simplicity.

has *probabilistic* public and private evaluation algorithms (denoted by prPub and prPriv) that may toss coins. In other words, the outputs of prPub and prPriv are probabilistic distributions over the hash value space. Regarding correctness, instead of requiring exact correctness as for (QA-)HPS, we require an approximate correctness for pr-QA-HPS. Moreover, we require a statistical indistinguishability of the two probabilistic evaluation algorithms. Secondly, pr-QA-HPS is defined for a *gap* language distribution. Some properties of pr-QA-HPS, e.g., the evaluation indistinguishability in Definition 7 and the one-time extracting in Definition 11, require the underlying language distribution to be a gap one.

Firstly, we present the syntax of probabilistic QA-HPS.

Definition 7 (Probabilistic QA-HPS). *A probabilistic QA-HPS (pr-QA-HPS) scheme* $\mathsf{prQAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{prPub}, \mathsf{prPriv})$ *for a gap language distribution* $\mathscr{L}$ *consists of four PPT algorithms:*

– $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_\$ \mathsf{Setup}_{\mathsf{HPS}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp}_{\mathsf{HPS}}$*, which serves as an implicit input of other algorithms.* $\mathsf{pp}_{\mathsf{HPS}}$ *implicitly defines a hashing key space* $\mathcal{SK}$*, a hash value space* $\mathcal{HV}$*, and a family of hash functions* $\Lambda_{(\cdot)} : \mathcal{X} \longrightarrow \mathcal{HV}$ *indexed by hashing keys* $sk \in \mathcal{SK}$*, where* $\mathcal{X}$ *is the universe for languages output by* $\mathscr{L}$*.*

We require that $\Lambda_{(\cdot)}$ *is efficiently computable and there are PPT algorithms for sampling* $sk \leftarrow_\$ \mathcal{SK}$ *uniformly and sampling* $hv \leftarrow_\$ \mathcal{HV}$ *uniformly. We also require the hash value space* $\mathcal{HV}$ *to be a metric space.*

– $pk_\rho \leftarrow \alpha_\rho(sk)$*: On input a hashing key* $sk \in \mathcal{SK}$*, the deterministic projection algorithm indexed by language parameter* ρ *outputs a projection key* pk_ρ*.*

– $hv \leftarrow_\$ \mathsf{prPub}(pk_\rho, x, w)$*: Taking as input a projection key* $pk_\rho = \alpha_\rho(sk)$ *specified by* ρ*, an instance* $x \in \widetilde{\mathcal{L}}_\rho$ *and a witness* w *for* $x \in \widetilde{\mathcal{L}}_\rho$*, the* probabilistic *public evaluation algorithm outputs a hash value* $hv \in \mathcal{HV}$*.*

– $hv \leftarrow_\$ \mathsf{prPriv}(sk, x)$*: On input a hashing key* $sk \in \mathcal{SK}$ *and an instance* $x \in \mathcal{X}$*, the* probabilistic *private evaluation algorithm outputs a hash value* $hv \in \mathcal{HV}$*.*

Moreover, there exist negligible functions $\varepsilon_{\mathsf{prPub}}$*,* $\varepsilon_{\mathsf{prPriv}}$ *and* $\varepsilon_{\mathsf{evalnd}}$ *(in* λ*), such that the following properties hold:*

- $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$**-Approximate Correctness for** $\mathcal{L}_\rho$**:** *For all* $(\rho, td_\rho) \in \mathscr{L}$*,* $\mathsf{pp}_{\mathsf{HPS}} \in \mathsf{Setup}_{\mathsf{HPS}}$*,* $sk \in \mathcal{SK}$*,* $x \in \mathcal{L}_\rho$ *with witness* w*, and* $pk_\rho := \alpha_\rho(sk)$*, it holds that* $\Pr[hv \leftarrow_\$ \mathsf{prPub}(pk_\rho, x, w) \ : \ hv \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPub}}}(\Lambda_{sk}(x))] = 1$
$$\text{and} \quad \Pr[hv \leftarrow_\$ \mathsf{prPriv}(sk, x) \ : \ hv \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPriv}}}(\Lambda_{sk}(x))] = 1.$$
Here $hv \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPub}}}(\Lambda_{sk}(x))$ *(resp.,* $hv \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPriv}}}(\Lambda_{sk}(x))$*) means that* hv *is within distance at most* $\varepsilon_{\mathsf{prPub}}$ *(resp.,* $\varepsilon_{\mathsf{prPriv}}$*) of the real hash value* $\Lambda_{sk}(x)$*.*
- $\varepsilon_{\mathsf{evalnd}}$**-Evaluation Indistinguishability for** $\widetilde{\mathcal{L}}_\rho$**:** *For all* $(\rho, td_\rho) \in \mathscr{L}$*,* $\mathsf{pp}_{\mathsf{HPS}} \in \mathsf{Setup}_{\mathsf{HPS}}$*,* $sk \in \mathcal{SK}$*,* $x \in \widetilde{\mathcal{L}}_\rho$ *with witness* w*, and* $pk_\rho := \alpha_\rho(sk)$*, it holds that*
$$\Delta\big(\mathsf{prPub}(pk_\rho, x, w),\ \mathsf{prPriv}(sk, x)\big) \leq \varepsilon_{\mathsf{evalnd}},$$
where the probability is only over the inner coin tosses of prPub *and* prPriv*.*

Note that the approximate correctness is required to hold for instances in $\mathcal{L}_\rho$, while the evaluation indistinguishability is required to hold for instances in $\widetilde{\mathcal{L}}_\rho$. Moreover, we can naturally define pr-QA-HPS for two gap language distributions $\mathscr{L}$ and $\mathscr{L}_0$, by requiring the above two properties to hold not only for language parameters ρ output by $\mathscr{L}$, but also for language parameters ρ_0 output by $\mathscr{L}_0$.

Next, we recall and adapt some useful properties defined in [28, 27] for QA-HPS to our pr-QA-HPS. We start by recalling a statistical property called $\langle\mathscr{L}, \mathscr{L}_0\rangle$*-key-switching* from [28], parameterized by two gap language distributions $\mathscr{L}$ and $\mathscr{L}_0$. Informally speaking, it stipulates that in the presence of a projection key $\alpha_\rho(sk)$ w.r.t. a language parameter ρ output by $\mathscr{L}$, the projection key $\alpha_{\rho_0}(sk)$ w.r.t. another language parameter ρ_0 output by $\mathscr{L}_0$ can be switched to $\alpha_{\rho_0}(sk')$ for an independent sk'.

Definition 8 ($\langle\mathscr{L}, \mathscr{L}_0\rangle$-Key-Switching). *Let $\mathscr{L}$ and $\mathscr{L}_0$ be two gap language distributions. A pr-QA-HPS scheme* $\mathsf{prQAHPS}$ *supports $\langle\mathscr{L}, \mathscr{L}_0\rangle$-key-switching, if for any (possibly unbounded) adversary $\mathcal{A}$, it holds that*

$$\epsilon^{\langle\mathscr{L},\mathscr{L}_0\rangle\text{-ks}}_{\mathsf{prQAHPS},\mathcal{A}} := \big|\Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho, \rho_0, \alpha_\rho(sk), \alpha_{\rho_0}(sk)) = 1] \\ - \Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho, \rho_0, \alpha_\rho(sk), \alpha_{\rho_0}(sk')) = 1]\big| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_\$ \mathsf{Setup}_{\mathsf{HPS}}$, $(\rho, td_\rho) \leftarrow_\$ \mathscr{L}$, $(\rho_0, td_{\rho_0}) \leftarrow_\$ \mathscr{L}_0$, *and* $sk, sk' \leftarrow_\$ \mathcal{SK}$.

We recall another statistical property from [27], called *projection key diversity (PK-diversity)*, which expresses statistical collision resistance of projection keys under different hashing keys.

Definition 9 (PK-Diversity). *A pr-QA-HPS scheme* $\mathsf{prQAHPS}$ *for $\mathscr{L}$ has projection key diversity (PK-diversity), if* $\epsilon^{\mathsf{pk\text{-}div}}_{\mathsf{prQAHPS}} := \Pr[\alpha_\rho(sk) = \alpha_\rho(sk')] \leq \mathsf{negl}(\lambda)$, *where* $(\rho, td_\rho) \leftarrow_\$ \mathscr{L}$, $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_\$ \mathsf{Setup}_{\mathsf{HPS}}$ *and* $sk, sk' \leftarrow_\$ \mathcal{SK}$.

In [28, 27], a computational property called $\mathscr{L}_0$*-multi-key-multi-extracting* is defined for QA-HPS, which demands the pseudorandomness of multiple hash values $\{\Lambda_{sk_i}(x_j)\}_{i,j}$ for multiple instances $\{x_j \leftarrow_\$ \mathcal{L}_{\rho_0}\}_j$ (where $\rho_0 \in \mathscr{L}_0$) under multiple keys $\{sk_i \leftarrow_\$ \mathcal{SK}\}_i$.

Below we adapt the property to pr-QA-HPS, by requiring the pseudorandomness of $\{\mathsf{prPriv}(sk_i, x_{i,j})\}_{i,j}$ for multiple instances $\{x_{i,j} \leftarrow_\$ \mathcal{L}_{\rho_0}\}_{i,j}$ under multiple keys $\{sk_i \leftarrow_\$ \mathcal{SK}\}_i$.

Definition 10 ($\mathscr{L}_0$-Multi-Key-Multi-Extracting). *A pr-QA-HPS scheme* $\mathsf{prQAHPS}$ *supports $\mathscr{L}_0$-multi-key-multi-extracting, if for any PPT adversary $\mathcal{A}$, any polynomial N and any polynomial Q, it holds that*

$$\mathsf{Adv}^{\mathscr{L}_0\text{-mk-mext}}_{\mathsf{prQAHPS},\mathcal{A},N,Q}(\lambda) := \big|\Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \{x_{i,j}, \mathsf{prPriv}(sk_i, x_{i,j})\}_{i\in[N],j\in[Q]}) = 1] \\ - \Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \{x_{i,j}, hv_{i,j}\}_{i\in[N],j\in[Q]}) = 1]\big| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_\$ \mathsf{Setup}_{\mathsf{HPS}}$, $(\rho_0, td_{\rho_0}) \leftarrow_\$ \mathscr{L}_0$, $sk_1, ..., sk_N \leftarrow_\$ \mathcal{SK}$, $x_{1,1}, ..., x_{N,Q} \leftarrow_\$ \mathcal{L}_{\rho_0}$ *and* $hv_{1,1}, ..., hv_{N,Q} \leftarrow_\$ \mathcal{HV}$.

In [27], a statistical property called $\langle \mathscr{L}_0, \mathscr{L} \rangle$*-one-time(OT)-extracting* is defined for QA-HPS. Informally speaking, it demands high min-entropy of $\Lambda_{sk}(x)$ for *any* $x \in \mathcal{L}_\rho$ with ρ output by $\mathscr{L}$, when sk is uniformly chosen from $\mathcal{SK}$, even in the presence of a projection key $\alpha_{\rho_0}(sk)$ w.r.t. ρ_0 output by $\mathscr{L}_0$. This min-entropy makes sure that any (unbounded) adversary is unable to guess the correct hash value $\Lambda_{sk}(x)$.

Below we generalize it to $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-Extracting for pr-QA-HPS, where $\varepsilon_{\mathsf{ext}} \geq 0$, by stipulating the hardness even for *any* $x \in \widetilde{\mathcal{L}}_\rho$ and even for finding a hash value hv close to $\Lambda_{sk}(x)$, i.e., finding $hv \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}\big(\Lambda_{sk}(x)\big)$.

Definition 11 ($\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-Extracting). *Let $\mathscr{L}_0$ and $\mathscr{L}$ be a pair of language distributions. A pr-QA-HPS scheme* $\mathsf{prQAHPS}$ *supports* $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-*OT-extracting, if for any (possibly unbounded) adversary $\mathcal{A}$, it holds that* $\epsilon^{\varepsilon_{\mathsf{ext}}\text{-}\langle \mathscr{L}_0, \mathscr{L} \rangle\text{-otext}}_{\mathsf{prQAHPS},\mathcal{A}}$

$$:= \Pr\left[\begin{array}{c} \mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}, (\rho_0, td_{\rho_0}) \leftarrow_{\$} \mathscr{L}_0, \\ (\rho, td_\rho) \leftarrow_{\$} \mathscr{L}, sk \leftarrow_{\$} \mathcal{SK}, \\ (x^*, hv^*) \leftarrow_{\$} \mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \rho, \alpha_{\rho_0}(sk)) \end{array} : \begin{array}{c} x^* \in \widetilde{\mathcal{L}}_\rho \ \wedge \\ hv^* \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}(\Lambda_{sk}(x^*)) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

4 Generic Constructions of SIG and PKE with Tight $\mathsf{MU^c}$ Security from Probabilistic QA-HPS

Recently, Han et al. [27] proposed generic constructions of digital signature (SIG) and public-key encryption (PKE) with tight $\mathsf{MU^c}$ security from QA-HPS and QA-NIZK. In this section, we propose a new generic SIG construction and extend their PKE construction, by using our probabilistic QA-HPS formalized in Sect. 3 as a central building block instead of QA-HPS, allowing instantiations from the LWE assumptions as shown later.

More precisely, we present our constructions of SIG with tight strong $\mathsf{MU^c}$-CMA security in Subsect. 4.1 and PKE with tight $\mathsf{MUMC^c}$-CCA security in Subsect. 4.2.

4.1 Generic Construction of SIG with Tight Strong $\mathsf{MU^c}$-CMA Security

We present our generic construction of strongly $\mathsf{MU^c}$-CMA secure SIG. Let $\mathcal{M}$ be an arbitrary message space. The underlying building blocks are as follows.

- Two gap language distributions $\mathscr{L}$ and $\mathscr{L}_0$, both of which have hard SMPs.
- A probabilistic $\mathsf{prQAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{prPub}, \mathsf{prPriv})$ for both $\mathscr{L}$ and $\mathscr{L}_0$ with hashing key space $\mathcal{SK}$, satisfying $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$-approximate correctness and $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting with $\varepsilon_{\mathsf{prPriv}} \leq \varepsilon_{\mathsf{ext}}$.
- A dual-mode gap commitment scheme $\mathsf{CMT} = (\mathsf{BSetup}, \mathsf{HSetup}, \mathsf{Com})$ with message spaces $\mathcal{M}_{\mathsf{CMT}} := \mathcal{SK} \subseteq \widetilde{\mathcal{SK}}$ and randomness spaces $\mathcal{R} \subseteq \widetilde{\mathcal{R}}$.
- A tag-based $\mathsf{QANIZK} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{SimGen}, \mathsf{Sim})$ for the gap language $\mathcal{GL}^{(\mathsf{QANIZK})}_{\rho'} = (\mathcal{L}^{(\mathsf{QANIZK})}_{\rho'}, \widetilde{\mathcal{L}}^{(\mathsf{QANIZK})}_{\rho'})$ defined in Fig. 1, with tag space $\mathcal{T}$. It is clear to see that $\mathcal{L}^{(\mathsf{QANIZK})}_{\rho'} \subseteq \widetilde{\mathcal{L}}^{(\mathsf{QANIZK})}_{\rho'}$ since $\mathcal{L}_\rho \subseteq \widetilde{\mathcal{L}}_\rho$, $\mathcal{SK} \subseteq \widetilde{\mathcal{SK}}$, $\mathcal{R} \subseteq \widetilde{\mathcal{R}}$ and $\varepsilon_{\mathsf{prPriv}} \leq \varepsilon_{\mathsf{ext}}$. (See Appendix A.3 for the definition.)

- A family of collision-resistant hash functions $\mathcal{H} = \{H : \mathcal{M} \longrightarrow \mathcal{T}\}$.

Our generic construction of $\mathsf{SIG} = (\mathsf{Setup}_{\mathsf{SIG}}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy}_{\mathsf{SIG}})$ is shown in Fig. 1. It is easy to see that the correctness of SIG follows from the $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$-approximate correctness of $\mathsf{prQAHPS}$ and the completeness of QANIZK, since d generated by $d \leftarrow_{\$} \mathsf{prPriv}(sk, x)$ always satisfies $d \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPriv}}}\big(\Lambda_{sk}(x)\big)$.

```
$\underline{\mathsf{pp}_{\mathsf{SIG}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{SIG}}:}$
$(\rho, td_\rho) \leftarrow_{\$} \mathscr{L}$, $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$, $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{BSetup}$.
$\rho' := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}})$ defines a gap language $\mathcal{GL}^{(\mathsf{QANIZK})}_{\rho'} = (\mathcal{L}^{(\mathsf{QANIZK})}_{\rho'}, \widetilde{\mathcal{L}}^{(\mathsf{QANIZK})}_{\rho'})$, where
    $\mathcal{L}^{(\mathsf{QANIZK})}_{\rho'} := \left\{ (x, vk, d) \;\middle|\; \exists\, (w, sk \in \mathcal{SK}, r \in \mathcal{R}), \text{ s.t. } \begin{array}{c} x \in \mathcal{L}_\rho \text{ with witness } w \\ \wedge\ vk = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk; r) \\ \wedge\ d \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPriv}}}(\Lambda_{sk}(x)) \end{array} \right\}$,
    $\widetilde{\mathcal{L}}^{(\mathsf{QANIZK})}_{\rho'} := \left\{ (x, vk, d) \;\middle|\; \exists\, (w, sk \in \widetilde{\mathcal{SK}}, r \in \widetilde{\mathcal{R}}), \text{ s.t. } \begin{array}{c} x \in \widetilde{\mathcal{L}}_\rho \text{ with witness } w \\ \wedge\ vk = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk; r) \\ \wedge\ d \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}(\Lambda_{sk}(x)) \end{array} \right\}$.
$\mathsf{crs} \leftarrow_{\$} \mathsf{CRSGen}(\rho')$.  $H \leftarrow_{\$} \mathcal{H}$.
Return $\mathsf{pp}_{\mathsf{SIG}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}}, \mathsf{crs}, H)$.
```

```
$\underline{(vk, sigk) \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp}_{\mathsf{SIG}}):}$
$sk \leftarrow_{\$} \mathcal{SK}$, $r \leftarrow_{\$} \mathcal{R}$.
$vk := \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk; r)$.
Return $(vk, sigk := (sk, r))$.
```

```
$\underline{\sigma \leftarrow_{\$} \mathsf{Sign}(sigk = (sk, r), m):}$
$x \leftarrow_{\$} \mathcal{L}_\rho$ with witness $w$.
$d \leftarrow_{\$} \mathsf{prPriv}(sk, x)$.
$vk := \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk; r)$.
$\tau := H(m) \in \mathcal{T}$.
$\pi \leftarrow_{\$} \mathsf{Prove}(\mathsf{crs}, \tau, (x, vk, d), (w, sk, r))$.
Return $\sigma := (x, d, \pi)$.
```

```
$\underline{0/1 \leftarrow \mathsf{Vrfy}_{\mathsf{SIG}}(vk, m, \sigma):}$
Parse $\sigma = (x, d, \pi)$.
$\tau := H(m) \in \mathcal{T}$.
If $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, (x, vk, d), \pi) = 1$:
    Return 1.
Else: Return 0.
```

Fig. 1. Generic construction of $\mathsf{SIG} = (\mathsf{Setup}_{\mathsf{SIG}}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy}_{\mathsf{SIG}})$ from $\mathsf{prQAHPS}$, CMT, tag-based QANIZK and $\mathcal{H}$. The message space is $\mathcal{M}$.

Next, we show the strong $\mathsf{MU^c}$-CMA security of SIG via the following theorem.

Theorem 1 (Strong $\mathsf{MU^c}$-CMA Security of SIG). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii) $\mathsf{prQAHPS}$ is a probabilistic QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$-approximate correctness, $\varepsilon_{\mathsf{evalnd}}$-evaluation indistinguishability, and supporting $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting, where $\varepsilon_{\mathsf{ext}} \geq \varepsilon_{\mathsf{prPriv}}$, (iii) CMT is a dual-mode gap commitment scheme that is $\varepsilon_{\mathsf{binding}}$-statistical binding and $\varepsilon_{\mathsf{hiding}}$-statistical hiding, (iv) QANIZK is a tag-based QA-NIZK for the gap language $\mathcal{GL}^{(\mathsf{QANIZK})}_{\rho'}$ defined in Fig. 1, satisfying both zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed SIG scheme in Fig. 1 is strongly $\mathsf{MU^c}$-CMA secure.*

Concretely, for any number N of users and any adversary $\mathcal{A}$ making at most Q_s times of $\mathcal{O}_{\mathrm{SIGN}}$ queries, there exist adversaries $\mathcal{B}_1, \cdots, \mathcal{B}_7$, s.t. $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_6) \approx \mathbf{T}(\mathcal{A}) + (N + Q_s) \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned} \mathsf{Adv}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N}(\lambda) \leq\ & \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda) \\ & + \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \mathsf{Adv}^{\mathsf{para\text{-}ind}}_{\mathsf{CMT},\mathcal{B}_6}(\lambda) + \textit{statist. loss}, \end{aligned}$$

where statist. loss $= 2 \cdot \varepsilon_{\mathsf{binding}} + Q_s \cdot \varepsilon_{\mathsf{evalnd}} + N \cdot \epsilon^{\varepsilon_{\mathsf{ext}}\text{-}\langle \mathscr{L}_0, \mathscr{L} \rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS},\mathcal{B}_7} + \varepsilon_{\mathsf{hiding}} + \frac{N(N-1)}{2} / |\mathcal{SK}|$.

We refer to Sect. 1 for an overview of the proof, and postpone the formal proof to Appendix B. Here we provide the game sequence G_0-G_7 used in the formal proof in Table 1. According to Theorem 1, SIG has tight strong $\mathsf{MU^c}$-CMA security as long as both the multi-fold SMPs related to $\mathscr{L}$ and $\mathscr{L}_0$ have tight reductions (e.g., to the LWE assumptions), and CMT and QANIZK are tightly secure.

Table 1. Brief Description of Games G_0-G_7 for the strong $\mathsf{MU^c}$-CMA security proof of SIG. Here column "$\mathcal{O}_{\text{SIGN}}$" suggests how a signature $\sigma = (x, d, \pi)$ is generated: sub-column "x from" refers to the language from which x is chosen; sub-column "d using" indicates the keys that are used in the computation of d; sub-column "π via" indicates the way (Prove or Sim) that π is computed. Column "$\mathcal{O}_{\text{COR}}$" shows the key returned by $\mathcal{O}_{\text{COR}}$. Column "Win's additional check for forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$" describes the additional check that $\mathcal{A}$'s forgery wins, besides the routine check $i^* \notin \mathcal{Q}_{\text{COR}} \wedge (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{SIGN}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1$, where $\tau^* := H(m^*)$.

	$\mathcal{O}_{\text{SIGN}}(i, m)$			$\mathcal{O}_{\text{COR}}(i)$	Win's additional check for forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$	Remark/Assumption
	x from	d using	π via			
G_0	$\mathcal{L}_\rho$	sk_i	Prove	sk_i		The strong $\mathsf{MU^c}$-CMA experiment
G_1	$\mathcal{L}_\rho$	sk_i	Prove	sk_i		Abort if verification keys collide: by *statistical binding* of CMT under BSetup & secret keys hardly collide
G_2	$\mathcal{L}_\rho$	sk_i	Sim	sk_i		By *zero-knowledge* of QANIZK
G_3	$\mathcal{L}_\rho$	sk_i	Sim	sk_i	$(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\text{SIM}}$	By *collision-resistance* of $\mathcal{H}$
G_4	$\mathcal{L}_{\rho_0}$	sk_i	Sim	sk_i	$(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\text{SIM}}$	By *multi-fold SMP* of $\mathscr{L}$ and $\mathscr{L}_0$
G_5	$\mathcal{L}_{\rho_0}$	sk_i	Sim	sk_i	$(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\text{SIM}}$, $x^* \in \widetilde{\mathcal{L}}_\rho,\ d^* \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}(\Lambda_{sk_{i^*}}(x^*))$	By *USS* of QANIZK & *statistical binding* of CMT under BSetup
G_6	$\mathcal{L}_{\rho_0}$	$\alpha_{\rho_0}(sk_i)$	Sim	sk_i	$(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\text{SIM}}$, $x^* \in \widetilde{\mathcal{L}}_\rho,\ d^* \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}(\Lambda_{sk_{i^*}}(x^*))$	By *evaluation indistinguishability* of $\mathsf{prQAHPS}$
G_7	$\mathcal{L}_{\rho_0}$	$\alpha_{\rho_0}(sk_i)$	Sim	sk_i	$(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\text{SIM}}$, $x^* \in \widetilde{\mathcal{L}}_\rho,\ d^* \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}(\Lambda_{sk_{i^*}}(x^*))$	Change to $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_\$ \mathsf{HSetup}$: by *parameter indistinguishability* of CMT $\Pr[\mathsf{Win}] = \mathsf{negl}$ in G_7: by $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-*OT-extracting* of $\mathsf{prQAHPS}$ & *statistical hiding* of CMT under HSetup

4.2 Generic Construction of PKE with Tight $\mathsf{MUMC^c}$-CCA Security

We present our generic construction of $\mathsf{MUMC^c}$-CCA secure PKE. Let $\mathcal{M}$ be an arbitrary message space. The underlying building blocks are as follows.

- Two gap language distributions $\mathscr{L}$ and $\mathscr{L}_0$, both of which have hard SMPs.
- A probabilistic $\mathsf{prQAHPS} = (\mathsf{Setup_{HPS}}, \alpha_{(\cdot)}, \mathsf{prPub}, \mathsf{prPriv})$ for both $\mathscr{L}$ and $\mathscr{L}_0$ with hashing key space $\mathcal{SK}$, projection key space $\mathcal{PK}$ and hash value space $\mathcal{HV}$, satisfying $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$-approximate correctness. We require $\mathcal{HV}$ to be an (additive) group.
- A tag-based $\mathsf{QANIZK} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy_{NIZK}}, \mathsf{SimGen}, \mathsf{Sim})$ for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ generated by $\mathscr{L}$, with tag space $\mathcal{T}$.
- A family of collision-resistant hash functions $\mathcal{H} = \{H : \mathcal{PK} \times \mathcal{HV} \longrightarrow \mathcal{T}\}$.

- An error-correcting code $\mathsf{ECC} = (\mathsf{Encode}, \mathsf{Decode})$ from $\mathcal{M}$ to $\mathcal{HV}$, which is able to correct $(\varepsilon_{\mathsf{prPub}} + \varepsilon_{\mathsf{prPriv}})$ errors efficiently. (See Appendix A.5 for the definition.)

Our generic construction of $\mathsf{PKE} = (\mathsf{Setup}_{\mathsf{PKE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is shown in Fig. 2. It is easy to check that the correctness of PKE follows from the $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$-approximate correctness of $\mathsf{prQAHPS}$, the $(\varepsilon_{\mathsf{prPub}} + \varepsilon_{\mathsf{prPriv}})$-correctness of ECC and the completeness of QANIZK: (1) by the $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$-approximate correctness of $\mathsf{prQAHPS}$, the hv generated by $hv \leftarrow_{\$} \mathsf{prPub}(pk, x, w)$ in Enc and the hv' generated by $hv' \leftarrow_{\$} \mathsf{prPriv}(sk, x)$ in Dec are within distance at most $(\varepsilon_{\mathsf{prPub}} + \varepsilon_{\mathsf{prPriv}})$, i.e., $hv' \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPub}}+\varepsilon_{\mathsf{prPriv}}}(hv)$, (2) then $d - hv' = hv - hv' + \mathsf{Encode}(m) \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPub}}+\varepsilon_{\mathsf{prPriv}}}(\mathsf{Encode}(m))$, and by the $(\varepsilon_{\mathsf{prPub}}+\varepsilon_{\mathsf{prPriv}})$-correctness of ECC, it follows that $\mathsf{Decode}(d - hv') = \mathsf{Decode}(hv - hv' + \mathsf{Encode}(m)) = m$.

```
$\mathsf{pp}_{\mathsf{PKE}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{PKE}}$:
$(\rho, td_\rho) \leftarrow_{\$} \mathscr{L}$.
$\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$.
$\mathsf{crs} \leftarrow_{\$} \mathsf{CRSGen}(\rho)$.
$H \leftarrow_{\$} \mathcal{H}$.
Return $\mathsf{pp}_{\mathsf{PKE}} :=$
  $(\rho, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$.
```

```
$(pk, sk) \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp}_{\mathsf{PKE}})$:
$sk \leftarrow_{\$} \mathcal{SK}$, $pk := \alpha_\rho(sk)$.
Return $(pk, sk)$.

$c \leftarrow_{\$} \mathsf{Enc}(pk, m)$:
$x \leftarrow_{\$} \mathcal{L}_\rho$ with witness $w$.
$hv \leftarrow_{\$} \mathsf{prPub}(pk, x, w)$.
$d := hv + \mathsf{Encode}(m)$.
$\tau := H(pk, d) \in \mathcal{T}$.
$\pi \leftarrow_{\$} \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$.
Return $c := (x, d, \pi)$.
```

```
$m'/\perp \leftarrow \mathsf{Dec}(sk, c)$:
Parse $c = (x, d, \pi)$.
$pk := \alpha_\rho(sk)$.
$\tau := H(pk, d) \in \mathcal{T}$.
If $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$:
    $hv' \leftarrow_{\$} \mathsf{prPriv}(sk, x)$.
    $m' := \mathsf{Decode}(d - hv')$.
    Return $m'$.
Else: Return $\perp$.
```

Fig. 2. Generic construction of $\mathsf{PKE} = (\mathsf{Setup}_{\mathsf{PKE}},\mathsf{Gen},\mathsf{Enc},\mathsf{Dec})$ from $\mathsf{prQAHPS}$, tag-based QANIZK, $\mathcal{H}$ and ECC. The message space is $\mathcal{M}$.

Next, we show the $\mathsf{MUMC^c}$-CCA security of PKE via the following theorem.

Theorem 2 ($\mathsf{MUMC^c}$-CCA **Security of** PKE). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii)* $\mathsf{prQAHPS}$ *is a probabilistic QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having $\varepsilon_{\mathsf{evalnd}}$-evaluation indistinguishability, PK-diversity, and supporting both $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching and $\mathscr{L}_0$-multi-key-multi-extracting, (iii)* QANIZK *is a tag-based QA-NIZK for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ generated by $\mathscr{L}$, satisfying both zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed* PKE *scheme in Fig. 2 is* $\mathsf{MUMC^c}$-CCA *secure.*

Concretely, for any number N of users and any adversary $\mathcal{A}$ who makes at most Q_e times of $\mathcal{O}_{\text{Enc}}$ queries and Q_d times of $\mathcal{O}_{\text{Dec}}$ queries, there exist adversaries $\mathcal{B}_1, \cdots, \mathcal{B}_7$, such that $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_6) \approx \mathbf{T}(\mathcal{A}) + (N + Q_e + Q_d) \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N}(\lambda) \leq\ & \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda) \\ & + \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{prQAHPS},\mathcal{B}_6,N,Q_e}(\lambda) + \textit{statist. loss},\end{aligned}$$

where statist. loss $= \frac{N(N-1)}{2} \cdot \epsilon^{\mathsf{pk\text{-}div}}_{\mathsf{prQAHPS}} + (3Q_e + 2Q_d) \cdot \varepsilon_{\mathsf{evalnd}} + N \cdot \epsilon^{\langle \mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{prQAHPS},\mathcal{B}_7}$.

We refer to Sect. 1 for an overview of the proof, and postpone the formal proof to Appendix C. Here we provide the game sequence G_0-G_9 used in the formal proof in Table 2. According to Theorem 2, PKE has tight $\mathsf{MUMC^c\text{-}CCA}$ security as long as both the multi-fold SMPs related to $\mathscr{L}$ and $\mathscr{L}_0$ have tight reductions, $\mathsf{prQAHPS}$ has tight $\mathscr{L}_0$-multi-key-multi-extracting, and QANIZK is tightly secure.

Table 2. Brief Description of Games G_0-G_9 for the $\mathsf{MUMC^c\text{-}CCA}$ security proof of PKE. Here column "$\mathcal{O}_{\text{ENC}}$" suggests how a challenge ciphertext $c^* = (x^*, d^*, \pi^*)$ is generated: sub-column "x^* from" refers to the language from which x^* is chosen; sub-column "hv^* using" indicates the keys that are used in the computation of hv^*; sub-column "π^* via" indicates the way (Prove or Sim) that π^* is computed. Column "$\mathcal{O}_{\text{DEC}}$" suggests how a decryption query $(i, c = (x, d, \pi))$ is answered: sub-column "additional check" describes the additional check made by $\mathcal{O}_{\text{DEC}}$ besides the routine check $(i, c) \notin \mathcal{Q}_{\text{ENC}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$, where $\tau := H(pk_i, d)$; $\mathcal{O}_{\text{DEC}}$ outputs $\perp$ if the check fails; sub-column "hv' using" indicates the keys that are used in the computation of hv'. Column "$\mathcal{O}_{\text{COR}}$" shows the key returned by $\mathcal{O}_{\text{COR}}$. Recall that it is not allowed to query $\mathcal{O}_{\text{ENC}}$ and $\mathcal{O}_{\text{COR}}$ for a same user index i.

	$\mathcal{O}_{\text{ENC}}(i^*, m_0, m_1)$			$\mathcal{O}_{\text{DEC}}(i, c)$		$\mathcal{O}_{\text{COR}}(i)$	Remark/Assumption
	x^* from	hv^* using	π^* via	additional check	hv' using		
G_0	$\mathcal{L}_\rho$	pk_{i^*}	Prove		sk_i	sk_i	The $\mathsf{MUMC^c\text{-}CCA}$ security experiment
G_1	$\mathcal{L}_\rho$	pk_{i^*}	Prove		sk_i	sk_i	Abort if public keys collide: by *PK-diversity* of $\mathsf{prQAHPS}$
G_2	$\mathcal{L}_\rho$	sk_{i^*}	Sim		sk_i	sk_i	By *evaluation indistinguishability* of $\mathsf{prQAHPS}$ & *zero-knowledge* of QANIZK
G_3	$\mathcal{L}_\rho$	sk_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}$	sk_i	sk_i	By *collision-resistance* of $\mathcal{H}$
G_4	$\mathcal{L}_{\rho_0}$	sk_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}$	sk_i	sk_i	By *multi-fold SMP* of $\mathscr{L}$ & $\mathscr{L}_0$
G_5	$\mathcal{L}_{\rho_0}$	sk_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}$, $x \in \widetilde{\mathcal{L}}_\rho$	sk_i	sk_i	By *USS* of QANIZK
G_6	$\mathcal{L}_{\rho_0}$	$\alpha_{\rho_0}(sk_{i^*})$	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}, x \in \widetilde{\mathcal{L}}_\rho$	$\alpha_\rho(sk_i)$	sk_i	By *evaluation indistinguishability* of $\mathsf{prQAHPS}$
$\{\mathsf{G}_{7.\eta}\}_{\eta \in [N]}$	$\mathcal{L}_{\rho_0}$	$\begin{cases} \alpha_{\rho_0}(sk'_{i^*}), & \text{if } i^* \leq \eta \\ \alpha_{\rho_0}(sk_{i^*}), & \text{if } i^* > \eta \end{cases}$	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}, x \in \widetilde{\mathcal{L}}_\rho$	$\alpha_\rho(sk_i)$	sk_i	By $\langle \mathscr{L}, \mathscr{L}_0 \rangle$*-key-switching* of $\mathsf{prQAHPS}$
$\mathsf{G}_{7.N}$	$\mathcal{L}_{\rho_0}$	$\alpha_{\rho_0}(sk'_{i^*})$	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}, x \in \widetilde{\mathcal{L}}_\rho$	$\alpha_\rho(sk_i)$	sk_i	–
G_8	$\mathcal{L}_{\rho_0}$	sk'_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}, x \in \widetilde{\mathcal{L}}_\rho$	sk_i	sk_i	By *evaluation indistinguishability* of $\mathsf{prQAHPS}$
G_9	$\mathcal{L}_{\rho_0}$	$= \mathsf{rand}$	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}, x \in \widetilde{\mathcal{L}}_\rho$	sk_i	sk_i	By $\mathscr{L}_0$*-multi-key-multi-extracting* of $\mathsf{prQAHPS}$ $\Pr[\mathsf{Win}] = \frac{1}{2}$ in G_9

5 Tighter Reduction from LWE to Multi-Secret LWE

In this section, we will show an almost tight reduction from LWE to multi-secret LWE, which supports the almost tight security of our LWE-based instantiations as shown later in Sect. 6. We note that similar results could be derived from [2]. Nevertheless, our proof is simpler, more flexible and results in tighter reduction compared with [2].

We first recall a useful lemma presenting the spectral norm upper bound of discrete Gaussian matrices. Then we recall the definitions of continuous Gaussian distribution D_σ and multi-secret LWE with continuous Gaussian distribution D_σ, which will serve as an intermediate assumption in our reduction to obtain better parameters by applying the noise lossiness approach in [15] (i.e., Lemma 11 and Lemma 13 in Appendix D.1). We also recall the randomized rounding technique due to Peikert [42]. Finally we show Theorem 3 that addresses the almost tight reduction from LWE to Multi-secret LWE for prime modulus. We also extend the result for composite modulus in Appendix D.3.

Lemma 7 ([38, Lemma 2.8, 2.9]). *Let* $\mathbf{F} \leftarrow_{\$} D_{\mathbb{Z},\gamma}^{n\times m}$ *and* $m \geq n$*. Then with all but* 2^{-m} *probability it holds that the spectral norm* $\sigma_{\mathbf{F}}$ *of* $\mathbf{F}$ *satisfies* $\sigma_{\mathbf{F}} \leq \gamma \cdot C \cdot \sqrt{m}$ *where* C *is a global constant.*

Definition 12 (Multi-secret LWE Assumption with Continuous Gaussian [15]). *For* $\sigma > 0$*, the continuous Gaussian distribution* D_σ *over* $\mathbb{R}$ *centered at* 0 *is defined by the probability density function* $D_\sigma(x) := \rho_\sigma(x)/\rho_\sigma(\mathbb{R})$ *for any* $x \in \mathbb{R}$*, where* $\rho_\sigma(x) := e^{-\pi x^2/\sigma^2}$ *and* $\rho_\sigma(\mathbb{R}) := \int_{\mathbb{R}} \rho_\sigma(z)dz = \sigma$*.*

Let $n, m, q, Q \in \mathbb{N}$*. The* $Q\text{-}\mathsf{LWE}_{n,q,D_\sigma,m}$*-assumption holds, if for any PPT* $\mathcal{A}$ *it holds that* $\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_\sigma,m],\mathcal{A}}(\lambda) := \big|\Pr[\mathcal{A}(\mathbf{A}, \mathbf{SA}+\mathbf{E}) = 1] - \Pr[\mathcal{A}(\mathbf{A}, \mathbf{U}+\mathbf{E}) = 1]\big| \leq \mathsf{negl}(\lambda)$*, where* $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$*,* $\mathbf{S} \leftarrow_{\$} \mathbb{Z}_q^{Q\times n}$*,* $\mathbf{E} \leftarrow_{\$} D_\sigma^{Q\times m}$ *and* $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{Q\times m}$*.*

Lemma 8 (Particular case of [42, Theorem 3.1]). *Let* $\sigma > 0$ *and* $r \geq \sqrt{\lambda}$*. For* $e \leftarrow_{\$} D_\sigma$ *and* $v \leftarrow_{\$} D_{\mathbb{Z}-e,r}$*, the distribution of* $e+v$ *is statistically close to* $D_{\mathbb{Z},\sqrt{\sigma^2+r^2}}$*, with statistical distance at most* $2^{-\lambda}$*.*

Theorem 3 (LWE ⇒ Multi-secret LWE with Prime Modulus). *Let* $n, m, \ell, q \in \mathbb{N}$*, and* q *be a prime. Let* $\sigma, \sigma_0, \sigma_1, r, \gamma > 0$ *such that* $\sigma = \sqrt{{\sigma_0}^2 + r^2}$*,* $\sigma_0 > \gamma\cdot C\cdot\sqrt{m}\cdot\sigma_1$*,* $\frac{q}{\sigma_1} \geq \sqrt{\frac{\ln(4n)}{\pi}}$ *and* $r \geq \sqrt{\lambda}$*, where* C *is the global constant from Lemma 7. For any adversary* $\mathcal{A}$*, there exists an adversary* $\mathcal{B}$*, such that* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q\cdot\mathsf{poly}(\lambda)$ *with* $\mathsf{poly}(\lambda)$ *independent of* $\mathbf{T}(\mathcal{A})$*, and* $\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\mathbb{Z},\sigma},m],\mathcal{A}}(\lambda) \leq 2cn\cdot\mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}}(\lambda) + \frac{Q(m+c+1)}{2^\lambda}$*, where* c *is an integer such that*

$$m' = \lfloor \tfrac{m}{c} \rfloor \quad \text{and} \quad n \geq (m'\log q + \ell\log q + 2\lambda + 1)/\log(\sigma_1). \tag{8}$$

Proof sketch. We will use the multi-secret LWE with continuous Gaussian D_{σ_0} defined in Definition 12 as an intermediate assumption, and show that there exists an adversary $\mathcal{B}'$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{B}') + Q\cdot\mathsf{poly}'(\lambda) \approx \mathbf{T}(\mathcal{A}) + Q\cdot\mathsf{poly}(\lambda)$ and

$$\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\mathbb{Z},\sigma},m],\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\sigma_0},m],\mathcal{B}'}(\lambda) + \tfrac{Qm}{2^\lambda}, \tag{9}$$

$$\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\sigma_0},m],\mathcal{B}'}(\lambda) \leq 2cn\cdot\mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}}(\lambda) + \tfrac{Q(c+1)}{2^\lambda}. \tag{10}$$

Then Theorem 3 follows directly from (9) and (10).

To prove (9), we construct $\mathcal{B}'$ to break the $Q\text{-}\mathsf{LWE}_{n,q,D_{\sigma_0},m}$-assumption by invoking $\mathcal{A}$. Given a challenge $(\mathbf{A}, \mathbf{B})$, $\mathcal{B}'$ wants to distinguish $\mathbf{B} = \mathbf{SA}+\mathbf{E}$ from $\mathbf{B} = \mathbf{U}+\mathbf{E}$, where $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$, $\mathbf{S} \leftarrow_{\$} \mathbb{Z}_q^{Q\times n}$, $\mathbf{E} \leftarrow_{\$} D_{\sigma_0}^{Q\times m}$ and $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_q^{Q\times m}$. To decide which case it is, $\mathcal{B}'$ parses $\mathbf{B} = (b_{i,j})_{i\in[Q],j\in[m]}$, samples $v_{i,j} \leftarrow_{\$} D_{\mathbb{Z}-b_{i,j},r}$ for all $i \in [Q], j \in [m]$, sets $\mathbf{B}' := (b_{i,j} + v_{i,j})_{i\in[Q],j\in[m]}$, feeds $(\mathbf{A}, \mathbf{B}')$ to $\mathcal{A}$, and returns whatever $\mathcal{A}$ outputs. We analyze the advantage of $\mathcal{B}'$.

In the case $\mathbf{B} = \mathbf{SA} + \mathbf{E}$*.* We parse $\mathbf{SA} = (t_{i,j})_{i\in[Q],j\in[m]}$ and $\mathbf{E} = (e_{i,j})_{i\in[Q],j\in[m]}$. Then we have $b_{i,j} = t_{i,j} + e_{i,j}$ and $\mathbf{B}' = (t_{i,j} + e_{i,j} + v_{i,j})_{i\in[Q],j\in[m]} = \mathbf{SA} + (e_{i,j} + v_{i,j})_{i\in[Q],j\in[m]}$. Since $t_{i,j} \in \mathbb{Z}$, $v_{i,j}$ follows the distribution $D_{\mathbb{Z}-b_{i,j},r} = D_{\mathbb{Z}-t_{i,j}-e_{i,j},r} = D_{\mathbb{Z}-e_{i,j},r}$. Then together with the fact that $e_{i,j}$ follows D_{σ_0}, by Lemma 8, the distribution of $e_{i,j} + v_{i,j}$ is within statistical distance $2^{-\lambda}$ of $D_{\mathbb{Z},\sigma} = D_{\mathbb{Z},\sqrt{{\sigma_0}^2+r^2}}$. Let $\mathbf{E}' := (e_{i,j} + v_{i,j})_{i\in[Q],j\in[m]}$. Then $\mathbf{B}' = \mathbf{SA} + \mathbf{E}'$ with $\mathbf{E}' = (e_{i,j} + v_{i,j})_{i\in[Q],j\in[m]}$ following a distribution statistically close to $D_{\mathbb{Z},\sigma}^{Q\times m}$, with statistical distance at most $Qm/2^{\lambda}$.

In the case $\mathbf{B} = \mathbf{U} + \mathbf{E}$*.* Similar to the above analysis, we can get that $\mathbf{B}' = \mathbf{U} + \mathbf{E}'$ with $\mathbf{E}' = (e_{i,j} + v_{i,j})_{i\in[Q],j\in[m]}$ distributed over $\mathbb{Z}_q^{Q\times m}$. Since $\mathbf{U}$ is uniformly distributed over $\mathbb{Z}_q^{Q\times m}$ and independent of $\mathbf{E}'$, $\mathbf{B}' = \mathbf{U} + \mathbf{E}'$ is also uniformly distributed over $\mathbb{Z}_q^{Q\times m}$.

Thus, $\mathcal{B}'$ successfully distinguishes $\mathbf{B} = \mathbf{SA} + \mathbf{E}$ from $\mathbf{B} = \mathbf{U} + \mathbf{E}$ as long as $\mathcal{A}$ can distinguish $\mathbf{B}' = \mathbf{SA} + \mathbf{E}'$ (with $\mathbf{E}'$ nearly following $D_{\mathbb{Z},\sigma}^{Q\times m}$) from the uniform distribution, i.e., breaking the Q-$\mathsf{LWE}_{n,q,D_{\mathbb{Z},\sigma},m}$-assumption. This proves (9).

Next we turn to the proof of (10). Here we describe the main ideas behind the proof, and postpone the formal proof of (10) to Appendix D. We aim to prove that the Q-$\mathsf{LWE}_{n,q,D_{\sigma_0},m}$-assumption holds, i.e.,

$$(\mathbf{A},\ \mathbf{SA} + \mathbf{E}) \overset{c}{\approx} (\mathbf{A},\ \mathbf{U} + \mathbf{E}), \tag{11}$$

based on the $\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\gamma},m}$-assumption, and determine the security loss factor. Here $\mathbf{A} \leftarrow_\$ \mathbb{Z}_q^{n\times m}$, $\mathbf{S} \leftarrow_\$ \mathbb{Z}_q^{Q\times n}$, $\mathbf{E} \leftarrow_\$ D_{\sigma_0}^{Q\times m}$ and $\mathbf{U} \leftarrow_\$ \mathbb{Z}_q^{Q\times m}$.

In the first step, we break $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ into $(\mathbf{A}_1|\bar{\mathbf{A}}_1) \in \mathbb{Z}_q^{n\times m'} \times \mathbb{Z}_q^{n\times(m-m')}$ and $\mathbf{E} \in D_{\sigma_0}^{Q\times m}$ into $(\mathbf{E}_1|\bar{\mathbf{E}}_1) \in D_{\sigma_0}^{Q\times m'} \times D_{\sigma_0}^{Q\times(m-m')}$, where the block $\mathbf{A}_1$ contains the first m' columns of $\mathbf{A}$. Then we change $\bar{\mathbf{A}}_1$ into a lossy one $\tilde{\mathbf{A}}_1 = \mathbf{CB} + \mathbf{F}$, where $\mathbf{C} \leftarrow_\$ \mathbb{Z}_q^{n\times\ell}, \mathbf{B} \leftarrow_\$ \mathbb{Z}_q^{\ell\times(m-m')}$ and $\mathbf{F} \in \mathbb{Z}_q^{n\times(m-m')}$ follows the error distribution $D_{\mathbb{Z},\gamma}^{n\times(m-m')}$. This change is indistinguishable due to the n-secret $\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\gamma},m-m'}$-assumption. Therefore,

$$(\mathbf{A},\mathbf{SA}{+}\mathbf{E}) = ((\mathbf{A}_1|\bar{\mathbf{A}}_1),(\mathbf{SA}_1{+}\mathbf{E}_1|\mathbf{S}\bar{\mathbf{A}}_1{+}\bar{\mathbf{E}}_1)) \overset{c}{\approx} ((\mathbf{A}_1|\tilde{\mathbf{A}}_1),(\mathbf{SA}_1{+}\mathbf{E}_1|\mathbf{S}\tilde{\mathbf{A}}_1{+}\bar{\mathbf{E}}_1))$$

but it incurs a loss factor of n since hybrid arguments yield $\mathsf{Adv}^{n\text{-}\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m-m']}(\lambda) \leq n \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m-m']}(\lambda) \leq n \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m]}(\lambda)$. Now given a lossy $\tilde{\mathbf{A}}_1$, the information of $\mathbf{S}$ leaked by $\mathbf{S}\tilde{\mathbf{A}}_1$ is bounded. By taking $\mathbf{A}_1$ as extractor, we can extract the remaining entropy of $\mathbf{S}$, and result in $\mathbf{SA}_1 \overset{s}{\approx} \mathbf{U}_1$, where $\mathbf{U}_1 \leftarrow_\$ \mathbb{Z}_q^{Q\times m'}$. So we have

$$((\mathbf{A}_1|\tilde{\mathbf{A}}_1),(\mathbf{SA}_1 + \mathbf{E}_1|\mathbf{S}\tilde{\mathbf{A}}_1 + \bar{\mathbf{E}}_1)) \overset{s}{\approx} ((\mathbf{A}_1|\tilde{\mathbf{A}}_1),(\mathbf{U}_1 + \mathbf{E}_1|\mathbf{S}\tilde{\mathbf{A}}_1 + \bar{\mathbf{E}}_1)).$$

Next, we change the lossy $\tilde{\mathbf{A}}_1$ back to uniform $\bar{\mathbf{A}}_1$, and have

$$((\mathbf{A}_1|\tilde{\mathbf{A}}_1),(\mathbf{U}_1 + \mathbf{E}_1|\mathbf{S}\tilde{\mathbf{A}}_1 + \bar{\mathbf{E}}_1)) \overset{c}{\approx} ((\mathbf{A}_1|\bar{\mathbf{A}}_1),(\mathbf{U}_1 + \mathbf{E}_1|\mathbf{S}\bar{\mathbf{A}}_1 + \bar{\mathbf{E}}_1)).$$

Then we have loss factor n again.

In the second step, we break $\mathbf{A} = (\mathbf{A}_1|\bar{\mathbf{A}}_1)$ further into $(\mathbf{A}_1|\mathbf{A}_2|\bar{\mathbf{A}}_2) \in \mathbb{Z}_q^{n\times m'} \times \mathbb{Z}_q^{n\times m'} \times \mathbb{Z}_q^{n\times(m-2m')}$ and $\mathbf{E} = (\mathbf{E}_1|\bar{\mathbf{E}}_1)$ into $(\mathbf{E}_1|\mathbf{E}_2|\bar{\mathbf{E}}_2) \in D_{\sigma_0}^{Q\times m'} \times D_{\sigma_0}^{Q\times m'} \times D_{\sigma_0}^{Q\times(m-2m')}$, where the block $\mathbf{A}_2$ contains the second m' columns of $\mathbf{A}$. Then we change $\bar{\mathbf{A}}_2$ to a lossy one $\tilde{\mathbf{A}}_2$ and have

$$\begin{aligned}((\mathbf{A}_1|\bar{\mathbf{A}}_1),(\mathbf{U}_1+\mathbf{E}_1|\mathbf{S}\bar{\mathbf{A}}_1+\bar{\mathbf{E}}_1)) &= ((\mathbf{A}_1|\mathbf{A}_2|\bar{\mathbf{A}}_2),(\mathbf{U}_1+\mathbf{E}_1|\mathbf{S}\mathbf{A}_2+\mathbf{E}_2|\mathbf{S}\bar{\mathbf{A}}_2+\bar{\mathbf{E}}_2)) \\ &\overset{c}{\approx} ((\mathbf{A}_1|\mathbf{A}_2|\tilde{\mathbf{A}}_2),(\mathbf{U}_1+\mathbf{E}_1|\mathbf{S}\mathbf{A}_2+\mathbf{E}_2|\mathbf{S}\tilde{\mathbf{A}}_2+\bar{\mathbf{E}}_2))\end{aligned}$$

with a lossy factor n. With a similar argument, the uniform $\mathbf{A}_2$ can extract the remaining entropy of $\mathbf{S}$ so that $\mathbf{S}\mathbf{A}_2 \overset{s}{\approx} \mathbf{U}_2$, where $\mathbf{U}_2 \leftarrow_{\$} \mathbb{Z}_q^{Q\times m'}$. So

$$((\mathbf{A}_1|\mathbf{A}_2|\tilde{\mathbf{A}}_2),(\mathbf{U}_1+\mathbf{E}_1|\mathbf{S}\mathbf{A}_2+\mathbf{E}_2|\mathbf{S}\tilde{\mathbf{A}}_2+\bar{\mathbf{E}}_2)) \overset{s}{\approx} ((\mathbf{A}_1|\mathbf{A}_2|\tilde{\mathbf{A}}_2),(\mathbf{U}_1+\mathbf{E}_1|\mathbf{U}_2+\mathbf{E}_2|\mathbf{S}\tilde{\mathbf{A}}_2+\bar{\mathbf{E}}_2)).$$

Changing lossy $\tilde{\mathbf{A}}_2$ back to uniform $\bar{\mathbf{A}}_2$ yields

$$((\mathbf{A}_1|\mathbf{A}_2|\tilde{\mathbf{A}}_2),(\mathbf{U}_1+\mathbf{E}_1|\mathbf{U}_2+\mathbf{E}_2|\mathbf{S}\tilde{\mathbf{A}}_2+\bar{\mathbf{E}}_2)) \overset{c}{\approx} ((\mathbf{A}_1|\mathbf{A}_2|\bar{\mathbf{A}}_2),(\mathbf{U}_1+\mathbf{E}_1|\mathbf{U}_2+\mathbf{E}_2|\mathbf{S}\bar{\mathbf{A}}_2+\bar{\mathbf{E}}_2))$$

with a price of another loss factor n.

Overall, with at most $c \approx \frac{m}{m'}$ steps, we can prove (11) with a loss factor of $2cn$. It should be noted that we analyze the entropy of $\mathbf{S}$ with the so-called "lossiness approach" in [15], which results in more flexible parameters. This finishes the proof sketch of (10), and we refer to Appendix D for the formal proof of (10).

Finally, taking (9) and (10) together, Theorem 3 holds. □

Some Useful Setting of Parameters. Our reduction holds for a wide range of parameters. Here we describe two settings of parameters in Table 3, both of which satisfy the constrains in the statements of Theorem 3.

Table 3. Parameter setting for Theorem 3, where C denotes the global constant in Lemma 7.

Parameters	n	m	ℓ	q	c	σ_1	γ	σ_0	r	σ
Setting I	36λ	72λ	λ	λ^6	40	$\sqrt{\lambda}$	$12\sqrt{\lambda}$	$102C\lambda^{1.5}$	$\sqrt{205}C\lambda^{1.5}$	$103C\lambda^{1.5}$
Setting II	4λ	λ^2	λ	$2^{2\sqrt{\lambda}}$	2λ	$2^{\sqrt{\lambda}}$	λ	$\frac{\sqrt{2}}{2}\lambda^{2.5}2^{\sqrt{\lambda}}$	$\frac{\sqrt{2}}{2}\lambda^{2.5}2^{\sqrt{\lambda}}$	$\lambda^{2.5}2^{\sqrt{\lambda}}$

Setting I in Table 3 allows a constant factor c, resulting in a loss factor as small as $O(\lambda)$. In many applications, more constrains of parameter setting are considered. For example, the number of LWE samples m should be set as $O(n\log q)$ when applying the leftover hash lemma (i.e., Lemma 2), and the modulus q should be set as $2^{\omega(\log\lambda)}$ when we use smudging lemma (i.e., Lemma 6). Setting II in Table 3 also takes these additional constrains into account. In this setting, the factor c can be set as $O(\lambda)$, resulting in a loss factor of $O(\lambda^2)$.

Remark 1 (Comparison with the almost tight reduction in [2]). If we use techniques in [2], we can also obtain an almost tight reduction from LWE to multi-secret LWE. However, the loss factor would be $O(mn)$, as shown in the technical overview in Sect. 1 of our paper.

In contrast, our reduction in the proof of Theorem 3 is fine-grained and tighter, where the loss factor is $O(cn)$ with $c \leq m$. In fact, due to the flexible setting of σ_1, we can always set $\log(\sigma_1) = O(\log q)$. Then the parameter c can be set as small as $O(\frac{m}{n})$ to satisfy the constrain $n \geq O((\frac{m}{c}\log q + \ell \log q)/\log(\sigma_1))$. Consequently, the loss factor of our reduction can be as small as $O(cn) = O(m)$, saving a factor at least $O(n)$ compared with [2]'s reduction loss factor.

For example, in Setting I and Setting II in Table 3, their loss factor should be $O(\lambda^2)$ and $O(\lambda^3)$ respectively, while ours are $O(\lambda)$ and $O(\lambda^2)$ respectively.

6 Instantiation from LWE

In this section, we instantiate our generic SIG and PKE constructions proposed in Sect. 4 from the LWE assumptions. More precisely, we will show how to instantiate the underlying building blocks, including gap language distributions in Subsect. 6.1, probabilistic QA-HPS in Subsect. 6.2, dual-mode gap commitment in Subsect. 6.3 and compatible tag-based QA-NIZK in Subsect. 6.4.

For simplicity, all instantiations in this section take LWE-related public parameters $\mathsf{pp}_{\mathsf{LWE}} = (n, m, \ell, q, \sigma, \gamma, \chi, B, \tilde{B}, B', \tilde{B}', \zeta, \zeta')$ as *implicit input*, where $n, m, \ell, q, \sigma, \gamma$ are parameters satisfying the constrains in Theorem 3, χ is the discrete Gaussian distribution $D_{\mathbb{Z},\sigma}$ as described in Theorem 3, $B, \tilde{B}, B', \tilde{B}' \in \mathbb{N}$ are error bounds such that χ is B-bounded, and ζ, ζ' are parameters for Gaussians. (Some instantiations use only part of $\mathsf{pp}_{\mathsf{LWE}}$.) According to Lemma 5 (the tail bound), $\chi = D_{\mathbb{Z},\sigma}$ is $\sqrt{\lambda} \cdot \sigma$-bounded, except with exponentially small probability $2^{-\lambda}$,[5] so we can set $B = \sqrt{\lambda} \cdot \sigma$. The requirements for these parameters will be stated in the following theorems, and the concrete choices satisfying all requirements will be suggested in Table 4 in Subsect. 6.5.

6.1 Gap Language Distributions from LWE

Let $\mathsf{pp}_{\mathsf{LWE}} = (n, m, \ell, q, \sigma, \gamma, \chi, B, \tilde{B}, \cdots)$ be the LWE-related public parameters that serve as implicit input to all algorithms and satisfy $B < \tilde{B} < q/(10m)$ and χ a B-bounded distribution. Our LWE-based gap language distribution $\mathscr{L}$ samples a language parameter ρ and its trapdoor td_ρ as follows.

– $\mathscr{L}$ invokes $(\mathbf{A}, \mathbf{T_A}) \leftarrow_{\$} \mathsf{TrapGen}(n, q, m)$ (cf. Lemma 3) and outputs $(\rho := \mathbf{A} \in \mathbb{Z}_q^{n \times m}, td_\rho := \mathbf{T_A} \in \mathbb{Z}_q^{m \times m})$.

According to Lemma 3, $\mathbf{A}$ is almost uniform over $\mathbb{Z}_q^{n \times m}$ and $\|\mathbf{T_A}\|_\infty = O(\sqrt{n \log q})$. The language parameter $\rho = \mathbf{A}$ determines a gap language $\mathcal{GL}_{\mathbf{A}} = (\mathcal{L}_{\mathbf{A}}, \widetilde{\mathcal{L}}_{\mathbf{A}})$,

[5] We will not mention this exponentially small probability hereafter for simplicity, and take for granted that χ is B-bounded.

where $\mathcal{L}_{\mathbf{A}}$ and $\widetilde{\mathcal{L}}_{\mathbf{A}}$ define "noisy linear" subspaces as follows[6]

$$\mathcal{L}_{\mathbf{A}} := \big\{ \mathbf{c} \in \mathbb{Z}_q^m \,\big|\, \exists\ \mathbf{s} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}, \mathbf{e} \in [-B, B]^m,\ \text{s.t. } \mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top \big\},$$
$$\widetilde{\mathcal{L}}_{\mathbf{A}} := \big\{ \mathbf{c} \in \mathbb{Z}_q^m \,\big|\, \exists\ \mathbf{s} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}, \mathbf{e} \in [-\tilde{B}, \tilde{B}]^m,\ \text{s.t. } \mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top \big\}.$$

Clearly, $\mathcal{L}_{\mathbf{A}} \subseteq \widetilde{\mathcal{L}}_{\mathbf{A}}$ and both of them are contained in the universal set $\mathcal{X} := \mathbb{Z}_q^m$.

The associated algorithms $(\mathsf{Sample}_{\mathcal{L}}, \mathsf{Sample}_{\mathcal{X}}, \mathsf{Check}_{\widetilde{\mathcal{L}}})$ are defined as follows.

- $(\mathbf{c}, w_{\mathbf{c}}) \leftarrow_{\$} \mathsf{Sample}_{\mathcal{L}}(\rho = \mathbf{A})$: It chooses $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_q^n$, $\mathbf{e} \leftarrow_{\$} \chi^m$, computes $\mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top$, and returns the instance $\mathbf{c}$ with its witness $w_{\mathbf{c}} := (\mathbf{s}, \mathbf{e})$.
- $\mathbf{c} \leftarrow_{\$} \mathsf{Sample}_{\mathcal{X}}$: It outputs a uniformly chosen $\mathbf{c} \leftarrow_{\$} \mathbb{Z}_q^m$.
- $0/1 \leftarrow \mathsf{Check}_{\widetilde{\mathcal{L}}}(\rho = \mathbf{A}, td_\rho = \mathbf{T}_{\mathbf{A}}, \mathbf{c})$: It invokes $(\mathbf{s}, \mathbf{e}) \leftarrow \mathsf{Invert}(\mathbf{T}_{\mathbf{A}}, \mathbf{c})$ (cf. Lemma 4), and outputs 1 if $\mathbf{e} \in [-\tilde{B}, \tilde{B}]^m$ and 0 otherwise.

Given that $\mathbf{e} \in [-\tilde{B}, \tilde{B}]^m$ and $\tilde{B} < q/(10m)$, we have $\|\mathbf{e}\| \le \sqrt{m}\tilde{B} \le q/(10\sqrt{m})$. Then according to Lemma 4, $\mathsf{Check}_{\widetilde{\mathcal{L}}}(\rho, td_\rho, \mathbf{c})$ outputs 1 iff $\mathbf{c} \in \widetilde{\mathcal{L}}_{\mathbf{A}}$.

The subset membership problem (SMP) for $\mathscr{L}$ is exactly the $\mathsf{LWE}_{n,q,\chi,m}$ problem, and the multi-fold SMP is just the multi-secret $\mathsf{LWE}_{n,q,\chi,m}$ problem. Since we set $\chi = D_{\mathbb{Z},\sigma}$, by the almost tight reduction from LWE to multi-secret LWE in Sect. 5, i.e., Theorem 3, we have the following lemma.

Lemma 9 **($\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\gamma},m}$ ⇒ Multi-fold SMP for $\mathscr{L}$).** *Let $\chi = D_{\mathbb{Z},\sigma}$ in $\mathsf{Sample}_{\mathcal{L}}$. For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and $\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{A},Q}(\lambda) \le 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}}(\lambda) + \frac{Q(m+c+1)}{2^\lambda}$, where $\chi = D_{\mathbb{Z},\sigma}$ and $D_{\mathbb{Z},\gamma}$ are the discrete Gaussian distributions as described in Theorem 3, and c is an integer satisfying (8).*

6.2 Probabilistic QA-HPS from LWE

In this subsection, we instantiate probabilistic QA-HPS from the LWE assumption. Let $\mathsf{pp}_{\mathsf{LWE}} = (n, m, \ell, q, \sigma, \gamma, \chi, B, \tilde{B}, B', \cdots)$ be the LWE-related public parameters that serve as implicit input to all algorithms. Let both $\mathscr{L}$ and $\mathscr{L}_0$ be the gap language distribution specified in Subsect. 6.1. Here we use two distributions $\mathscr{L}$ and $\mathscr{L}_0$ to indicate the independence of them. We present our LWE-based scheme $\mathsf{prQAHPS}_{\mathsf{LWE}} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{prPub}, \mathsf{prPriv})$ for $\mathscr{L}$ in Fig. 3. The hash value space $\mathcal{HV} = \mathbb{Z}_q$ is a metric space with metric $\mathsf{dist}(hv, hv') := |hv - hv'|$ for $hv, hv' \in \mathbb{Z}_q$. Then $\mathsf{Ball}_\varepsilon(hv) := \{hv' \in \mathbb{Z}_q \mid |hv - hv'| \le \varepsilon\}$.

Firstly we prove that $\mathsf{prQAHPS}_{\mathsf{LWE}}$ is a pr-QA-HPS scheme in Theorem 4.

Theorem 4. *The $\mathsf{prQAHPS}_{\mathsf{LWE}}$ proposed in Fig. 3 is a pr-QA-HPS scheme that has $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$-approximate correctness and $\varepsilon_{\mathsf{evalnd}}$-evaluation indistinguishability with $\varepsilon_{\mathsf{prPub}} = B' + mB$, $\varepsilon_{\mathsf{prPriv}} = B'$ and $\varepsilon_{\mathsf{evalnd}} = m\tilde{B}/B'$.*

[6] For technical reasons (concretely, for the $\varepsilon_{\mathsf{ext}}$-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property of the pr-QA-HPS scheme constructed later), the vector $\mathbf{0}$ must be excluded from the set $\mathbb{Z}_q^n$ that $\mathbf{s}$ is chosen from. For simplicity, we forgo making this explicit in the sequel.

```
$\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$:
Return $\mathsf{pp}_{\mathsf{HPS}} := \mathsf{pp}_{\mathsf{LWE}}$, which implicitly defines
        $(\mathcal{SK} := \{0,1\}^m,\ \mathcal{HV} := \mathbb{Z}_q,\ \Lambda_{(\cdot)})$,
where $\Lambda_{sk}(\mathbf{c}) := \mathbf{c}^\top \cdot \mathbf{k} \in \mathbb{Z}_q$
        for $sk = \mathbf{k} \in \mathcal{SK}$ and $\mathbf{c} \in \mathcal{X} = \mathbb{Z}_q^m$.

$pk_\rho \leftarrow \alpha_\rho(sk)$, where $\rho = \mathbf{A} \in \mathbb{Z}_q^{n \times m}$:
Parse $sk = \mathbf{k} \in \{0,1\}^m$.
$\mathbf{p} := \mathbf{A} \cdot \mathbf{k} \in \mathbb{Z}_q^n$.
Return $pk_\rho := \mathbf{p}$.

$hv \leftarrow_{\$} \mathsf{prPub}(pk_\rho, \mathbf{c}, w_{\mathbf{c}} = (\mathbf{s}, \mathbf{e}))$,
where $\mathbf{c} \in \widetilde{\mathcal{L}}_\rho$ for $\rho = \mathbf{A}$:   // $\mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top$
Parse $pk_\rho = \mathbf{p} \in \mathbb{Z}_q^n$.
$e' \leftarrow_{\$} [-B', B']$.
Return $hv := \mathbf{s}^\top \cdot \mathbf{p} + e' \in \mathbb{Z}_q$.

$hv \leftarrow_{\$} \mathsf{prPriv}(sk, \mathbf{c} \in \mathcal{X})$:
Parse $sk = \mathbf{k} \in \{0,1\}^m$.
$e' \leftarrow_{\$} [-B', B']$.
Return $hv := \mathbf{c}^\top \cdot \mathbf{k} + e' \in \mathbb{Z}_q$.
```

Fig. 3. The probabilistic QA-HPS scheme $\mathsf{prQAHPS}_{\mathsf{LWE}}$ from LWE.

See the technical overview in Sect. 1 for a proof sketch of evaluation indistinguishability. We postpone the proof of Theorem 4 to Appendix E.1.

Through the following theorems, we show the $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching, PK-diversity and $\mathscr{L}_0$-multi-key-multi-extracting of $\mathsf{prQAHPS}_{\mathsf{LWE}}$, as needed for the $\mathsf{MUMC^c}$-CCA security of our PKE in Subsect. 4.2 (cf. Theorem 2), then show the $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting of $\mathsf{prQAHPS}_{\mathsf{LWE}}$, where $\varepsilon_{\mathsf{ext}} \geq \varepsilon_{\mathsf{prPriv}}$, as needed for the strong $\mathsf{MU^c}$-CMA security of our SIG in Subsect. 4.1 (cf. Theorem 1).

The high-level ideas behind the proofs of these theorems are implicitly contained in the security proof sketches for our SIG and PKE schemes using LWE-based pr-QA-HPS as a building block in Sect. 1. We postpone the proofs of these theorems to Appendix E.2, E.3, E.4 and E.5, respectively.

Theorem 5 **($\langle \mathscr{L}, \mathscr{L}_0 \rangle$-Key-Switching of $\mathsf{prQAHPS}_{\mathsf{LWE}}$).** *Let $m > 3n \log q + 2(\lambda + 1)$. The proposed $\mathsf{prQAHPS}_{\mathsf{LWE}}$ in Fig. 3 supports $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching with $\boldsymbol{\epsilon}^{\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{prQAHPS}, \mathcal{A}} \leq 2^{-\lambda}$ for any (possibly unbounded) adversary $\mathcal{A}$.*

Theorem 6 **(PK-Diversity of $\mathsf{prQAHPS}_{\mathsf{LWE}}$).** *The proposed $\mathsf{prQAHPS}_{\mathsf{LWE}}$ in Fig. 3 has PK-diversity with $\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{prQAHPS}} = 2^{-m} + q^{-n}$.*

Theorem 7 **(Almost Tight $\mathscr{L}_0$-Multi-Key-Multi-Extracting of $\mathsf{prQAHPS}_{\mathsf{LWE}}$).** *Let $m > 2n \log q + 2\lambda$. If the $\mathsf{LWE}_{\ell, q, D_{\mathbb{Z},\gamma}, m}$ assumptions hold, then the proposed $\mathsf{prQAHPS}_{\mathsf{LWE}}$ in Fig. 3 supports $\mathscr{L}_0$-multi-key-multi-extracting. Concretely, for any adversary $\mathcal{A}$, any N and any Q, there exist adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$, such that $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{A}) + NQ \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and $\mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{prQAHPS}, \mathcal{A}, N, Q}(\lambda) \leq 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell, q, D_{\mathbb{Z},\gamma}, m], \mathcal{B}_1}(\lambda) + 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell, q, D_{\mathbb{Z},\gamma}, m], \mathcal{B}_2}(\lambda) + \frac{2NQ(m+c+2)+N}{2^\lambda} + NQ \cdot (m+1)B/B'$, where c is an integer satisfying (8).*

Theorem 8 **($\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-Extracting of $\mathsf{prQAHPS}_{\mathsf{LWE}}$).** *Let $\varepsilon_{\mathsf{ext}} \geq \varepsilon_{\mathsf{prPriv}}$, $m > 3n \log q + 2\lambda$ and q be a prime. The proposed $\mathsf{prQAHPS}_{\mathsf{LWE}}$ in Fig. 3 supports $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting with $\boldsymbol{\epsilon}^{\varepsilon_{\mathsf{ext}}\text{-}\langle \mathscr{L}_0, \mathscr{L} \rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS}, \mathcal{A}} \leq 2^{-\lambda} + m\tilde{B}/B' + (2\varepsilon_{\mathsf{ext}} + 2B' + 1)/q$ for any (possibly unbounded) adversary $\mathcal{A}$.*

6.3 Commitment Scheme from LWE

Let $\mathsf{pp}_{\mathsf{LWE}} = (n, m, \ell, q, \sigma, \gamma, \chi, B, \tilde{B}, \cdots)$ be the LWE-related public parameters that serve as implicit input to all algorithms. We present our LWE-based dual-mode gap commitment scheme $\mathsf{CMT}_{\mathsf{LWE}} = (\mathsf{BSetup}, \mathsf{HSetup}, \mathsf{Com})$ in Fig. 4, with two message spaces $\mathcal{M} = \{0,1\}^m \subseteq \widetilde{\mathcal{M}} = [-\tilde{B}, \tilde{B}]^m$ and two randomness spaces $\mathcal{R} = \{0,1\}^{m\times m} \subseteq \widetilde{\mathcal{R}} = [-\tilde{B}, \tilde{B}]^{m\times m}$. The scheme uses a modulus q^2.

```
$\underline{\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{BSetup}:}$   //Binding mode
$\overline{\mathbf{X}} \leftarrow_{\$} \mathbb{Z}_{q^2}^{n\times m}$.
$\mathbf{s} \leftarrow_{\$} \mathbb{Z}_{q^2}^{n}$, $\mathbf{e} \leftarrow_{\$} \chi^m$.
$\mathbf{b}^\top := \mathbf{s}^\top \overline{\mathbf{X}} + \mathbf{e}^\top \bmod q^2$.
$\mathbf{X} := \binom{\overline{\mathbf{X}}}{\mathbf{b}^\top} \in \mathbb{Z}_{q^2}^{(n+1)\times m}$.
Return $\mathsf{pp}_{\mathsf{CMT}} := \mathbf{X}$.

$\underline{\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{HSetup}:}$   //Hiding mode
$\mathbf{X} \leftarrow_{\$} \mathbb{Z}_{q^2}^{(n+1)\times m}$.
Return $\mathsf{pp}_{\mathsf{CMT}} := \mathbf{X}$.

$\underline{\mathsf{com} \leftarrow \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}} = \mathbf{X}, \mathbf{m}; \mathbf{R}):}$   //$\mathbf{m} \in [-\tilde{B}, \tilde{B}]^m$, $\mathbf{R} \in [-\tilde{B}, \tilde{B}]^{m\times m}$
$\mathsf{com} := \mathbf{X} \cdot \mathbf{R} + \binom{\mathbf{0}}{q\cdot \mathbf{m}^\top} \in \mathbb{Z}_{q^2}^{(n+1)\times m}$.   //Here $\mathbf{0}$ is an $n \times m$ zero matrix
Return $\mathsf{com}$.
```

Fig. 4. The dual-mode gap commitment scheme $\mathsf{CMT}_{\mathsf{LWE}}$ from LWE.

This commitment scheme is essentially adapted from the Regev's PKE scheme [45]. Here, the public parameter in the binding mode is just the public key of Regev's scheme, while the committing algorithm is just Regev encryption algorithm. The decryption correctness of Regev's PKE guarantees the property of statistical binding. According to the LWE assumption, the public key of Regev's scheme is computationally indistinguishable from a uniform matrix, which serves as the public parameter in the hiding mode. The statistical hiding property in the hiding mode relies on the fact that a uniform matrix is a good extractor (cf. Lemma 2). Formally, we have Theorem 9 with proof appeared in Appendix F.

Theorem 9. *Let $q > 2mB\tilde{B}$ and $m > 4(n{+}1)\log q{+}2(\lambda{+}1)$. If the $\mathsf{LWE}_{n,q^2,\chi,m}$ assumption holds, then the proposed $\mathsf{CMT}_{\mathsf{LWE}}$ in Fig. 4 is a dual-mode gap commitment scheme that has $\varepsilon_{\mathsf{binding}}$-statistical binding and $\varepsilon_{\mathsf{hiding}}$-statistical hiding with $\varepsilon_{\mathsf{binding}} = 0$ and $\varepsilon_{\mathsf{hiding}} = m \cdot 2^{-\lambda}$. Moreover, for any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ s.t. $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{para\text{-}ind}}_{\mathsf{CMT},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{LWE}}_{[n,q^2,\chi,m],\mathcal{B}}(\lambda)$.*

6.4 QA-NIZK from LWE

In this subsection, we instantiate tag-based QA-NIZK for gap language based on the LWE assumptions. We will follow the generic transformation proposed by Libert et al. in [34, Subsect. 4.2] that compiles any trapdoor Σ-protocol for gap language into tag-based QA-NIZK for the same gap language, and moreover, the transformation is tightness-preserving, i.e., the resulting tag-based QA-NIZK has tight zero-knowledge and tight USS as long as the building blocks are tightly secure. The formal definitions of the building blocks including trapdoor Σ-protocol are provided in Appendix G.1. Therefore, all we need to do is to instantiate trapdoor Σ-protocol for gap language from LWE.

The Gap Language for QA-NIZK. Note that the gap languages needed in our generic SIG and PKE constructions are different. More precisely, for the SIG construction in Subsect. 4.1, the gap language is the $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ defined in Fig. 1, which is determined by the gap language distribution $\mathscr{L}$, the pr-QA-HPS scheme $\mathsf{prQAHPS}$ and the commitment scheme CMT, while for the PKE construction in Subsect. 4.2, the gap language is exactly the $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ generated by $\mathscr{L}$, as defined in Subsect. 6.1.

We make the gap language $\mathcal{GL}_{\rho'}^{\mathsf{(QANIZK)}} = (\mathcal{L}_{\rho'}^{\mathsf{(QANIZK)}}, \widetilde{\mathcal{L}}_{\rho'}^{\mathsf{(QANIZK)}})$ concrete by instantiating it with our LWE-based $\mathscr{L}$ in Subsect. 6.1, $\mathsf{prQAHPS}_{\widetilde{\mathsf{LWE}}}$ in Subsect. 6.2 and $\mathsf{CMT}_{\mathsf{LWE}}$ in Subsect. 6.3. Let $\mathsf{pp}_{\mathsf{LWE}} = (n, m, \ell, q, \sigma, \gamma, \chi, B, \tilde{B}, B', \tilde{B}', \cdots)$ be the LWE-related public parameters that serve as implicit input to all algorithms, where $B < \tilde{B}$ and $B' < \tilde{B}'$. More precisely, let $\rho = \mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a language parameter output by $\mathscr{L}$, and let $\mathsf{pp}_{\mathsf{CMT}} = \mathbf{X} \in \mathbb{Z}_{q^2}^{(n+1) \times m}$ be a parameter generated by BSetup. Then according to Fig. 1, we have $\rho' = (\mathbf{A}, \mathbf{X})$ and the gap language $\mathcal{GL}_{\rho'}^{\mathsf{(QANIZK)}} = (\mathcal{L}_{\rho'}^{\mathsf{(QANIZK)}}, \widetilde{\mathcal{L}}_{\rho'}^{\mathsf{(QANIZK)}})$ is instantiated as follows:

$$\mathcal{L}_{\rho'}^{\mathsf{(QANIZK)}} = \left\{ (\mathbf{c}, vk, d) \;\middle|\; \begin{array}{c} \exists\ (\mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m, \\ \mathbf{R} \in \{0,1\}^{m \times m}, \mathbf{k} \in \{0,1\}^m, \\ e' \in [-B', B']) \end{array} \text{ s.t. } \begin{array}{c} \mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top \\ \wedge\ vk = \mathbf{X} \cdot \mathbf{R} + \binom{\mathbf{0}}{q \cdot \mathbf{k}^\top} \\ \wedge\ d = \mathbf{c}^\top \cdot \mathbf{k} + e' \end{array} \right\}, \quad (12)$$

$$\widetilde{\mathcal{L}}_{\rho'}^{\mathsf{(QANIZK)}} = \left\{ (\mathbf{c}, vk, d) \;\middle|\; \begin{array}{c} \exists\ (\mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-\tilde{B}, \tilde{B}]^m, \\ \mathbf{R} \in [-\tilde{B}, \tilde{B}]^{m \times m}, \mathbf{k} \in [-\tilde{B}, \tilde{B}]^m, \\ e' \in [-\tilde{B}', \tilde{B}']) \end{array} \text{ s.t. } \begin{array}{c} \mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top \\ \wedge\ vk = \mathbf{X} \cdot \mathbf{R} + \binom{\mathbf{0}}{q \cdot \mathbf{k}^\top} \\ \wedge\ d = \mathbf{c}^\top \cdot \mathbf{k} + e' \end{array} \right\}. \quad (13)$$

The Trapdoor Σ-protocol from LWE. Observe that no matter the gap language $\mathcal{GL}_{\rho'}^{\mathsf{(QANIZK)}} = (\mathcal{L}_{\rho'}^{\mathsf{(QANIZK)}}, \widetilde{\mathcal{L}}_{\rho'}^{\mathsf{(QANIZK)}})$ defined in (12) and (13) or the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ defined in Subsect. 6.1, both of them are defined with linear equations, i.e., the instance is linear in the witness, and parts of the witness are bounded. To build trapdoor Σ-protocol for these gap languages, we are inspired by the trapdoor Σ-protocol for ACPS ciphertexts [4] with tight security constructed by Libert et al. in [34, Sect. 5], where the gap languages defined by ACPS ciphertexts enjoy similar properties described as above.

Roughly speaking, our trapdoor Σ-protocol for $\mathcal{GL}_{\rho'}^{\mathsf{(QANIZK)}}$ works as follows. To prove $(\mathbf{c}, vk, d) \in \mathcal{L}_{\rho'}^{\mathsf{(QANIZK)}}$ with the help of a witness $(\mathbf{s}, \mathbf{e}, \mathbf{R}, \mathbf{k}, e')$, the prover first generates a fresh instance $(\mathbf{c}_0, vk_0, d_0)$ by sampling witness $(\mathbf{s}_0, \mathbf{e}_0, \mathbf{R}_0, \mathbf{k}_0, e'_0)$ appropriately and sends it to the verifier, then the verifier chooses a challenge $\mathsf{ch} \in \{0, 1\}$ uniformly at random. According to the linear properties, the "mixed" $(\mathbf{s}_0 + \mathsf{ch} \cdot \mathbf{s}, \mathbf{e}_0 + \mathsf{ch} \cdot \mathbf{e}, \mathbf{R}_0 + \mathsf{ch} \cdot \mathbf{R}, \mathbf{k}_0 + \mathsf{ch} \cdot \mathbf{k}, e'_0 + \mathsf{ch} \cdot e')$ is also a witness for the "mixed" instance $(\mathbf{c}_0 + \mathsf{ch} \cdot \mathbf{c}, vk_0 + \mathsf{ch} \cdot vk, d_0 + \mathsf{ch} \cdot d)$ to satisfy the equations in (12) and (13). Therefore, the prover sends the "mixed" witness to the verifier, and the verifier checks the equations in (12) and (13) for the "mixed" instance and witness and also checks whether the corresponding parts of the "mixed" witness (namely $\mathbf{e}_0 + \mathsf{ch} \cdot \mathbf{e}, \mathbf{R}_0 + \mathsf{ch} \cdot \mathbf{R}, \mathbf{k}_0 + \mathsf{ch} \cdot \mathbf{k}, e'_0 + \mathsf{ch} \cdot e'$) are bounded.

The trapdoor Σ-protocol for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ is a simplified version of that for $\mathcal{GL}_{\rho'}^{\mathsf{(QANIZK)}} = (\mathcal{L}_{\rho'}^{\mathsf{(QANIZK)}}, \widetilde{\mathcal{L}}_{\rho'}^{\mathsf{(QANIZK)}})$, since $\mathcal{GL}_\rho$ is much simper.

We put the formal descriptions of the LWE-based trapdoor Σ-protocols and their security proof in Appendix G.3.

The QA-NIZK from LWE. Finally, by compiling the LWE-based trapdoor Σ-protocols via the generic transformation proposed by Libert et al. in [34, Subsect. 4.2], we are able to obtain tag-based QA-NIZK schemes for the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ and $\mathcal{GL}_{\rho} = (\mathcal{L}_{\rho}, \widetilde{\mathcal{L}}_{\rho})$ from the LWE assumptions, serving as building blocks for our SIG and PKE constructions.

For completeness, in Appendix G.4, we first recall the generic transformation in [34, Subsect. 4.2], then describe how to compile our LWE-based trapdoor Σ-protocols into tag-based QA-NIZK schemes for gap languages. Especially, we obtain the following corollary in Appendix G.4.

Corollary 1 (Almost Tight Security of LWE-based QA-NIZK) *We obtain a tag-based QA-NIZK scheme for the gap language* $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ *specified by (12) and (13) and a tag-based QA-NIZK scheme for the gap language* $\mathcal{GL}_{\rho} = (\mathcal{L}_{\rho}, \widetilde{\mathcal{L}}_{\rho})$ *specified in Subsect. 6.1, both of which have almost tight zero-knowledge and USS based on the LWE assumption.*

Concretely, the advantage of zero-knowledge for any (even all powerful) adversary $\mathcal{A}'$ *is given by* $\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{A}'}(\lambda) \leq 2^{-\Omega(\lambda)}$. *Meanwhile, the advantage of USS for any PPT adversary* $\mathcal{A}$ *is given by*

$$\mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SIS}}_{[n,q,m,\beta],\mathcal{B}_1}(\lambda) + 2\lambda^2 \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\lambda,q,\chi,m],\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)},$$

where PPT algorithms $\mathcal{B}_1$ *and* $\mathcal{B}_2$ *run in about the same time as* $\mathcal{A}$.

6.5 Setting the Parameters

We give a suggestion for parameters $\mathsf{pp}_{\mathsf{LWE}} = (n, m, \ell, q, \sigma, \gamma, \chi, B, \tilde{B}, B', \tilde{B}', \zeta, \zeta')$ in Table 4, so that all conditions of the theorems in the section can be met. Moreover, our parameter suggestion in Table 4 corresponds to the parameter Setting II in Table 3, thus the conditions in Theorem 3 (almost tight reduction from LWE to multi-secret LWE) are also satisfied. By instantiating our generic constructions in Sect. 4 with the LWE-based building blocks proposed in this section, we obtain LWE-based SIG and PKE schemes with almost tight strong $\mathsf{MU^c}$-CMA and $\mathsf{MUMC^c}$-CCA security, respectively. Under the parameters in Table 4, the security loss factor of our schemes is $O(\lambda^2)$.

Table 4. Parameter setting, where λ denotes the security parameter.

Parameters	n	m	ℓ	q	σ	γ	χ
Setting	4λ	λ^2	λ	$2^{2\sqrt{\lambda}}$	$\lambda^{2.5} \cdot 2^{\sqrt{\lambda}}$	λ	$D_{\mathbb{Z},\lambda^{2.5} \cdot 2^{\sqrt{\lambda}}}$

Parameters	B	$\tilde{B}$	B'	$\tilde{B}'$	ζ	ζ'
Setting	$\lambda^3 \cdot 2^{\sqrt{\lambda}}$	$\lambda^6 \cdot 2^{\sqrt{\lambda}}$	$2^{1.5\sqrt{\lambda}}$	$\lambda \cdot 2^{1.5\sqrt{\lambda}}$	$\lambda^{4.5} \cdot 2^{\sqrt{\lambda}}$	$\sqrt{\lambda} \cdot 2^{1.5\sqrt{\lambda}}$

Acknowledgments. We would like to thank the reviewers for their valuable comments. Shuai Han and Shengli Liu were partially supported by Guangdong Major Project of Basic and Applied Basic Research (2019B030302008), National

Natural Science Foundation of China (Grant Nos. 62002223, 61925207), the National Key R&D Program of China under Grant 2022YFB2701500, Shanghai Sailing Program (20YF1421100), and Young Elite Scientists Sponsorship Program by China Association for Science and Technology (YESS20200185). Zhedong Wang was partially supported by National Natural Science Foundation of China (Grant No. 62202305), Young Elite Scientists Sponsorship Program by China Association for Science and Technology (YESS20220150) and Shanghai Pujiang Program under Grant 22PJ1407700. Dawu Gu was partially supported by the National Key R&D Program of China under Grant 2020YFA0712302.

References

[1] Ajtai, M.: Generating hard instances of lattice problems. In: STOC 1996, pp. 99–108

[2] Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited - new reduction, properties and applications. In: CRYPTO 2013, pp. 57–74

[3] An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: EUROCRYPT 2002, pp. 83–107

[4] Applebaum, B., Cash, D., Peikert, C., Sahai, A.: Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In: CRYPTO 2009, pp. 595–618

[5] Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: EUROCRYPT 2012, pp. 483–501

[6] Bader, C., Hofheinz, D., Jager, T., Kiltz, E., Li, Y.: Tightly-secure authenticated key exchange. In: TCC 2015, pp. 629–658

[7] Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: EUROCRYPT 2016, pp. 273–304

[8] Badertscher, C., Banfi, F., Maurer, U.: A constructive perspective on signcryption security. In: SCN 18, pp. 102–120

[9] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: CRYPTO 1993, pp. 232–249

[10] Bellare, M., Stepanovs, I.: Security under message-derived keys: Signcryption in iMessage. In: EUROCRYPT 2020, pp. 507–537

[11] Benhamouda, F., Blazy, O., Ducas, L., Quach, W.: Hash proof systems over lattices revisited. In: PKC 2018, pp. 644–674

[12] Böhl, F., Hofheinz, D., Jager, T., Koch, J., Seo, J.H., Striecks, C.: Practical signatures from standard assumptions. In: EUROCRYPT 2013, pp. 461–485

[13] Boneh, D., Lewi, K., Montgomery, H.W., Raghunathan, A.: Key homomorphic PRFs and their applications. In: CRYPTO 2013, pp. 410–428

[14] Boyen, X.: Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In: PKC 2010, pp. 499–517

[15] Brakerski, Z., Döttling, N.: Hardness of LWE on general entropic distributions. In: EUROCRYPT 2020, pp. 551–575

[16] Canetti, R., Chen, Y., Holmgren, J., Lombardi, A., Rothblum, G.N., Rothblum, R.D., Wichs, D.: Fiat-Shamir: from practice to theory. In: STOC 2019, pp. 1082–1090

[17] Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: EUROCRYPT 2001, pp. 453–474

[18] Canetti, R., Lombardi, A., Wichs, D.: Non-interactive zero knowledge and correlation intractability from circular-secure FHE. Cryptology ePrint Archive, Report 2018/1248 (2018), https://eprint.iacr.org/2018/1248

[19] Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: EUROCRYPT 2002, pp. 45–64

[20] Diemert, D., Gellert, K., Jager, T., Lyu, L.: More efficient digital signatures with tight multi-user security. In: PKC 2021, pp. 1–31

[21] 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)

[22] Ducas, L., Micciancio, D.: Improved short lattice signatures in the standard model. In: CRYPTO 2014, pp. 335–352

[23] Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly CCA-secure encryption without pairings. In: EUROCRYPT 2016, pp. 1–27

[24] Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. ACM Transactions on Information and System Security 9(2), 181–234 (2006)

[25] Gjøsteen, K., Jager, T.: Practical and tightly-secure digital signatures and authenticated key exchange. In: CRYPTO 2018, pp. 95–125

[26] Han, S., Jager, T., Kiltz, E., Liu, S., Pan, J., Riepel, D., Schäge, S.: Authenticated key exchange and signatures with tight security in the standard model. In: CRYPTO 2021, pp. 670–700

[27] Han, S., Liu, S., Gu, D.: Almost tight multi-user security under adaptive corruptions & leakages in the standard model. In: EUROCRYPT 2023, pp. 132–162

[28] Han, S., Liu, S., Lyu, L., Gu, D.: Tight leakage-resilient CCA-security from quasi-adaptive hash proof system. In: CRYPTO 2019, pp. 417–447

[29] Jiang, S., Gong, G., He, J., Nguyen, K., Wang, H.: PAKEs: New framework, new techniques and more efficient lattice-based constructions in the standard model. In: PKC 2020, pp. 396–427

[30] Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: ASIACRYPT 2013, pp. 1–20

[31] Katz, J., Vaikuntanathan, V.: Smooth projective hashing and password-based authenticated key exchange from lattices. In: ASIACRYPT 2009, pp. 636–652

[32] LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger security of authenticated key exchange. In: ProvSec 2007, pp. 1–16

[33] Lee, Y., Lee, D.H., Park, J.H.: Tightly CCA-secure encryption scheme in a multi-user setting with corruptions. Des. Codes Cryptogr. 88(11), 2433–2452 (2020)

[34] Libert, B., Nguyen, K., Passelègue, A., Titiu, R.: Simulation-sound arguments for LWE and applications to KDM-CCA2 security. In: ASIACRYPT 2020, pp. 128–158

[35] Libert, B., Sakzad, A., Stehlé, D., Steinfeld, R.: All-but-many lossy trapdoor functions and selective opening chosen-ciphertext security from LWE. In: CRYPTO 2017, pp. 332–364

[36] Lyubashevsky, V.: Lattice signatures without trapdoors. In: EUROCRYPT 2012, pp. 738–755

[37] Micciancio, D., Mol, P.: Pseudorandom knapsacks and the sample complexity of LWE search-to-decision reductions. In: CRYPTO 2011, pp. 465–484

[38] Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In: EUROCRYPT 2012, pp. 700–718

[39] Morgan, A., Pass, R., Shi, E.: On the adaptive security of MACs and PRFs. In: ASIACRYPT 2020, pp. 724–753

[40] Pan, J., Wagner, B.: Lattice-based signatures with tight adaptive corruptions and more. In: PKC 2022, pp. 347–378

[41] Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: STOC 2009, pp. 333–342

[42] Peikert, C.: An efficient and parallel Gaussian sampler for lattices. In: CRYPTO 2010, pp. 80–97

[43] Peikert, C., Shiehian, S.: Noninteractive zero knowledge for NP from (plain) learning with errors. In: CRYPTO 2019, pp. 89–114

[44] Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008, pp. 187–196

[45] Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84–93

[46] Zhang, J., Yu, Y.: Two-round PAKE from approximate SPH and instantiations from lattices. In: ASIACRYPT 2017, pp. 37–67

Appendix

A Additional Preliminaries

A.1 Digital Signature and Its Strong $\mathsf{MU^c\text{-}CMA}$ Security

Definition 13 (SIG). *A signature (SIG) scheme* $\mathsf{SIG} = (\mathsf{Setup_{SIG}}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy_{SIG}})$ *with message space* $\mathcal{M}$ *consists of four PPT algorithms:*

- $\mathsf{pp_{SIG}} \leftarrow_{\$} \mathsf{Setup_{SIG}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp_{SIG}}$*, which serves as an implicit input of other algorithms.*
- $(vk, sigk) \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp_{SIG}})$*: Taking* $\mathsf{pp_{SIG}}$ *as input, the key generation algorithm outputs a pair of verification key and signing key* $(vk, sigk)$*.*
- $\sigma \leftarrow_{\$} \mathsf{Sign}(sigk, m)$*: Taking as input a signing key sigk and a message* $m \in \mathcal{M}$*, the signing algorithm outputs a signature* σ*.*
- $0/1 \leftarrow \mathsf{Vrfy_{SIG}}(vk, m, \sigma)$*: Taking as input a verification key vk, a message* $m \in \mathcal{M}$ *and a signature* σ*, the deterministic verification algorithm outputs a bit indicating whether* σ *is a valid signature for* m *w.r.t.* vk*.*

Correctness requires that for all $\mathsf{pp_{SIG}} \in \mathsf{Setup_{SIG}}$*,* $(vk, sigk) \in \mathsf{Gen}(\mathsf{pp_{SIG}})$*,* $m \in \mathcal{M}$*, it holds that* $\Pr[\sigma \leftarrow_{\$} \mathsf{Sign}(sigk, m) : \mathsf{Vrfy_{SIG}}(vk, m, \sigma) = 1] \geq 1 - \mathsf{negl}(\lambda)$*.*

In [6], Bader et al. defined existential unforgeability for digital signatures under chosen-message attacks (CMA) in a Multi-User setting with adaptive corruptions of secret keys ($\mathsf{MU^c\text{-}CMA}$). Moreover, *strong* $\mathsf{MU^c\text{-}CMA}$ requires that the adversary cannot even forge a new signature for a message that it has ever queried. Below we present the definition of strong $\mathsf{MU^c\text{-}CMA}$ security.

Definition 14 (Strong $\mathsf{MU^c\text{-}CMA}$ Security for SIG). *A signature scheme* SIG *is strongly* $\mathsf{MU^c\text{-}CMA}$ *secure, if for any PPT* $\mathcal{A}$ *and any polynomial* N*, it holds that* $\mathsf{Adv}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N}$ *is defined in Fig. 5.*

```
$\mathsf{Exp}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N}$:
$\mathsf{pp_{SIG}} \leftarrow_{\$} \mathsf{Setup_{SIG}}$
For $i \in [N]$: $(vk_i, sigk_i) \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp_{SIG}})$
$\mathcal{Q}_{\text{SIGN}} := \emptyset$                                  //Record the signing queries
$\mathcal{Q}_{\text{COR}} := \emptyset$                               //Record the corruption queries
$(i^* \in [n], m^*, \sigma^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{SIGN}}(\cdot,\cdot),\mathcal{O}_{\text{COR}}(\cdot)}(\mathsf{pp_{SIG}}, \{vk_i\}_{i\in[N]})$
If $(i^* \notin \mathcal{Q}_{\text{COR}}) \wedge ((i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{SIGN}}) \wedge (\mathsf{Vrfy_{SIG}}(vk_{i^*}, m^*, \sigma^*) = 1)$:
    Return 1;
Else: Return 0

$\mathcal{O}_{\text{SIGN}}(i, m)$:
    $\sigma \leftarrow_{\$} \mathsf{Sign}(sigk_i, m)$
    $\mathcal{Q}_{\text{SIGN}} := \mathcal{Q}_{\text{SIGN}} \cup \{(i, m, \sigma)\}$
    Return $\sigma$

$\mathcal{O}_{\text{COR}}(i)$:
    $\mathcal{Q}_{\text{COR}} := \mathcal{Q}_{\text{COR}} \cup \{i\}$
    Return $sigk_i$
```

Fig. 5. The strong $\mathsf{MU^c\text{-}CMA}$ security experiment $\mathsf{Exp}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N}$ for SIG.

A.2 Public-Key Encryption and Its $\mathsf{MUMC^c\text{-}CCA}$ Security

Definition 15 (PKE). *A public-key encryption (PKE) scheme* $\mathsf{PKE} = (\mathsf{Setup_{PKE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}$ *consists of four PPT algorithms:*

- $\mathsf{pp_{PKE}} \leftarrow_\$ \mathsf{Setup_{PKE}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp_{PKE}}$*, which serves as an implicit input of other algorithms.*
- $(pk, sk) \leftarrow_\$ \mathsf{Gen}(\mathsf{pp_{PKE}})$*: Taking* $\mathsf{pp_{PKE}}$ *as input, the key generation algorithm outputs a pair of public key and secret key* (pk, sk)*.*
- $c \leftarrow_\$ \mathsf{Enc}(pk, m)$*: Taking as input a public key* pk *and a message* $m \in \mathcal{M}$*, the encryption algorithm outputs a ciphertext* c*.*
- $m'/\perp \leftarrow \mathsf{Dec}(sk, c)$*: Taking as input a secret key* sk *and a ciphertext* c*, the deterministic decryption algorithm outputs either a message* $m' \in \mathcal{M}$ *or a special symbol* $\perp$ *indicating the failure of decryption.*

Correctness requires that for all $\mathsf{pp_{PKE}} \in \mathsf{Setup_{PKE}}$*,* $(pk, sk) \in \mathsf{Gen}(\mathsf{pp_{PKE}})$ *and* $m \in \mathcal{M}$*, it holds that* $\Pr[c \leftarrow_\$ \mathsf{Enc}(pk, m) : \mathsf{Dec}(sk, c) = m] \geq 1 - \mathsf{negl}(\lambda)$*.*

In [33], Lee et al. defined indistinguishability for PKE schemes under chosen-ciphertext attacks (CCA) in a Multi-User Multi-Challenge setting with adaptive corruptions of secret keys, which was originally called MUC$^+$ in [33] and is denoted by $\mathsf{MUMC^c\text{-}CCA}$ in this paper. Below we present the definition of $\mathsf{MUMC^c\text{-}CCA}$ security.

Definition 16 ($\mathsf{MUMC^c\text{-}CCA}$ Security for PKE). *A PKE scheme* PKE *is* $\mathsf{MUMC^c\text{-}CCA}$ *secure, if for any PPT* $\mathcal{A}$ *and any polynomial* N*, it holds that* $\mathsf{Adv}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N} \Rightarrow 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N}$ *is defined in Fig. 6.*

```
$\mathsf{Exp}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N}$:
$\mathsf{pp_{PKE}} \leftarrow_\$ \mathsf{Setup_{PKE}}$
For $i \in [N]$:  $(pk_i, sk_i) \leftarrow_\$ \mathsf{Gen}(\mathsf{pp_{PKE}})$
$\mathcal{Q}_{\text{ENC}} := \emptyset$          //Record the encryption queries
$\mathcal{Q}_{\text{COR}} := \emptyset$          //Record the corruption queries
$\beta \leftarrow_\$ \{0,1\}$                     //Single challenge bit
$\beta' \leftarrow_\$ \mathcal{A}^{\mathcal{O}_{\text{ENC}}(\cdot,\cdot,\cdot),\mathcal{O}_{\text{DEC}}(\cdot,\cdot),\mathcal{O}_{\text{COR}}(\cdot)}(\mathsf{pp_{PKE}}, \{pk_i\}_{i\in[N]})$
If $\beta' = \beta$:  Return 1;   Else:  Return 0

$\mathcal{O}_{\text{ENC}}(i^*, m_0, m_1)$:
    If $|m_0| \neq |m_1|$: Return $\perp$
    If $i^* \in \mathcal{Q}_{\text{COR}}$: Return $\perp$
    $c^* \leftarrow_\$ \mathsf{Enc}(pk_{i^*}, m_\beta)$
    $\mathcal{Q}_{\text{ENC}} := \mathcal{Q}_{\text{ENC}} \cup \{(i^*, c^*)\}$
    Return $c^*$

$\mathcal{O}_{\text{DEC}}(i, c)$:
    If $(i, c) \in \mathcal{Q}_{\text{ENC}}$: Return $\perp$
    Return $\mathsf{Dec}(sk_i, c)$

$\mathcal{O}_{\text{COR}}(i)$:
    If $(i, \cdot) \in \mathcal{Q}_{\text{ENC}}$: Return $\perp$
    $\mathcal{Q}_{\text{COR}} := \mathcal{Q}_{\text{COR}} \cup \{i\}$
    Return $sk_i$
```

Fig. 6. The $\mathsf{MUMC^c\text{-}CCA}$ security experiment $\mathsf{Exp}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N}$ for PKE. Note that to avoid trivial attacks, $\mathcal{A}$ is not allowed to submit a same user index i to both $\mathcal{O}_{\text{ENC}}$ and $\mathcal{O}_{\text{COR}}$.

A.3 Quasi-Adaptive Non-Interactive Zero-Knowledge Argument

Quasi-Adaptive Non-Interactive Zero-Knowledge argument (QA-NIZK) was proposed by Jutla and Roy [30], where the common reference string (CRS) may depend on the specific language $\mathcal{L}_\rho$ for which proofs are generated. Tag-based

QA-NIZK additionally takes a tag as input when generating and verifying proofs. Below we formalize tag-based QA-NIZK according to [34], for a gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ $(\mathcal{L}_\rho \subseteq \widetilde{\mathcal{L}}_\rho)$ indexed by language parameter ρ. Intuitively, completeness and zero-knowledge of QA-NIZK are guaranteed for instances in $\mathcal{L}_\rho$, while soundness is guaranteed for instances outside $\widetilde{\mathcal{L}}_\rho$.

Definition 17 (Tag-based QA-NIZK for Gap Language). *A tag-based QA-NIZK scheme* $\mathsf{QANIZK} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{SimGen}, \mathsf{Sim})$ *for a gap language* $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ *with tag space* $\mathcal{T}$ *consists of five PPT algorithms:*

- $\mathsf{crs} \leftarrow_\$ \mathsf{CRSGen}(\rho)$*: Taking as input the language parameter* ρ*, the CRS generation algorithm outputs a common reference string (CRS)* crs*.*
- $\pi \leftarrow_\$ \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$*: Taking as input* crs*, a tag* $\tau \in \mathcal{T}$*,* $x \in \mathcal{L}_\rho$ *and a witness* w *for* $x \in \mathcal{L}_\rho$*, the proof generation algorithm outputs a proof* π*.*
- $0/1 \leftarrow \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi)$*: Taking as input* crs*, a tag* $\tau \in \mathcal{T}$*,* $x \in \mathcal{X}$ *and a proof* π*, the deterministic verification algorithm outputs a bit indicating whether* π *is a valid proof.*
- $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_\$ \mathsf{SimGen}(\rho)$*: Taking as input the parameter* ρ*, the simulated CRS generation algorithm outputs a* crs *and a simulation trapdoor* $\mathsf{td}_{\mathsf{crs}}$*.*
- $\pi \leftarrow_\$ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$*: Taking as input* crs*, a simulation trapdoor* $\mathsf{td}_{\mathsf{crs}}$*, a tag* $\tau \in \mathcal{T}$ *and* $x \in \mathcal{X}$*, the simulation algorithm outputs a simulated proof* π*.*

Completeness requires: for all $\mathsf{crs} \in \mathsf{CRSGen}(\rho)$*,* $\tau \in \mathcal{T}$ *and* $x \in \mathcal{L}_\rho$ *with witness* w*, it holds* $\Pr[\pi \leftarrow_\$ \mathsf{Prove}(\mathsf{crs}, \tau, x, w) : \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1] \geq 1 - \mathsf{negl}(\lambda)$*.*

Below we define the *zero-knowledge* and the *unbounded simulation-soundness* (USS) according to [34].

Definition 18 (Zero-Knowledge of Tag-based QA-NIZK). *A tag-based QA-NIZK scheme* QANIZK *for gap language* $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ *has zero-knowledge, if for any PPT* $\mathcal{A}$*, it holds that* $\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{A}}(\lambda) := \big|\Pr[\mathsf{Exp}^{\mathsf{zk},(0)}_{\mathsf{QANIZK},\mathcal{A}} \Rightarrow 1] - \Pr[\mathsf{Exp}^{\mathsf{zk},(1)}_{\mathsf{QANIZK},\mathcal{A}} \Rightarrow 1]\big| \leq \mathsf{negl}(\lambda)$*, where the experiments* $\mathsf{Exp}^{\mathsf{zk},(0)}_{\mathsf{QANIZK},\mathcal{A}}$ *and* $\mathsf{Exp}^{\mathsf{zk},(1)}_{\mathsf{QANIZK},\mathcal{A}}$ *are defined in Fig. 7.*

```
$\mathsf{Exp}^{\mathsf{zk},(0)}_{\mathsf{QANIZK},\mathcal{A}}$:
$\mathsf{crs} \leftarrow_\$ \mathsf{CRSGen}(\rho)$
$\beta \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{(0)}_{\text{PRV}}(\cdot,\cdot,\cdot)}(\rho, \mathsf{crs})$
Return $\beta$

$\mathcal{O}^{(0)}_{\text{PRV}}(\tau, x, w)$:
    If $w$ is not a witness for $x \in \mathcal{L}_\rho$: Return $\perp$
    Else: $\pi \leftarrow_\$ \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$, Return $\pi$
```

```
$\mathsf{Exp}^{\mathsf{zk},(1)}_{\mathsf{QANIZK},\mathcal{A}}$:
$(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_\$ \mathsf{SimGen}(\rho)$
$\beta \leftarrow_\$ \mathcal{A}^{\mathcal{O}^{(1)}_{\text{PRV}}(\cdot,\cdot,\cdot)}(\rho, \mathsf{crs})$
Return $\beta$

$\mathcal{O}^{(1)}_{\text{PRV}}(\tau, x, w)$:
    If $w$ is not a witness for $x \in \mathcal{L}_\rho$: Return $\perp$
    Else: $\pi \leftarrow_\$ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$, Return $\pi$
```

Fig. 7. The zero-knowledge experiments $\mathsf{Exp}^{\mathsf{zk},(0)}_{\mathsf{QANIZK},\mathcal{A}}$ and $\mathsf{Exp}^{\mathsf{zk},(1)}_{\mathsf{QANIZK},\mathcal{A}}$ for QANIZK.

We note that the above definition captures a notion of *multi-theorem* zero-knowledge, which allows the adversary to obtain proofs for multiple statements.

Definition 19 (USS of Tag-based QA-NIZK). *A tag-based QA-NIZK scheme* QANIZK *for gap language* $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ *has unbounded simulation-soundness (USS), if for any PPT* $\mathcal{A}$*, it holds that* $\mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}} \Rightarrow 1] \le \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}$ *is defined in Fig. 8.*

```
$\underline{\mathsf{Exp}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}:}$
$(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{SimGen}(\rho)$
$\mathcal{Q}_{\text{SIM}} := \emptyset$                                   //Record the simulation queries
$(\tau^*, x^*, \pi^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{SIM}}(\cdot,\cdot)}(\rho, td_\rho, \mathsf{crs})$      //Recall that $td_\rho$ is a trapdoor
                                                   for testing membership of $\widetilde{\mathcal{L}}_\rho$
If $(x^* \notin \widetilde{\mathcal{L}}_\rho) \wedge ((\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}}) \wedge (\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1)$: Return 1;
Else: Return 0

$\underline{\mathcal{O}_{\text{SIM}}(\tau, x):}$
    $\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$
    $\mathcal{Q}_{\text{SIM}} := \mathcal{Q}_{\text{SIM}} \cup \{(\tau, x, \pi)\}$
    Return $\pi$
```

Fig. 8. The unbounded simulation-soundness experiment $\mathsf{Exp}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}$ for QANIZK.

We note that the above USS definition is different from the usual one in [23] in the following three aspects.

- Firstly, $\mathcal{A}$ is given the trapdoor td_ρ of the language parameter ρ. Recall that td_ρ contains enough information for deciding whether an instance x is in $\widetilde{\mathcal{L}}_\rho$.
- Secondly, $\mathcal{A}$ is allowed to output a forgery with a reused tag.
- Thirdly, the instance x^* in $\mathcal{A}$'s forgery should be outside $\widetilde{\mathcal{L}}_\rho$ rather than $\mathcal{L}_\rho$.

A.4 Collision-Resistant Hash Functions

Definition 20 (Collision-Resistant Hash Functions). *A family of hash functions* $\mathcal{H}$ *is collision-resistant, if for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{A}}(\lambda) := \Pr[H \leftarrow_{\$} \mathcal{H}, (x_1, x_2) \leftarrow_{\$} \mathcal{A}(H) : x_1 \neq x_2 \wedge H(x_1) = H(x_2)] \le \mathsf{negl}(\lambda).$$

A.5 Error-Correcting Code

Definition 21 (Error-Correcting Code). *An error-correcting code* $\mathsf{ECC} = (\mathsf{Encode}, \mathsf{Decode})$ *from a message set* $\mathcal{M}$ *to a codeword set* $\mathcal{C}$ *consists of two deterministic polynomial-time algorithms:*

- $c \leftarrow \mathsf{Encode}(m)$*: Taking a message* $m \in \mathcal{M}$ *as input, the encoding algorithm outputs a codeword* $c \in \mathcal{C}$*.*
- $m'/\perp \leftarrow \mathsf{Decode}(c)$*: Taking an element* $c \in \mathcal{C}$ *as input, the decoding algorithm outputs either a message* $m' \in \mathcal{M}$ *or a special symbol* $\perp$ *indicating the failure of decoding.*

We say that ECC *is able to correct* ε *errors (*ε*-correctness), if for all* $m \in \mathcal{M}$*,* $c := \mathsf{Encode}(m)$ *and* $c' \in \mathsf{Ball}_\varepsilon(c)$*, it holds that* $m = \mathsf{Decode}(c')$*.*

A.6 Quasi-Adaptive Hash Proof System

We recall the formal definition of QA-HPS according to [28].

Definition 22 (QA-HPS). *A quasi-adaptive hash proof system (QA-HPS) scheme* $\mathsf{QAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{Pub}, \mathsf{Priv})$ *for a language distribution* $\mathscr{L}$ *consists of four PPT algorithms:*

- $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp}_{\mathsf{HPS}}$*, which implicitly defines a hashing key space* $\mathcal{SK}$*, a hash value space* $\mathcal{HV}$*, and a family of hash functions* $\Lambda_{(\cdot)} : \mathcal{X} \longrightarrow \mathcal{HV}$ *indexed by hashing keys* $sk \in \mathcal{SK}$*, where* $\mathcal{X}$ *is the universe for languages output by* $\mathscr{L}$*.*

  We require that $\Lambda_{(\cdot)}$ *is efficiently computable and there are PPT algorithms for sampling* $sk \leftarrow_{\$} \mathcal{SK}$ *uniformly and sampling* $hv \leftarrow_{\$} \mathcal{HV}$ *uniformly. We require* $\mathsf{pp}_{\mathsf{HPS}}$ *to be an implicit input of other algorithms.*
- $pk_\rho \leftarrow \alpha_\rho(sk)$*: Taking as input a hashing key* $sk \in \mathcal{SK}$*, the projection algorithm indexed by language parameter* ρ *outputs a projection key* pk_ρ*.*
- $hv \leftarrow \mathsf{Pub}(pk_\rho, x, w)$*: Taking as input a projection key* $pk_\rho = \alpha_\rho(sk)$ *specified by* ρ*, an instance* $x \in \mathcal{L}_\rho$ *and a witness* w *for* $x \in \mathcal{L}_\rho$*, the deterministic public evaluation algorithm outputs a hash value* $hv \in \mathcal{HV}$*.*
- $hv \leftarrow \mathsf{Priv}(sk, x)$*: Taking as input a hashing key* sk *and an instance* $x \in \mathcal{X}$*, the deterministic private evaluation algorithm outputs a hash value* $hv \in \mathcal{HV}$*.*

Correctness requires: for all $(\rho, td_\rho) \in \mathscr{L}$*,* $\mathsf{pp}_{\mathsf{HPS}} \in \mathsf{Setup}_{\mathsf{HPS}}$*,* $sk \in \mathcal{SK}$*,* $x \in \mathcal{L}_\rho$ *with witness* w*,* $pk_\rho := \alpha_\rho(sk)$*, it holds that* $\mathsf{Pub}(pk_\rho, x, w) = \Lambda_{sk}(x) = \mathsf{Priv}(sk, x)$*.*

B Proof of Theorem 1 (Strong MU$^{\mathsf{c}}$-CMA Security of SIG)

Theorem 1 (Strong $\mathsf{MU^c\text{-}CMA}$ **Security of** SIG**)** *Assume that (i)* $\mathscr{L}$ *and* $\mathscr{L}_0$ *have hard SMPs, (ii)* $\mathsf{prQAHPS}$ *is a probabilistic QA-HPS for both* $\mathscr{L}$ *and* $\mathscr{L}_0$*, having* $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$*-approximate correctness,* $\varepsilon_{\mathsf{evalnd}}$*-evaluation indistinguishability, and supporting* $\varepsilon_{\mathsf{ext}}$*-*$\langle \mathscr{L}_0, \mathscr{L} \rangle$*-OT-extracting, where* $\varepsilon_{\mathsf{ext}} \geq \varepsilon_{\mathsf{prPriv}}$*, (iii)* CMT *is a dual-mode gap commitment scheme that is* $\varepsilon_{\mathsf{binding}}$*-statistical binding and* $\varepsilon_{\mathsf{hiding}}$*-statistical hiding, (iv)* QANIZK *is a tag-based QA-NIZK for the gap language* $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})}$ *defined in Fig. 1, satisfying both zero-knowledge and unbounded simulation-soundness, (iv)* $\mathcal{H}$ *is collision-resistant. Then the proposed* SIG *scheme in Fig. 1 is strongly* $\mathsf{MU^c\text{-}CMA}$ *secure.*

Concretely, for any number N *of users and any adversary* $\mathcal{A}$ *making at most* Q_s *times of* $\mathcal{O}_{\text{SIGN}}$ *queries, there exist adversaries* $\mathcal{B}_1, \cdots, \mathcal{B}_7$*, s.t.* $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_6) \approx \mathbf{T}(\mathcal{A}) + (N + Q_s) \cdot \mathsf{poly}(\lambda)$*, with* $\mathsf{poly}(\lambda)$ *independent of* $\mathbf{T}(\mathcal{A})$*, and*

$$\mathsf{Adv}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N}(\lambda) \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda)$$
$$+ \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \mathsf{Adv}^{\mathsf{para\text{-}ind}}_{\mathsf{CMT},\mathcal{B}_6}(\lambda) + \textit{statist. loss},$$

where statist. loss $= 2 \cdot \varepsilon_{\mathsf{binding}} + Q_s \cdot \varepsilon_{\mathsf{evalnd}} + N \cdot \epsilon^{\varepsilon_{\mathsf{ext}}\text{-}\langle \mathscr{L}_0, \mathscr{L} \rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS},\mathcal{B}_7} + \varepsilon_{\mathsf{hiding}} + \frac{N(N-1)}{2} / |\mathcal{SK}|$.

Proof of Theorem 1. We prove the theorem by defining a sequence of games G_0–G_7 and showing adjacent games indistinguishable. A brief description of differences between adjacent games is summarized in Table 1. By $\Pr_i[\cdot]$ we denote the probability of a particular event occurring in game G_i.

Game G_0: This is the $\mathsf{Exp}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N}$ experiment (cf. Fig. 5).

Let $(vk_i, sigk_i = (sk_i, r_i))$ denote the verification/signing key pair of user $i \in [N]$. In this game, when answering an $\mathcal{O}_{\text{SIGN}}$ query (i, m), the challenger samples $x \leftarrow_{\$} \mathcal{L}_\rho$ with witness w, computes $d \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$, $\tau := H(m)$ and $\pi \leftarrow_{\$} \mathsf{Prove}(\mathsf{crs}, \tau, (x, vk_i, d), (w, sk_i, r_i))$. Then, the challenger returns $\sigma := (x, d, \pi)$ to $\mathcal{A}$ and puts (i, m, σ) to set $\mathcal{Q}_{\text{SIGN}}$. For an $\mathcal{O}_{\text{COR}}$ query i, the challenger returns $sigk_i = (sk_i, r_i)$ to $\mathcal{A}$ and puts i to set $\mathcal{Q}_{\text{COR}}$.

At the end of the game, $\mathcal{A}$ outputs a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$. Let Win denote the event that

$$i^* \notin \mathcal{Q}_{\text{COR}} \wedge (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{SIGN}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1,$$

where $\tau^* := H(m^*)$. By definition, $\mathsf{Adv}^{\mathsf{str\text{-}cma\text{-}c}}_{\mathsf{SIG},\mathcal{A},N}(\lambda) = \Pr_0[\mathsf{Win}]$.

Game G_1: It is the same as G_0, except that, after generating n pairs of verification/signing keys $\{(vk_i, sigk_i = (sk_i, r_i))\}_{i \in [N]}$, the challenger aborts immediately if there are two verification keys collide, i.e., $\exists 1 \le i < j \le N$, s.t. $vk_i = vk_j$.

Claim 1. $\big|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\big| \le \varepsilon_{\mathsf{binding}} + \frac{N(N-1)}{2}/|\mathcal{SK}|$.

Proof. Let VKColl denote the event that $\exists 1 \le i < j \le N$, s.t. $vk_i = vk_j$, and let SKColl denote the event that $\exists 1 \le i < j \le N$, s.t. $sk_i = sk_j$. Clearly, G_0 and G_1 are the same until VKColl occurs, thus

$$\big|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\big| \le \Pr_1[\mathsf{VKColl}] \le \Pr_1[\mathsf{SKColl}] + \Pr_1[\neg\mathsf{SKColl} \wedge \mathsf{VKColl}]. \tag{14}$$

It suffices to bound $\Pr_1[\mathsf{SKColl}]$ and $\Pr_1[\neg\mathsf{SKColl} \wedge \mathsf{VKColl}]$.

- Since sk_i and sk_j are independently and uniformly chosen from $\mathcal{SK}$, by a union bound, we have $\Pr_1[\mathsf{SKColl}] \le \sum_{1 \le i < j \le N} \Pr[sk_i = sk_j] \le \frac{N(N-1)}{2}/|\mathcal{SK}|$.
- Since $vk_i = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_i; r_i)$ and $vk_j = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_j; r_j)$, the event $\neg\mathsf{SKColl} \wedge \mathsf{VKColl}$ means $\exists 1 \le i < j \le N$, s.t. $sk_i \ne sk_j$ but $vk_i = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_i; r_i) = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_j; r_j) = vk_j$. By the $\varepsilon_{\mathsf{binding}}$-statistical binding property of CMT under BSetup (the binding mode), this can happen with probability at most $\varepsilon_{\mathsf{binding}}$. Therefore $\Pr_1[\neg\mathsf{SKColl} \wedge \mathsf{VKColl}] \le \varepsilon_{\mathsf{binding}}$.

Overall, Claim 1 holds by plugging the above two bounds into (14). ▌

Game G_2: It is the same as G_1, except that, at the beginning of the game, the challenger generates crs via $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{SimGen}(\rho')$ instead of $\mathsf{crs} \leftarrow_{\$} \mathsf{CRSGen}(\rho')$. Moreover, when answering $\mathcal{O}_{\text{SIGN}}(i, m)$, the challenger computes π via the Sim algorithm of QANIZK by using the simulation trapdoor $\mathsf{td}_{\mathsf{crs}}$:

- $\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, (x, vk_i, d))$.

Note that the witness w for $x \in \mathcal{L}_\rho$ is no longer needed.

Claim 2. $\big|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda)$.

Proof. Note that when answering $\mathcal{O}_{\mathrm{SIGN}}(i, m)$, (1) x is chosen from $\mathcal{L}_\rho$ with witness w, (2) $vk_i = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_i; r_i)$, and (3) $d \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$, which satisfies $d \in \mathsf{Ball}_{\varepsilon_{\mathsf{prPriv}}}\big(\Lambda_{sk_i}(x)\big)$ by the $(\varepsilon_{\mathsf{prPub}}, \varepsilon_{\mathsf{prPriv}})$-approximate correctness of $\mathsf{prQAHPS}$. Therefore, $(x, vk_i, d) \in \mathcal{L}^{(\mathsf{QANIZK})}_{\rho'}$ with witness (w, sk_i, r_i). Then by the zero-knowledge of QANIZK (cf. Definition 18), the crs generated via SimGen and the π's generated via Sim in G_2 are computationally indistinguishable from the crs generated via CRSGen and the π's generated via Prove in G_1. Consequently, we have $\big|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda)$ and Claim 2 follows. ▮

Game G_3: It is the same as G_2, except that, when answering $\mathcal{O}_{\mathrm{SIGN}}(i, m)$, the challenger also puts $(\tau, (x, vk_i, d), \pi)$ to a set $\mathcal{Q}_{\mathrm{SIM}}$, and for the forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ output by $\mathcal{A}$, the event Win is now defined as

$$i^* \notin \mathcal{Q}_{\mathrm{COR}} \ \wedge\ (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\mathrm{SIGN}} \ \wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1$$
$$\wedge\ (\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}}.$$

Claim 3. $\big|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)$.

Proof. By Bad denote the event that $\mathcal{A}$'s forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ satisfying $\exists\ (i, m, \sigma = (x, d, \pi)) \in \mathcal{Q}_{\mathrm{SIGN}}$, s.t.

$$i^* \notin \mathcal{Q}_{\mathrm{COR}} \ \wedge\ (i^*, m^*, \sigma^* = (x^*, d^*, \pi^*)) \neq (i, m, \sigma = (x, d, \pi)) \in \mathcal{Q}_{\mathrm{SIGN}}$$
$$\wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1 \ \wedge\ (\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = (\tau, (x, vk_i, d), \pi) \in \mathcal{Q}_{\mathrm{SIM}},$$

where $\tau^* := H(m^*)$ and $\tau := H(m)$. Clearly, G_2 and G_3 are the same until Bad occurs, thus $\big|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]\big| \leq \Pr_3[\mathsf{Bad}]$.

To bound $\Pr_3[\mathsf{Bad}]$, we first note that $(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = (\tau, (x, vk_i, d), \pi)$ in Bad implies $(\tau^*, x^*, i^*, d^*, \pi^*) = (\tau, x, i, d, \pi)$, since there are no verification key collisions (due to the game change in G_1). Together with $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*)) \neq (i, m, \sigma = (x, d, \pi))$ in Bad, it follows that $m^* \neq m$ but $\tau^* = H(m^*) = H(m) = \tau$. Therefore, Bad suggests a collision of H. It is straightforward to construct an adversary $\mathcal{B}_2$ so that $\Pr_3[\mathsf{Bad}] \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)$. ($\mathcal{B}_2$ can sample all signing keys itself, simulate G_3 honestly for $\mathcal{A}$, and successfully find a collision as long as Bad happens.)

Overall, $\big|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]\big| \leq \Pr_3[\mathsf{Bad}] \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)$. ▮

Game G_4: It is the same as G_3, except that, at the beginning of the game, the challenger picks $(\rho_0, td_{\rho_0}) \leftarrow_{\$} \mathscr{L}_0$ besides $(\rho, td_\rho) \leftarrow_{\$} \mathscr{L}$, and for all the $\mathcal{O}_{\mathrm{SIGN}}$ queries, the challenger samples $x \leftarrow_{\$} \mathcal{L}_{\rho_0}$ instead of $x \leftarrow_{\$} \mathcal{L}_\rho$. We stress that the challenger still uses ρ to define the QANIZK's gap language parameter $\rho' := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}})$ for which $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{SimGen}(\rho')$ is generated.

Claim 4. $\big|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda)$.

Proof. We introduce an intermediate **Game** $\mathsf{G}_{3.5}$ between G_3 and G_4, where the challenger samples $x \leftarrow_\$ \mathcal{X}$ for all the $\mathcal{O}_{\mathrm{SIGN}}$ queries.

Since witness w for x is not used at all in G_3, $\mathsf{G}_{3.5}$ and G_4 (due to the game change in G_2), we can directly construct two adversaries $\mathcal{B}_3$ and $\mathcal{B}_4$ for solving the multi-fold SMP related to $\mathscr{L}$ and the multi-fold SMP related to $\mathscr{L}_0$ respectively, s.t. $\big|\Pr_3[\mathsf{Win}] - \Pr_{3.5}[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda)$ and $\big|\Pr_{3.5}[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda)$. The full description of $\mathcal{B}_3$ and $\mathcal{B}_4$ can be found in Appendix B.1. ($\mathcal{B}_3$ and $\mathcal{B}_4$ can sample all signing keys themselves, simulate $\mathsf{G}_3/\mathsf{G}_{3.5}/\mathsf{G}_4$ honestly for $\mathcal{A}$ depending on the challenges that $\mathcal{B}_3$ and $\mathcal{B}_4$ receive, and succeed as long as $\mathcal{A}$ behaves differently in these games.) ▮

Game G_5**:** It is the same as G_4, except that, the event Win is now defined as

$$\begin{aligned} & i^* \notin \mathcal{Q}_{\mathrm{COR}} \ \wedge \ (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\mathrm{SIGN}} \ \wedge \ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1 \\ & \ \wedge \ (\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}} \ \wedge \ x^* \in \widetilde{\mathcal{L}}_\rho \ \wedge \ d^* \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}\big(\Lambda_{sk_{i^*}}(x^*)\big) \,. \end{aligned}$$

Claim 5. $\big|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \varepsilon_{\mathsf{binding}}$.

Proof. By Forge denote the event that $\mathcal{A}$'s forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ s.t.

$$\begin{aligned} & i^* \notin \mathcal{Q}_{\mathrm{COR}} \ \wedge \ (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\mathrm{SIGN}} \ \wedge \ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1 \\ & \ \wedge \ (\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}} \ \wedge \ (x^* \notin \widetilde{\mathcal{L}}_\rho \ \vee \ d^* \notin \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}\big(\Lambda_{sk_{i^*}}(x^*)\big)). \end{aligned}$$

G_4 and G_5 are the same unless Forge occurs, so $\big|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\big| \leq \Pr_5[\mathsf{Forge}]$.

Note that by the $\varepsilon_{\mathsf{binding}}$-statistical binding property of CMT under BSetup, $vk_{i^*} = \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_{i^*}; r_{i^*})$ cannot be a commitment of messages in $\widetilde{\mathcal{SK}}$ other than sk_{i^*}, except with probability at most $\varepsilon_{\mathsf{binding}}$. We take this for granted in the following analysis. Therefore, the event $(x^* \notin \widetilde{\mathcal{L}}_\rho \ \vee \ d^* \notin \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}\big(\Lambda_{sk_{i^*}}(x^*)\big))$ in Forge implies $(x^*, vk_{i^*}, d^*) \notin \widetilde{\mathcal{L}}^{(\mathsf{QANIZK})}_{\rho'}$ where $\rho' = (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}})$. Consequently, Forge implies $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1 \wedge (\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}} \ \wedge \ (x^*, vk_{i^*}, d^*) \notin \widetilde{\mathcal{L}}^{(\mathsf{QANIZK})}_{\rho'}$, which directly breaks the USS property of tag-based QANIZK. Formally, we can build an adversary $\mathcal{B}_5$ such that $\Pr_5[\mathsf{Forge}] \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$. $\mathcal{B}_5$ can sample all signing keys itself, simulate G_5 honestly for $\mathcal{A}$ (using its own oracle $\mathcal{O}_{\mathrm{SIM}}$ defined in Fig. 8 to generate simulated proofs π when answering $\mathcal{O}_{\mathrm{SIGN}}$ queries for $\mathcal{A}$), output the $(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*)$ extracted from $\mathcal{A}$'s forgery to its own challenger, and succeed as long as Forge occurs. We also provide a full description of $\mathcal{B}_5$ in Appendix B.2.

By taking the aforementioned $\varepsilon_{\mathsf{binding}}$ into account, we have $\Pr_5[\mathsf{Forge}] \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \varepsilon_{\mathsf{binding}}$. This completes the proof of Claim 5. ▮

Game G_6**:** It is the same as G_5, except that, when answering $\mathcal{O}_{\mathrm{SIGN}}(i, m)$, the challenger computes d via the prPub algorithm of $\mathsf{prQAHPS}$ by using the projection key $\alpha_{\rho_0}(sk_i)$ and a witness w of $x \in \mathcal{L}_{\rho_0}$:

- $d \leftarrow_{\$} \mathsf{prPub}(\alpha_{\rho_0}(sk_i), x, w)$.

Since x is chosen from $\mathcal{L}_{\rho_0} \subseteq \widetilde{\mathcal{L}}_{\rho_0}$ with witness w, by the $\varepsilon_{\mathsf{evalInd}}$-evaluation indistinguishability of $\mathsf{prQAHPS}$ (cf. Definition 7), the $d \leftarrow_{\$} \mathsf{prPub}(\alpha_{\rho_0}(sk_i), x, w)$ in G_6 is statistically close to the $d \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$ in G_5, with statistical distance at most $\varepsilon_{\mathsf{evalInd}}$. By a union bound over all $\mathcal{O}_{\mathrm{SIGN}}$ queries, we have $\big|\Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}]\big| \le Q_s \cdot \varepsilon_{\mathsf{evalInd}}$.

Game G_7: It is the same as G_6, except that, at the beginning of the game, the challenger generates $\mathsf{pp}_{\mathsf{CMT}}$ via $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{HSetup}$ (the hiding mode) instead of $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{BSetup}$ (the binding mode).

By the parameter indistinguishability of the two modes of CMT, G_6 and G_7 are computationally indistinguishable, and it is straightforward to construct an adversary $\mathcal{B}_6$ so that $\big|\Pr_6[\mathsf{Win}] - \Pr_7[\mathsf{Win}]\big| \le \mathsf{Adv}^{\mathsf{para\text{-}ind}}_{\mathsf{CMT},\mathcal{B}_6}(\lambda)$. ($\mathcal{B}_6$ receives $\mathsf{pp}_{\mathsf{CMT}}$ from its own challenger, simulates $\mathsf{G}_6/\mathsf{G}_7$ honestly for $\mathcal{A}$ by using the $\mathsf{pp}_{\mathsf{CMT}}$ it received and by sampling all signing keys itself, and successfully distinguishes the two modes as long as $\mathcal{A}$ behaves differently in G_6 and G_7.)

Finally, we have the following claim regarding $\Pr_7[\mathsf{Win}]$.

Claim 6. $\Pr_7[\mathsf{Win}] \le N \cdot \epsilon^{\varepsilon_{\mathsf{ext}}\text{-}\langle \mathscr{L}_0, \mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS},\mathcal{B}_7} + \varepsilon_{\mathsf{hiding}}$.

Proof. Let i^* denote the user index contained in $\mathcal{A}$'s forgery. In the case that $\mathcal{A}$ corrupts user i^* (i.e., $i^* \in \mathcal{Q}_{\mathrm{COR}}$), Win does not occur, thus the claim trivially holds. Next we prove the claim in the case that $\mathcal{A}$ never corrupts user i^* (i.e., $i^* \notin \mathcal{Q}_{\mathrm{COR}}$). We analyze the information about sk_{i^*} that $\mathcal{A}$ may obtain in G_7.

- Firstly, the verification keys contain $vk_{i^*} = \mathsf{CMT}(\mathsf{pp}_{\mathsf{CMT}}, sk_{i^*}; r_{i^*})$.
  Due to the game change in G_7, $\mathsf{pp}_{\mathsf{CMT}}$ is generated by HSetup. By the $\varepsilon_{\mathsf{hiding}}$-statistical hiding property of CMT under HSetup (the hiding mode), $vk_{i^*} = \mathsf{CMT}(\mathsf{pp}_{\mathsf{CMT}}, sk_{i^*}; r_{i^*})$ is statistically close to a commitment $\mathsf{CMT}(\mathsf{pp}_{\mathsf{CMT}}, \widetilde{sk}; \widetilde{r})$ of any $\widetilde{sk} \in \mathcal{SK}$ with $\widetilde{r} \leftarrow_{\$} \mathcal{R}$. Therefore, $vk_{i^*} = \mathsf{CMT}(\mathsf{pp}_{\mathsf{CMT}}, sk_{i^*}; r_{i^*})$ statistically hides the information about sk_{i^*}.
- Due to the game change in G_6, $\mathcal{O}_{\mathrm{SIGN}}(i^*, m)$ for user i^* uses only $\alpha_{\rho_0}(sk_{i^*})$ instead of the whole sk_{i^*}.
- Since $i^* \notin \mathcal{Q}_{\mathrm{COR}}$, $\mathcal{A}$ never queries $\mathcal{O}_{\mathrm{COR}}(i^*)$.

Overall, the information about sk_{i^*} that $\mathcal{A}$ learns in G_7 is limited in $\alpha_{\rho_0}(sk_{i^*})$.

Then we analyze the probability $\Pr_7[\mathsf{Win}]$. For $\mathcal{A}$'s forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$, Win will not occur unless $x^* \in \widetilde{\mathcal{L}}_\rho \ \wedge \ d^* \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}\big(\Lambda_{sk_{i^*}}(x^*)\big)$. Intuitively, by the $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property of $\mathsf{prQAHPS}$ (cf. Definition 11), we know that $x^* \in \widetilde{\mathcal{L}}_\rho \ \wedge \ d^* \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}\big(\Lambda_{sk_{i^*}}(x^*)\big)$ holds with only a negligible probability, even in the presence of $\alpha_{\rho_0}(sk_{i^*})$. Hence Win hardly happens in G_7.

Formally, we build an (unbounded) adversary $\mathcal{B}_7$ against the $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property of $\mathsf{prQAHPS}$. $\mathcal{B}_7$ is given $(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \rho, \alpha_{\rho_0}(sk))$, where $sk \leftarrow_{\$} \mathcal{SK}$ is chosen by its own challenger. $\mathcal{B}_7$ will simulate G_7 for $\mathcal{A}$. Firstly,

$\mathcal{B}_7$ guesses the user index i^* for which $\mathcal{A}$ forges a signature (with a security loss N) and implicitly sets the signing key of user i^* as the sk chosen by its own challenger. $\mathcal{B}_7$ samples $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{HSetup}$ and computes the verification key of user i^* as $vk_{i^*} := \mathsf{CMT}(\mathsf{pp}_{\mathsf{CMT}}, \widetilde{sk}; \widetilde{r})$ for an arbitrary $\widetilde{sk} \in \mathcal{SK}$, where $\widetilde{r} \leftarrow_{\$} \mathcal{R}$. By the $\varepsilon_{\mathsf{hiding}}$-statistical hiding property of CMT under HSetup (the hiding mode), this simulation is statistically close to G_7, with statistical distance at most $\varepsilon_{\mathsf{hiding}}$. For the remaining $N-1$ users, $\mathcal{B}_7$ samples signing keys itself, thus can honestly answer $\mathcal{O}_{\text{SIGN}}$ and $\mathcal{O}_{\text{COR}}$ queries made by $\mathcal{A}$ for these users. For user i^*, $\mathcal{B}_7$ can answer $\mathcal{O}_{\text{SIGN}}$ queries using the projection key $\alpha_{\rho_0}(sk)$ contained in its own input (since $x \leftarrow_{\$} \mathcal{L}_{\rho_0}$) and abort immediately if $\mathcal{A}$ corrupts i^*. Finally, $\mathcal{B}_7$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$, and returns (x^*, d^*) to its own challenger. Overall, $\mathcal{B}_7$'s simulation is statistically close to G_7 and $\mathcal{B}_7$ succeeds (i.e., $x^* \in \widetilde{\mathcal{L}}_\rho \ \wedge \ d^* \in \mathsf{Ball}_{\varepsilon_{\mathsf{ext}}}\big(\Lambda_{sk_{i^*}}(x^*)\big)$) as long as i^* is correctly guessed and Win occurs, thus $\epsilon^{\varepsilon_{\mathsf{ext}}\text{-}\langle \mathscr{L}_0, \mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS}, \mathcal{B}_7} \geq \frac{1}{N} \cdot \Pr[\mathsf{Win}\text{ occurs in }\mathcal{B}_7\text{'s simulation}] \geq \frac{1}{N} \cdot \big(\Pr_7[\mathsf{Win}] - \varepsilon_{\mathsf{hiding}}\big)$. We also provide a full description of $\mathcal{B}_7$ in Appendix B.3. ▌

Taking all things together, Theorem 1 follows. □

B.1 Full Description of Reductions $\mathcal{B}_3$ and $\mathcal{B}_4$ for Claim 4

We introduce an intermediate game $\mathsf{G}_{3.5}$ between G_3 and G_4:

- **Game $\mathsf{G}_{3.5}$:** It is the same as game G_3, except that, for all the $\mathcal{O}_{\text{SIGN}}$ queries, the challenger samples $x \leftarrow_{\$} \mathcal{X}$.

Note that the witness w for x is not used at all in games G_3, $\mathsf{G}_{3.5}$ and G_4 (due to the game change in G_2).

Below we construct two adversaries $\mathcal{B}_3$ and $\mathcal{B}_4$ for solving the multi-fold SMP related to $\mathscr{L}$ and the multi-fold SMP related to $\mathscr{L}_0$ respectively, s.t. $\big|\Pr_3[\mathsf{Win}] - \Pr_{3.5}[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}, \mathcal{B}_3, Q_s}(\lambda)$ and $\big|\Pr_{3.5}[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0, \mathcal{B}_4, Q_s}(\lambda)$.

We first provide the full description of $\mathcal{B}_3$ for solving the multi-fold SMP related to $\mathscr{L}$ (cf. Definition 3). $\mathcal{B}_3$ is given $(\rho, \{x_j\}_{j \in [Q_s]})$, where $(\rho, td_\rho) \leftarrow_{\$} \mathscr{L}$, and $\mathcal{B}_3$ aims to decide whether $x_1, ..., x_{Q_s} \leftarrow_{\$} \mathcal{L}_\rho$ or $x_1, ..., x_{Q_s} \leftarrow_{\$} \mathcal{X}$. $\mathcal{B}_3$ will simulate G_3 or $\mathsf{G}_{3.5}$ for $\mathcal{A}$, depending on the input that $\mathcal{B}_3$ receives.

- Firstly, $\mathcal{B}_3$ invokes $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$, $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{BSetup}$, and sets $\rho' := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}})$ which defines the gap language $\mathcal{GL}^{(\mathsf{QANIZK})}_{\rho'}$ as in Fig. 1. Then $\mathcal{B}_3$ invokes $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{SimGen}(\rho')$, samples $H \leftarrow_{\$} \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{SIG}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}}, \mathsf{crs}, H)$. Then for each user $i \in [N]$, $\mathcal{B}_3$ sets the signing key $sigk_i := (sk_i, r_i)$ itself with $sk_i \leftarrow_{\$} \mathcal{SK}$ and $r_i \leftarrow_{\$} \mathcal{R}$, and computes the corresponding $vk_i := \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_i; r_i)$. $\mathcal{B}_3$ sends $(\mathsf{pp}_{\mathsf{SIG}}, \{vk_i\}_{i \in [N]})$ to $\mathcal{A}$.
- For $\mathcal{O}_{\text{SIGN}}$ queries, when answering the j-th ($j \in [Q_s]$) $\mathcal{O}_{\text{SIGN}}$ query (i, m), $\mathcal{B}_3$ sets x as the x_j in its own input, and computes $d \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$, $\tau := H(m)$ and $\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, (x, vk_i, d))$, without knowing a witness of x. $\mathcal{B}_3$ returns $\sigma := (x, d, \pi)$ to $\mathcal{A}$, puts (i, m, σ) to $\mathcal{Q}_{\text{SIGN}}$ and puts $(\tau, (x, vk_i, d), \pi)$ to $\mathcal{Q}_{\text{SIM}}$.

In the case that $x = x_j$ is uniformly chosen from $\mathcal{L}_\rho$, $\mathcal{B}_3$ perfectly simulates G_3 for $\mathcal{A}$; in the case that $x = x_j$ is uniformly chosen from $\mathcal{X}$, $\mathcal{B}_3$ perfectly simulates $\mathsf{G}_{3.5}$ for $\mathcal{A}$.

- For an $\mathcal{O}_{\text{COR}}$ query i, $\mathcal{B}_3$ returns $sigk_i = (sk_i, r_i)$ to $\mathcal{A}$ and puts i to $\mathcal{Q}_{\text{COR}}$, the same way as G_3 and $\mathsf{G}_{3.5}$.
- Finally, $\mathcal{B}_3$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$. $\mathcal{B}_3$ uses the signing keys $\{sigk_i\}_{i\in[N]}$ to decide whether the event Win defined in G_3 (which is the same as that defined in $\mathsf{G}_{3.5}$ and G_4) occurs, i.e.,

$$i^* \notin \mathcal{Q}_{\text{COR}} \ \wedge \ (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{SIGN}} \ \wedge \ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1$$
$$\wedge \ (\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\text{SIM}}.$$

$\mathcal{B}_3$ returns 1 to its own challenger if and only if Win occurs.

Overall, $\mathcal{B}_3$ simulates G_3 for $\mathcal{A}$ in the case $x_1, ..., x_{Q_s} \leftarrow_{\$} \mathcal{L}_\rho$ and simulates $\mathsf{G}_{3.5}$ for $\mathcal{A}$ in the case $x_1, ..., x_{Q_s} \leftarrow_{\$} \mathcal{X}$, thus $\mathcal{B}_3$ successfully distinguishes the two cases as long as the probability that Win occurs in G_3 differs non-negligibly from that in $\mathsf{G}_{3.5}$. Consequently, we have $\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda) \geq \big|\Pr_3[\mathsf{Win}] - \Pr_{3.5}[\mathsf{Win}]\big|$.

Next, we provide the description of $\mathcal{B}_4$ for solving the multi-fold SMP related to $\mathscr{L}_0$ (cf. Definition 3). $\mathcal{B}_4$ is given $(\rho_0, \{x_j\}_{j\in[Q_s]})$, where $(\rho_0, td_{\rho_0}) \leftarrow_{\$} \mathscr{L}_0$, and $\mathcal{B}_4$ aims to decide whether $x_1, ..., x_{Q_s} \leftarrow_{\$} \mathcal{L}_{\rho_0}$ or $x_1, ..., x_{Q_s} \leftarrow_{\$} \mathcal{X}$. $\mathcal{B}_4$ simulates exactly the same way as $\mathcal{B}_3$ does, except that, $\mathcal{B}_4$ samples $(\rho, td_\rho) \leftarrow_{\$} \mathscr{L}$ itself to generate the ρ contained in $\mathsf{pp}_{\mathsf{SIG}}$. In particular, when answering the j-th $(j \in [Q_s])$ $\mathcal{O}_{\text{SIGN}}$ query (i, m) made by $\mathcal{A}$, $\mathcal{B}_4$ sets x as the x_j in its own input. In the case that $x = x_j$ is uniformly chosen from $\mathcal{L}_{\rho_0}$, $\mathcal{B}_4$ perfectly simulates G_4 for $\mathcal{A}$; in the case that $x = x_j$ is uniformly chosen from $\mathcal{X}$, $\mathcal{B}_4$ perfectly simulates $\mathsf{G}_{3.5}$ for $\mathcal{A}$. Therefore, $\mathcal{B}_4$ successfully distinguishes $x_1, ..., x_{Q_s} \leftarrow_{\$} \mathcal{L}_{\rho_0}$ from $x_1, ..., x_{Q_s} \leftarrow_{\$} \mathcal{X}$ as long as the probability that Win occurs in G_4 differs non-negligibly from that in $\mathsf{G}_{3.5}$. Consequently, we have $\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda) \geq \big|\Pr_{3.5}[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\big|$.

This completes the proof of Claim 4. ▌

B.2 Full Description of Reduction $\mathcal{B}_5$ for Claim 5

To bound $\Pr_5[\mathsf{Forge}]$, we construct an adversary $\mathcal{B}_5$ against the USS of tag-based QANIZK (cf. Definition 19) for the gap language $\mathcal{GL}^{(\mathsf{QANIZK})}_{\rho'} = (\mathcal{L}^{(\mathsf{QANIZK})}_{\rho'}, \widetilde{\mathcal{L}}^{(\mathsf{QANIZK})}_{\rho'})$ defined in Fig. 1, where $\rho' = (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}})$. The full description of $\mathcal{B}_5$ is as follows. $\mathcal{B}_5$ is given $(\rho' = (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}}), td_{\rho'}, \mathsf{crs})$ and has access to the oracle $\mathcal{O}_{\text{SIM}}$ defined in Fig. 8 (cf. Definition 19). $\mathcal{B}_5$ simulates G_5 for $\mathcal{A}$ as follows.

- Firstly, $\mathcal{B}_5$ samples $H \leftarrow_{\$} \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{SIG}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}}, \mathsf{crs}, H)$. $\mathcal{B}_5$ also invokes $(\rho_0, td_{\rho_0}) \leftarrow_{\$} \mathscr{L}_0$. Then for each user $i \in [N]$, $\mathcal{B}_5$ samples $sk_i \leftarrow_{\$} \mathcal{SK}$ and $r_i \leftarrow_{\$} \mathcal{R}$ itself, sets $sigk_i := (sk_i, r_i)$, and computes the corresponding $vk_i := \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_i; r_i)$. $\mathcal{B}_5$ sends $(\mathsf{pp}_{\mathsf{SIG}}, \{vk_i\}_{i\in[N]})$ to $\mathcal{A}$.
- For an $\mathcal{O}_{\text{SIGN}}$ query (i, m) made by $\mathcal{A}$, $\mathcal{B}_5$ samples $x \leftarrow_{\$} \mathcal{L}_{\rho_0}$, computes $d \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$ and $\tau := H(m)$. Then $\mathcal{B}_5$ sends $(\tau, (x, vk_i, d))$ to its own

$\mathcal{O}_{\text{SIM}}$ oracle and obtains π, which is generated by $\mathcal{O}_{\text{SIM}}$ via $\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, (x, vk_i, d))$. $\mathcal{B}_5$ returns $\sigma := (x, d, \pi)$ to $\mathcal{A}$, puts (i, m, σ) to $\mathcal{Q}_{\text{SIGN}}$ and puts $(\tau, (x, vk_i, d), \pi)$ to $\mathcal{Q}_{\text{SIM}}$.
- For an $\mathcal{O}_{\text{COR}}$ query i, $\mathcal{B}_5$ returns $sigk_i = (sk_i, r_i)$ to $\mathcal{A}$ and puts i to $\mathcal{Q}_{\text{COR}}$.
- Finally, $\mathcal{B}_5$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$. $\mathcal{B}_5$ computes $\tau^* := H(m^*)$, and outputs $(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*)$ to its own challenger.

It is clear to see that $\mathcal{B}_5$ simulates G_5 perfectly for $\mathcal{A}$, and $\mathcal{B}_5$ outputs a successful forgery $(\tau^*, (x^*, vk_{i^*}, d^*), \pi^*)$ to its own challenger so that $(x^*, vk_{i^*}, d^*) \notin \widehat{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})} \wedge (\tau^*, (x^*, vk_{i^*}, d^*), \pi^*) \notin \mathcal{Q}_{\text{SIM}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, (x^*, vk_{i^*}, d^*), \pi^*) = 1$ as long as Forge occurs. By taking the aforementioned statistical binding parameter $\varepsilon_{\mathsf{binding}}$ into account, we have $\Pr_5[\mathsf{Forge}] \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \varepsilon_{\mathsf{binding}}$. This completes the proof of Claim 5. ▌

B.3 Full Description of Reduction $\mathcal{B}_7$ for Claim 6

To bound $\Pr_7[\mathsf{Win}]$, we construct an (unbounded) adversary $\mathcal{B}_7$ against the $\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting property of $\mathsf{prQAHPS}$ (cf. Definition 11). The full description of $\mathcal{B}_7$ is as follows. $\mathcal{B}_7$ is given $(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \rho, \alpha_{\rho_0}(sk))$, where $sk \leftarrow_{\$} \mathcal{SK}$ is chosen by its own challenger. $\mathcal{B}_7$ simulates G_7 for $\mathcal{A}$ as follows.

- Firstly, $\mathcal{B}_7$ invokes $\mathsf{pp}_{\mathsf{CMT}} \leftarrow_{\$} \mathsf{HSetup}$, and sets $\rho' := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}})$ which defines the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})}$ as in Fig. 1. Then $\mathcal{B}_7$ invokes $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{SimGen}(\rho')$, $H \leftarrow_{\$} \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{SIG}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{CMT}}, \mathsf{crs}, H)$.
  $\mathcal{B}_7$ samples an index $\widehat{i} \leftarrow_{\$} [N]$ uniformly, sets $sk_{\widehat{i}} := sk$ implicitly for user $\widehat{i}$, where sk is the hashing key chosen by $\mathcal{B}_7$'s own challenger, and computes the verification key of user $\widehat{i}$ as $vk_{\widehat{i}} := \mathsf{CMT}(\mathsf{pp}_{\mathsf{CMT}}, \widetilde{sk}; \widetilde{r})$ for an arbitrary $\widetilde{sk} \in \mathcal{SK}$, with $\widetilde{r} \leftarrow_{\$} \mathcal{R}$. By the $\varepsilon_{\mathsf{hiding}}$-statistical hiding property of CMT under HSetup (the hiding mode), this simulation is statistically close to G_7, with statistical distance at most $\varepsilon_{\mathsf{hiding}}$.
  For all other users $i \in [N] \setminus \{\widehat{i}\}$, $\mathcal{B}_7$ samples $sk_i \leftarrow_{\$} \mathcal{SK}$ and $r_i \leftarrow_{\$} \mathcal{R}$ itself, sets $sigk_i := (sk_i, r_i)$, and computes $vk_i := \mathsf{Com}(\mathsf{pp}_{\mathsf{CMT}}, sk_i; r_i)$.
  $\mathcal{B}_7$ sends $(\mathsf{pp}_{\mathsf{SIG}}, \{vk_i\}_{i \in [N]})$ to $\mathcal{A}$.
- For an $\mathcal{O}_{\text{SIGN}}$ query (i, m) made by $\mathcal{A}$, $\mathcal{B}_7$ computes a signature σ as follows.
  $\mathcal{B}_7$ first samples $x \leftarrow_{\$} \mathcal{L}_{\rho_0}$ with witness w. If $i \neq \widehat{i}$, $\mathcal{B}_7$ computes d via $d \leftarrow_{\$} \mathsf{prPub}(\alpha_{\rho_0}(sk_i), x, w)$ using $\alpha_{\rho_0}(sk_i)$, the same as G_7; if $i = \widehat{i}$, $\mathcal{B}_7$ computes d via $d \leftarrow_{\$} \mathsf{prPub}(\alpha_{\rho_0}(sk), x, w)$ using the projection key $\alpha_{\rho_0}(sk)$ contained in its own input, which is also the same as G_7. Then, $\mathcal{B}_7$ computes $\tau := H(m)$, invokes $\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, (x, vk_i, d))$ and sets $\sigma := (x, d, \pi)$.
  $\mathcal{B}_7$ returns σ to $\mathcal{A}$, puts (i, m, σ) to $\mathcal{Q}_{\text{SIGN}}$ and $(\tau, (x, vk_i, d), \pi)$ to $\mathcal{Q}_{\text{SIM}}$.
- For an $\mathcal{O}_{\text{COR}}$ query i made by $\mathcal{A}$, if $i \neq \widehat{i}$, $\mathcal{B}_7$ returns $sigk_i = (sk_i, r_i)$ to $\mathcal{A}$; if $i = \widehat{i}$, $\mathcal{B}_7$ aborts immediately.
- Finally, $\mathcal{B}_7$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$. If $i^* = \widehat{i}$, $\mathcal{B}_7$ outputs (x^*, d^*) to its own challenger; if $i^* \neq \widehat{i}$, $\mathcal{B}_7$ aborts the game.

It is clear to see that if $\widehat{i} = i^*$ (which happens with probability $\frac{1}{N}$) and $\mathcal{A}$ never corrupts i^*, $\mathcal{B}_7$'s simulation is statistically close to G_7, and $\mathcal{B}_7$'s output (x^*, d^*) succeeds (i.e., $x^* \in \widetilde{\mathcal{L}}_\rho \ \wedge \ d^* \in \mathsf{Ball}_{\boldsymbol{\varepsilon}_{\mathsf{ext}}}\big(\Lambda_{sk}(x^*)\big)$) as long as Win occurs. Thus, $\boldsymbol{\epsilon}^{\boldsymbol{\varepsilon}_{\mathsf{ext}}\text{-}\langle \mathscr{L}_0, \mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS}, \mathcal{B}_7} \geq \frac{1}{N} \cdot \Pr[\mathsf{Win} \text{ occurs in } \mathcal{B}_7\text{'s simulation}] \geq \frac{1}{N} \cdot \big(\Pr_7[\mathsf{Win}] - \boldsymbol{\varepsilon}_{\mathsf{hiding}}\big)$ and Claim 6 follows. ▌

C Proof of Theorem 2 (MUMC$^{\mathsf{c}}$-CCA Security of PKE)

Theorem 2 (MUMC$^{\mathsf{c}}$-CCA **Security of** PKE) *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii)* prQAHPS *is a probabilistic QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having $\boldsymbol{\varepsilon}_{\mathsf{evalnd}}$-evaluation indistinguishability, PK-diversity, and supporting both $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching and $\mathscr{L}_0$-multi-key-multi-extracting, (iii)* QANIZK *is a tag-based QA-NIZK for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ generated by $\mathscr{L}$, satisfying both zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed* PKE *scheme in Fig. 2 is* MUMC$^{\mathsf{c}}$-CCA *secure.*

Concretely, for any number N of users and any adversary $\mathcal{A}$ who makes at most Q_e times of $\mathcal{O}_{\mathrm{ENC}}$ queries and Q_d times of $\mathcal{O}_{\mathrm{DEC}}$ queries, there exist adversaries $\mathcal{B}_1, \cdots, \mathcal{B}_7$, such that $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_6) \approx \mathbf{T}(\mathcal{A}) + (N + Q_e + Q_d) \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N}(\lambda) \leq \ & \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda) \\ & + \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{prQAHPS},\mathcal{B}_6,N,Q_e}(\lambda) + \textit{statist. loss},\end{aligned}$$

where statist. loss $= \frac{N(N-1)}{2} \cdot \boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{prQAHPS}} + (3Q_e + 2Q_d) \cdot \boldsymbol{\varepsilon}_{\mathsf{evalnd}} + N \cdot \boldsymbol{\epsilon}^{\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{prQAHPS},\mathcal{B}_7}$.

Proof of Theorem 2. We prove Theorem 2 by defining a sequence of games G_0–G_9 and showing adjacent games indistinguishable. A brief description of differences between adjacent games is summarized in Table 2. By $\Pr_i[\cdot]$ we denote the probability of a particular event occurring in game G_i.

Game G_0: This is the $\mathsf{Exp}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N}$ experiment (cf. Fig. 6). Let Win denote the event that $\beta' = \beta$. By definition, $\mathsf{Adv}^{\mathsf{cca\text{-}c}}_{\mathsf{PKE},\mathcal{A},N}(\lambda) = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

Let (pk_i, sk_i) denote the public/secret key pair of user $i \in [N]$. In this game, when answering an $\mathcal{O}_{\mathrm{ENC}}$ query (i^*, m_0, m_1), the challenger samples $x^* \leftarrow_{\$} \mathcal{L}_\rho$ with witness w^*, computes $hv^* \leftarrow_{\$} \mathsf{prPub}(pk_{i^*}, x^*, w^*)$, $d^* := hv^* + \mathsf{Encode}(m_\beta)$, $\tau^* := H(pk_{i^*}, d^*)$ and $\pi^* \leftarrow_{\$} \mathsf{Prove}(\mathsf{crs}, \tau^*, x^*, w^*)$. Then, the challenger returns the challenge ciphertext $c^* := (x^*, d^*, \pi^*)$ to $\mathcal{A}$ and puts (i^*, c^*) to set $\mathcal{Q}_{\mathrm{ENC}}$. Upon an $\mathcal{O}_{\mathrm{DEC}}$ query $(i, c = (x, d, \pi))$, the challenger computes $\tau := H(pk_i, d)$, $hv' \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$, returns $m := \mathsf{Decode}(d - hv')$ to $\mathcal{A}$ if $(i, c) \notin \mathcal{Q}_{\mathrm{ENC}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$ holds, and returns $\perp$ otherwise. For an $\mathcal{O}_{\mathrm{COR}}$ query i, the challenger returns sk_i to $\mathcal{A}$ and puts i to set $\mathcal{Q}_{\mathrm{COR}}$.

Game G_1: It is the same as G_0, except that, the challenger aborts immediately if there are collisions in $\{pk_i\}_{i \in [N]}$, i.e., $\exists 1 \leq i < j \leq N$, s.t. $pk_i = pk_j$.

Since sk_i and sk_j are chosen independently from $\mathcal{SK}$ for each $1 \leq i < j \leq N$, by a union bound and by the PK-diversity of $\mathsf{prQAHPS}$, it follows that $\big|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\big| \leq \sum_{1 \leq i < j \leq N} \Pr[\alpha_\rho(sk_i) = \alpha_\rho(sk_j)] \leq \frac{N(N-1)}{2} \cdot \epsilon^{\mathsf{pk\text{-}div}}_{\mathsf{prQAHPS}}$.

Game G_2: It is the same as G_1, except that, at the beginning of the game, the challenger generates crs via $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_\$ \mathsf{SimGen}(\rho)$ instead of $\mathsf{crs} \leftarrow_\$ \mathsf{CRSGen}(\rho)$. Moreover, when answering $\mathcal{O}_{\mathrm{ENC}}(i^*, m_0, m_1)$, the challenger computes hv^* and π^* without using the witness w^* for $x^* \in \mathcal{L}_\rho$:

- $hv^* \leftarrow_\$ \mathsf{prPriv}(sk_{i^*}, x^*)$,
- $\pi^* \leftarrow_\$ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*, x^*)$.

Claim 7. $\big|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda) + Q_e \cdot \varepsilon_{\mathsf{evalInd}}$.

Proof. Since x^* is chosen from $\mathcal{L}_\rho$ with witness w^*, by the zero-knowledge of QANIZK (cf. Definition 18), the crs generated via SimGen and the π^*'s generated via Sim in G_2 are computationally indistinguishable from the crs generated via CRSGen and the π^*'s generated via Prove in G_1, and more precisely, $\mathcal{A}$ can distinguish them with probability at most $\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda)$.

Moreover, by the $\varepsilon_{\mathsf{evalInd}}$-evaluation indistinguishability of $\mathsf{prQAHPS}$ (cf. Definition 7), the $hv^* \leftarrow_\$ \mathsf{prPriv}(sk_{i^*}, x^*)$ in G_2 is statistically close to the $hv^* \leftarrow_\$ \mathsf{prPub}(pk_{i^*}, x^*, w^*)$ in G_1, with statistical distance at most $\varepsilon_{\mathsf{evalInd}}$. Then by a union bound over all $\mathcal{O}_{\mathrm{ENC}}$ queries, all hv^*'s generated via prPriv in G_2 are statistically indistinguishable from the hv^*'s generated via prPub in G_1, with statistical distance at most $Q_e \cdot \varepsilon_{\mathsf{evalInd}}$.

Overall, we have $\big|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{B}_1}(\lambda) + Q_e \cdot \varepsilon_{\mathsf{evalInd}}$. ▮

Game G_3: It is the same as G_2, except that, when answering $\mathcal{O}_{\mathrm{ENC}}(i^*, m_0, m_1)$, the challenger also puts (τ^*, x^*, π^*) to a set $\mathcal{Q}_{\mathrm{SIM}}$, and when answering $\mathcal{O}_{\mathrm{DEC}}(i, c = (x, d, \pi))$, the challenger adds the following new rejection rule:

- If $(\tau, x, \pi) \in \mathcal{Q}_{\mathrm{SIM}}$, return $\perp$ directly.

Clearly, G_2 and G_3 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\mathrm{DEC}}(i, c = (x, d, \pi))$ s.t.

$$\exists\, (i^*, c^* = (x^*, d^*, \pi^*)) \in \mathcal{Q}_{\mathrm{ENC}},\ \text{s.t.}\ (i, c = (x, d, \pi)) \neq (i^*, c^* = (x^*, d^*, \pi^*)) \\ \wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1\ \wedge\ (\tau, x, \pi) = (\tau^*, x^*, \pi^*) \in \mathcal{Q}_{\mathrm{SIM}},$$

where $\tau := H(pk_i, d)$ and $\tau^* := H(pk_{i^*}, d^*)$.

Note that by $(i, c = (x, d, \pi)) \neq (i^*, c^* = (x^*, d^*, \pi^*))$ and $(\tau, x, \pi) = (\tau^*, x^*, \pi^*)$, it follows that $(i, d) \neq (i^*, d^*)$ and $\tau = H(pk_i, d) = H(pk_{i^*}, d^*) = \tau^*$. Since there are no public key collisions (due to the game change in G_1), $(i, d) \neq (i^*, d^*)$ implies $(pk_i, d) \neq (pk_{i^*}, d^*)$. Consequently, the above event suggests a collision of H, and we have $\big|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)$.

Game G_4: It is the same as G_3, except that, at the beginning of the game, the challenger picks $(\rho_0, td_{\rho_0}) \leftarrow_\$ \mathscr{L}_0$ besides $(\rho, td_\rho) \leftarrow_\$ \mathscr{L}$, and for all the $\mathcal{O}_{\mathrm{ENC}}$ queries, the challenger samples $x^* \leftarrow_\$ \mathcal{L}_{\rho_0}$ instead of $x^* \leftarrow_\$ \mathcal{L}_\rho$.

By the multi-fold SMP related to $\mathscr{L}$ and by the multi-fold SMP related to $\mathscr{L}_0$, we can first change G_3 to an intermediate game $\mathsf{G}_{3.5}$ where the challenger samples $x^* \leftarrow_\$ \mathcal{X}$ for all the $\mathcal{O}_{\rm ENC}$ queries, then further change $\mathsf{G}_{3.5}$ to G_4. Overall, we have the following claim.

Claim 8. $\big|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda)$.

The proof is similar to that for Claim 4 in the proof of Theorem 1, thus we omit it.

Game G_5: It is the same as G_4, except that, when answering $\mathcal{O}_{\rm DEC}(i, c = (x, d, \pi))$, the challenger adds another new rejection rule:

- If $x \notin \widetilde{\mathcal{L}}_\rho$, return $\perp$ directly.

Note that the challenger can use the trapdoor td_ρ to check $x \notin \widetilde{\mathcal{L}}_\rho$ efficiently.

Clearly, G_4 and G_5 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\rm DEC}(i, c = (\,x, d, \pi\,)\,)$ s.t.

$$(i, c = (\,x, d, \pi\,)) \notin \mathcal{Q}_{\rm ENC} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\rm SIM} \wedge x \notin \widetilde{\mathcal{L}}_\rho.$$

This event implies $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\rm SIM} \wedge x \notin \widetilde{\mathcal{L}}_\rho$. Thus by the USS of QANIZK, we have the following claim.

Claim 9. $\big|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$.

We provide a formal proof for Claim 9 in Appendix C.1. A subtlety is that $\mathcal{B}_5$ obtains the language trapdoor td_ρ from its own challenger, thus can use td_ρ to efficiently decide the membership of $\widetilde{\mathcal{L}}_\rho$ when answering $\mathcal{O}_{\rm DEC}$ queries for $\mathcal{A}$.

Game G_6: It is the same as G_5, except that, when answering $\mathcal{O}_{\rm ENC}(i^*, m_0, m_1)$, the challenger computes hv^* via the prPub algorithm of $\mathsf{prQAHPS}$ by using the projection key $\alpha_{\rho_0}(sk_{i^*})$ and a witness w^* of $x^* \in \mathcal{L}_{\rho_0}$:

- $hv^* \leftarrow_\$ \mathsf{prPub}(\alpha_{\rho_0}(sk_{i^*}), x^*, w^*)$.

Moreover, when answering $\mathcal{O}_{\rm DEC}(i, c = (x, d, \pi))$, the challenger computes $\tau := H(pk_i, d)$, checks whether $(i, c) \notin \mathcal{Q}_{\rm ENC} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\rm SIM} \wedge x \in \widetilde{\mathcal{L}}_\rho$ holds, and returns $\perp$ to $\mathcal{A}$ directly if the check fails. If the check passes, the challenger uses brute force to find a witness w for $x \in \widetilde{\mathcal{L}}_\rho$, and computes hv' via the prPub algorithm by using the projection key $\alpha_\rho(sk_i)$:

- $hv' \leftarrow_\$ \mathsf{prPub}(\alpha_\rho(sk_i), x, w)$,

and returns $m := \mathsf{Decode}(d - hv')$ to $\mathcal{A}$.

We note that the challenger in this game may not be PPT. This does not matter, since the following arguments (before the challenger is changed back to PPT) are statistical.

Below we show that G_6 is statistically close to G_5. For $\mathcal{O}_{\rm ENC}$ queries, since w^* is a witness for $x^* \in \mathcal{L}_{\rho_0} \subseteq \widetilde{\mathcal{L}}_{\rho_0}$, by the $\varepsilon_{\mathsf{evalnd}}$-evaluation indistinguishability of $\mathsf{prQAHPS}$ (cf. Definition 7), the $hv^* \leftarrow_\$ \mathsf{prPub}(\alpha_{\rho_0}(sk_{i^*}), x^*, w^*)$ in G_6 is statistically close to the $hv^* \leftarrow_\$ \mathsf{prPriv}(sk_{i^*}, x^*)$ in G_5, with statistical distance at most $\varepsilon_{\mathsf{evalInd}}$. Similarly, for $\mathcal{O}_{\mathrm{DEC}}$ queries, since w is a witness for $x \in \widetilde{\mathcal{L}}_\rho$, the $hv' \leftarrow_{\$} \mathsf{prPub}(\alpha_\rho(sk_i), x, w)$ in G_6 is statistically close to the $hv' \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$ in G_5, with statistical distance at most $\varepsilon_{\mathsf{evalInd}}$. By a union bound over all $\mathcal{O}_{\mathrm{ENC}}$ queries and all $\mathcal{O}_{\mathrm{DEC}}$ queries, we have $\big|\Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}]\big| \le (Q_e + Q_d) \cdot \varepsilon_{\mathsf{evalInd}}$.

Game $\mathsf{G}_{7.\eta}$, $0 \le \eta \le N$: It is the same as G_6, except that, at the beginning of the game, the challenger picks another $sk'_i \leftarrow_{\$} \mathcal{SK}$ besides sk_i for each user $i \in [N]$. Moreover, when answering $\mathcal{O}_{\mathrm{ENC}}(i^*, m_0, m_1)$ for users $i^* \le \eta$, the challenger switches sk_{i^*} to the new secret key sk'_{i^*} in computing hv^*:

- $hv^* \leftarrow_{\$} \mathsf{prPub}(\alpha_{\rho_0}(sk'_{i^*}), x^*, w^*)$,

where w^* is a witness of $x^* \in \mathcal{L}_{\rho_0}$. The challenger still uses $\{sk_i\}_{i\in[N]}$ to compute the public keys for all users $i \in [N]$, to answer $\mathcal{O}_{\mathrm{ENC}}$ queries for users $i^* > \eta$, and to answer $\mathcal{O}_{\mathrm{DEC}}$ and $\mathcal{O}_{\mathrm{COR}}$ queries for all users $i \in [N]$.

It is clearly that $\mathsf{G}_{7.0}$ is identical to G_6, thus $\Pr_6[\mathsf{Win}] = \Pr_{7.0}[\mathsf{Win}]$.

For each $\eta \in [N]$, note that the difference between $\mathsf{G}_{7.\eta-1}$ and $\mathsf{G}_{7.\eta}$ lies in the $\mathcal{O}_{\mathrm{ENC}}$ oracle for user η: in $\mathsf{G}_{7.\eta-1}$, $\mathcal{O}_{\mathrm{ENC}}$ computes $hv^* \leftarrow_{\$} \mathsf{prPub}(\alpha_{\rho_0}(sk_\eta), x^*, w^*)$ using sk_η, while in $\mathsf{G}_{7.\eta}$, $\mathcal{O}_{\mathrm{ENC}}$ computes $hv^* \leftarrow_{\$} \mathsf{prPub}(\alpha_{\rho_0}(sk'_\eta), x^*, w^*)$ using sk'_η. By the $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching property of $\mathsf{prQAHPS}$ (cf. Definition 8), the challenger can safely switch sk_η to sk'_η when answering $\mathcal{O}_{\mathrm{ENC}}$ for user η, and we have the following claim.

Claim 10. For each $\eta \in [N]$, $|\Pr_{7.\eta-1}[\mathsf{Win}] - \Pr_{7.\eta}[\mathsf{Win}]| \le \epsilon^{\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{prQAHPS}, \mathcal{B}_7}$.

We provide a formal proof for Claim 10 in Appendix C.2.

Game G_8: It is the same as $\mathsf{G}_{7.N}$, except that, when answering $\mathcal{O}_{\mathrm{ENC}}(i^*, m_0, m_1)$, the challenger computes hv^* via the prPriv algorithm of $\mathsf{prQAHPS}$ by using sk'_{i^*}, without using a witness w^* of $x^* \in \mathcal{L}_{\rho_0}$:

- $hv^* \leftarrow_{\$} \mathsf{prPriv}(sk'_{i^*}, x^*)$.

Moreover, when answering $\mathcal{O}_{\mathrm{DEC}}(i, c = (x, d, \pi))$, the challenger computes $\tau := H(pk_i, d)$, checks whether $(i, c) \notin \mathcal{Q}_{\mathrm{ENC}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\mathrm{SIM}} \wedge x \in \widetilde{\mathcal{L}}_\rho$ holds, and returns $\perp$ to $\mathcal{A}$ directly if the check fails. If the check passes, the challenger does not use brute force anymore, but computes hv' via the prPriv algorithm of $\mathsf{prQAHPS}$, without knowing a witness w for $x \in \widetilde{\mathcal{L}}_\rho$:

- $hv' \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$,

and returns $m := \mathsf{Decode}(d - hv')$ to $\mathcal{A}$.

We note that the challenger in this game is now PPT again, since it can use the language trapdoor td_ρ to decide the membership of $\widetilde{\mathcal{L}}_\rho$ efficiently.

The change from $\mathsf{G}_{7.N}$ to G_8 is reverse to that from G_5 to G_6. By a similar argument, we have $\big|\Pr_{7.N}[\mathsf{Win}] - \Pr_8[\mathsf{Win}]\big| \le (Q_e + Q_d) \cdot \varepsilon_{\mathsf{evalInd}}$.

Game G_9: It is the same as G_8, except that, for all the $\mathcal{O}_{\mathrm{ENC}}$ queries, the challenger samples $hv^* \leftarrow_{\$} \mathcal{HV}$ uniformly, instead of computing hv^* with $\{sk'_i\}_{i\in[N]}$.

Note that the only place that G_8 differs from G_9 lies in the computations of hv^* in the $\mathcal{O}_{\text{ENC}}$ oracle for all users $i^* \in [N]$, where $hv^* \leftarrow_{\$} \mathsf{prPriv}(sk'_{i^*}, x^*)$ in G_8 while $hv^* \leftarrow_{\$} \mathcal{HV}$ in G_9. Since $\{sk'_i\}_{i\in[N]}$ is used only in the computations of hv^* in $\mathcal{O}_{\text{ENC}}$, and x^* in $\mathcal{O}_{\text{ENC}}$ are uniformly chosen from $\mathcal{L}_{\rho_0}$, by the $\mathscr{L}_0$-multi-key-multi-extracting property of $\mathsf{prQAHPS}$ (cf. Definition 10), we have the following claim.

Claim 11. $\left|\Pr_8[\mathsf{Win}] - \Pr_9[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathscr{L}_0\text{-mk-mext}}_{\mathsf{prQAHPS},\mathcal{B}_6,N,Q_e}(\lambda)$.

We provide a formal proof for Claim 11 in Appendix C.3.

Finally in G_9, hv^* is uniformly chosen from $\mathcal{HV}$ and $d^* := hv^* + \mathsf{Encode}(m_\beta)$, thus the challenge bit β is completely hidden to $\mathcal{A}$. Then $\Pr_9[\mathsf{Win}] = \frac{1}{2}$.

Taking all things together, Theorem 2 follows. □

C.1 Proof of Claim 9

Claim 9. $\left|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$.

Proof. By Forge denote the event that $\mathcal{A}$ ever queries $\mathcal{O}_{\text{DEC}}(i, c = (x, d, \pi))$ s.t.

$$(i, c = (x, d, \pi)) \notin \mathcal{Q}_{\text{ENC}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}} \wedge x \notin \widetilde{\mathcal{L}}_\rho.$$

G_4 and G_5 are the same until Forge occurs, so $\left|\Pr_4[\mathsf{Win}]-\Pr_5[\mathsf{Win}]\right| \leq \Pr_5[\mathsf{Forge}]$.

To bound $\Pr_5[\mathsf{Forge}]$, we construct an adversary $\mathcal{B}_5$ against the USS of tag-based QANIZK (cf. Definition 19) for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ as follows. $\mathcal{B}_5$ is given $(\rho, td_\rho, \mathsf{crs})$ and has access to the oracle $\mathcal{O}_{\text{SIM}}$ defined in Fig. 8 (cf. Definition 19). $\mathcal{B}_5$ simulates G_5 for $\mathcal{A}$ as follows.

- Firstly, $\mathcal{B}_5$ invokes $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$, samples $H \leftarrow_{\$} \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{PKE}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{crs}, H)$. Then for each user $i \in [N]$, $\mathcal{B}_5$ samples secret key $sk_i \leftarrow_{\$} \mathcal{SK}$ itself and computes the corresponding public key $pk_i := \alpha_\rho(sk_i)$. $\mathcal{B}_5$ also picks $(\rho_0, td_{\rho_0}) \leftarrow_{\$} \mathscr{L}_0$. $\mathcal{B}_5$ sends $(\mathsf{pp}_{\mathsf{PKE}}, \{pk_i\}_{i\in[N]})$ to $\mathcal{A}$.
- For an $\mathcal{O}_{\text{ENC}}$ query (i^*, m_0, m_1) made by $\mathcal{A}$, $\mathcal{B}_5$ samples $x^* \leftarrow_{\$} \mathcal{L}_{\rho_0}$, computes $hv^* \leftarrow_{\$} \mathsf{prPriv}(sk_{i^*}, x^*)$, $d^* := hv^* + \mathsf{Encode}(m_\beta)$ and $\tau^* := H(pk_{i^*}, d^*)$. Then $\mathcal{B}_5$ sends (τ^*, x^*) to its own $\mathcal{O}_{\text{SIM}}$ oracle and obtains π^*, which is generated by $\mathcal{O}_{\text{SIM}}$ via $\pi^* \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*, x^*)$. $\mathcal{B}_5$ returns $c^* := (x^*, d^*, \pi^*)$ to $\mathcal{A}$, puts (i^*, c^*) to $\mathcal{Q}_{\text{ENC}}$ and puts (τ^*, x^*, π^*) to $\mathcal{Q}_{\text{SIM}}$.
- For an $\mathcal{O}_{\text{DEC}}$ query $(i, c = (x, d, \pi))$ made by $\mathcal{A}$, $\mathcal{B}_5$ computes $\tau := H(pk_i, d)$, checks whether $(i, c) \notin \mathcal{Q}_{\text{ENC}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}}$, and returns $\perp$ to $\mathcal{A}$ if the check fails. Then $\mathcal{B}_5$ uses td_ρ to further check whether $x \in \widetilde{\mathcal{L}}_\rho$. If $x \notin \widetilde{\mathcal{L}}_\rho$, $\mathcal{B}_5$ returns $\perp$ to $\mathcal{A}$, the same as G_5, and sends (τ, x, π) to its own challenger as its forgery. If $x \in \widetilde{\mathcal{L}}_\rho$, $\mathcal{B}_5$ computes $hv' \leftarrow_{\$} \mathsf{prPriv}(sk_i, x)$, and returns $m := \mathsf{Decode}(d - hv')$ to $\mathcal{A}$, the same as G_5.
- $\mathcal{B}_5$ uses $\{sk_i\}_{i\in[N]}$ to answer $\mathcal{O}_{\text{COR}}$ queries for $\mathcal{A}$, the same as G_5.

It is clear to see that $\mathcal{B}_5$ simulates G_5 perfectly for $\mathcal{A}$, and $\mathcal{B}_5$ outputs a successful forgery (τ, x, π) to its own challenger so that $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}} \wedge x \notin \widetilde{\mathcal{L}}_\rho$ as long as Forge occurs. Therefore, $\Pr_5[\mathsf{Forge}] \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$ and Claim 9 follows. ▮

C.2 Proof of Claim 10

Claim 10. For each $\eta \in [N]$, $|\Pr_{7.\eta-1}[\mathsf{Win}] - \Pr_{7.\eta}[\mathsf{Win}]| \leq \epsilon^{\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-ks}}_{\mathsf{prQAHPS}, \mathcal{B}_7}$.

Proof. Note that the only difference between $\mathsf{G}_{7.\eta-1}$ and $\mathsf{G}_{7.\eta}$ lies in the $\mathcal{O}_{\mathrm{ENC}}$ oracle for user η: in $\mathsf{G}_{7.\eta-1}$, $\mathcal{O}_{\mathrm{ENC}}$ computes $hv^* \leftarrow_\$ \mathsf{prPub}(\alpha_{\rho_0}(sk_\eta), x^*, w^*)$ using sk_η, while in $\mathsf{G}_{7.\eta}$, $\mathcal{O}_{\mathrm{ENC}}$ computes $hv^* \leftarrow_\$ \mathsf{prPub}(\alpha_{\rho_0}(sk'_\eta), x^*, w^*)$ using sk'_η.

Let Cor_η denote the event that $\mathcal{A}$ corrupts user η, i.e., $\mathcal{A}$ ever queries $\mathcal{O}_{\mathrm{COR}}(\eta)$ when $(\eta, \cdot) \notin \mathcal{Q}_{\mathrm{ENC}}$ and obtains sk_η. In the case that Cor_η occurs, η is appended to $\mathcal{Q}_{\mathrm{COR}}$, thus $\mathcal{A}$ is not allowed to query $\mathcal{O}_{\mathrm{ENC}}(\eta, m_0, m_1)$ for user η, and $\mathsf{G}_{7.\eta-1}$ is identical to $\mathsf{G}_{7.\eta}$. Consequently,

$$|\Pr_{7.\eta-1}[\mathsf{Win}] - \Pr_{7.\eta}[\mathsf{Win}]| = |\Pr_{7.\eta-1}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta] - \Pr_{7.\eta}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta]|. \quad (15)$$

To bound (15), we first analyze the information about sk_η (resp. sk_η and sk'_η) that $\mathcal{A}$ may obtain in $\mathsf{G}_{7.\eta-1}$ (resp. $\mathsf{G}_{7.\eta}$) in the case that $\neg\mathsf{Cor}_\eta$ occurs.

- Firstly, the public keys contain $pk_\eta = \alpha_\rho(sk_\eta)$.
- In $\mathcal{O}_{\mathrm{ENC}}(\eta, m_0, m_1)$, due to the game change in G_6, the behavior of $\mathcal{O}_{\mathrm{ENC}}$ for user η is determined by $\alpha_{\rho_0}(sk_\eta)$ (resp. $\alpha_{\rho_0}(sk'_\eta)$).
- In $\mathcal{O}_{\mathrm{DEC}}(\eta, c)$, due to the game change in G_6, the behavior of $\mathcal{O}_{\mathrm{DEC}}$ for user η is determined by $\alpha_\rho(sk_\eta)$.
- In the case that $\neg\mathsf{Cor}_\eta$, $\mathcal{A}$ never queries $\mathcal{O}_{\mathrm{COR}}(\eta)$.

Overall, the information about sk_η (resp. sk_η and sk'_η) that $\mathcal{A}$ learns in $\mathsf{G}_{7.\eta-1}$ (resp. $\mathsf{G}_{7.\eta}$) is limited in $\alpha_\rho(sk_\eta)$ and $\alpha_{\rho_0}(sk_\eta)$ (resp. $\alpha_{\rho_0}(sk'_\eta)$).

Then we analyze (15). Intuitively, by the $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching property of $\mathsf{prQAHPS}$ (cf. Definition 8), $\alpha_{\rho_0}(sk_\eta)$ is statistically close to $\alpha_{\rho_0}(sk'_\eta)$, even in the presence of $\alpha_\rho(sk_\eta)$. Thus, the $\mathcal{O}_{\mathrm{ENC}}$ for user η in $\mathsf{G}_{7.\eta-1}$ (using sk_η) is statistically close to that in $\mathsf{G}_{7.\eta}$ (using sk'_η).

Formally, we build an (unbounded) adversary $\mathcal{B}_7$ against the $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching property of $\mathsf{prQAHPS}$. $\mathcal{B}_7$ is given a challenge $(\mathsf{pp}_{\mathsf{HPS}}, \rho, \rho_0, \alpha_\rho(sk), \alpha_{\rho_0}(sk))$ (say $b = 0$) or $(\mathsf{pp}_{\mathsf{HPS}}, \rho, \rho_0, \alpha_\rho(sk), \alpha_{\rho_0}(sk'))$ (say $b = 1$), where $sk, sk' \leftarrow_\$ \mathcal{SK}$ are chosen by its own challenger, and $\mathcal{B}_7$ wants to decide which case it is. To this end, $\mathcal{B}_7$ will simulate $\mathsf{G}_{7.\eta-1}$ (or $\mathsf{G}_{7.\eta}$) for $\mathcal{A}$. $\mathcal{B}_7$ picks a challenge bit $\beta \leftarrow_\$ \{0, 1\}$. Intuitively, $\mathcal{B}_7$ will implicitly set sk_η as sk and set sk'_η as sk' for user η, where sk and sk' are the hashing keys chosen by its own challenger, and explicitly define the public key of user η as the $\alpha_\rho(sk)$ contained in its input. For the remaining $N - 1$ users $i \in [N] \setminus \{\eta\}$, $\mathcal{B}_7$ samples secret keys sk_i, sk'_i itself, thus can honestly answer $\mathcal{O}_{\mathrm{ENC}}$ queries (sampling x^* from $\mathcal{L}_{\rho_0}$), $\mathcal{O}_{\mathrm{DEC}}$ queries (using brute force to decide the membership of $\widetilde{\mathcal{L}}_\rho$ and find witness) and $\mathcal{O}_{\mathrm{COR}}$ queries made by $\mathcal{A}$ for these users. For user η, $\mathcal{B}_7$ can answer $\mathcal{O}_{\mathrm{DEC}}$ queries using the projection key $\alpha_\rho(sk)$ contained in its own input (since $\mathcal{O}_{\mathrm{DEC}}$ will output $\perp$

unless $x \in \widetilde{\mathcal{L}}_\rho$ & $\mathcal{B}_7$ can decide the membership of $\widetilde{\mathcal{L}}_\rho$ and find witness using brute force), and aborts immediately if $\mathcal{A}$ corrupts η. To answer $\mathcal{O}_{\text{ENC}}$ queries of user η, $\mathcal{B}_7$ samples x^* from $\mathcal{L}_{\rho_0}$, and uses the projection key $\alpha_{\rho_0}(sk)$ (or $\alpha_{\rho_0}(sk')$) contained in its own challenge to compute hv^*. Finally, $\mathcal{B}_7$ receives a bit β' from $\mathcal{A}$ and returns 1 to its own challenger as the guessing of b if and only if $\beta' = \beta$ and $\neg\mathsf{Cor}_\eta$ occurs (i.e., $\mathcal{A}$ never corrupts user η). Overall, $\mathcal{B}_7$ simulates $\mathsf{G}_{7.\eta-1}$ perfectly for $\mathcal{A}$ if $b = 0$ and $\neg\mathsf{Cor}_\eta$ occurs, and simulates $\mathsf{G}_{7.\eta}$ perfectly for $\mathcal{A}$ if $b = 1$ and $\neg\mathsf{Cor}_\eta$ occurs. Therefore, $\mathcal{B}_7$ successfully distinguishes $b = 0$ from $b = 1$ as long as the probability that $\beta' = \beta$ in $\mathsf{G}_{7.\eta-1}$ differs non-negligibly from that in $\mathsf{G}_{7.\eta}$ in the case $\neg\mathsf{Cor}_\eta$, and consequently, we have $\epsilon^{\langle \mathscr{L},\mathscr{L}_0\rangle\text{-ks}}_{\mathsf{prQAHPS},\mathcal{B}_7} \geq |\,\Pr_{7.\eta-1}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta] - \Pr_{7.\eta}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta]|$.

The full description of $\mathcal{B}_7$ is as follows.

- $\mathcal{B}_7$ is given a challenge $(\mathsf{pp}_{\mathsf{HPS}}, \rho, \rho_0, \alpha_\rho(sk), \widetilde{pk}_b)$, where $\widetilde{pk}_0 := \alpha_{\rho_0}(sk)$ and $\widetilde{pk}_1 := \alpha_{\rho_0}(sk')$ with $sk, sk' \leftarrow_\$ \mathcal{SK}$ are chosen by $\mathcal{B}_7$'s own challenger.
- Firstly, $\mathcal{B}_7$ invokes $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_\$ \mathsf{SimGen}(\rho)$, samples $H \leftarrow_\$ \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{PKE}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{crs}, H)$. $\mathcal{B}_7$ also samples $(\rho_0, td_{\rho_0}) \leftarrow_\$ \mathscr{L}_0$, and samples a challenge bit $\beta \leftarrow_\$ \{0, 1\}$ for $\mathcal{A}$.

  For user η, $\mathcal{B}_7$ sets $sk_\eta := sk$ and $sk'_\eta := sk'$ implicitly and defines $pk_\eta := \alpha_\rho(sk)$ explicitly, where sk and sk' are the hashing keys chosen by $\mathcal{B}_7$'s own challenger and $\alpha_\rho(sk)$ is part of $\mathcal{B}_7$'s own input. For all other users $i \in [N] \setminus \{\eta\}$, $\mathcal{B}_7$ samples secret keys $sk_i, sk'_i \leftarrow_\$ \mathcal{SK}$ itself and computes $pk_i := \alpha_\rho(sk_i)$. $\mathcal{B}_7$ sends $(\mathsf{pp}_{\mathsf{PKE}}, \{pk_i\}_{i\in[N]})$ to $\mathcal{A}$.
- When answering an $\mathcal{O}_{\text{ENC}}$ query (i^*, m_0, m_1) for user $i^* \neq \eta$ made by $\mathcal{A}$, $\mathcal{B}_7$ computes a challenge ciphertext c^* the same way as $\mathsf{G}_{7.\eta-1}$ and $\mathsf{G}_{7.\eta}$.

  More precisely, $\mathcal{B}_7$ samples $x^* \leftarrow_\$ \mathcal{L}_{\rho_0}$ with witness w^*, computes $hv^* \leftarrow_\$ \mathsf{prPub}(\alpha_{\rho_0}(sk'_{i^*}), x^*, w^*)$ using sk'_{i^*} if $i^* < \eta$ and computes $hv^* \leftarrow_\$ \mathsf{prPub}(\alpha_{\rho_0}(sk_{i^*}), x^*, w^*)$ using sk_{i^*} if $i^* > \eta$. Then $\mathcal{B}_7$ computes $d^* := hv^* + \mathsf{Encode}(m_\beta)$, $\tau^* := H(pk_{i^*}, d^*)$, invokes $\pi^* \leftarrow_\$ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*, x^*)$ and sets $c^* := (x^*, d^*, \pi^*)$.

  $\mathcal{B}_7$ returns c^* to $\mathcal{A}$, puts (i^*, c^*) to $\mathcal{Q}_{\text{ENC}}$ and puts (τ^*, x^*, π^*) to $\mathcal{Q}_{\text{SIM}}$.
- When answering an $\mathcal{O}_{\text{ENC}}$ query (η, m_0, m_1) for user η made by $\mathcal{A}$, $\mathcal{B}_7$ computes a challenge ciphertext c^* as follows.

  $\mathcal{B}_7$ samples $x^* \leftarrow_\$ \mathcal{L}_{\rho_0}$ with witness w^*, and computes $hv^* \leftarrow_\$ \mathsf{prPub}(\widetilde{pk}_b, x^*, w^*)$ using the projection key $\widetilde{pk}_b$ contained in $\mathcal{B}_7$'s challenge. Then $\mathcal{B}_7$ computes $d^* := hv^* + \mathsf{Encode}(m_\beta)$, $\tau^* := H(pk_\eta, d^*)$, $\pi^* \leftarrow_\$ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*, x^*)$ and sets $c^* := (x^*, d^*, \pi^*)$.

  $\mathcal{B}_7$ returns c^* to $\mathcal{A}$, puts (η, c^*) to $\mathcal{Q}_{\text{ENC}}$ and puts (τ^*, x^*, π^*) to $\mathcal{Q}_{\text{SIM}}$.

  In the case $b = 0$, note that $\widetilde{pk}_0 = \alpha_{\rho_0}(sk)$ and $\mathcal{B}_7$ implicitly sets $sk_\eta := sk$, it follows that $hv^* \leftarrow_\$ \mathsf{prPub}(\widetilde{pk}_b, x^*, w^*) = \mathsf{prPub}(\alpha_{\rho_0}(sk_\eta), x^*, w^*)$, thus $\mathcal{B}_7$ perfectly simulates $\mathsf{G}_{7.\eta-1}$ for $\mathcal{A}$; in the case $b = 1$, note that $\widetilde{pk}_1 = \alpha_{\rho_0}(sk')$ and $\mathcal{B}_7$ implicitly sets $sk'_\eta := sk'$, it follows that $hv^* \leftarrow_\$ \mathsf{prPub}(\widetilde{pk}_b, x^*, w^*) = \mathsf{prPub}(\alpha_{\rho_0}(sk'_\eta), x^*, w^*)$, thus $\mathcal{B}_7$ perfectly simulates $\mathsf{G}_{7.\eta}$ for $\mathcal{A}$.

- For an $\mathcal{O}_{\text{Dec}}$ query $(i, c = (x, d, \pi))$ made by $\mathcal{A}$, $\mathcal{B}_7$ decrypts the same way as $\mathsf{G}_{7.\eta-1}$ and $\mathsf{G}_{7.\eta}$.

  More precisely, $\mathcal{B}_7$ computes $\tau := H(pk_i, d)$, checks whether $(i, c) \notin \mathcal{Q}_{\text{Enc}}$ $\wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\text{Sim}}$, and returns $\perp$ to $\mathcal{A}$ if the check fails. Then $\mathcal{B}_7$ uses brute force to further decide whether $x \in \widetilde{\mathcal{L}}_\rho$. If $x \notin \widetilde{\mathcal{L}}_\rho$, $\mathcal{B}_7$ returns $\perp$ to $\mathcal{A}$. If $x \in \widetilde{\mathcal{L}}_\rho$, $\mathcal{B}_7$ uses brute force to find a witness w for $x \in \widetilde{\mathcal{L}}_\rho$, computes $hv' \leftarrow_{\$} \mathsf{prPub}(\alpha_\rho(sk_i), x, w)$ using sk_i if $i \neq \eta$ and computes $hv' \leftarrow_{\$} \mathsf{prPub}(\alpha_\rho(sk), x, w)$ using the projection key $\alpha_\rho(sk)$ contained in its own input if $i = \eta$, and returns $m := \mathsf{Decode}(d - hv')$ to $\mathcal{A}$.
- For an $\mathcal{O}_{\text{Cor}}$ query i made by $\mathcal{A}$, if $i \neq \eta$, $\mathcal{B}_7$ returns sk_i to $\mathcal{A}$; if $i = \eta$, $\mathcal{B}_7$ aborts immediately.
- Finally, $\mathcal{B}_7$ receives a bit β' from $\mathcal{A}$, and outputs 1 to its own challenger as the guessing of b if and only if $\beta' = \beta$ and $\mathcal{A}$ never corrupts η (i.e., $\neg\mathsf{Cor}_\eta$).

It is clearly that $\mathcal{B}_7$ simulates oracles $\mathcal{O}_{\text{Enc}}$ w.r.t. users $i^* \neq \eta$ and $\mathcal{O}_{\text{Dec}}$ perfectly for $\mathcal{A}$, and simulates oracle $\mathcal{O}_{\text{Cor}}$ perfectly for $\mathcal{A}$ as well in the case of $\neg\mathsf{Cor}_\eta$. Moreover, $\mathcal{B}_7$'s simulation of oracle $\mathcal{O}_{\text{Enc}}$ w.r.t. user η is the same as $\mathsf{G}_{7.\eta-1}$ in the case $b = 0$ and the same as $\mathsf{G}_{7.\eta}$ in the case $b = 1$. Overall, $\mathcal{B}_7$ simulates $\mathsf{G}_{7.\eta-1}$ perfectly for $\mathcal{A}$ in the case $b = 0$ and $\neg\mathsf{Cor}_\eta$, and simulates $\mathsf{G}_{7.\eta}$ perfectly for $\mathcal{A}$ in the case $b = 1$ and $\neg\mathsf{Cor}_\eta$. Therefore, we have

$$\begin{aligned} \epsilon^{\langle \mathscr{L}, \mathscr{L}_0\rangle\text{-ks}}_{\mathsf{prQAHPS}, \mathcal{B}_7} &= |\Pr[\mathcal{B}_7 \Rightarrow 1 | b = 0] - \Pr[\mathcal{B}_7 \Rightarrow 1 | b = 1]| \\ &= |\Pr[\beta' = \beta \ \wedge \ \neg\mathsf{Cor}_\eta | b = 0] - \Pr[\beta' = \beta \ \wedge \ \neg\mathsf{Cor}_\eta | b = 1]| \qquad (16) \\ &= |\Pr_{7.\eta-1}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta] - \Pr_{7.\eta}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta]|. \end{aligned}$$

Taking (15) and (16) together, Claim 10 follows. ∎

C.3 Proof of Claim 11

Claim 11. $\big|\Pr_8[\mathsf{Win}] - \Pr_9[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathscr{L}_0\text{-mk-mext}}_{\mathsf{prQAHPS}, \mathcal{B}_6, N, Q_e}(\lambda)$.

Proof. The only place that G_8 differs from G_9 lies in $\mathcal{O}_{\text{Enc}}$. For an $\mathcal{O}_{\text{Enc}}(i^*, m_0, m_1)$ query, the challenger samples $x^* \leftarrow_{\$} \mathcal{L}_{\rho_0}$, and computes $hv^* \leftarrow_{\$} \mathsf{prPriv}(sk'_{i^*}, x^*)$ in G_8 while samples $hv^* \leftarrow_{\$} \mathcal{HV}$ in G_9.

Let us fix some notations. Let i^*_j, x^*_j, hv^*_j denote the i^*, x^*, hv^* in the j-th $\mathcal{O}_{\text{Enc}}$ query, respectively, where $j \in [Q_e]$. The difference between G_8 and G_9 can be characterized by the following two distributions:

- G_8: $\big(\ x^*_j \leftarrow_{\$} \mathcal{L}_{\rho_0},\ hv^*_j \leftarrow_{\$} \mathsf{prPriv}(sk'_{i^*_j}, x^*_j)\ \big)_{j \in [Q_e]}$,
- G_9: $\big(\ x^*_j \leftarrow_{\$} \mathcal{L}_{\rho_0},\ hv^*_j \leftarrow_{\$} \mathcal{HV}\ \big)_{j \in [Q_e]}$.

Since $\{sk'_i\}_{i \in [N]}$ is used only in the computations of $\{hv^*_j\}_{j \in [Q_e]}$ in $\mathcal{O}_{\text{Enc}}$, and $\{x^*_j\}_{j \in [Q_e]}$ in $\mathcal{O}_{\text{Enc}}$ are uniformly chosen from $\mathcal{L}_{\rho_0}$, by the $\mathscr{L}_0$-multi-key-multi-extracting property of $\mathsf{prQAHPS}$ (cf. Definition 10), the above two distributions are computationally indistinguishable.

Formally, we build an adversary $\mathcal{B}_6$ against the $\mathscr{L}_0$-multi-key-multi-extracting property of $\mathsf{prQAHPS}$ by invoking $\mathcal{A}$. $\mathcal{B}_6$ is given $(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \{x_{i,j}, hv_{i,j}\}_{i\in[N],j\in[Q_e]})$, where $(\rho_0, td_{\rho_0}) \leftarrow_{\$} \mathscr{L}_0$, $sk'_1, ..., sk'_n \leftarrow_{\$} \mathcal{SK}$, and $x_{1,1}, ..., x_{N,Q_e} \leftarrow_{\$} \mathcal{L}_{\rho_0}$. $\mathcal{B}_6$ aims to decide whether $hv_{i,j} \leftarrow_{\$} \mathsf{prPriv}(sk'_i, x_{i,j})$ for all $i \in [N]$ and $j \in [Q_e]$ (say $b = 0$) or $hv_{1,1}, ..., hv_{N,Q} \leftarrow_{\$} \mathcal{HV}$ (say $b = 1$). $\mathcal{B}_6$ will simulate G_8 or G_9 for $\mathcal{A}$, depending on the value of b.

- Firstly, $\mathcal{B}_6$ invokes $(\rho, td_\rho) \leftarrow_{\$} \mathscr{L}$, $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{SimGen}(\rho)$, samples $H \leftarrow_{\$} \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{PKE}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{crs}, H)$. Then for each user $i \in [N]$, $\mathcal{B}_6$ samples secret key $sk_i \leftarrow_{\$} \mathcal{SK}$ itself and computes the corresponding public key $pk_i := \alpha_\rho(sk_i)$. $\mathcal{B}_6$ sends $(\mathsf{pp}_{\mathsf{PKE}}, \{pk_i\}_{i\in[N]})$ to $\mathcal{A}$. $\mathcal{B}_6$ also picks a challenge bit $\beta \leftarrow_{\$} \{0, 1\}$ for $\mathcal{A}$.
- $\mathcal{B}_6$ has the secret keys sk_i of all users, thus can honestly answer $\mathcal{O}_{\text{DEC}}$ queries (using td_ρ to decide the membership of $\widetilde{\mathcal{L}_\rho}$) and $\mathcal{O}_{\text{COR}}$ queries made by $\mathcal{A}$, the same way as G_8 and G_9.
- As for $\mathcal{O}_{\text{ENC}}$ queries, when answering the j-th ($j \in [Q_e]$) $\mathcal{O}_{\text{ENC}}$ query $(i^*_j, m_{0,j}, m_{1,j})$, $\mathcal{B}_6$ sets x^*_j as the $x_{i^*_j,j}$ in its own input, and sets hv^*_j as the $hv_{i^*_j,j}$ in its own input. Then $\mathcal{B}_6$ computes $d^*_j := hv^*_j + \mathsf{Encode}(m_{\beta,j})$, $\tau^*_j := H(pk_{i^*_j}, d^*_j)$ and $\pi^*_j \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*_j, x^*_j)$, without knowing a witness of x^*_j. $\mathcal{B}_6$ returns $c^*_j := (x^*_j, d^*_j, \pi^*_j)$ to $\mathcal{A}$, puts (i^*_j, c^*_j) to $\mathcal{Q}_{\text{ENC}}$ and puts $(\tau^*_j, x^*_j, \pi^*_j)$ to $\mathcal{Q}_{\text{SIM}}$.

  In the case $b = 0$, $hv^*_j = hv_{i^*_j,j}$ is generated by $\mathsf{prPriv}(sk'_{i^*_j}, x_{i^*_j,j}) = \mathsf{prPriv}(sk'_{i^*_j}, x^*_j)$, thus $\mathcal{B}_6$ perfectly simulates G_8 for $\mathcal{A}$; in the case $b = 1$, $hv^*_j = hv_{i^*_j,j}$ is uniformly random over $\mathcal{HV}$, thus $\mathcal{B}_6$ perfectly simulates G_9.
- Finally, $\mathcal{B}_6$ receives a bit β' from $\mathcal{A}$ and returns 1 to its own challenger if and only if $\beta' = \beta$.

Overall, $\mathcal{B}_6$ simulates G_8 for $\mathcal{A}$ in the case $b = 0$ and simulates G_9 for $\mathcal{A}$ in the case $b = 1$, thus $\mathcal{B}_6$ successfully distinguishes $b = 0$ from $b = 1$ as long as the probability that $\beta' = \beta$ in G_8 differs non-negligibly from that in G_9. Consequently, we have $\mathsf{Adv}^{\mathscr{L}_0\text{-mk-mext}}_{\mathsf{prQAHPS},\mathcal{B}_6,N,Q_e}(\lambda) \geq \big|\Pr_8[\mathsf{Win}] - \Pr_9[\mathsf{Win}]\big|$.

This completes the proof of Claim 11. ▌

D Missing Details in Sect. 5 and Proof of Theorem 3 (Tighter Reduction from LWE to Multi-secret LWE)

In this section, we provide the missing details in Sect. 5, and in particular, the formal proof of Theorem 3.

Before presenting the proof, we first specify some notations involved in this section. For two distribution ensembles X, Y and a positive real number ϵ, we use the notation "$X \overset{c}{\approx} Y$ with ϵ" to denote $|\Pr[\mathcal{D}(X) = 1] - \Pr[\mathcal{D}(Y) = 1]| \leq \epsilon$ for all PPT distinguishers $\mathcal{D}$. For a matrix $\mathbf{M}$, we use $\sigma_{\mathbf{M}}$ to denote its spectral norm.

The rest of this section is organized as follows. In Appendix D.1, we introduce some definitions and lemmas need in our formal proof. Then in Appendix D.2, we present the formal proof of Theorem 3. Finally, in Appendix D.3, we extend Theorem 3 to Theorem 10, which addresses the almost tight reduction from LWE to Multi-secret LWE for arbitrary modulus instead of just the prime modulus.

D.1 Additional Backgrounds on Lattices

Firstly, we recall the definition of "Lossy Sampler".

Definition 23 (Lossy Sampler [2, Definition 3.1]). *Let λ be the security parameter, n, m, ℓ, q be integers (functions of λ), and $\chi = \chi(\lambda)$ be a distribution over $\mathbb{Z}_q$. We define the following efficient lossy sampler $\tilde{\mathbf{A}} \leftarrow_{\$} \mathsf{Lossy}(1^n, 1^m, 1^\ell, q, \chi)$ as: Sample $\mathbf{B} \leftarrow_{\$} \mathbb{Z}_q^{\ell \times m}$, $\mathbf{C} \leftarrow_{\$} \mathbb{Z}_q^{n \times \ell}$, $\mathbf{F} \leftarrow_{\$} \chi^{n \times m}$ and output $\tilde{\mathbf{A}} = \mathbf{C} \cdot \mathbf{B} + \mathbf{F}$.*

The following lemma shows that the output of lossy sampler is computationally indistinguishable from random matrix.

Lemma 10 ([2]). *Let $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n \times m}$, and let $\tilde{\mathbf{A}} \leftarrow_{\$} \mathsf{Lossy}(1^n, 1^m, 1^\ell, q, \chi)$. Then, we have: $\mathbf{A} \overset{c}{\approx} \tilde{\mathbf{A}}$ with $\mathsf{Adv}^{n\text{-}\mathsf{LWE}}_{[\ell, q, \chi, m]}(\lambda)$.*

The following lemma shows the decomposition of continuous Gaussian vector.

Lemma 11 ([15, Proposition 3.2]). *Let $\mathbf{F} \in \mathbb{Z}^{n \times m}$ be an arbitrary matrix with spectral norm $\sigma_{\mathbf{F}}$. Let $\sigma_0, \sigma_1 > 0$ be s.t. $\sigma_0 > \sigma_1 \cdot \sigma_{\mathbf{F}}$. Let $\mathbf{e}_1^\top \leftarrow_{\$} D_{\sigma_1}^n$ and let $\mathbf{e}_2 \leftarrow_{\$} D_{\sqrt{\mathbf{\Sigma}}}$ for $\mathbf{\Sigma} = {\sigma_0}^2 \mathbf{I} - {\sigma_1}^2 \mathbf{F}^\top \mathbf{F}$. Then the random variable $\mathbf{e}^\top = \mathbf{e}_1^\top \mathbf{F} + \mathbf{e}_2^\top$ is distributed according to $D_{\sigma_0}^m$.*

With the results above, we can derive the following conditional min-entropy lower bound. The proof is similar to that of [15].

Lemma 12. *Let n, m, ℓ, q be positive integers. Let $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_q^n$, $\tilde{\mathbf{A}} \leftarrow_{\$} \mathsf{Lossy}(1^n, 1^m, 1^\ell, q, D_{\mathbb{Z},\gamma})$, $\mathbf{e}_1 \leftarrow_{\$} D_{\sigma_1}^n$, and $\mathbf{e} \leftarrow_{\$} D_{\sigma_0}^m$ such that $\sigma_0 > \gamma \cdot C \cdot \sqrt{m} \cdot \sigma_1$, where C is the global constant from Lemma 7. Then we have:*

$$\widetilde{\mathbf{H}}_\infty(\mathbf{s} \mid (\tilde{\mathbf{A}}, \mathbf{s}^\top \cdot \tilde{\mathbf{A}} + \mathbf{e}^\top)) \geq \widetilde{\mathbf{H}}_\infty(\mathbf{s} |\ \mathbf{s} + \mathbf{e}_1) - \ell \cdot \log q.$$

Proof. The proof is similar to that of [15, Theorem 4.1]. According to Definition 23, we know $\tilde{\mathbf{A}} = \mathbf{C} \cdot \mathbf{B} + \mathbf{F}$, and hence $\mathbf{s}^\top \cdot \tilde{\mathbf{A}} + \mathbf{e}^\top = \mathbf{s}^\top \cdot \mathbf{C} \cdot \mathbf{B} + \mathbf{s}^\top \mathbf{F} + \mathbf{e}^\top$.

Furthermore, by Lemma 11, we know $\mathbf{e}^\top = \mathbf{e}_1^\top \mathbf{F} + \mathbf{e}_2^\top$, so

$$\mathbf{s}^\top \cdot \mathbf{C} \cdot \mathbf{B} + \mathbf{s}^\top \mathbf{F} + \mathbf{e}^\top = \mathbf{s}^\top \cdot \mathbf{C} \cdot \mathbf{B} + \mathbf{s}^\top \mathbf{F} + \mathbf{e}_1^\top \mathbf{F} + \mathbf{e}_2^\top = \mathbf{s}^\top \cdot \mathbf{C} \cdot \mathbf{B} + (\mathbf{s}^\top + \mathbf{e}_1^\top)\mathbf{F} + \mathbf{e}_2^\top.$$

Note that $\tilde{\mathbf{A}}$ and $\mathbf{s}^\top \cdot \tilde{\mathbf{A}} + \mathbf{e}^\top$ can be reconstructed completely given $\mathbf{C}, \mathbf{B}, \mathbf{F}, \mathbf{s}^\top \cdot \mathbf{C}, \mathbf{s} + \mathbf{e}_1, \mathbf{e}_2$. Together with the fact that $\mathbf{s}^\top \cdot \mathbf{C}$ leaks at most $\ell \log q$ bits of information about $\mathbf{s}$, by Lemma 1, we have

$$\begin{aligned} \widetilde{\mathbf{H}}_\infty(\mathbf{s} \mid (\tilde{\mathbf{A}}, \mathbf{s}^\top \cdot \tilde{\mathbf{A}} + \mathbf{e}^\top)) \geq & \widetilde{\mathbf{H}}_\infty(\mathbf{s} \mid (\mathbf{C}, \mathbf{B}, \mathbf{F}, \mathbf{s}^\top \cdot \mathbf{C}, \mathbf{s} + \mathbf{e}_1, \mathbf{e}_2)) \\ = & \widetilde{\mathbf{H}}_\infty(\mathbf{s} \mid (\mathbf{s}^\top \cdot \mathbf{C}, \mathbf{s} + \mathbf{e}_1)) \geq \widetilde{\mathbf{H}}_\infty(\mathbf{s} \mid \mathbf{s} + \mathbf{e}_1) - \ell \cdot \log q. \quad \square \end{aligned}$$

The following lemma states the lower bound of the so-called "noise lossiness" of uniformly random vectors.

Lemma 13 ([15, Lemma 5.2]). *Let n be an integer, let q be a modulus and σ_1 be a parameter for a Gaussian. Assume that $\frac{q}{\sigma_1} \geq \sqrt{\frac{\ln(4n)}{\pi}}$. Let $\mathbf{s} \leftarrow_{\$} \mathbb{Z}_q^n$ and $\mathbf{e}_1 \leftarrow_{\$} D_{\sigma_1}^n$. Then it holds that $\widetilde{\mathbf{H}}_\infty(\mathbf{s} \mid \mathbf{s}+\mathbf{e}_1) \geq n \cdot \log(\sigma_1) - 1$.*

D.2 Proof of Theorem 3

Now we recall Theorem 3 and present its formal proof.

Theorem 3 (LWE ⇒ Multi-secret LWE with Prime Modulus) *Let $n, m, \ell, q \in \mathbb{N}$, and q be a prime. Let $\sigma, \sigma_0, \sigma_1, r, \gamma > 0$ such that $\sigma = \sqrt{\sigma_0{}^2 + r^2}$, $\sigma_0 > \gamma \cdot C \cdot \sqrt{m} \cdot \sigma_1$, $\frac{q}{\sigma_1} \geq \sqrt{\frac{\ln(4n)}{\pi}}$ and $r \geq \sqrt{\lambda}$, where C is the global constant from Lemma 7. For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$, such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and $\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\mathbb{Z},\sigma},m],\mathcal{A}}(\lambda) \leq 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}}(\lambda) + \frac{Q(m+c+1)}{2^\lambda}$, where c is an integer such that $m' = \lfloor \frac{m}{c} \rfloor$ and $n \geq (m' \log q + \ell \log q + 2\lambda + 1)/\log(\sigma_1)$.*

Proof of Theorem 3. We will use the multi-secret LWE with continuous Gaussian D_{σ_0} defined in Definition 12 as an intermediate assumption, and show that there exists an adversary $\mathcal{B}'$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{B}') + Q \cdot \mathsf{poly}'(\lambda) \approx \mathbf{T}(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ and

$$\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\mathbb{Z},\sigma},m],\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\sigma_0},m],\mathcal{B}'}(\lambda) + \frac{Qm}{2^\lambda}, \tag{9}$$

$$\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\sigma_0},m],\mathcal{B}'}(\lambda) \leq 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}}(\lambda) + \frac{Q(c+1)}{2^\lambda}. \tag{10}$$

Then Theorem 3 follows directly from (9) and (10).

We already proved (9) in proof sketch in Sect. 5. It remains to show (10). Below we present the formal proof of (10). Our target is to prove that the Q-$\mathsf{LWE}_{n,q,D_{\sigma_0},m}$-assumption holds, i.e.,

$$\left(\mathbf{A}, \mathbf{s}_1^\top \cdot \mathbf{A} + \mathbf{e}_1^\top, \ldots, \mathbf{s}_Q^\top \cdot \mathbf{A} + \mathbf{e}_Q^\top\right) \overset{c}{\approx} \left(\mathbf{A}, \mathbf{u}_1^\top + \mathbf{e}_1^\top, \ldots, \mathbf{u}_Q^\top + \mathbf{e}_Q^\top\right), \tag{17}$$

based on the the $\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\gamma},m}$-assumption. Here $\mathbf{s}_1, \cdots, \mathbf{s}_Q$ and $\mathbf{u}_1, \cdots, \mathbf{u}_Q$ are independent and uniformly random in $\mathbb{Z}_q^n$ and $\mathbb{Z}_q^m$ respectively, and $\mathbf{e}_1, \cdots, \mathbf{e}_Q$ are independently sampled from $D_{\sigma_0}^m$.

Given the matrix $\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$ in (17), we can parse $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_{c+1})$, where $\mathbf{A}_j \in \mathbb{Z}_q^{n\times m_1}$ for $1 \leq j \leq c$, and $\mathbf{A}_{c+1} \in \mathbb{Z}_q^{n\times m_2}$ with $m_1 = \lfloor \frac{m}{c} \rfloor$ and $m_2 = m - c\lfloor \frac{m}{c} \rfloor$. Then the left part of (17) can be rewritten as

$$\begin{aligned}&\left(\mathbf{A}, \mathbf{s}_1^\top \cdot \mathbf{A} + \mathbf{e}_1^\top, \ldots, \mathbf{s}_Q^\top \cdot \mathbf{A} + \mathbf{e}_Q^\top\right) = \left(\{\mathbf{A}_j\}_{j\in[c+1]}, \{\mathbf{s}_i^\top \cdot \mathbf{A}_j + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[c+1]}\right)\\ &= \left(\mathbf{A}_1, \{\mathbf{s}_i^\top \cdot \mathbf{A}_1 + \mathbf{e}_{i,1}^\top\}_{i\in[Q]}, \ldots, \mathbf{A}_{c+1}, \{\mathbf{s}_i^\top \cdot \mathbf{A}_{c+1} + \mathbf{e}_{i,c+1}^\top\}_{i\in[Q]}\right),\end{aligned} \tag{18}$$

where $\mathbf{e}_{i,j} \leftarrow_{\$} D_{\sigma_0}^{m_1}$ for $1 \leq j \leq c$ and $\mathbf{e}_{i,c+1} \leftarrow_{\$} D_{\sigma_0}^{m_2}$.

Then, we will use the standard hybrid argument to prove (17). According to (18), the related hybrids are defined as follows.

– H_0: $\big(\mathbf{A}_1, \{\mathbf{s}_i^\top \cdot \mathbf{A}_1 + \mathbf{e}_{i,1}^\top\}_{i\in[Q]}, \ldots, \mathbf{A}_{c+1}, \{\mathbf{s}_i^\top \cdot \mathbf{A}_{c+1} + \mathbf{e}_{i,c+1}^\top\}_{i\in[Q]}\big)$.

– H_z for $1 \leq z \leq c$:

$$\Big(\boxed{\{\mathbf{A}_i\}_{i\in[z]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z]}}, \{\mathbf{A}_j\}_{(z+1)\leq j\leq c}, \{\mathbf{s}_i^\top \cdot \mathbf{A}_j + \mathbf{e}_{i,j}^\top\}_{i\in[Q],(z+1)\leq j\leq c},$$
$$\mathbf{A}_{c+1}, \{\mathbf{s}_i^\top \cdot \mathbf{A}_{c+1} + \mathbf{e}_{i,c+1}^\top\}_{i\in[Q]}\Big),$$

where $\mathbf{u}_{i,j} \leftarrow_\$ \mathbb{Z}_q^{m_1}$ for $i \in [Q], j \in [z]$.

– H_{c+1}: $\big(\{\mathbf{A}_i\}_{i\in[c+1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[c+1]}\big)$.

Therefore, we have that

$$\begin{aligned}\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\sigma_0},m],\mathcal{B}'}(\lambda) &= \big|\Pr[\mathcal{B}'(H_0)=1] - \Pr[\mathcal{B}'(H_{c+1})=1]\big| \\ &\leq \sum_{z=1}^{c+1} \big|\Pr[\mathcal{B}'(H_{z-1})=1] - \Pr[\mathcal{B}'(H_z)=1]\big|.\end{aligned} \tag{19}$$

Next, we use the following two claims to show the indistinguishability of these neighboring hybrids.

Claim 12. For each $1 \leq z \leq c$*, we have* $\big|\Pr[\mathcal{B}'(H_{z-1})=1] - \Pr[\mathcal{B}'(H_z)=1]\big| \leq 2n \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}}(\lambda) + Q \cdot 2^{-\lambda}$ *for an adversary* $\mathcal{B}$ *against the* $\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\gamma},m}$ *assumption with* $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{B}') + Q \cdot \mathsf{poly}'(\lambda) \approx \mathbf{T}(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$*.*

Proof. For $1 \leq z \leq c$, let $m'_z = m - zm_1$, $\mathbf{A}'_z = (\mathbf{A}_{z+1}, \ldots, \mathbf{A}_{c+1}) \in \mathbb{Z}_q^{n\times m'_z}$, $\mathbf{e}'_{i,z} = (\mathbf{e}_{i,z+1}, \ldots, \mathbf{e}_{i,c+1}) \in D_{\sigma_0}^{m'_z}$. Then we can parse $\mathbf{A} = (\mathbf{A}_1, \ldots, \mathbf{A}_{z-1}, \mathbf{A}_z, \mathbf{A}'_z)$. In this case, we can re-write hybrids H_{z-1} and H_z in the following way:

$$\begin{aligned}H_{z-1} :\Big(&\{\mathbf{A}_i\}_{i\in[z-1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z-1]}, \boxed{\mathbf{A}_z, \{\mathbf{s}_i^\top \cdot \mathbf{A}_z + \mathbf{e}_{i,z}^\top\}_{i\in[Q]}}, \mathbf{A}'_z, \\ &\{\mathbf{s}_i^\top \cdot \mathbf{A}'_z + \mathbf{e}'^\top_{i,z}\}_{i\in[Q]}\Big),\end{aligned} \tag{20}$$

$$\begin{aligned}H_z :\Big(&\{\mathbf{A}_i\}_{i\in[z-1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z-1]}, \boxed{\mathbf{A}_z, \{\mathbf{u}_{i,z}^\top + \mathbf{e}_{i,z}^\top\}_{i\in[Q]}}, \mathbf{A}'_z, \\ &\{\mathbf{s}_i^\top \cdot \mathbf{A}'_z + \mathbf{e}'^\top_{i,z}\}_{i\in[Q]}\Big),\end{aligned} \tag{21}$$

where $\mathbf{u}_{i,z} \leftarrow_\$ \mathbb{Z}_q^{m_1}$ for $i \in [Q]$. Hence, the target of this claim is to prove that (20) and (21) are computationally indistinguishable.

To do this, we take $n, m'_z, \ell, q, D_{\mathbb{Z},\gamma}$ as input and run lossy sampler $\mathsf{Lossy}(1^n, 1^{m'_z}, 1^\ell, q, D_{\mathbb{Z},\gamma})$ to get $\tilde{\mathbf{A}}'_z = \mathbf{C}\cdot\mathbf{B}+\mathbf{F}$, where $\mathbf{B} \leftarrow_\$ \mathbb{Z}_q^{\ell\times m'_z}$, $\mathbf{C} \leftarrow_\$ \mathbb{Z}_q^{n\times\ell}$, $\mathbf{F} \leftarrow_\$ D_{\mathbb{Z},\gamma}^{n\times m'_z}$. According to Lemma 10, we have $\mathbf{A}'_z \overset{c}{\approx} \tilde{\mathbf{A}}'_z$ with $\mathsf{Adv}^{n\text{-}\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m'_z]}(\lambda)$. For (20), we have:

$$\begin{aligned}&\Big(\{\{\mathbf{A}_i\}_{i\in[z-1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z-1]}, \mathbf{A}_z, \{\mathbf{s}_i^\top \cdot \mathbf{A}_z + \mathbf{e}_{i,z}^\top\}_{i\in[Q]}, \\ &\boxed{\mathbf{A}'_z, \{\mathbf{s}_i^\top \cdot \mathbf{A}'_z + \mathbf{e}'^\top_{i,z}\}_{i\in[Q]}}\Big) \\ \overset{c}{\approx}&\Big(\{\mathbf{A}_i\}_{i\in[z-1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z-1]}, \mathbf{A}_z, \{\mathbf{s}_i^\top \cdot \mathbf{A}_z + \mathbf{e}_{i,z}^\top\}_{i\in[Q]}, \\ &\boxed{\tilde{\mathbf{A}}'_z, \{\mathbf{s}_i^\top \cdot \tilde{\mathbf{A}}'_z + \mathbf{e}'^\top_{i,z}\}_{i\in[Q]}}\Big),\end{aligned} \tag{22}$$

with $\mathsf{Adv}^{n\text{-}\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m'_z]}(\lambda)$.

Then, according to Lemma 12, Lemma 13 and our parameter setting as the theorem statement, it holds

$$\begin{aligned}\widetilde{\mathbf{H}}_\infty(\mathbf{s}_i \mid (\tilde{\mathbf{A}}'_z, \mathbf{s}_i^\top \cdot \tilde{\mathbf{A}}'_z + \mathbf{e}'^\top_{i,z})) &\geq \widetilde{\mathbf{H}}_\infty(\mathbf{s}_i \mid \mathbf{s}_i + \mathbf{e}_{i,1}) - \ell \cdot \log q \\ &\geq n \log(\sigma_1) - \ell \cdot \log q - 1 \\ &\geq m_1 \log q + 2\lambda.\end{aligned} \tag{23}$$

Moreover, by Lemma 2 and (23), for every $i \in [Q]$ and $\mathbf{u}_{i,z} \leftarrow\!\!\$\ \mathbb{Z}_q^{m_1}$, we have

$$\varDelta\left(\left(\mathbf{A}_z, \mathbf{s}_i^\top \cdot \mathbf{A}_z\right), \left(\mathbf{A}_z, \mathbf{u}_{i,z}^\top\right)\right) \leq 2^{-\lambda}. \tag{24}$$

In this case, through putting all $i \in [Q]$ in (24) together, we have

$$\begin{aligned}&\varDelta\big((\{\mathbf{A}_i\}_{i\in[z-1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z-1]}, \boxed{\mathbf{A}_z, \{\mathbf{s}_i^\top \cdot \mathbf{A}_z + \mathbf{e}_{i,z}^\top\}_{i\in[Q]}}, \tilde{\mathbf{A}}'_z, \\ &\{\mathbf{s}_i^\top \cdot \tilde{\mathbf{A}}'_z + \mathbf{e}'^\top_{i,z}\}_{i\in[Q]}), (\{\mathbf{A}_i\}_{i\in[z-1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z-1]}, \\ &\boxed{\mathbf{A}_z, \{\mathbf{u}_{i,z}^\top + \mathbf{e}_{i,z}^\top\}_{i\in[Q]}}, \tilde{\mathbf{A}}'_z, \{\mathbf{s}_i^\top \cdot \tilde{\mathbf{A}}'_z + \mathbf{e}'^\top_{i,z}\}_{i\in[Q]})\big) \leq Q \cdot 2^{-\lambda},\end{aligned} \tag{25}$$

since every $\{\mathbf{s}_i\}_{i\in[Q]}$ is sampled independently, and $\{\mathbf{A}_i\}_{i\in[z-1]}$ and $\tilde{\mathbf{A}}'_z$ are independent of $\mathbf{A}_z$.

Then, similar to (22), we can use Lemma 10 again to change $\tilde{\mathbf{A}}'_z$ back to $\mathbf{A}'_z$. Hence, it holds that

$$\begin{aligned}&\Big(\{\mathbf{A}_i\}_{i\in[z-1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z-1]}, \mathbf{A}_z, \{\mathbf{u}_{i,z}^\top + \mathbf{e}_{i,z}^\top\}_{i\in[Q]}, \\ &\boxed{\tilde{\mathbf{A}}'_z, \{\mathbf{s}_i^\top \cdot \tilde{\mathbf{A}}'_z + \mathbf{e}'^\top_{i,z}\}_{i\in[Q]}}\Big) \\ \overset{c}{\approx}&\Big(\{\mathbf{A}_i\}_{i\in[z-1]}, \{\mathbf{u}_{i,j}^\top + \mathbf{e}_{i,j}^\top\}_{i\in[Q],j\in[z-1]}, \mathbf{A}_z, \{\mathbf{u}_{i,z}^\top + \mathbf{e}_{i,z}^\top\}_{i\in[Q]}, \\ &\boxed{\mathbf{A}'_z, \{\mathbf{s}_i^\top \cdot \mathbf{A}'_z + \mathbf{e}'^\top_{i,z}\}_{i\in[Q]}}\Big),\end{aligned} \tag{26}$$

with $\mathsf{Adv}^{n\text{-}\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m'_z]}(\lambda)$.

Finally, through combining (20), (21) (22), (25) and (26) together, we get:

$$H_{z-1} \overset{c}{\approx} H_z,$$

with

$$\begin{aligned}2 \cdot \mathsf{Adv}^{n\text{-}\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m'_z]}(\lambda) + Q \cdot 2^{-\lambda} &\leq 2 \cdot \mathsf{Adv}^{n\text{-}\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m]}(\lambda) + Q \cdot 2^{-\lambda} \\ &\leq 2n \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m]}(\lambda) + Q \cdot 2^{-\lambda},\end{aligned}$$

where the last inequality follows from a simple hybrid argument. More specifically, we can construct an adversary $\mathcal{B}$, such that $\big|\Pr[\mathcal{B}'(H_{z-1}) = 1] - \Pr[\mathcal{B}'(H_z) = 1]\big| \leq 2n \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}}(\lambda) + Q \cdot 2^{-\lambda}$. This completes the proof of the claim. ▌

Claim 13. H_c and H_{c+1} are statistically indistinguishable. More specifically, the statistical distance between H_c and H_{c+1} is at most $Q \cdot 2^{-\lambda}$.

Proof. The difference between H_c and H_{c+1} can be noticed more clearly from the following descriptions:

$$H_c : \left(\{\mathbf{A}_i\}_{i\in[c]}, \{\mathbf{u}_{i,j}^{\top} + \mathbf{e}_{i,j}^{\top}\}_{i\in[Q],j\in[c]}, \boxed{\mathbf{A}_{c+1}, \{\mathbf{s}_i^{\top} \cdot \mathbf{A}_{c+1} + \mathbf{e}_{i,c+1}^{\top}\}_{i\in[Q]}} \right),$$

$$H_{c+1} : \left(\{\mathbf{A}_i\}_{i\in[c]}, \{\mathbf{u}_{i,j}^{\top} + \mathbf{e}_{i,j}^{\top}\}_{i\in[Q],j\in[c]}, \boxed{\mathbf{A}_{c+1}, \{\mathbf{u}_{i,c+1}^{\top} + \mathbf{e}_{i,c+1}^{\top}\}_{i\in[Q]}} \right).$$

In this case, it suffices to prove the statistical distance between $(\mathbf{A}_{c+1}, \mathbf{s}_i^{\top} \cdot \mathbf{A}_{c+1})$ and $(\mathbf{A}_{c+1}, \mathbf{u}_{i,c+1}^{\top})$ is negligible in λ, i.e.,

$$\Delta\left(\left(\mathbf{A}_{c+1}, \mathbf{s}_i^{\top} \cdot \mathbf{A}_{c+1} + \mathbf{e}_i\right), \left(\mathbf{A}_{c+1}, \mathbf{u}_{i,c+1}^{\top}\right) \right) \leq 2^{-\lambda}, \tag{27}$$

for all $i \in [Q]$, since every $\{\mathbf{s}_i\}_{i\in[Q]}$ is sampled independently, and $\{\mathbf{A}_i\}_{i\in[c]}$ are independent of $\mathbf{A}_{c+1}$. Furthermore, according to Lemma 2 and the lower bound on min-entropy $\widetilde{\mathbf{H}}_\infty(\mathbf{s}_i)$ from the theorem statement, (27) clearly holds. As a result, this claim follows. ▮

Now, by plugging Claim 12 and Claim 13 into (19), (10) is clearly set up.

Finally, taking (9) and (10) together, Theorem 3 holds. □

D.3 Almost Tight Reduction for Arbitrary Modulus

Similar to Theorem 3, we have the following theorem that addresses the almost tight reduction from LWE to Multi-secret LWE for arbitrary modulus.

Theorem 10 (LWE ⇒ Multi-secret LWE with Arbitrary Modulus). *Let $n, m, \ell, q \in \mathbb{N}$. Let $\sigma, \sigma_0, \sigma_1, r, \gamma > 0$ such that $\sigma = \sqrt{{\sigma_0}^2 + r^2}$, $\sigma_0 > \gamma \cdot C \cdot \sqrt{m} \cdot \sigma_1$, $\frac{q}{\sigma_1} \geq \sqrt{\frac{\ln(4n)}{\pi}}$ and $r \geq \sqrt{\lambda}$, where C is the global constant from Lemma 7. For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$, such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + Q \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and $\mathsf{Adv}^{Q\text{-}\mathsf{LWE}}_{[n,q,D_{\mathbb{Z},\sigma},m],\mathcal{A}}(\lambda) \leq 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}}(\lambda) + \frac{Q(m+c+1)}{2^{\lambda}}$, where c is an integer such that $m' = \lfloor \frac{m}{c} \rfloor$ and $n \geq (2m' \log q + \ell \log q + 2\lambda + 1)/\log(\frac{p\sigma_1}{q})$ for any q's prime factor p.*

The proof of Theorem 10 is almost identical to that of Theorem 3, except that in all places where we use (the first result of) Lemma 2 in the prime modulus setting, we now use the second result of Lemma 2 to deal with composite modulus.

E Missing Proofs in Subsect. 6.2 (Probabilistic QA-HPS from LWE)

E.1 Proof of Theorem 4 (Approximate Correctness & Evaluation Indistinguishability of $\mathsf{prQAHPS}_{\mathsf{LWE}}$)

First, we show the approximate correctness for instances in $\mathcal{L}_\rho = \mathcal{L}_{\mathbf{A}}$. Note that for any $sk = \mathbf{k} \in \{0,1\}^m$, $pk_\rho = \mathbf{p} = \mathbf{A}\mathbf{k}$ and $\mathbf{c} = (\mathbf{s}^{\top}\mathbf{A} + \mathbf{e}^{\top})^{\top} \in \mathcal{L}_{\mathbf{A}}$ with witness $w_{\mathbf{c}} = (\mathbf{s}, \mathbf{e} \in [-B, B]^m)$, we have

$$\begin{aligned} \mathsf{prPub}(pk_\rho, \mathbf{c}, w_{\mathbf{c}}) - \Lambda_{sk}(\mathbf{c}) &= \mathbf{s}^\top(\mathbf{A}\mathbf{k}) + e' - (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{k} = e' - \mathbf{e}^\top\mathbf{k}, \\ \mathsf{prPriv}(sk, \mathbf{c}) - \Lambda_{sk}(\mathbf{c}) &= \mathbf{c}^\top\mathbf{k} + e' - \mathbf{c}^\top\mathbf{k} = e', \end{aligned}$$

where $e' \leftarrow_\$ [-B', B']$. Since $|e'| \leq B'$, $\|\mathbf{e}\|_\infty \leq B$ and $\|\mathbf{k}\|_\infty \leq 1$, it follows that $|e' - \mathbf{e}^\top\mathbf{k}| \leq B' + mB$. Thus, $\mathsf{prPub}(pk_\rho, \mathbf{c}, w_{\mathbf{c}})$ always lies in $\mathsf{Ball}_{\varepsilon_{\mathsf{prPub}}}\big(\Lambda_{sk}(\mathbf{c})\big)$ with $\varepsilon_{\mathsf{prPub}} = B' + mB$ and $\mathsf{prPriv}(sk, \mathbf{c})$ lies in $\mathsf{Ball}_{\varepsilon_{\mathsf{prPriv}}}\big(\Lambda_{sk}(\mathbf{c})\big)$ with $\varepsilon_{\mathsf{prPriv}} = B'$.

Next, we evaluate the statistical distance between the probabilistic public evaluation and private evaluation for instances in $\widetilde{\mathcal{L}}_\rho = \widetilde{\mathcal{L}}_{\mathbf{A}}$. For any (fixed) $sk = \mathbf{k} \in \{0,1\}^m$, $pk_\rho = \mathbf{p} = \mathbf{A}\mathbf{k}$ and $\mathbf{c} = (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)^\top \in \widetilde{\mathcal{L}}_{\mathbf{A}}$ with witness $w_{\mathbf{c}} = (\mathbf{s}, \mathbf{e} \in [-\tilde{B}, \tilde{B}]^m)$, we have

$$\Delta(\mathsf{prPub}(pk_\rho, \mathbf{c}, w_{\mathbf{c}}),\ \mathsf{prPriv}(sk, \mathbf{c})) = \Delta(\mathbf{s}^\top(\mathbf{A}\mathbf{k}) + e',\ (\mathbf{s}^\top\mathbf{A} + \mathbf{e}^\top)\mathbf{k} + e')$$

$$\overset{(*)}{=} \Delta(\cancel{\mathbf{s}^\top\mathbf{A}\mathbf{k}} + e',\ \cancel{\mathbf{s}^\top\mathbf{A}\mathbf{k}} + \mathbf{e}^\top\mathbf{k} + e') \overset{(**)}{\leq} m\tilde{B}/B',$$

where the probability is over $e' \leftarrow_\$ [-B', B']$. Here $(*)$ holds since $\mathbf{s}^\top\mathbf{A}\mathbf{k}$ is a common constant, and $(**)$ follows from the fact that $|\mathbf{e}^\top\mathbf{k}| \leq m\tilde{B}$ (due to $\|\mathbf{e}\|_\infty \leq \tilde{B}$ and $\|\mathbf{k}\|_\infty \leq 1$) and Lemma 6 (the Smudging Lemma). Therefore, $\mathsf{prQAHPS}_{\mathsf{LWE}}$ has $\varepsilon_{\mathsf{evalnd}}$-evaluation indistinguishability with $\varepsilon_{\mathsf{evalnd}} = m\tilde{B}/B'$. □

E.2 Proof of Theorem 5 ($\langle\mathscr{L}, \mathscr{L}_0\rangle$-Key-Switching of $\mathsf{prQAHPS}_{\mathsf{LWE}}$)

For any adversary $\mathcal{A}$, we aim to prove $\boldsymbol{\epsilon}^{\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{prQAHPS},\mathcal{A}} :=$

$$\begin{aligned} &\big|\Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho = \mathbf{A}, \rho_0 = \mathbf{A}_0, \alpha_\rho(sk) = \mathbf{A}\mathbf{k}, \boxed{\alpha_{\rho_0}(sk) = \mathbf{A}_0\mathbf{k}}) = 1] \\ &- \Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho = \mathbf{A}, \rho_0 = \mathbf{A}_0, \alpha_\rho(sk) = \mathbf{A}\mathbf{k}, \boxed{\alpha_{\rho_0}(sk') = \mathbf{A}_0\mathbf{k}'}) = 1]\big| \leq 2^{-\lambda}. \end{aligned} \tag{28}$$

where $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow_\$ \mathscr{L}$, $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow_\$ \mathscr{L}_0$, $sk = \mathbf{k} \leftarrow_\$ \{0,1\}^m$, $sk' = \mathbf{k}' \leftarrow_\$ \{0,1\}^m$.

Let p be any prime factor of q. Since $\mathbf{k}$ and $\mathbf{k}'$ are chosen uniformly at random from $\{0,1\}^m$, we have $\widetilde{\mathbf{H}}_\infty(\mathbf{k} \bmod p) = \widetilde{\mathbf{H}}_\infty(\mathbf{k}' \bmod p) = m$. Note that $\mathbf{A}\mathbf{k} \in \mathbb{Z}_q^n$ leaks at most $n \log q$ bits of information about $\mathbf{k}$, but leaks nothing about $\mathbf{k}'$. Thus, according to Lemma 1, we have

$$\widetilde{\mathbf{H}}_\infty(\mathbf{k} \bmod p \,|\, \mathbf{A}\mathbf{k}) \geq m - n\log q, \quad \widetilde{\mathbf{H}}_\infty(\mathbf{k}' \bmod p \,|\, \mathbf{A}\mathbf{k}) = m.$$

According to Lemma 2, we know that uniform matrix $\mathbf{A}_0$ is a good extractor. Concretely, by applying Lemma 2 with $\epsilon = 2^{-(\lambda+1)}$ and by the condition $m > 3n\log q + 2(\lambda + 1)$, we have

$$\Delta((\mathbf{A}_0, \mathbf{A}_0\mathbf{k}), (\mathbf{A}_0, \mathbf{u}) \mid \mathbf{A}\mathbf{k}) \leq 2^{-(\lambda+1)}, \quad \Delta((\mathbf{A}_0, \mathbf{A}_0\mathbf{k}'), (\mathbf{A}_0, \mathbf{u}) \mid \mathbf{A}\mathbf{k}) \leq 2^{-(\lambda+1)},$$

where $\mathbf{u}$ is uniformly chosen from $\mathbb{Z}_q^n$. Then by the triangle inequality, we have

$$\Delta((\mathbf{A}_0, \boxed{\mathbf{A}_0\mathbf{k}}), (\mathbf{A}_0, \boxed{\mathbf{A}_0\mathbf{k}'}) \mid \mathbf{A}\mathbf{k}) \leq 2^{-\lambda}. \tag{29}$$

Finally, (28) follows from (29) by noting that $\mathbf{A}$ is independent of $\mathbf{A}_0$, $\mathbf{k}$, $\mathbf{k}'$. □

E.3 Proof of Theorem 6 (PK-Diversity of $\mathsf{prQAHPS}_{\mathsf{LWE}}$)

For $(\rho = \mathbf{A}, td_\rho = \mathbf{T_A}) \leftarrow_\$ \mathscr{L}$, $\mathbf{k}, \mathbf{k}' \leftarrow_\$ \{0,1\}^m$, we have

$$\begin{aligned}\epsilon^{\mathsf{pk\text{-}div}}_{\mathsf{prQAHPS}} &:= \Pr[\alpha_\rho(sk) = \mathbf{Ak} = \mathbf{Ak}' = \alpha_\rho(sk')] \\ &\leq \Pr[\mathbf{k}' = \mathbf{k}] + \Pr[\mathbf{Ak} = \mathbf{Ak}' \mid \mathbf{k}' \neq \mathbf{k}] = 2^{-m} + q^{-n},\end{aligned}$$

where the last equality is explained below.

- The uniformity of $\mathbf{k}$ and $\mathbf{k}'$ over $\{0,1\}^m$ implies that $\Pr[\mathbf{k}' = \mathbf{k}] = 2^{-m}$.
- Parse $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m)$ with each $\mathbf{a}_i \in \mathbb{Z}_q^n$, and parse $(\mathbf{k} - \mathbf{k}') = (b_1, b_2, \ldots, b_m)^\top \in \{-1,0,1\}^m$. Note that the condition $\mathbf{k}' \neq \mathbf{k}$ means $b_j \in \{-1,1\}$ for some $j \in [m]$, and the event $\mathbf{Ak} = \mathbf{Ak}'$ means $\sum_{i=1}^m b_i \cdot \mathbf{a}_i = \mathbf{0}$. By the uniformity of $\mathbf{a}_j$ over $\mathbb{Z}_q^n$ and by the condition that $b_j \in \{-1,1\}$, it follows that $\sum_{i=1}^m b_i \cdot \mathbf{a}_i = b_j \cdot \mathbf{a}_j + \sum_{i=1, i\neq j}^m b_i \cdot \mathbf{a}_i$ is uniformly distributed over $\mathbb{Z}_q^n$. Thus, the probability that $\sum_{i=1}^m b_i \cdot \mathbf{a}_i = \mathbf{0}$ conditioned on $b_j \in \{-1,1\}$ is exactly q^{-n}, and consequently, we get $\Pr[\mathbf{Ak} = \mathbf{Ak}' \mid \mathbf{k}' \neq \mathbf{k}] = q^{-n}$. □

E.4 Proof of Theorem 7 (Almost Tight $\mathscr{L}_0$-Multi-Key-Multi-Extracting of $\mathsf{prQAHPS}_{\mathsf{LWE}}$)

We prove the theorem by defining a sequence of distributions $\mathsf{D}_0 - \mathsf{D}_5$ and showing adjacent distributions indistinguishable.

Let $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_\$ \mathsf{Setup}_{\mathsf{HPS}}$, $(\rho_0 = \mathbf{A}_0, td_{\rho_0} = \mathbf{T}_{\mathbf{A}_0}) \leftarrow_\$ \mathscr{L}_0$, $sk_i = \mathbf{k}_i \leftarrow_\$ \{0,1\}^m$ for all $i \in [N]$, and $\mathbf{c}_{i,j} \leftarrow_\$ \mathcal{L}_{\rho_0}$ with $\mathbf{c}_{i,j}^\top = \mathbf{s}_{i,j}^\top \mathbf{A}_0 + \mathbf{e}_{i,j}^\top$ for all $i \in [N]$ and $j \in [Q]$. The distributions are defined as follows, where the differences are highlighted.

- $\mathsf{D}_0 := (\mathsf{pp}_{\mathsf{HPS}}, \mathbf{A}_0, \{\mathbf{c}_{i,j}, \boxed{hv_{i,j} = \mathsf{prPriv}(sk_i, \mathbf{c}_{i,j}) = (\mathbf{s}_{i,j}^\top \mathbf{A}_0 + \mathbf{e}_{i,j}^\top)\mathbf{k}_i + e'_{i,j}}\}_{i\in[N], j\in[Q]})$,
  where $e'_{i,j} \leftarrow_\$ [-B', B']$ for all $i \in [N], j \in [Q]$.
- $\mathsf{D}_1 := (\mathsf{pp}_{\mathsf{HPS}}, \mathbf{A}_0, \{\mathbf{c}_{i,j}, hv_{i,j} = \mathbf{s}_{i,j}^\top \mathbf{A}_0 \mathbf{k}_i + \cancel{\mathbf{e}_{i,j}^\top \mathbf{k}_i} + e'_{i,j}\}_{i\in[N], j\in[Q]})$.
- $\mathsf{D}_2 := (\mathsf{pp}_{\mathsf{HPS}}, \mathbf{A}_0, \{\mathbf{c}_{i,j}, hv_{i,j} = \mathbf{s}_{i,j}^\top \mathbf{A}_0 \mathbf{k}_i + \widetilde{e}_{i,j} + e'_{i,j}\}_{i\in[N], j\in[Q]})$,
  where $\widetilde{e}_{i,j} \leftarrow_\$ \chi$ for all $i \in [N], j \in [Q]$.
- $\mathsf{D}_3 := (\mathsf{pp}_{\mathsf{HPS}}, \mathbf{A}_0, \{\mathbf{c}_{i,j}, hv_{i,j} = \mathbf{s}_{i,j}^\top \mathbf{b}_i + \widetilde{e}_{i,j} + e'_{i,j}\}_{i\in[N], j\in[Q]})$,
  where $\mathbf{b}_i \leftarrow_\$ \mathbb{Z}_q^n$ for all $i \in [N]$.
- $\mathsf{D}_4 := (\mathsf{pp}_{\mathsf{HPS}}, \mathbf{A}_0, \{ \mathbf{c}_{i,j} \leftarrow_\$ \mathbb{Z}_q^m,\ hv_{i,j} \leftarrow_\$ \mathbb{Z}_q \}_{i\in[N], j\in[Q]})$.
- $\mathsf{D}_5 := (\mathsf{pp}_{\mathsf{HPS}}, \mathbf{A}_0, \{ \mathbf{c}_{i,j} \leftarrow_\$ \mathcal{L}_{\rho_0}, \boxed{hv_{i,j} \leftarrow_\$ \mathbb{Z}_q}\}_{i\in[N], j\in[Q]})$.

By definition, $\mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{prQAHPS},\mathcal{A},n,Q}(\lambda) = \big| \Pr[\mathcal{A}(\mathsf{D}_0) = 1] - \Pr[\mathcal{A}(\mathsf{D}_5) = 1] \big|$.

We prove adjacent distributions indistinguishable via the following claims.

Claim 14. $\big| \Pr[\mathcal{A}(\mathsf{D}_0) = 1] - \Pr[\mathcal{A}(\mathsf{D}_1) = 1] \big| \leq \Delta(\mathsf{D}_0, \mathsf{D}_1) \leq NQ \cdot mB/B'$.

Proof. Let us first fix all random variables except $e'_{i,j} \leftarrow_{\$} [-B', B']$ for all $i \in [N], j \in [Q]$, and analyze $\Delta(\mathsf{D}_0, \mathsf{D}_1)$:

$$\begin{aligned}
\Delta(\mathsf{D}_0, \mathsf{D}_1) &= \Delta(\{\mathbf{e}_{i,j}^\top \mathbf{k}_i + e'_{i,j}\}_{i\in[N],j\in[Q]},\ \{e'_{i,j}\}_{i\in[N],j\in[Q]}) && (30)\\
&\le \textstyle\sum_{i\in[N],j\in[Q]} \Delta(\mathbf{e}_{i,j}^\top \mathbf{k}_i + e'_{i,j},\ e'_{i,j}) && (31)\\
&\le NQ \cdot mB/B', && (32)
\end{aligned}$$

where (30) holds since all other terms in D_0 and D_1 are fixed values and are identical in D_0 and D_1, (31) follows from a hybrid argument, and (32) follows from the fact that $|\mathbf{e}_{i,j}^\top \mathbf{k}_i| \le mB$ (due to $\|\mathbf{e}_{i,j}\|_\infty \le B$ and $\|\mathbf{k}_i\|_\infty \le 1$), $e'_{i,j} \leftarrow_{\$} [-B', B']$ and Lemma 6 (the Smudging Lemma).

Then by an averaging argument over all random variables, we still have $\Delta(\mathsf{D}_0, \mathsf{D}_1) \le NQ \cdot mB/B'$. ∎

Claim 15. $\big|\Pr[\mathcal{A}(\mathsf{D}_1) = 1] - \Pr[\mathcal{A}(\mathsf{D}_2) = 1]\big| \le \Delta(\mathsf{D}_1, \mathsf{D}_2) \le NQ \cdot B/B'$.

Proof. The proof is similar to that of Claim 14. Firstly, let us fix all random variables except $e'_{i,j} \leftarrow_{\$} [-B', B']$ for all $i \in [N], j \in [Q]$ and analyze $\Delta(\mathsf{D}_1, \mathsf{D}_2)$:

$$\begin{aligned}
\Delta(\mathsf{D}_1, \mathsf{D}_2) &= \Delta(\{e'_{i,j}\}_{i\in[N],j\in[Q]},\ \{\widetilde{e}_{i,j} + e'_{i,j}\}_{i\in[N],j\in[Q]})\\
&\le \textstyle\sum_{i\in[N],j\in[Q]} \Delta(e'_{i,j},\ \widetilde{e}_{i,j} + e'_{i,j}) \le NQ \cdot B/B',
\end{aligned}$$

which follows from similar arguments as those in the proof of Claim 14 (with one difference that $|\widetilde{e}_{i,j}| \le B$ for any fixed $\widetilde{e}_{i,j} \leftarrow_{\$} \chi$).

Then Claim 15 follows from an averaging argument. ∎

Claim 16. $\big|\Pr[\mathcal{A}(\mathsf{D}_2) = 1] - \Pr[\mathcal{A}(\mathsf{D}_3) = 1]\big| \le \Delta(\mathsf{D}_2, \mathsf{D}_3) \le N \cdot 2^{-\lambda}$.

Proof. Let p be any prime factor of q. Since each $\mathbf{k}_i$ is chosen uniformly at random from $\{0,1\}^m$, we have $\widetilde{\mathbf{H}}_\infty(\mathbf{k}_i \bmod p) = m > 2n \log q + 2\lambda$. According to Lemma 2, uniform matrix $\mathbf{A}_0$ is a good extractor. Then it follows that

$$\begin{aligned}
\Delta(\mathsf{D}_2, \mathsf{D}_3) &\le \Delta((\mathbf{A}_0, \{\mathbf{A}_0\mathbf{k}_i\}_{i\in[N]}),\ (\mathbf{A}_0, \{\mathbf{b}_i\}_{i\in[N]})) && (33)\\
&\le \textstyle\sum_{i\in[N]} \Delta((\mathbf{A}_0, \mathbf{A}_0\mathbf{k}_i),\ (\mathbf{A}_0, \mathbf{b}_i)) && (34)\\
&\le N \cdot 2^{-\lambda}, && (35)
\end{aligned}$$

where (33) holds since D_2 (resp., D_3) can be constructed from $(\mathbf{A}_0, \{\mathbf{A}_0\mathbf{k}_i\}_{i\in[N]})$ (resp., $(\mathbf{A}_0, \{\mathbf{b}_i\}_{i\in[N]})$) along with $\{\mathbf{s}_{i,j}^\top, \mathbf{e}_{i,j}^\top, \widetilde{e}_{i,j}, e'_{i,j}\}_{i\in[N],j\in[Q]}$, (34) follows from a hybrid argument, and (35) holds by applying Lemma 2 with $\epsilon = 2^{-\lambda}$. ∎

Claim 17. $\big|\Pr[\mathcal{A}(\mathsf{D}_3) = 1] - \Pr[\mathcal{A}(\mathsf{D}_4) = 1]\big| \le 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}_1}(\lambda) + \frac{NQ(m+c+2)}{2^\lambda}$.

Proof. In D_3, for all $i \in [N]$ and $j \in [Q]$, we have $\mathbf{c}_{i,j}^\top = \mathbf{s}_{i,j}^\top \mathbf{A}_0 + \mathbf{e}_{i,j}^\top$ and $hv_{i,j} = \mathbf{s}_{i,j}^\top \mathbf{b}_i + \widetilde{e}_{i,j} + e'_{i,j}$. For the ease of our analysis, for each $i \in [N]$, we use the following notations:

$$\mathbf{C}_i := \begin{pmatrix} \mathbf{c}_{i,1}^\top \\ \mathbf{c}_{i,2}^\top \\ \vdots \\ \mathbf{c}_{i,Q}^\top \end{pmatrix} \in \mathbb{Z}_q^{Q\times m}, \quad \mathbf{S}_i := \begin{pmatrix} \mathbf{s}_{i,1}^\top \\ \mathbf{s}_{i,2}^\top \\ \vdots \\ \mathbf{s}_{i,Q}^\top \end{pmatrix} \in \mathbb{Z}_q^{Q\times n}, \quad \mathbf{E}_i := \begin{pmatrix} \mathbf{e}_{i,1}^\top \\ \mathbf{e}_{i,2}^\top \\ \vdots \\ \mathbf{e}_{i,Q}^\top \end{pmatrix} \in \mathbb{Z}_q^{Q\times m},$$

$$\mathbf{hv}_i := \begin{pmatrix} hv_{i,1} \\ hv_{i,2} \\ \vdots \\ hv_{i,Q} \end{pmatrix} \in \mathbb{Z}_q^{Q}, \quad \widetilde{\mathbf{e}}_i := \begin{pmatrix} \widetilde{e}_{i,1} \\ \widetilde{e}_{i,2} \\ \vdots \\ \widetilde{e}_{i,Q} \end{pmatrix} \in \mathbb{Z}_q^{Q}, \quad \mathbf{e}'_i := \begin{pmatrix} e'_{i,1} \\ e'_{i,2} \\ \vdots \\ e'_{i,Q} \end{pmatrix} \in \mathbb{Z}_q^{Q}.$$

Then for all $i \in [N]$, we have $\mathbf{C}_i = \mathbf{S}_i \mathbf{A}_0 + \mathbf{E}_i$ and $\mathbf{hv}_i = \mathbf{S}_i \mathbf{b}_i + \widetilde{\mathbf{e}}_i + \mathbf{e}'_i$, i.e.,

$$(\mathbf{hv}_i | \mathbf{C}_i) = \mathbf{S}_i(\mathbf{b}_i | \mathbf{A}_0) + (\widetilde{\mathbf{e}}_i | \mathbf{E}_i) + (\mathbf{e}'_i | \mathbf{0}),$$

where $\mathbf{S}_i \leftarrow_\$ \mathbb{Z}_q^{Q\times n}$, $\mathbf{b}_i \leftarrow_\$ \mathbb{Z}_q^n$, $\mathbf{E}_i \leftarrow_\$ \chi^{Q\times m}$, $\widetilde{\mathbf{e}}_i \leftarrow_\$ \chi^Q$ and $\mathbf{e}'_i \leftarrow_\$ [-B', B']^Q$.

In D_4, for all $i \in [N]$ and $j \in [Q]$, we have $\mathbf{c}_{i,j} \leftarrow_\$ \mathbb{Z}_q^m$ and $hv_{i,j} \leftarrow_\$ \mathbb{Z}_q$. By using the above notations, for all $i \in [N]$, we have

$$(\mathbf{hv}_i | \mathbf{C}_i) \leftarrow_\$ \mathbb{Z}_q^{Q\times(m+1)}.$$

Therefore, it suffices to show

$$\begin{aligned} &\mathsf{D}_3: \ (\mathbf{A}_0, \{\mathbf{S}_i(\mathbf{b}_i|\mathbf{A}_0) + (\widetilde{\mathbf{e}}_i|\mathbf{E}_i)\}_{i\in[N]}) \\ \overset{c}{\approx}\ &\mathsf{D}_4: \ (\mathbf{A}_0, \{\ \mathbf{U}_i\ \}_{i\in[N]}), \end{aligned} \tag{36}$$

where $\mathbf{A}_0 \leftarrow_\$ \mathbb{Z}_q^{n\times m}$, and $\mathbf{S}_i \leftarrow_\$ \mathbb{Z}_q^{Q\times n}$, $\mathbf{b}_i \leftarrow_\$ \mathbb{Z}_q^n$, $(\widetilde{\mathbf{e}}_i|\mathbf{E}_i) \leftarrow_\$ \chi^{Q\times(m+1)} = D_{\mathbb{Z},\sigma}^{Q\times(m+1)}$ and $\mathbf{U}_i \leftarrow_\$ \mathbb{Z}_q^{Q\times(m+1)}$ for each $i \in [N]$. We will prove (36) based on the $\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\gamma},m}$-assumption.

Firstly, we note that if all $\mathbf{b}_i$'s are the same, i.e., $\mathbf{b}_1 = \mathbf{b}_2 = \cdots = \mathbf{b}_N \leftarrow_\$ \mathbb{Z}_q^n$, then the problem of distinguishing (36) is just the (NQ)-$\mathsf{LWE}_{n,q,\chi,m+1}$ problem. Since we set $\chi = D_{\mathbb{Z},\sigma}$, by the almost tight reduction from LWE to multi-secret LWE (Theorem 3), we know that $\big|\Pr[\mathcal{A}(\mathsf{D}_3) = 1] - \Pr[\mathcal{A}(\mathsf{D}_4) = 1]\big| \leq \mathsf{Adv}^{NQ\text{-}\mathsf{LWE}}_{[n,q,\chi,m+1],\mathcal{B}'_1}(\lambda) \leq 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}_1}(\lambda) + \frac{NQ(m+c+2)}{2^\lambda}$, and Claim 17 follows.

However, $\mathbf{b}_1, \cdots, \mathbf{b}_N$ in (36) are independently chosen, so the problem of distinguishing (36) is not exactly the same as (but very close to) the (NQ)-$\mathsf{LWE}_{n,q,\chi,m+1}$ problem. Nevertheless, for the problem of distinguishing (36), we can basically use the same techniques as in the proof of Theorem 3 to show that $\big|\Pr[\mathcal{A}(\mathsf{D}_3) = 1] - \Pr[\mathcal{A}(\mathsf{D}_4) = 1]\big| \leq 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}_1}(\lambda) + \frac{NQ(m+c+2)}{2^\lambda}$. Below we give a proof sketch.

Similar to the proof of Theorem 3 , we first introduce an intermediate problem of distinguishing the following D_3' and D_4' with errors sampled according to the continuous Gaussian D_{σ_0} (recall that $\sigma = \sqrt{{\sigma_0}^2 + r^2}$ for $r \geq \sqrt{\lambda}$):

$$\begin{aligned} & \mathsf{D}_3' : \ (\mathbf{A}_0, \{\mathbf{S}_i(\mathbf{b}_i|\mathbf{A}_0) + (\widetilde{\mathbf{e}}_i|\mathbf{E}_i)\}_{i\in[N]}) \\ \overset{c}{\approx} \ & \mathsf{D}_4' : \ (\mathbf{A}_0, \{ \mathbf{U}_i + (\widetilde{\mathbf{e}}_i|\mathbf{E}_i)\}_{i\in[N]}), \end{aligned} \tag{37}$$

where $\mathbf{A}_0 \leftarrow_{\$} \mathbb{Z}_q^{n\times m}$, and $\mathbf{S}_i \leftarrow_{\$} \mathbb{Z}_q^{Q\times n}$, $\mathbf{b}_i \leftarrow_{\$} \mathbb{Z}_q^n$, $(\widetilde{\mathbf{e}}_i|\mathbf{E}_i) \leftarrow_{\$} D_{\sigma_0}^{Q\times(m+1)}$ and $\mathbf{U}_i \leftarrow_{\$} \mathbb{Z}_q^{Q\times(m+1)}$ for each $i \in [N]$. Then we will prove the claim by showing that there exists an adversary $\mathcal{B}_1'$ such that $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_1') + NQ \cdot \mathsf{poly}'(\lambda) \approx \mathbf{T}(\mathcal{A}) + NQ \cdot \mathsf{poly}(\lambda)$ and

$$\big|\Pr[\mathcal{A}(\mathsf{D}_3) = 1] - \Pr[\mathcal{A}(\mathsf{D}_4) = 1]\big| \leq \big|\Pr[\mathcal{B}_1'(\mathsf{D}_3') = 1] - \Pr[\mathcal{B}_1'(\mathsf{D}_4') = 1]\big| + \tfrac{NQm}{2^\lambda}, \tag{38}$$

$$\big|\Pr[\mathcal{B}_1'(\mathsf{D}_3') = 1] - \Pr[\mathcal{B}_1'(\mathsf{D}_4') = 1]\big| \leq 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}_1}(\lambda) + \tfrac{NQ(c+2)}{2^\lambda}. \tag{39}$$

The proof of (38) is almost identical to that of (9) in the proof of Theorem 3, by using the randomized rounding technique due to Peikert [42] (i.e., Lemma 8), thus we omit it here.

Next we turn to the proof of (39). That is, we aim to prove (37) based on the $\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\gamma},m}$-assumption, and determine the security loss factor. Its proof is almost identical to that of (10) in the proof of Theorem 3, with only the first step being slightly different, as shown below.

In the first step, we break $\mathbf{A}_0 \in \mathbb{Z}_q^{n\times m}$ into $(\mathbf{A}_{0,1}|\bar{\mathbf{A}}_{0,1}) \in \mathbb{Z}_q^{n\times m'} \times \mathbb{Z}_q^{n\times(m-m')}$ and $\mathbf{E}_i \in D_{\sigma_0}^{Q\times m}$ into $(\mathbf{E}_{i,1}|\bar{\mathbf{E}}_{i,1}) \in D_{\sigma_0}^{Q\times m'} \times D_{\sigma_0}^{Q\times(m-m')}$ for each $i \in [N]$, where the block $\mathbf{A}_{0,1}$ contains the first m' columns of $\mathbf{A}_0$. Then we change $\bar{\mathbf{A}}_{0,1}$ into a lossy one $\tilde{\mathbf{A}}_{0,1} = \mathbf{CB} + \mathbf{F}$, where $\mathbf{C} \leftarrow_{\$} \mathbb{Z}_q^{n\times\ell}, \mathbf{B} \leftarrow_{\$} \mathbb{Z}_q^{\ell\times(m-m')}$ and $\mathbf{F} \in \mathbb{Z}_q^{n\times(m-m')}$ follows the error distribution $D_{\mathbb{Z},\gamma}^{n\times(m-m')}$. This change is indistinguishable due to the n-secret $\mathsf{LWE}_{\ell,q,D_{\mathbb{Z},\gamma},m-m'}$-assumption. Therefore,

$$\begin{aligned} & \mathsf{D}_3' : \ \Big(\mathbf{A}_0, \Big\{\mathbf{S}_i(\mathbf{b}_i|\mathbf{A}_0) + (\widetilde{\mathbf{e}}_i|\mathbf{E}_i)\Big\}_{i\in[N]}\Big) \\ = & \bigg((\mathbf{A}_{0,1}|\bar{\mathbf{A}}_{0,1}), \Big\{ (\mathbf{S}_i(\mathbf{b}_i|\mathbf{A}_{0,1}) + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\bar{\mathbf{A}}_{0,1} + \bar{\mathbf{E}}_{i,1})\Big\}_{i\in[N]}\bigg) \\ \overset{c}{\approx} & \bigg((\mathbf{A}_{0,1}|\tilde{\mathbf{A}}_{0,1}), \Big\{ (\mathbf{S}_i(\mathbf{b}_i|\mathbf{A}_{0,1}) + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\tilde{\mathbf{A}}_{0,1} + \bar{\mathbf{E}}_{i,1})\Big\}_{i\in[N]}\bigg) \end{aligned}$$

but it incurs a loss factor of n since hybrid arguments yield $\mathsf{Adv}^{n\text{-}\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m-m']}(\lambda) \leq n \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m-m']}(\lambda) \leq n \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m]}(\lambda)$. Now given a lossy $\tilde{\mathbf{A}}_{0,1}$, for each $i \in [N]$, the information of $\mathbf{S}_i$ leaked by $\mathbf{S}_i\tilde{\mathbf{A}}_{0,1}$ is bounded. Then for each $i \in [N]$, since $(\mathbf{b}_i|\mathbf{A}_{0,1})$ is uniformly distributed over $\mathbb{Z}_q^{n\times(m'+1)}$, by taking it as extractor, we can extract the remaining entropy of $\mathbf{S}_i$ to obtain $\mathbf{S}_i(\mathbf{b}_i|\mathbf{A}_{0,1}) \overset{s}{\approx} \mathbf{U}_{i,1}$, where $\mathbf{U}_{i,1} \leftarrow_{\$} \mathbb{Z}_q^{Q\times(m'+1)}$. So we have

$$\begin{aligned}
&\left((\mathbf{A}_{0,1}|\tilde{\mathbf{A}}_{0,1}), \left\{ (\mathbf{S}_i(\mathbf{b}_i|\mathbf{A}_{0,1}) + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\tilde{\mathbf{A}}_{0,1} + \bar{\mathbf{E}}_{i,1})\right\}_{i\in[N]}\right)\\
\overset{s}{\approx}\; &\left((\mathbf{A}_{0,1}|\tilde{\mathbf{A}}_{0,1}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\tilde{\mathbf{A}}_{0,1} + \bar{\mathbf{E}}_{i,1})\right\}_{i\in[N]}\right).
\end{aligned}$$

Next, we change the lossy $\tilde{\mathbf{A}}_{0,1}$ back to uniform $\bar{\mathbf{A}}_{0,1}$, and have

$$\begin{aligned}
&\left((\mathbf{A}_{0,1}|\tilde{\mathbf{A}}_{0,1}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\tilde{\mathbf{A}}_{0,1} + \bar{\mathbf{E}}_{i,1})\right\}_{i\in[N]}\right)\\
\overset{c}{\approx}\; &\left((\mathbf{A}_{0,1}|\bar{\mathbf{A}}_{0,1}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\bar{\mathbf{A}}_{0,1} + \bar{\mathbf{E}}_{i,1})\right\}_{i\in[N]}\right).
\end{aligned}$$

Then we have loss factor n again.

In the second step, we break $\mathbf{A}_0 = (\mathbf{A}_{0,1}|\bar{\mathbf{A}}_{0,1})$ further into $(\mathbf{A}_{0,1}|\mathbf{A}_{0,2}|\bar{\mathbf{A}}_{0,2}) \in \mathbb{Z}_q^{n\times m'} \times \mathbb{Z}_q^{n\times m'} \times \mathbb{Z}_q^{n\times(m-2m')}$ and $\mathbf{E}_i = (\mathbf{E}_{i,1}|\bar{\mathbf{E}}_{i,1})$ into $(\mathbf{E}_{i,1}|\mathbf{E}_{i,2}|\bar{\mathbf{E}}_{i,2}) \in D_{\sigma_0}^{Q\times m'} \times D_{\sigma_0}^{Q\times m'} \times D_{\sigma_0}^{Q\times(m-2m')}$ for each $i \in [N]$, where the block $\mathbf{A}_{0,2}$ contains the second m' columns of $\mathbf{A}_0$. Then we change $\bar{\mathbf{A}}_{0,2}$ to a lossy one $\tilde{\mathbf{A}}_{0,2}$ and have

$$\begin{aligned}
&\left((\mathbf{A}_{0,1}|\bar{\mathbf{A}}_{0,1}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\bar{\mathbf{A}}_{0,1} + \bar{\mathbf{E}}_{i,1})\right\}_{i\in[N]}\right)\\
=\; &\left((\mathbf{A}_{0,1}|\mathbf{A}_{0,2}|\bar{\mathbf{A}}_{0,2}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\mathbf{A}_{0,2} + \mathbf{E}_{i,2})\Big|(\mathbf{S}_i\bar{\mathbf{A}}_{0,2} + \bar{\mathbf{E}}_{i,2})\right\}_{i\in[N]}\right)\\
\overset{c}{\approx}\; &\left((\mathbf{A}_{0,1}|\mathbf{A}_{0,2}|\tilde{\mathbf{A}}_{0,2}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\mathbf{A}_{0,2} + \mathbf{E}_{i,2})\Big|(\mathbf{S}_i\tilde{\mathbf{A}}_{0,2} + \bar{\mathbf{E}}_{i,2})\right\}_{i\in[N]}\right)
\end{aligned}$$

with a lossy factor n. With a similar argument, the uniform $\mathbf{A}_{0,2}$ can extract the remaining entropy of $\mathbf{S}_i$ for each $i \in [N]$ so that $\mathbf{S}_i\mathbf{A}_{0,2} \overset{s}{\approx} \mathbf{U}_{i,2}$, where $\mathbf{U}_{i,2} \leftarrow_{\$} \mathbb{Z}_q^{Q\times m'}$. So

$$\begin{aligned}
&\left((\mathbf{A}_{0,1}|\mathbf{A}_{0,2}|\tilde{\mathbf{A}}_{0,2}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{S}_i\mathbf{A}_{0,2} + \mathbf{E}_{i,2})\Big|(\mathbf{S}_i\tilde{\mathbf{A}}_{0,2} + \bar{\mathbf{E}}_{i,2})\right\}_{i\in[N]}\right)\\
\overset{s}{\approx}\; &\left((\mathbf{A}_{0,1}|\mathbf{A}_{0,2}|\tilde{\mathbf{A}}_{0,2}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{U}_{i,2} + \mathbf{E}_{i,2})\Big|(\mathbf{S}_i\tilde{\mathbf{A}}_{0,2} + \bar{\mathbf{E}}_{i,2})\right\}_{i\in[N]}\right).
\end{aligned}$$

Changing lossy $\tilde{\mathbf{A}}_{0,2}$ back to uniform $\bar{\mathbf{A}}_{0,2}$ yields

$$\begin{aligned}
&\left((\mathbf{A}_{0,1}|\mathbf{A}_{0,2}|\tilde{\mathbf{A}}_{0,2}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{U}_{i,2} + \mathbf{E}_{i,2})\Big|(\mathbf{S}_i\tilde{\mathbf{A}}_{0,2} + \bar{\mathbf{E}}_{i,2})\right\}_{i\in[N]}\right)\\
\overset{c}{\approx}\; &\left((\mathbf{A}_{0,1}|\mathbf{A}_{0,2}|\bar{\mathbf{A}}_{0,2}), \left\{ (\mathbf{U}_{i,1} + (\widetilde{\mathbf{e}}_i|\mathbf{E}_{i,1}))\Big|(\mathbf{U}_{i,2} + \mathbf{E}_{i,2})\Big|(\mathbf{S}_i\bar{\mathbf{A}}_{0,2} + \bar{\mathbf{E}}_{i,2})\right\}_{i\in[N]}\right)
\end{aligned}$$

with a price of another loss factor n.

Overall, with at most $c \approx \frac{m}{m'}$ steps, we can prove (37) with a loss factor of $2cn$, and thus obtain (39).

Finally, taking (38) and (39) together, Claim 17 holds. ▌

Claim 18. $\big|\Pr[\mathcal{A}(\mathsf{D}_4) = 1] - \Pr[\mathcal{A}(\mathsf{D}_5) = 1]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}'_2,NQ}(\lambda) \leq 2cn \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\ell,q,D_{\mathbb{Z},\gamma},m],\mathcal{B}_2}(\lambda) + \frac{NQ(m+c+1)}{2^\lambda}$.

Proof. Note that all $hv_{i,j}$'s are uniformly chosen from $\mathbb{Z}_q$ both in D_4 and D_5. The only difference between D_4 and D_5 is the $\mathbf{c}_{i,j}$'s ($i \in [N]$, $j \in [Q]$), which are chosen from $\mathbb{Z}_q^m = \mathcal{X}$ in D_4 and from $\mathcal{L}_{\rho_0} = \mathcal{L}_{\mathbf{A}_0}$ in D_5. Thus, D_4 and D_5 are computationally indistinguishable by the multi-fold SMP for $\mathscr{L}_0$, and $\big|\Pr[\mathcal{A}(\mathsf{D}_4) = 1] - \Pr[\mathcal{A}(\mathsf{D}_5) = 1]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}'_2,NQ}(\lambda)$ for an adversary $\mathcal{B}'_2$. Then by Lemma 9 (since $\mathscr{L}_0$ is the distribution specified in Subsect. 6.1), Claim 18 follows. ▌

Finally, by taking Claims 14–18 together, Theorem 7 follows. □

E.5 Proof of Theorem 8 ($\varepsilon_{\mathsf{ext}}$-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-Extracting of $\mathsf{prQAHPS}_{\mathsf{LWE}}$)

By definition, we have $\boldsymbol{\epsilon}^{\varepsilon_{\mathsf{ext}}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS},\mathcal{A}} :=$

$$\Pr\left[(\mathbf{c}^*, hv^*) \leftarrow_{\$} \mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho_0 = \mathbf{A}_0, \rho = \mathbf{A}, \alpha_{\rho_0}(sk) = \mathbf{A}_0\mathbf{k}) : \begin{array}{c} \mathbf{c}^* \in \widetilde{\mathcal{L}}_{\mathbf{A}} \wedge \\ |hv^* - \Lambda_{sk}(\mathbf{c}^*)| \leq \varepsilon_{\mathsf{ext}} \end{array}\right],$$

where $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow_{\$} \mathscr{L}$, $(\mathbf{A}_0, \mathbf{T}_{\mathbf{A}_0}) \leftarrow_{\$} \mathscr{L}_0$ and $sk = \mathbf{k} \leftarrow_{\$} \{0,1\}^m$.

In the case $\mathbf{c}^* \notin \widetilde{\mathcal{L}}_{\mathbf{A}}$, $\boldsymbol{\epsilon}^{\varepsilon_{\mathsf{ext}}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS},\mathcal{A}} = 0$, then the theorem trivially holds. Next, we prove the theorem in the case $\mathbf{c}^* \in \widetilde{\mathcal{L}}_{\mathbf{A}}$. To this end, we first claim that in the view of $\mathcal{A}$, $\Lambda_{sk}(\mathbf{c}^*) + e'$ with $e' \leftarrow_{\$} [-B', B']$ is statistically close to the uniform distribution over $\mathbb{Z}_q$, i.e.,

$$\Delta(\Lambda_{sk}(\mathbf{c}^*) + e', u \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k})) \leq 2^{-\lambda} + m\tilde{B}/B', \tag{40}$$

where $u \leftarrow_{\$} \mathbb{Z}_q$. Assuming that the claim (40) holds, Theorem 8 follows due to

$$\begin{aligned} \boldsymbol{\epsilon}^{\varepsilon_{\mathsf{ext}}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{prQAHPS},\mathcal{A}} &\leq \Pr\left[(\mathbf{c}^*, hv^*) \leftarrow_{\$} \mathcal{A}(\cdots) \ : \ |hv^* - (\Lambda_{sk}(\mathbf{c}^*) + e')| \leq \varepsilon_{\mathsf{ext}} + B'\right] \\ &\leq 2^{-\lambda} + m\tilde{B}/B' + \Pr\left[(\mathbf{c}^*, hv^*) \leftarrow_{\$} \mathcal{A}(\cdots) \ : \ |hv^* - u| \leq \varepsilon_{\mathsf{ext}} + B'\right] \\ &= 2^{-\lambda} + m\tilde{B}/B' + \Pr\left[(\mathbf{c}^*, hv^*) \leftarrow_{\$} \mathcal{A}(\cdots) \ : \ u \in [hv^* - \varepsilon_{\mathsf{ext}} - B', hv^* + \varepsilon_{\mathsf{ext}} + B']\right] \\ &= 2^{-\lambda} + m\tilde{B}/B' + (2\varepsilon_{\mathsf{ext}} + 2B' + 1)/q. \end{aligned}$$

It remains to prove (40). Since $\mathbf{c}^* \in \widetilde{\mathcal{L}}_{\mathbf{A}}$, we can write $\mathbf{c}^* = (\mathbf{s}^{*\top}\mathbf{A} + \mathbf{e}^{*\top})^\top$ for some $\mathbf{s}^* \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}$ and $\mathbf{e}^* \in [-\tilde{B}, \tilde{B}]^m$. By the triangle inequality, we have

$$\begin{aligned} &\Delta(\Lambda_{sk}(\mathbf{c}^*) + e', u \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k})) = \Delta((\mathbf{s}^{*\top}\mathbf{A} + \mathbf{e}^{*\top})\mathbf{k} + e', u \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k})) \\ &\leq \Delta(\mathbf{s}^{*\top}\mathbf{A}\mathbf{k} + \mathbf{e}^{*\top}\mathbf{k} + e', \mathbf{s}^{*\top}\mathbf{A}\mathbf{k} + e' \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k})) \qquad (41) \\ &\quad + \Delta(\mathbf{s}^{*\top}\mathbf{A}\mathbf{k} + e', u \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k})). \qquad (42) \end{aligned}$$

Next, we analyze the two statistical distances in (41) and (42) separately. The analysis of the statistical distance in (41) is as follows

$$\begin{aligned}
&\mathit{\Delta}(\mathbf{s}^{*\top}\mathbf{A}\mathbf{k} + \mathbf{e}^{*\top}\mathbf{k} + e', \mathbf{s}^{*\top}\mathbf{A}\mathbf{k} + e' \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k}))\\
\leq\ &\mathit{\Delta}(\cancel{\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}} + \mathbf{e}^{*\top}\mathbf{k} + e', \cancel{\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}} + e' \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k}, \mathbf{s}^{*\top}\mathbf{A}\mathbf{k})) \qquad (43)\\
\leq\ &m\tilde{B}/B', \qquad (44)
\end{aligned}$$

where (43) holds since $\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}+\mathbf{e}^{*\top}\mathbf{k}+e'$ (resp., $\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}+e'$) can be constructed with $\mathbf{e}^{*\top}\mathbf{k}+e'$ (resp., e') and $\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}$, and (44) follows from the fact that $|\mathbf{e}^{*\top}\mathbf{k}| \leq m\tilde{B}$ (due to $\|\mathbf{e}^*\|_\infty \leq \tilde{B}$ and $\|\mathbf{k}\|_\infty \leq 1$) and Lemma 6 (the Smudging Lemma). The analysis of the statistical distance in (42) is as follows

$$\begin{aligned}
&\mathit{\Delta}(\mathbf{s}^{*\top}\mathbf{A}\mathbf{k} + e', u \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k}))\\
\leq\ &\mathit{\Delta}(\mathbf{s}^{*\top}\mathbf{A}\mathbf{k} + \cancel{e'}, u - e' \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k}, e')) \qquad (45)\\
=\ &\mathit{\Delta}(\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}, u - \cancel{e'} \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k}, \cancel{e'})) \qquad (46)\\
=\ &\mathit{\Delta}(\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}, \mathbf{s}^{*\top}\mathbf{u} \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k})) \qquad (47)\\
\leq\ &\mathit{\Delta}(\mathbf{A}\mathbf{k}, \mathbf{u} \mid (\mathbf{A}_0, \mathbf{A}, \mathbf{A}_0\mathbf{k})) \qquad (48)\\
\leq\ &2^{-\lambda}, \qquad (49)
\end{aligned}$$

where $\mathbf{u} \leftarrow_{\$} \mathbb{Z}_q^n$. Here (45) holds since $\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}+e'$ (resp., u) can be constructed with $\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}$ (resp., $u-e'$) and e', (46) holds since u is uniformly distributed over $\mathbb{Z}_q$ and e' is independent of other variables, (47) holds due to the uniformity of $\mathbf{u}$ and the fact that $\mathbf{s}^* \neq \mathbf{0}$, and (48) holds since $\mathbf{s}^{*\top}\mathbf{A}\mathbf{k}$ (resp., $\mathbf{s}^{*\top}\mathbf{u}$) can be constructed from $\mathbf{A}\mathbf{k}$ (resp., $\mathbf{u}$) along with $\mathbf{s}^*$. The justification of (49) is as follows. Let p be any prime factor of q. Since $\mathbf{k}$ is chosen uniformly at random from $\{0,1\}^m$, we have $\widetilde{\mathbf{H}}_\infty(\mathbf{k} \bmod p) = m$. Note that $\mathbf{A}_0\mathbf{k} \in \mathbb{Z}_q^n$ leaks at most $n\log q$ bits of information about $\mathbf{k}$. Thus, according to Lemma 1 and by the condition $m > 3n\log q + 2\lambda$, we have

$$\widetilde{\mathbf{H}}_\infty(\mathbf{k} \bmod p \mid (\mathbf{A}_0, \mathbf{A}_0\mathbf{k})) \geq m - n\log q > 2n\log q + 2\lambda.$$

According to Lemma 2, we know that uniform matrix $\mathbf{A}$ is a good extractor. Concretely, by applying Lemma 2 with $\epsilon = 2^{-\lambda}$, we have

$$\mathit{\Delta}((\mathbf{A}, \mathbf{A}\mathbf{k}), (\mathbf{A}, \mathbf{u}) \mid (\mathbf{A}_0, \mathbf{A}_0\mathbf{k})) \leq 2^{-\lambda}.$$

Thus (49) holds. Finally, by bounding the statistical distances in (41) and (42) with (44) and (49), we obtain (40) and complete the proof of Theorem 8. □

F Proof of Theorem 9 (Security of $\mathsf{Com}_{\mathsf{LWE}}$)

We prove the three security properties for $\mathsf{Com}_{\mathsf{LWE}}$ as follows.

Parameter Indistinguishability. The public parameter $\mathsf{pp}_{\mathsf{CMT}}$ in the binding mode is $\mathbf{X} := \binom{\overline{\mathbf{X}}}{\mathbf{b}^\top} = \binom{\overline{\mathbf{X}}}{\mathbf{s}^\top\overline{\mathbf{X}}+\mathbf{e}^\top}$. The public parameter $\mathsf{pp}_{\mathsf{CMT}}$ in the hiding mode is $\mathbf{X} \leftarrow_{\$} \mathbb{Z}_{q^2}^{(n+1)\times m}$, which can be equivalently set by $\mathbf{X} := \binom{\overline{\mathbf{X}}}{\mathbf{u}^\top}$ with $\overline{\mathbf{X}} \leftarrow_{\$} \mathbb{Z}_{q^2}^{n\times m}$ and $\mathbf{u} \leftarrow_{\$} \mathbb{Z}_{q^2}^{m}$. By the $\mathsf{LWE}_{n,q^2,\chi,m}$ assumption, $\binom{\overline{\mathbf{X}}}{\mathbf{s}^\top\overline{\mathbf{X}}+\mathbf{e}^\top}$ is computationally indistinguishable from $\binom{\overline{\mathbf{X}}}{\mathbf{u}^\top}$, thus we have $\mathsf{Adv}^{\mathsf{para\text{-}ind}}_{\mathsf{CMT},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{LWE}}_{[n,q^2,\chi,m],\mathcal{B}}(\lambda)$.

Statistical Binding for $\widetilde{\mathcal{M}}$ under BSetup. For any public parameter in the binding mode $\mathbf{X} = \binom{\overline{\mathbf{X}}}{\mathbf{s}^\top\overline{\mathbf{X}}+\mathbf{e}^\top}$, we show that it is impossible to have $\mathbf{m} \neq \mathbf{m}' \in \widetilde{\mathcal{M}}$ and $\mathbf{R}, \mathbf{R}' \in \widetilde{\mathcal{R}}$ such that $\mathsf{Com}(\mathbf{X}, \mathbf{m}; \mathbf{R}) = \mathsf{Com}(\mathbf{X}, \mathbf{m}'; \mathbf{R}')$, thus $\varepsilon_{\mathsf{binding}} = 0$.

Suppose towards a contradiction that there exist $\mathbf{m} \neq \mathbf{m}' \in \widetilde{\mathcal{M}}$ and $\mathbf{R}, \mathbf{R}' \in \widetilde{\mathcal{R}}$ such that $\mathsf{Com}(\mathbf{X}, \mathbf{m}; \mathbf{R}) = \mathbf{X}\cdot\mathbf{R} + \binom{\mathbf{0}}{q\cdot\mathbf{m}^\top} = \mathbf{X}\cdot\mathbf{R}' + \binom{\mathbf{0}}{q\cdot\mathbf{m}'^\top} = \mathsf{Com}(\mathbf{X}, \mathbf{m}'; \mathbf{R}')$. Then

$$\mathbf{X} \cdot (\mathbf{R}' - \mathbf{R}) = \binom{\mathbf{0}}{q\cdot(\mathbf{m}-\mathbf{m}')^\top}.$$

By multiplying $(-\mathbf{s}^\top, 1)$ to the both sides of the above equation, we obtain $\mathbf{e}^\top \cdot (\mathbf{R}' - \mathbf{R}) = q \cdot (\mathbf{m} - \mathbf{m}')^\top$, which further implies that

$$\left\|\mathbf{e}^\top \cdot (\mathbf{R}' - \mathbf{R})\right\|_\infty = \left\|q \cdot (\mathbf{m} - \mathbf{m}')^\top\right\|_\infty. \tag{50}$$

However, on the left-hand side of (50), $\|\mathbf{e}\|_\infty \leq B$ (since χ is B-bounded) and $\|\mathbf{R}' - \mathbf{R}\|_\infty \leq 2\widetilde{B}$ (since $\mathbf{R}, \mathbf{R}' \in \widetilde{\mathcal{R}} = [-\widetilde{B}, \widetilde{B}]^{m\times m}$), so $\left\|\mathbf{e}^\top \cdot (\mathbf{R}' - \mathbf{R})\right\|_\infty \leq 2mB\widetilde{B}$. On the right-hand side, $\left\|q \cdot (\mathbf{m} - \mathbf{m}')^\top\right\|_\infty = q \cdot \left\|(\mathbf{m} - \mathbf{m}')^\top\right\|_\infty \geq q$ (since $\mathbf{m} \neq \mathbf{m}' \in \widetilde{\mathcal{M}} = [-\widetilde{B}, \widetilde{B}]^m$). According to the condition $q > 2mB\widetilde{B}$, (50) is impossible to hold, which yields a contradiction.

Statistical Hiding for $\mathcal{M}$ under HSetup. Let $\mathbf{m}_0, \mathbf{m}_1$ be any pair of messages in $\mathcal{M} = \{0,1\}^m$. We aim to prove that

$$\varDelta\big((\mathbf{X}, \underbrace{\mathbf{X}\cdot\mathbf{R} + \tbinom{\mathbf{0}}{q\cdot\mathbf{m}_0^\top}}_{\mathsf{Com}(\mathbf{X},\mathbf{m}_0;\mathbf{R})}), (\mathbf{X}, \underbrace{\mathbf{X}\cdot\mathbf{R} + \tbinom{\mathbf{0}}{q\cdot\mathbf{m}_1^\top}}_{\mathsf{Com}(\mathbf{X},\mathbf{m}_1;\mathbf{R})})\big) \leq \varepsilon_{\mathsf{hiding}} = m \cdot 2^{-\lambda}, \tag{51}$$

where the probability is over $\mathbf{X} \leftarrow_{\$} \mathbb{Z}_{q^2}^{(n+1)\times m}$ (the public parameter in the hiding mode) and $\mathbf{R} \leftarrow_{\$} \mathcal{R} = \{0,1\}^{m\times m}$.

Let us parse $\mathbf{R} = \{0,1\}^{m\times m}$ as $\mathbf{R} = (\mathbf{r}_1, \ldots, \mathbf{r}_m)$ with each $\mathbf{r}_i \in \{0,1\}^m$. Due to the uniformity of $\mathbf{R}$, each $\mathbf{r}_i$ is uniformly distributed over $\{0,1\}^m$, hence $\widetilde{\mathbf{H}}_\infty(\mathbf{r}_i \bmod p) = m$ for any prime factor p of q^2. According to Lemma 2, we know that uniform matrix $\mathbf{X}$ is a good extractor. Concretely, by applying Lemma 2 with $\epsilon = 2^{-(\lambda+1)}$ and by the condition $m > 4(n+1)\log q + 2(\lambda+1)$, we have

$$\varDelta((\mathbf{X}, \mathbf{X}\cdot\mathbf{r}_i), (\mathbf{X}, \mathbf{u}_i)) \leq 2^{-(\lambda+1)}$$

for each $i \in [m]$, where $\mathbf{u}_i \leftarrow_{\$} \mathbb{Z}_{q^2}^{n+1}$. By a simple hybrid argument, it yields that

$$\varDelta((\mathbf{X}, \mathbf{X}\cdot\mathbf{R}), (\mathbf{X}, \mathbf{U})) \leq m \cdot 2^{-(\lambda+1)},$$

where $\mathbf{U} \leftarrow_{\$} \mathbb{Z}_{q^2}^{(n+1)\times m}$. Hence no matter for $\mathbf{m} = \mathbf{m}_0$ or $\mathbf{m} = \mathbf{m}_1$, we have

$$\varDelta\big((\mathbf{X}, \mathbf{X} \cdot \mathbf{R} + \tbinom{\mathbf{0}}{q \cdot \mathbf{m}^\top}), (\mathbf{X}, \mathbf{U})\big) \leq m \cdot 2^{-(\lambda+1)}. \tag{52}$$

Finally, (51) follows from (52) by the triangle inequality.

This completes the proof of Theorem 9. □

G Full Details of QA-NIZK from LWE in Subsect. 6.4

In this section, we present full details of Subsect. 6.4 and show how to build tag-based QA-NIZK for gap language based on the LWE assumptions, in order to serve as building blocks for our SIG and PKE constructions together with our LWE-based $\mathsf{prQAHPS}_{\mathsf{LWE}}$ and $\mathsf{CMT}_{\mathsf{LWE}}$ schemes.

We will follow the generic transformation proposed by Libert et al. in [34, Subsect. 4.2] that compiles any trapdoor $\varSigma$-protocol for gap language into tag-based QA-NIZK for the same gap language, with the help of correlation intractable (CI) hash function and lossy PKE. Moreover, the transformation is tightness-preserving, i.e., the resulting tag-based QA-NIZK has tight zero-knowledge and tight USS as long as the building blocks are tightly secure. Given the fact that there are already CI hash and lossy PKE from LWE (see Appendix G.1 for their LWE-based instantiations), all we need to do is to instantiate trapdoor $\varSigma$-protocol for gap language from LWE.

The roadmap of this section is as follows. In Appendix G.1, we recall the definitions of the building blocks including trapdoor $\varSigma$-protocol, CI hash and lossy PKE, and provide the instantiations of CI hash and lossy PKE based on LWE. In Appendix G.2, we provide additional lattice backgrounds. In Appendix G.3, we present the instantiations of trapdoor $\varSigma$-protocol based on LWE. Finally, in Appendix G.4, we recall the generic transformation in [34, Subsect. 4.2] for completeness, and describe how to compile our LWE-based trapdoor $\varSigma$-protocols into tag-based QA-NIZK schemes for gap languages.

G.1 Building Blocks: Definitions and Instantiations

In this subsection, we present the formal definitions of the building blocks of the generic transformation proposed in [34, Subsect. 4.2], including trapdoor $\varSigma$-protocol, correlation intractable (CI) hash function, lossy PKE, pseudorandom function (PRF) and one-time signature (OTS). We also recall the existing LWE-based instantiations for all the building blocks except trapdoor $\varSigma$-protocol, whose instantiations will be given in Appendix G.3.

Building Block 1 – Trapdoor $\varSigma$-Protocol for Gap Language: Syntax and Security Requirements.

Definition 24 (Trapdoor $\varSigma$-Protocol for Gap Language [34]). *Let $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ be a gap language parameterized by language parameter ρ, and let td_ρ denote some trapdoor information for $\mathcal{GL}_\rho$. A trapdoor $\varSigma$-protocol for gap language*

$\mathcal{GL}_\rho$ *consists of PPT algorithms* $\Sigma = (\Sigma.\mathsf{CRSGen}, \Sigma.\mathsf{Prove}_1, \Sigma.\mathsf{Prove}_2, \Sigma.\mathsf{Vrfy}, \Sigma.\mathsf{Sim}, \Sigma.\mathsf{TrapGen}, \Sigma.\mathsf{BadChallenge})$.

- $\mathsf{crs} \leftarrow_\$ \Sigma.\mathsf{CRSGen}(\rho)$: *Taking as input a language parameter* ρ*, the CRS generation algorithm outputs a common reference string (CRS)* crs*, which implicitly defines a challenge space* $\mathcal{CH}$*.*
- **3-Move Protocol.** *The* 3*-move protocol is executed between a "prover" and a "verifier". The prover and the verifier both take a CRS* crs *and an instance* $x \in \mathcal{L}_\rho$ *as input. The prover also takes as input a witness* w *of* x*.*
  - $(\mathsf{a}, \mathsf{st}) \leftarrow_\$ \Sigma.\mathsf{Prove}_1(\mathsf{crs}, x, w)$*: The prover invokes* $\Sigma.\mathsf{Prove}_1$ *to get a message* a *and a state* st*. Then the prover keeps* st *as its own state information and sends* a *to the verifier.*
  - $\mathsf{ch} \leftarrow_\$ \mathcal{CH}$*: After receiving* a*, the verifier chooses* $\mathsf{ch} \leftarrow_\$ \mathcal{CH}$ *uniformly at random as the challenge and sends* ch *to the prover.*
  - $\mathsf{z} \leftarrow_\$ \Sigma.\mathsf{Prove}_2(\mathsf{crs}, x, w, \mathsf{a}, \mathsf{st}, \mathsf{ch})$*: After obtaining* ch*, the prover invokes* $\Sigma.\mathsf{Prove}_2$ *to get a message* z *and sends* z *to the verifier.*
  - $0/1 \leftarrow \Sigma.\mathsf{Vrfy}(\mathsf{crs}, x, \mathsf{a}, \mathsf{ch}, \mathsf{z})$*: After getting* z*, the verifier invokes* $\Sigma.\mathsf{Vrfy}$ *to obtain a decision bit.*
- $(\tilde{\mathsf{a}}, \tilde{\mathsf{z}}) \leftarrow_\$ \Sigma.\mathsf{Sim}(\mathsf{crs}, x, \mathsf{ch})$*: Taking as input* crs*, an instance* x *and a challenge* $\mathsf{ch} \in \mathcal{CH}$*, the simulation algorithm outputs a simulated* $(\tilde{\mathsf{a}}, \tilde{\mathsf{z}})$*. Here* $(\tilde{\mathsf{a}}, \mathsf{ch}, \tilde{\mathsf{z}})$ *serves as a simulated transcript.*
- $(\mathsf{crs}, \mathsf{td}_\Sigma) \leftarrow_\$ \Sigma.\mathsf{TrapGen}(\rho, td_\rho)$*: Taking as input a language parameter* ρ *and a trapdoor information* td_ρ *for the gap language* $\mathcal{GL}_\rho$*, the trapdoor CRS generation algorithm outputs a* crs *and a trapdoor* td_Σ *for the scheme.*
- $\mathsf{ch} \leftarrow_\$ \Sigma.\mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}_\Sigma, x, \mathsf{a})$*: Taking as input* crs*, a trapdoor* td_Σ*, an instance* x *and a first message* a*, the bad challenge algorithm outputs a challenge* ch*.*

The following properties are required:

- **Completeness:** *For all* $x \in \mathcal{L}_\rho$ *with witness* w *and all* $\mathsf{crs} \leftarrow_\$ \Sigma.\mathsf{CRSGen}(\rho)$*, it holds that*

$$\Pr\left[\begin{array}{c}(\mathsf{a}, \mathsf{st}) \leftarrow_\$ \Sigma.\mathsf{Prove}_1(\mathsf{crs}, x, w), \\ \mathsf{ch} \leftarrow_\$ \mathcal{CH}, \\ \mathsf{z} \leftarrow_\$ \Sigma.\mathsf{Prove}_2(\mathsf{crs}, x, w, \mathsf{a}, \mathsf{st}, \mathsf{ch})\end{array} : \Sigma.\mathsf{Vrfy}(\mathsf{crs}, x, \mathsf{a}, \mathsf{ch}, \mathsf{z}) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

- **Special Soundness:** *For any* $x \notin \widetilde{\mathcal{L}}_\rho$*, any* $\mathsf{crs} \in \Sigma.\mathsf{CRSGen}(\rho)$ *and any first message* a*, there is at most one challenge* $\mathsf{ch} \in \mathcal{CH}$ *such that* $\Sigma.\mathsf{Vrfy}(\mathsf{crs}, x, \mathsf{a}, \mathsf{ch}, \mathsf{z}) = 1$ *for some third message* z*. Moreover, we define a "bad challenge function"* f *with* $f(\mathsf{crs}, x, \mathsf{a}) := \mathsf{ch}$ *if there exists such a unique* ch *and* $f(\mathsf{crs}, x, \mathsf{a}) := \perp$ *otherwise. Note that* f *might not be efficiently computable.*
- **Special Zero-Knowledge:** *For all* $x \in \mathcal{L}_\rho$ *with witness* w*, all* $\mathsf{crs} \leftarrow_\$ \Sigma.\mathsf{CRSGen}(\rho)$ *and all* $\mathsf{ch} \in \mathcal{CH}$*, it holds that*

$$\Delta((\mathsf{a}, \mathsf{z}), (\tilde{\mathsf{a}}, \tilde{\mathsf{z}})) \leq \mathsf{negl}(\lambda),$$

where the probability is over $(\mathsf{a}, \mathsf{st}) \leftarrow \Sigma.\mathsf{Prove}_1(\mathsf{crs}, x, w)$*,* $\mathsf{z} \leftarrow \Sigma.\mathsf{Prove}_2(\mathsf{crs}, x, w, \mathsf{a}, \mathsf{st}, \mathsf{ch}))$ *and* $(\tilde{\mathsf{a}}, \tilde{\mathsf{z}}) \leftarrow \Sigma.\mathsf{Sim}(\mathsf{crs}, x, \mathsf{ch})$.

- **Perfect CRS Indistinguishability:** *The* crs *generated by* $\mathsf{crs} \leftarrow_\$ \Sigma.\mathsf{CRSGen}(\rho)$ *is identically distributed as the* crs *generated by* $(\mathsf{crs}, \mathsf{td}_\Sigma) \leftarrow_\$ \Sigma.\mathsf{TrapGen}(\rho, td_\rho)$.
- **Correctness of** $\Sigma.\mathsf{BadChallenge}$**:** *For all* $x \notin \widetilde{\mathcal{L}}_\rho$*, all* $(\mathsf{crs}, \mathsf{td}_\Sigma) \in \mathsf{TrapGen}(\rho, td_\rho)$ *and all first messages* a*, it holds that* $\Sigma.\mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}_\Sigma, x, \mathsf{a}) = f(\mathsf{crs}, x, \mathsf{a})$ *if* $f(\mathsf{crs}, x, \mathsf{a}) \neq \perp$*. Here* f *is the bad challenge function.*

Building Block 2 – Correlation Intractable Hash: Syntax, Security Requirements, and LWE-based Construction.

Definition 25 (Searchable Relation). *A relation* $R \subseteq \mathcal{X} \times \mathcal{Y}$ *is searchable in time* T *if there exists a function* $f : \mathcal{X} \longrightarrow \mathcal{Y}$ *which is computable in time* T *and satisfies that, if there exists* y *s.t.* $(x, y) \in R$*, then* $f(x) = y$*.*

Definition 26 (Somewhere Statistically Correlation Intractable Hash [16]). *Given a relation ensemble* $\mathcal{R} = \{R \subseteq \mathcal{X} \times \mathcal{Y}\}$*, a keyed hash family* $\mathcal{H} = \{h : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}\}$ *with key space* $\mathcal{K}$ *is somewhere statistically correlation intractable (CI) w.r.t.* $\mathcal{R}$ *if there exist PPT algorithms* $\mathsf{CIH} = (\mathsf{CIH.Gen}, \mathsf{CIH.StGen})$ *defined as follows:*

– $k \leftarrow_\$ \mathsf{CIH.Gen}$*: It outputs a hashing key* $k \in \mathcal{K}$*.*
– $k \leftarrow_\$ \mathsf{CIH.StGen}(\mathsf{aux})$*: It takes an auxiliary string* aux *as input and outputs a hashing key* $k \in \mathcal{K}$*.*

For any relation $R \in \mathcal{R}$*, there exists an auxiliary string* aux_R *with the following two properties:*

- **Key Indistinguishability:** *For any PPT algorithm* $\mathcal{A}$*, it holds that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{CIH},\mathcal{A}}(\lambda) :=\big| &\Pr[k \leftarrow_\$ \mathsf{CIH.Gen} : \mathcal{A}(k, \mathsf{aux}_R) = 1] \\ &- \Pr[k \leftarrow_\$ \mathsf{CIH.StGen}(\mathsf{aux}_R) : \mathcal{A}(k, \mathsf{aux}_R) = 1]\big| \leq \mathsf{negl}(\lambda).\end{aligned}$$

- **Statistical Correlation Intractability:** *It requires that*

$$\Pr\left[k \leftarrow_\$ \mathsf{CIH.StGen}(\mathsf{aux}_R) \;:\; \exists\, x \in \mathcal{X} \text{ s.t. } (x, h(k, x)) \in R\right] \leq 2^{-\Omega(\lambda)}.$$

In [43], Peikert and Shiehian proposed a CI-Hash for any searchable relation defined by functions f of bounded depth (in the sense of Definition 25) based on the standard LWE assumption. We summarize the result in the following theorem.

Theorem 11 ([43]). *Assuming the hardness of* $\mathsf{LWE}_{n-1,q,\chi,m+1}$ *for a* $\mathsf{poly}(n)$*-bounded* χ *and a sufficiently large* $q = m^{O(d)}$*, the CI-Hash scheme proposed in [43] supports arbitrary input length, and its output length is exactly* $m = n\lceil \log q \rceil$*. It is somewhere statistically correlation intractable for the class of functions with output length* m *that can be implemented by depth-*d *Boolean circuits, and each circuit serves as the auxiliary input for itself. Concretely, for any PPT adversary* $\mathcal{A}$*, there exists a PPT adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{CIH},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{LWE}}_{[n-1,q,\chi,m+1],\mathcal{B}}(\lambda).$$

With the help of fully homomorphic encryption (FHE) scheme, the CIH function can be constructed to support statistically correlation intractable function which is implemented by circuits of any polynomial size. In this case, the security of CIH is tightly reduced to the LWE assumption and CPA security of FHE, which can be further tightly reduced to the LWE assumption.

Building Block 3 – Lossy PKE: Syntax, Security Requirements, and LWE-based Construction.

Definition 27 (**R-Lossy PKE with Efficient Opening** [34, 18]). *Let $R \subseteq \mathcal{K}_\lambda \times \mathcal{T}_\lambda$ be an efficiently computable binary relation. An R-lossy PKE scheme R-$\mathsf{LPKE} = (\mathsf{Gen}, \mathsf{LGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Opener}, \mathsf{LOpener})$ consists of PPT algorithms and is associated with message space $\mathcal{M}$, tag space $\mathcal{T}_\lambda$, initialization value space $\mathcal{K}_\lambda$ and randomness space $\mathcal{R}_{\mathsf{LPKE}}$. The randomness distribution over $\mathcal{R}_{\mathsf{LPKE}}$ used for encryption is denoted by $D_{\mathcal{R}_{\mathsf{LPKE}}}$.*

- $(pk, sk, tk) \leftarrow_{\$} \mathsf{Gen}(K)$*: The key generation algorithm takes as input an initialization value $K \in \mathcal{K}_\lambda$, and outputs an injective public key pk, a decryption key sk and a trapdoor key tk.*
- $(pk, sk, tk) \leftarrow_{\$} \mathsf{LGen}(K)$*: The lossy key generation algorithm takes as input an initialization value $K \in \mathcal{K}_\lambda$, and outputs a lossy public key pk, a lossy secret key sk and a trapdoor key tk.*
- $c \leftarrow_{\$} \mathsf{Enc}(pk, t, m)$*: The encryption algorithm takes as input a public key pk, a tag $t \in \mathcal{T}_\lambda$ and a message $m \in \mathcal{M}$, and outputs a ciphertext c.*
- $m'/\bot \leftarrow \mathsf{Dec}(sk, t, c)$*: The decryption algorithm takes as input a decryption key sk, a tag $t \in \mathcal{T}_\lambda$ and a ciphertext c, and outputs $m' \in \mathcal{M}$ or $\bot$.*
- $r' \leftarrow_{\$} \mathsf{Opener}(pk, tk, t, c, m')$*: The opening algorithm takes as input a public key pk, a trapdoor key tk, a tag $t \in \mathcal{T}_\lambda$, a ciphertext c and a message m', and outputs a randomness $r' \in \mathcal{R}_{\mathsf{LPKE}}$.*
- $r' \leftarrow_{\$} \mathsf{LOpener}(sk, t, c, m')$*: The lossy opening algorithm takes as input a secret key sk, a tag $t \in \mathcal{T}_\lambda$, a ciphertext c and a message m', and outputs a randomness $r' \in \mathcal{R}_{\mathsf{LPKE}}$.*

The following properties should be satisfied:

- **Decryption Correctness under Injective Tags:** *For any initialization value K and any tag t such that $(K, t) \in R$, and any $m \in \mathcal{M}$, it holds that*

$$\Pr\left[(pk, sk, tk) \leftarrow_{\$} \mathsf{Gen}(K) \ : \ \begin{array}{c} \exists\ r \in \mathcal{R}_{\mathsf{LPKE}},\ s.t. \\ \mathsf{Dec}(sk, t, \mathsf{Enc}(pk, t, m; r)) \neq m \end{array}\right] \leq \mathsf{negl}(\lambda).$$

- **Key Indistinguishability:** *There are two requirements. One is the indistinguishability of public/trapdoor key pairs outputted by the normal algorithm* Gen *and the lossy algorithm* LGen*. The other is the indistinguishability of public/secret key pairs output by* LGen *under different initialization values.*

  (i) For any initialization value $K \in \mathcal{K}_\lambda$, and any PPT $\mathcal{A}$, it holds that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind\text{-}1}}_{R\text{-}\mathsf{LPKE},\mathcal{A}}(\lambda) := \big|\Pr[(pk, sk, tk) \leftarrow_{\$} \mathsf{Gen}(K) : \mathcal{A}(pk, tk) = 1] \\ - \Pr[(pk, sk, tk) \leftarrow_{\$} \mathsf{LGen}(K) : \mathcal{A}(pk, tk) = 1]\big| \leq \mathsf{negl}(\lambda).\end{aligned}$$

(ii) For any distinct values $K, K' \in \mathcal{K}_\lambda$, and any PPT $\mathcal{A}$, it holds that

$$\begin{aligned}\mathsf{Adv}^{\mathsf{ind\text{-}2}}_{R\text{-}\mathsf{LPKE},\mathcal{A}}(\lambda) := \big|\Pr[(pk, sk, tk) \leftarrow_\$ \mathsf{LGen}(K) : \mathcal{A}(pk, sk) = 1] \\ - \Pr[(pk, sk, tk) \leftarrow_\$ \mathsf{LGen}(K') : \mathcal{A}(pk, sk) = 1]\big| \leq \mathsf{negl}(\lambda).\end{aligned}$$

- **Lossiness under Lossy Tags:** *For any value $K \in \mathcal{K}_\lambda$ and tag $t \in \mathcal{T}_\lambda$ such that $(K, t) \notin R$, any $(pk, sk, tk) \leftarrow_\$ \mathsf{Gen}(K)$, and any $m_0, m_1 \in \mathcal{M}$, it holds*

  $$\Delta(c_0, c_1) \leq \mathsf{negl}(\lambda),$$

  where the probability is over $c_0 \leftarrow \mathsf{Enc}(pk, t, m_0)$ and $c_1 \leftarrow \mathsf{Enc}(pk, t, m_1)$.
- **Efficient Opening via Opener under Lossy Tags:** *Let $D_{\mathcal{R}_{\mathsf{LPKE}}}$ be the randomness distribution over $\mathcal{R}_{\mathsf{LPKE}}$, from which the random coins r used by Enc are sampled. For any public key pk, tag t, message m and ciphertext c, let $D_{pk,m,c,t}$ denote the probability distribution on $\mathcal{R}_{\mathsf{LPKE}}$ with support*

  $$S_{pk,m,c,t} = \left\{\bar{r} \in \mathcal{R}_{\mathsf{LPKE}} \;\middle|\; \mathsf{Enc}(pk, t, m; \bar{r}) = c\right\},$$

  and such that, for any $\bar{r} \in S_{pk,m,c,t}$, we have

  $$D_{pk,m,c,t}(\bar{r}) = \Pr_{r \leftarrow_\$ D_{\mathcal{R}_{\mathsf{LPKE}}}}\left[r = \bar{r} \;\middle|\; \mathsf{Enc}(pk, t, m; r) = c\right].$$

  For any $K \in \mathcal{K}_\lambda$, any keys $(pk, sk, tk) \leftarrow_\$ \mathsf{Gen}(K)$ and $(pk, sk, tk) \leftarrow_\$ \mathsf{LGen}(K)$, any tag $t \in \mathcal{T}_\lambda$ such that $(K, t) \notin R$, any messages $m_0, m_1 \in \mathcal{M}$, and any $r \leftarrow_\$ D_{\mathcal{R}_{\mathsf{LPKE}}}$, let $c = \mathsf{Enc}(pk, t, m_0; r)$. Then it holds that

  $$\Delta(r', \bar{r}) \leq \mathsf{negl}(\lambda),$$

  where $r' \leftarrow_\$ \mathsf{Opener}(pk, tk, t, c, m_1)$ and $\bar{r}$ follows the distribution $D_{pk,m_1,c,t}$.
- **Efficient Opening via $\mathsf{LOpener}$ under Lossy Keys:** *For any $K \in \mathcal{K}_\lambda$, any $(pk, sk, tk) \leftarrow_\$ \mathsf{LGen}(K)$, any tag $t \in \mathcal{T}_\lambda$, any messages $m_0, m_1 \in \mathcal{M}$, and any $r \leftarrow_\$ D_{\mathcal{R}_{\mathsf{LPKE}}}$, let $c = \mathsf{Enc}(pk, t, m_0; r)$. Then it holds that*

  $$\Delta(r', \bar{r}) \leq \mathsf{negl}(\lambda),$$

  where $r' \leftarrow_\$ \mathsf{LOpener}(sk, t, c, m_1)$ and $\bar{r}$ follows the distribution $D_{pk,m_1,c,t}$.

In [34], Libert et al. proposed a R-LPKE scheme with security tightly reduced to the multi-secret LWE assumption. We summarize the result in the following theorem.

Theorem 12 ([34]). *Let $q = \mathsf{poly}(\lambda)$ be a prime modulus, $\mathcal{M} = \{0, 1\}^{n_0}$ be the message space, $n = n_0 + \Omega(\lambda)$, $m = 2n\lceil \log q \rceil + O(\lambda)$ and $\sigma = O(m) \cdot \lambda$. Then the R-LPKE scheme proposed in [34] is a lossy PKE scheme with message space $\mathcal{M} = \{0, 1\}^{n_0}$. Concretely, for any PPT adversary $\mathcal{A}$, there exists a PPT adversary $\mathcal{B}$ s.t. $\mathsf{Adv}^{\mathsf{ind\text{-}1}}_{R\text{-}\mathsf{LPKE},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{n_0\text{-}\mathsf{LWE}}_{[n-n_0,q,D_{\mathbb{Z},\sigma},m],\mathcal{B}}(\lambda)$ and $\mathsf{Adv}^{\mathsf{ind\text{-}2}}_{R\text{-}\mathsf{LPKE},\mathcal{A}}(\lambda) \leq 2^{-\Omega(\lambda)}$.*

Building Block 4 – Pseudorandom Function: Syntax, Security Requirements, and LWE-based Construction.

Definition 28 (Pseudorandom Function). *Let* $\mathsf{PRF} : \mathcal{K} \times \mathcal{X} \longrightarrow \mathcal{Y}$ *be a function with key space* $\mathcal{K}$*, input space* $\mathcal{X}$ *and output space* $\mathcal{Y}$*.* PRF *is a pseudorandom function, if for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{pse}}_{\mathsf{PRF},\mathcal{A}}(\lambda) := \left| \Pr[\mathcal{A}^{\mathsf{PRF}(K,\cdot)} = 1] - \Pr[\mathcal{A}^{f(\cdot)} = 1] \right| \leq \mathsf{negl}(\lambda),$$

where $K \leftarrow_{\$} \mathcal{K}$*,* f *is uniformly chosen from the set of all functions mapping* $\mathcal{X}$ *to* $\mathcal{Y}$*, and* $\mathcal{A}$ *has oracle access to either* $\mathsf{PRF}(K,\cdot)$ *or* $f(\cdot)$*.*

As suggested by Libert et al. [34], the Key-homomorphic PRF scheme in [13] is a good choice, and the pseudorandomness of PRF is based on the LWE assumption with security loss factor linear to the input length of the PRF. We conclude as follows.

Theorem 13 ([34, 13]). *Let* $q = O(\sqrt{n}/\alpha)$ *and* $m = \lceil n \log q \rceil$*. If the PRF scheme proposed in [13] supports* ℓ*-bit input, then for any PPT adversary* $\mathcal{A}$*, there exists a PPT adversary* $\mathcal{B}$ *such that*

$$\mathsf{Adv}^{\mathsf{pse}}_{\mathsf{PRF},\mathcal{A}}(\lambda) \leq \ell \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[n,q,\chi,m],\mathcal{B}}(\lambda).$$

Building Block 5 – One-Time Signature: Syntax, Security Requirements, and LWE-based Construction.

The syntax of one-time signature (OTS) is the same as signature defined in Definition 13. Below we define the strong one-time security for one-time signature in the Multi-User setting (strong MU-OT).

Definition 29 (Strong MU-OT Security for One-Time Signature). *A signature scheme* $\mathsf{OTS} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is strongly* MU-OT *secure, if for any PPT adversary* $\mathcal{A}$ *and any polynomial* N*, it holds that* $\mathsf{Adv}^{\mathsf{str\text{-}ot}}_{\mathsf{OTS},\mathcal{A},N}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{str\text{-}ot}}_{\mathsf{OTS},\mathcal{A},N} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*, where* $\mathsf{Exp}^{\mathsf{str\text{-}ot}}_{\mathsf{OTS},\mathcal{A},N}$ *is defined in Fig. 9.*

```
$\mathsf{Exp}^{\mathsf{str\text{-}ot}}_{\mathsf{OTS},\mathcal{A},N}$:
$\mathsf{pp}_{\mathsf{SIG}} \leftarrow_{\$} \mathsf{Setup}$
For $i \in [N]$:  $(vk_i, sigk_i) \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp}_{\mathsf{SIG}})$
$\mathcal{Q}_{\text{SIGN}} := \emptyset$                //Record the signing queries
$(i^* \in [n], m^*, \sigma^*) \leftarrow_{\$} \mathcal{A}^{\mathcal{O}_{\text{SIGN}}(\cdot,\cdot)}(\mathsf{pp}_{\mathsf{SIG}}, \{vk_i\}_{i \in [N]})$
If $((i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{SIGN}}) \wedge (\mathsf{Vrfy}(vk_{i^*}, m^*, \sigma^*) = 1)$:
    Return 1;
Else: Return 0

$\mathcal{O}_{\text{SIGN}}(i, m)$:  //at most one query per user $i$
    $\sigma \leftarrow_{\$} \mathsf{Sign}(sigk_i, m)$
    $\mathcal{Q}_{\text{SIGN}} := \mathcal{Q}_{\text{SIGN}} \cup \{(i, m, \sigma)\}$
    Return $\sigma$
```

Fig. 9. The MU-OT security experiment $\mathsf{Exp}^{\mathsf{str\text{-}ot}}_{\mathsf{OTS},\mathcal{A},N}$ for OTS.

In [35], Libert et al. presented a one-time signature with strong MU-OT security, which is tightly reduced to the Short Integer Solution (SIS) assumption. We recall the definition of SIS and conclude their OTS as follows.

Definition 30 (SIS Assumption). *The Short Integer Solution* $\mathsf{SIS}_{n,q,m,\beta}$ *assumption holds if for any PPT adversary* $\mathcal{A}$*, it holds that*

$$\mathsf{Adv}^{\mathsf{SIS}}_{[n,q,m,\beta],\mathcal{A}}(\lambda) := \Pr\left[\begin{array}{c}\mathbf{A} \leftarrow_{\$} \mathbb{Z}_q^{n\times m}, \\ \mathbf{x}\in\mathbb{Z}^m \leftarrow_{\$} \mathcal{A}(\mathbf{A})\end{array} : \begin{array}{c}\mathbf{A}\cdot\mathbf{x} = \mathbf{0} \bmod q \\ \wedge\ \mathbf{x}\neq\mathbf{0}\ \wedge\ \ \|\mathbf{x}\|\leq\beta\end{array}\right] \leq \mathsf{negl}(\lambda).$$

Theorem 14 ([35]). *The* OTS *scheme proposed in [35] is strongly* MU-OT *secure based on the SIS assumption. Let* $n, m, q \in \mathbb{N}$ *be public parameters of* OTS *such that* $m > 4n\log q$*. Let* σ *be the discrete Gaussian parameter in* OTS*, and let* $\beta = m(1+2\sigma)$*. The message space of* OTS *is* $\mathcal{M} = \{0,1\}^m$*. Then for any PPT adversary* $\mathcal{A}$ *and any polynomial* N*, there exists a PPT adversary* $\mathcal{B}$ *such that* $\mathsf{Adv}^{\mathsf{str\text{-}ot}}_{\mathsf{OTS},\mathcal{A},N}(\lambda) \leq \mathsf{Adv}^{\mathsf{SIS}}_{[n,q,m,\beta],\mathcal{B}}(\lambda)$.

G.2 Additional Backgrounds on Lattices

Lemma 14 ([36, Theorem 4.6]). *Let* $\mathcal{V}$ *be a subset of* $\mathbb{Z}^m$ *in which all elements have* ℓ_2 *norms less than* T*,* ζ *be a real number such that* $\zeta = \omega(T\sqrt{\log m})$ *and* V *be a distribution over* $\mathcal{V}$*. Then, there exists a real number* M *such that the distributions of the following algorithms* $\mathcal{A}$ *and* $\mathcal{F}$ *has statistical distance at most* $\frac{2^{-\omega(\log m)}}{M}$*:*

- $\mathcal{A}$*: sample* $\mathbf{v} \leftarrow_{\$} V$*,* $\mathbf{z} \leftarrow_{\$} D_{\mathbb{Z}^m,\zeta,\mathbf{v}}$ *and output* $(\mathbf{z},\mathbf{v})$ *with probability*
$$\min\left(\frac{D_{\mathbb{Z}^m,\zeta}(\mathbf{z})}{M\cdot D_{\mathbb{Z}^m,\zeta,\mathbf{v}}(\mathbf{z})}, 1\right);$$
- $\mathcal{F}$*: sample* $\mathbf{v} \leftarrow_{\$} V$*,* $\mathbf{z} \leftarrow_{\$} D_{\mathbb{Z}^m,\zeta}$ *and output* $(\mathbf{z},\mathbf{v})$ *with probability* $1/M$*.*

Moreover, the probability that $\mathcal{A}$ *outputs something is at least* $\frac{1-2^{-\omega(\log m)}}{M}$*.*

More concretely, if $\zeta = \alpha T$ *for any positive* α*, then* $M = e^{12/\alpha + 1/(2\alpha^2)}$*, the above statistical distance is at most* $2^{-100}/M$*, and the probability that* $\mathcal{A}$ *outputs something is at least* $(1-2^{-100})/M$*.*

G.3 Trapdoor Σ-protocol from LWE

In order to construct tag-based QA-NIZK schemes for gap languages, in this subsection, we will first construct trapdoor Σ-protocols for the same gap languages based on the LWE assumptions, then in next subsection (Appendix G.4), we show how to compile them into tag-based QA-NIZK schemes via the generic transformation proposed in [34, Subsect. 4.2].

The Gap Language. As discussed in Subsect. 6.4, we note that the gap languages needed in our generic SIG and PKE constructions are different.

For the SIG construction in Subsect. 4.1, the gap language is the $\mathcal{GL}^{(\mathsf{QANIZK})}_{\rho'} = (\mathcal{L}^{(\mathsf{QANIZK})}_{\rho'}, \widetilde{\mathcal{L}}^{(\mathsf{QANIZK})}_{\rho'})$ defined in Fig. 1, which is determined by the gap language distribution $\mathscr{L}$, the pr-QA-HPS scheme $\mathsf{prQAHPS}$ and the commitment scheme CMT. We make the gap language concrete by instantiating with our

LWE-based $\mathscr{L}$ in Subsect. 6.1, $\mathsf{prQAHPS}_{\mathsf{LWE}}$ in Subsect. 6.2 and $\mathsf{CMT}_{\mathsf{LWE}}$ in Subsect. 6.3. Let $\mathsf{pp}_{\mathsf{LWE}} = (n, m, \ell, q, \sigma, \gamma, \chi, B, \tilde{B}, B', \tilde{B}', \zeta, \zeta')$ be the LWE-related public parameters that serve as implicit input to all algorithms, where $B < \tilde{B}$ and $B' < \tilde{B}'$. More precisely, let $\rho = \mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a language parameter output by $\mathscr{L}$, which is generated by $(\mathbf{A}, \mathbf{T}_{\mathbf{A}}) \leftarrow_{\$} \mathsf{TrapGen}(n, q, m)$ (cf. Lemma 3), and let $\mathsf{pp}_{\mathsf{CMT}} = \mathbf{X} \in \mathbb{Z}_{q^2}^{(n+1)\times m}$ be a parameter generated by BSetup. Then according to Fig. 1, $\rho' = (\mathbf{A}, \mathbf{X})$ and the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ is instantiated as follows:

$$\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})} = \left\{ (\mathbf{c}, vk, d) \;\middle|\; \begin{matrix} \exists\, (\mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-B, B]^m, \\ \mathbf{R} \in \{0,1\}^{m \times m}, \mathbf{k} \in \{0,1\}^m, \\ e' \in [-B', B']) \end{matrix} \text{ s.t. } \begin{matrix} \mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top \\ \wedge\, vk = \mathbf{X} \cdot \mathbf{R} + \binom{\mathbf{0}}{q \cdot \mathbf{k}^\top} \\ \wedge\, d = \mathbf{c}^\top \cdot \mathbf{k} + e' \end{matrix} \right\}, \tag{53}$$

$$\widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})} = \left\{ (\mathbf{c}, vk, d) \;\middle|\; \begin{matrix} \exists\, (\mathbf{s} \in \mathbb{Z}_q^n, \mathbf{e} \in [-\tilde{B}, \tilde{B}]^m, \\ \mathbf{R} \in [-\tilde{B}, \tilde{B}]^{m \times m}, \mathbf{k} \in [-\tilde{B}, \tilde{B}]^m, \\ e' \in [-\tilde{B}', \tilde{B}']) \end{matrix} \text{ s.t. } \begin{matrix} \mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top \\ \wedge\, vk = \mathbf{X} \cdot \mathbf{R} + \binom{\mathbf{0}}{q \cdot \mathbf{k}^\top} \\ \wedge\, d = \mathbf{c}^\top \cdot \mathbf{k} + e' \end{matrix} \right\}. \tag{54}$$

We set $td_{\rho'} := \mathbf{T}_{\mathbf{A}}$ as the trapdoor information of the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})}$, where $\mathbf{T}_{\mathbf{A}}$ is generated along with $\mathbf{A}$ by $\mathsf{TrapGen}(n, q, m)$.

For the PKE construction in Subsect. 4.2, the gap language is exactly the $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ generated by $\mathscr{L}$, as defined in Subsect. 6.1, i.e., $\rho = \mathbf{A}$ and

$$\mathcal{L}_\rho := \{ \mathbf{c} \in \mathbb{Z}_q^m \,|\, \exists\ \mathbf{s} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}, \mathbf{e} \in [-B, B]^m, \text{ s.t. } \mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top \}, \tag{55}$$

$$\widetilde{\mathcal{L}}_\rho := \{ \mathbf{c} \in \mathbb{Z}_q^m \,|\, \exists\ \mathbf{s} \in \mathbb{Z}_q^n \setminus \{\mathbf{0}\}, \mathbf{e} \in [-\tilde{B}, \tilde{B}]^m, \text{ s.t. } \mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top \}. \tag{56}$$

Next, we will construct trapdoor Σ-protocols for the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ and for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ based on the LWE assumptions, respectively, serving as building blocks for our SIG and PKE constructions. Our constructions are inspired by the trapdoor Σ-protocol for ACPS ciphertexts [4] constructed in [34, Sect. 5], by observing that both the gap languages $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})}$ and $\mathcal{GL}_\rho$ are defined with linear equations, i.e., the instance is linear in the witness, and parts of the witness are bounded.

The Trapdoor Σ-protocol for $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})}$. The syntax of trapdoor Σ-protocol is shown in Definition 24. Below we construct an LWE-based trapdoor Σ-protocol $\Sigma = (\Sigma.\mathsf{CRSGen}, \Sigma.\mathsf{Prove}_1, \Sigma.\mathsf{Prove}_2, \Sigma.\mathsf{Vrfy}, \Sigma.\mathsf{Sim}, \Sigma.\mathsf{TrapGen}, \Sigma.\mathsf{BadChallenge})$ for the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ specified by (53) and (54), with challenge set $\mathcal{CH} = \{0, 1\}$.

- $\underline{\mathsf{crs} \leftarrow_{\$} \Sigma.\mathsf{CRSGen}(\rho')}$**:** On input of language parameter $\rho' = (\mathbf{A}, \mathbf{X})$, return $\mathsf{crs} := (\mathbf{A}, \mathbf{X})$.
- $\underline{(\mathsf{a}, \mathsf{st}) \leftarrow_{\$} \Sigma.\mathsf{Prove}_1(\mathsf{crs}, x, w)}$**:** Parse $\mathsf{crs} = (\mathbf{A}, \mathbf{X})$ and $x = (\mathbf{c}, vk, d)$. Choose

$$\mathbf{s}_0 \leftarrow_{\$} \mathbb{Z}_q^n,\ \mathbf{e}_0 \leftarrow_{\$} D_{\mathbb{Z}^m, \zeta},\ \mathbf{R}_0 \leftarrow_{\$} D_{\mathbb{Z}^{m \times m}, \zeta},\ \mathbf{k}_0 \leftarrow_{\$} D_{\mathbb{Z}^m, \zeta},\ e'_0 \leftarrow_{\$} D_{\mathbb{Z}, \zeta'}.$$

Compute

$$\mathbf{c}_0^\top := \mathbf{s}_0^\top \cdot \mathbf{A} + \mathbf{e}_0^\top \bmod q, \quad vk_0 := \mathbf{X} \cdot \mathbf{R}_0 + \binom{\mathbf{0}}{q \cdot \mathbf{k}_0^\top} \bmod q^2,$$
$$d_0 := \mathbf{c}^\top \cdot \mathbf{k}_0 + e_0' \bmod q.$$

Return $\mathsf{a} := (\mathbf{c}_0, vk_0, d_0)$ and $\mathsf{st} := (\mathbf{s}_0, \mathbf{e}_0, \mathbf{R}_0, \mathbf{k}_0, e_0')$.

- <u>$\mathsf{z} \leftarrow_\$ \Sigma.\mathsf{Prove}_2(\mathsf{crs}, x, w, \mathsf{a}, \mathsf{st}, \mathsf{ch} \in \{0,1\})$:</u> Parse $w = (\mathbf{s}, \mathbf{e}, \mathbf{R}, \mathbf{k}, e')$ and $\mathsf{st} = (\mathbf{s}_0, \mathbf{e}_0, \mathbf{R}_0, \mathbf{k}_0, e_0')$. Compute

$$\mathbf{s}_{mix} := \mathbf{s}_0 + \mathsf{ch} \cdot \mathbf{s} \bmod q, \quad \mathbf{e}_{mix} := \mathbf{e}_0 + \mathsf{ch} \cdot \mathbf{e} \bmod q,$$
$$\mathbf{R}_{mix} := \mathbf{R}_0 + \mathsf{ch} \cdot \mathbf{R} \bmod q^2, \quad \mathbf{k}_{mix} := \mathbf{k}_0 + \mathsf{ch} \cdot \mathbf{k} \bmod q,$$
$$e'_{mix} := e'_0 + \mathsf{ch} \cdot e' \bmod q.$$

Return $\mathsf{z} := (\mathbf{s}_{mix}, \mathbf{e}_{mix}, \mathbf{R}_{mix}, \mathbf{k}_{mix}, e'_{mix})$ with probability θ and abort otherwise, where the probability θ is defined by $\theta := \min\left(\frac{D_{\mathbb{Z}^m,\zeta}(\mathbf{e}_{mix})}{M \cdot D_{\mathbb{Z}^m,\zeta,\mathsf{ch}\cdot\mathbf{e}}(\mathbf{e}_{mix})} \cdot \frac{D_{\mathbb{Z}^{m\times m},\zeta}(\mathbf{R}_{mix})}{M \cdot D_{\mathbb{Z}^{m\times m},\zeta,\mathsf{ch}\cdot\mathbf{R}}(\mathbf{R}_{mix})} \cdot \frac{D_{\mathbb{Z}^m,\zeta}(\mathbf{k}_{mix})}{M \cdot D_{\mathbb{Z}^m,\zeta,\mathsf{ch}\cdot\mathbf{k}}(\mathbf{k}_{mix})} \cdot \frac{D_{\mathbb{Z},\zeta'}(e'_{mix})}{M' \cdot D_{\mathbb{Z},\zeta',ch\cdot e'}(e'_{mix})}, 1\right)$ with $M := e^{12\sqrt{m}B/\zeta + mB^2/(2\zeta^2)}$ and $M' := e^{12B'/\zeta' + B'^2/(2\zeta'^2)}$.

- <u>$0/1 \leftarrow \Sigma.\mathsf{Vrfy}(\mathsf{crs}, x, \mathsf{a}, \mathsf{ch}, \mathsf{z})$:</u> Parse $\mathsf{crs} = (\mathbf{A}, \mathbf{X})$, $x = (\mathbf{c}, vk, d)$, $\mathsf{a} = (\mathbf{c}_0, vk_0, d_0)$ and $\mathsf{z} = (\mathbf{s}_{mix}, \mathbf{e}_{mix}, \mathbf{R}_{mix}, \mathbf{k}_{mix}, e'_{mix})$. Check if

$$\|\mathbf{e}_{mix}\|_\infty \le \tilde{B}/2, \ \|\mathbf{R}_{mix}\|_\infty \le \tilde{B}/2, \ \|\mathbf{k}_{mix}\|_\infty \le \tilde{B}/2, \ |e'_{mix}| \le \tilde{B}'/2,$$

and check if

$$\mathbf{c}_0^\top + \mathsf{ch} \cdot \mathbf{c}^\top = \mathbf{s}_{mix}^\top \cdot \mathbf{A} + \mathbf{e}_{mix}^\top \bmod q,$$
$$vk_0 + \mathsf{ch} \cdot vk = \mathbf{X} \cdot \mathbf{R}_{mix} + \binom{\mathbf{0}}{q \cdot \mathbf{k}_{mix}^\top} \bmod q^2,$$
$$d_0 + \mathsf{ch} \cdot d = \mathbf{c}^\top \cdot \mathbf{k}_{mix} + e'_{mix} \bmod q.$$

If all these checks pass, return 1; otherwise, return 0.

- <u>$(\mathsf{crs}, \mathsf{td}_\Sigma) \leftarrow_\$ \Sigma.\mathsf{TrapGen}(\rho', td_{\rho'})$:</u> On input of language parameter $\rho' = (\mathbf{A}, \mathbf{X})$ and trapdoor information $td_{\rho'} = \mathbf{T_A}$ for $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})}$, return $\mathsf{crs} := (\mathbf{A}, \mathbf{X})$ and $\mathsf{td}_\Sigma := \mathbf{T_A}$.

- <u>$\mathsf{ch} \leftarrow \Sigma.\mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}_\Sigma = \mathbf{T_A}, x, \mathsf{a})$:</u> Parse $x = (\mathbf{c}, vk, d)$ and $\mathsf{a} = (\mathbf{c}_0, vk_0, d_0)$. Invoke $(\mathbf{s}, \mathbf{e}) \leftarrow_\$ \mathsf{Invert}(\mathbf{c}_0, \mathbf{T_A})$ (cf. Lemma 4) and if $\|\mathbf{e}\|_\infty \le \tilde{B}/2$ then return $\mathsf{ch} := 0$.
  Invoke $(\mathbf{s}, \mathbf{e}) \leftarrow_\$ \mathsf{Invert}(\mathbf{c}_0 + \mathbf{c}, \mathbf{T_A})$ and if $\|\mathbf{e}\|_\infty \le \tilde{B}/2$ then return $\mathsf{ch} := 1$.
  Otherwise, return $\mathsf{ch} := \perp$.

- <u>$(\tilde{\mathsf{a}}, \tilde{\mathsf{z}}) \leftarrow_\$ \Sigma.\mathsf{Sim}(\mathsf{crs}, x, \mathsf{ch})$:</u> Parse $\mathsf{crs} = (\mathbf{A}, \mathbf{X})$ and $x = (\mathbf{c}, vk, d)$. Choose

$$\tilde{\mathbf{s}}_{mix} \leftarrow_\$ \mathbb{Z}_q^n, \ \tilde{\mathbf{e}}_{mix} \leftarrow_\$ D_{\mathbb{Z}^m,\zeta}, \ \tilde{\mathbf{R}}_{mix} \leftarrow_\$ D_{\mathbb{Z}^{m\times m},\zeta}, \ \tilde{\mathbf{k}}_{mix} \leftarrow_\$ D_{\mathbb{Z}^m,\zeta}, \ \tilde{e}'_{mix} \leftarrow_\$ D_{\mathbb{Z},\zeta'}.$$

Compute

$$\begin{aligned}
\tilde{\mathbf{c}}_0^\top &:= \big(\tilde{\mathbf{s}}_{mix}^\top \cdot \mathbf{A} + \tilde{\mathbf{e}}_{mix}^\top\big) - \mathsf{ch} \cdot \mathbf{c}^\top \bmod q,\\
\widetilde{vk}_0 &:= \big(\mathbf{X} \cdot \tilde{\mathbf{R}}_{mix} + \big(\begin{smallmatrix}\mathbf{0}\\ q\cdot\tilde{\mathbf{k}}_{mix}^\top\end{smallmatrix}\big)\big) - \mathsf{ch} \cdot vk \bmod q^2,\\
\tilde{d}_0 &:= (\mathbf{c}^\top \cdot \tilde{\mathbf{k}}_{mix} + \tilde{e}'_{mix}) - \mathsf{ch} \cdot d \bmod q.
\end{aligned}$$

Return $\tilde{\mathsf{a}} := (\tilde{\mathbf{c}}_0, \widetilde{vk}_0, \tilde{d}_0)$ and $\tilde{\mathsf{z}} := (\tilde{\mathbf{s}}_{mix}, \tilde{\mathbf{e}}_{mix}, \tilde{\mathbf{R}}_{mix}, \tilde{\mathbf{k}}_{mix}, \tilde{e}'_{mix})$ with probability $\tilde{\theta}$ and abort otherwise, where the probability $\tilde{\theta}$ is defined by $\tilde{\theta} := 1/(M^3 M')$ with $M := e^{12\sqrt{m}B/\zeta + mB^2/(2\zeta^2)}$ and $M' := e^{12B'/\zeta' + B'^2/(2\zeta'^2)}$.

Theorem 15 (Trapdoor Σ-protocol for $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})}$). *Let $\zeta = \sqrt{m}B \cdot \omega(\sqrt{\log m})$, $\zeta' = \omega(B')$, $\tilde{B} = 2 \cdot (\zeta\sqrt{m} \cdot \omega(\sqrt{\log \lambda}) + B)$, $\tilde{B}' = 2 \cdot (\zeta' \cdot \omega(\sqrt{\log \lambda}) + B')$ and $q \geq 5m\tilde{B}$. Then the above construction is a trapdoor Σ-protocol for the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ specified by (53) and (54).*

Proof of Theorem 15. By the choices of $\tilde{B} = 2 \cdot (\zeta\sqrt{m} \cdot \omega(\sqrt{\log \lambda}) + B)$ and $\tilde{B}' = 2 \cdot (\zeta' \cdot \omega(\sqrt{\log \lambda}) + B')$, according to Lemma 5, we know that $D_{\mathbb{Z}^m,\zeta}$ and $D_{\mathbb{Z}^{m\times m},\zeta}$ are $(\frac{\tilde{B}}{2} - B) = \zeta\sqrt{m} \cdot \omega(\sqrt{\log \lambda})$-bounded and $D_{\mathbb{Z},\zeta'}$ is $(\frac{\tilde{B}'}{2} - B')$ $= \zeta' \cdot \omega(\sqrt{\log \lambda})$-bounded, except with negligible probability.

Completeness. For any instance $x = (\mathbf{c}, vk, d) \in \mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}$ with witness $w = (\mathbf{s}, \mathbf{e}, \mathbf{R}, \mathbf{k}, e')$, any proof $\mathsf{a} = (\mathbf{c}_0, vk_0, d_0), \mathsf{z} = (\mathbf{s}_{mix}, \mathbf{e}_{mix}, \mathbf{R}_{mix}, \mathbf{k}_{mix}, e'_{mix})$ and state $\mathsf{st} = (\mathbf{s}_0, \mathbf{e}_0, \mathbf{R}_0, \mathbf{k}_0, e'_0)$ generated honestly by prover, we know that $\mathbf{e}_0 \leftarrow_\$ D_{\mathbb{Z}^m,\zeta}$, $\mathbf{R}_0 \leftarrow_\$ D_{\mathbb{Z}^{m\times m},\zeta}$, $\mathbf{k}_0 \leftarrow_\$ D_{\mathbb{Z}^m,\zeta}$ and $e'_0 \leftarrow_\$ D_{\mathbb{Z},\zeta'}$. The above analysis shows that $\mathbf{e}_0$, $\mathbf{R}_0$ and $\mathbf{k}_0$ are all $(\frac{\tilde{B}}{2} - B)$-bounded and e'_0 is $(\frac{\tilde{B}'}{2} - B')$-bounded except with negligible probability. Hence for any challenge $\mathsf{ch} \in \{0, 1\}$, it holds that $\|\mathbf{e}_{mix}\|_\infty \leq \|\mathbf{e}_0\|_\infty + \|\mathbf{e}\|_\infty \leq \tilde{B}/2$, $\|\mathbf{R}_{mix}\|_\infty \leq \|\mathbf{R}_0\|_\infty + \|\mathbf{R}\|_\infty \leq \tilde{B}/2$, $\|\mathbf{k}_{mix}\|_\infty \leq \|\mathbf{k}_0\|_\infty + \|\mathbf{k}\|_\infty \leq \tilde{B}/2$ and $|e'_{mix}| \leq |e'_0| + |e'| \leq \tilde{B}'/2$, except with negligible probability. Meanwhile, we have

$$\begin{aligned}
\mathbf{c}_0^\top + \mathsf{ch} \cdot \mathbf{c}^\top &= (\mathbf{s}_0^\top + \mathsf{ch} \cdot \mathbf{s}^\top)\mathbf{A} + ({\mathbf{e}_0}^\top + \mathsf{ch} \cdot \mathbf{e}^\top) = \mathbf{s}_{mix}^\top \cdot \mathbf{A} + \mathbf{e}_{mix}^\top \bmod q,\\
vk_0 + \mathsf{ch} \cdot vk &= \mathbf{X}(\mathbf{R}_0 + \mathsf{ch} \cdot \mathbf{R}) + \big(\begin{smallmatrix}\mathbf{0}\\ q(\mathbf{k}_0^\top + \mathsf{ch}\cdot\mathbf{k}^\top)\end{smallmatrix}\big) = \mathbf{X} \cdot \mathbf{R}_{mix} + \big(\begin{smallmatrix}\mathbf{0}\\ q\cdot\mathbf{k}_{mix}^\top\end{smallmatrix}\big) \bmod q^2,\\
d_0 + \mathsf{ch} \cdot d &= \mathbf{c}^\top(\mathbf{k}_0 + \mathsf{ch} \cdot \mathbf{k}) + (e'_0 + \mathsf{ch} \cdot e') = \mathbf{c}^\top \cdot \mathbf{k}_{mix} + e'_{mix} \bmod q.
\end{aligned}$$

Therefore, the verification passes except with negligible probability.

Special Soundness. Special soundness requires that for any $x \notin \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})}$ and any first message a, there exists at most one challenge $\mathsf{ch} \in \{0, 1\}$ such that $\Sigma.\mathsf{Vrfy}(\mathsf{crs}, x, \mathsf{a}, \mathsf{ch}, \mathsf{z}) = 1$ for some third message z. Suppose, toward contradiction, there exist $x = (\mathbf{c}, vk, d) \notin \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})}$ and $\mathsf{a} = (\mathbf{c}_0, vk_0, d_0)$ such that $\Sigma.\mathsf{Vrfy}(\mathsf{crs}, x, \mathsf{a}, \mathsf{ch}, \mathsf{z}^{(\mathsf{ch})}) = 1$ for both $\mathsf{ch} = 0$ and $\mathsf{ch} = 1$ for some $\mathsf{z}^{(0)} = (\mathbf{s}_{mix}^{(0)}, \mathbf{e}_{mix}^{(0)}, \mathbf{R}_{mix}^{(0)}, \mathbf{k}_{mix}^{(0)}, e_{mix}'^{(0)})$ and $\mathsf{z}^{(1)} = (\mathbf{s}_{mix}^{(1)}, \mathbf{e}_{mix}^{(1)}, \mathbf{R}_{mix}^{(1)}, \mathbf{k}_{mix}^{(1)}, e_{mix}'^{(1)})$. That is, for both $\mathsf{ch} = 0$ and $\mathsf{ch} = 1$ it holds that $\left\|\mathbf{e}_{mix}^{(\mathsf{ch})}\right\|_\infty \leq \tilde{B}/2$, $\left\|\mathbf{R}_{mix}^{(\mathsf{ch})}\right\|_\infty \leq \tilde{B}/2$,

$\left\|\mathbf{k}_{mix}^{(\mathsf{ch})}\right\|_\infty \leq \tilde{B}/2$, $|e'^{(\mathsf{ch})}_{mix}| \leq \tilde{B}'/2$ and

$$\begin{aligned}
\mathbf{c}_0^\top + 0 \cdot \mathbf{c}^\top &= \mathbf{s}_{mix}^{(0)}{}^\top \cdot \mathbf{A} + \mathbf{e}_{mix}^{(0)}{}^\top \bmod q, && (57)\\
\mathbf{c}_0^\top + 1 \cdot \mathbf{c}^\top &= \mathbf{s}_{mix}^{(1)}{}^\top \cdot \mathbf{A} + \mathbf{e}_{mix}^{(1)}{}^\top \bmod q, && (58)\\
vk_0 + 0 \cdot vk &= \mathbf{X} \cdot \mathbf{R}_{mix}^{(0)} + \left(\begin{smallmatrix}\mathbf{0}\\ q\cdot\mathbf{k}_{mix}^{(0)}{}^\top\end{smallmatrix}\right) \bmod q^2, && (59)\\
vk_0 + 1 \cdot vk &= \mathbf{X} \cdot \mathbf{R}_{mix}^{(1)} + \left(\begin{smallmatrix}\mathbf{0}\\ q\cdot\mathbf{k}_{mix}^{(1)}{}^\top\end{smallmatrix}\right) \bmod q^2, && (60)\\
d_0 + 0 \cdot d &= \mathbf{c}^\top \cdot \mathbf{k}_{mix}^{(0)} + e'^{(0)}_{mix} \bmod q, && (61)\\
d_0 + 1 \cdot d &= \mathbf{c}^\top \cdot \mathbf{k}_{mix}^{(1)} + e'^{(1)}_{mix} \bmod q. && (62)
\end{aligned}$$

By subtracting (57) from (58), (59) from (60), and (61) from (62) respectively, we have

$$\begin{aligned}
\mathbf{c}^\top &= \left(\mathbf{s}_{mix}^{(1)} - \mathbf{s}_{mix}^{(0)}\right)^\top \cdot \mathbf{A} + \left(\mathbf{e}_{mix}^{(1)} - \mathbf{e}_{mix}^{(0)}\right)^\top \bmod q,\\
vk &= \mathbf{X} \cdot (\mathbf{R}_{mix}^{(1)} - \mathbf{R}_{mix}^{(0)}) + \left(\begin{smallmatrix}\mathbf{0}\\ q\cdot(\mathbf{k}_{mix}^{(1)} - \mathbf{k}_{mix}^{(0)})^\top\end{smallmatrix}\right) \bmod q^2,\\
d &= \mathbf{c}^\top \cdot (\mathbf{k}_{mix}^{(1)} - \mathbf{k}_{mix}^{(0)}) + (e'^{(1)}_{mix} - e'^{(0)}_{mix}) \bmod q.
\end{aligned}$$

Note that $\left\|\mathbf{e}_{mix}^{(1)} - \mathbf{e}_{mix}^{(0)}\right\|_\infty \leq \left\|\mathbf{e}_{mix}^{(1)}\right\|_\infty + \left\|\mathbf{e}_{mix}^{(0)}\right\|_\infty \leq \tilde{B}$, $\left\|\mathbf{R}_{mix}^{(1)} - \mathbf{R}_{mix}^{(0)}\right\|_\infty \leq \left\|\mathbf{R}_{mix}^{(1)}\right\|_\infty + \left\|\mathbf{R}_{mix}^{(0)}\right\|_\infty \leq \tilde{B}$, $\left\|\mathbf{k}_{mix}^{(1)} - \mathbf{k}_{mix}^{(0)}\right\|_\infty \leq \left\|\mathbf{k}_{mix}^{(1)}\right\|_\infty + \left\|\mathbf{k}_{mix}^{(0)}\right\|_\infty \leq \tilde{B}$ and $|e'^{(1)}_{mix} - e'^{(0)}_{mix}| \leq |e'^{(1)}_{mix}| + |e'^{(0)}_{mix}| \leq \tilde{B}'$. As a result, $\left(\mathbf{s}_{mix}^{(1)} - \mathbf{s}_{mix}^{(0)}, \mathbf{e}_{mix}^{(1)} - \mathbf{e}_{mix}^{(0)}, \mathbf{R}_{mix}^{(1)} - \mathbf{R}_{mix}^{(0)}, \mathbf{k}_{mix}^{(1)} - \mathbf{k}_{mix}^{(0)}, e'^{(1)}_{mix} - e'^{(0)}_{mix}\right)$ constitutes a witness for $x = (\mathbf{c}, vk, d) \in \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})}$, which yields a contradiction.

Correctness of Σ.$\mathsf{BadChallenge}$. For any $x = (\mathbf{c}, vk, d) \notin \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})}$ and any $\mathsf{a} = (\mathbf{c}_0, vk_0, d_0)$, if the bad challenge function $f(\mathsf{crs}, x, \mathsf{a}) \neq \perp$, we aim to prove that $\Sigma.\mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}_\Sigma, x, a) = f(\mathsf{crs}, x, a)$. Suppose that $f(\mathsf{crs}, x, a) = \mathsf{ch}$ for some $\mathsf{ch} \in \{0, 1\}$, then by the definition of f, ch is the unique challenge such that $\Sigma.\mathsf{Vrfy}(\mathsf{crs}, x, \mathsf{a}, \mathsf{ch}, \mathsf{z}) = 1$ for some $\mathsf{z} = (\mathbf{s}_{mix}, \mathbf{e}_{mix}, \mathbf{R}_{mix}, \mathbf{k}_{mix}, e'_{mix})$. Thus $\|\mathbf{e}_{mix}\|_\infty \leq \tilde{B}/2$ and $\mathbf{c}_0^\top + \mathsf{ch} \cdot \mathbf{c}^\top = \mathbf{s}_{mix}^\top \cdot \mathbf{A} + \mathbf{e}_{mix}^\top$. Note that $q \geq 5m\tilde{B}$, so $\|\mathbf{e}_{mix}\| \leq \sqrt{m}\tilde{B}/2 \leq q/(10\sqrt{m})$. According to Lemma 4, it must hold that $\mathsf{Invert}(\mathbf{T}_\mathbf{A}, \mathbf{c}_0 + \mathsf{ch} \cdot \mathbf{c}) = (\mathbf{s}_{mix}, \mathbf{e}_{mix})$. Consequently, $\Sigma.\mathsf{BadChallenge}(\mathsf{crs}, \mathsf{td}_\Sigma, x, \mathsf{a})$ outputs ch, the same as $f(\mathsf{crs}, x, \mathsf{a})$. The correctness of $\Sigma.\mathsf{BadChallenge}$ follows.

Special Zero-Knowledge. We aim to bound the statistical distance between the real proof $(\mathsf{a} = (\mathbf{c}_0, vk_0, d_0), \mathsf{z} = (\mathbf{s}_{mix}, \mathbf{e}_{mix}, \mathbf{R}_{mix}, \mathbf{k}_{mix}, e'_{mix}))$ and the simulated proof $(\tilde{\mathsf{a}} = (\tilde{\mathbf{c}}_0, \widetilde{vk}_0, \tilde{d}_0), \tilde{\mathsf{z}} = (\tilde{\mathbf{s}}_{mix}, \tilde{\mathbf{e}}_{mix}, \tilde{\mathbf{R}}_{mix}, \tilde{\mathbf{k}}_{mix}, \tilde{e}'_{mix}))$ for any instance $x = (\mathbf{c}, vk, d) \in \mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}$ with witness $w = (\mathbf{s}, \mathbf{e}, \mathbf{R}, \mathbf{k}, e')$.

Note that both the real proof and the simulated proof satisfy the verification equations, i.e.,

$$\begin{aligned}
&\mathbf{c}_0^\top + \mathsf{ch} \cdot \mathbf{c}^\top = \mathbf{s}_{mix}^\top \cdot \mathbf{A} + \mathbf{e}_{mix}^\top \bmod q, \\
&vk_0 + \mathsf{ch} \cdot vk = \mathbf{X} \cdot \mathbf{R}_{mix} + \left(\begin{smallmatrix} \mathbf{0} \\ q \cdot \mathbf{k}_{mix}^\top \end{smallmatrix}\right) \bmod q^2, \\
&d_0 + \mathsf{ch} \cdot d = \mathbf{c}^\top \cdot \mathbf{k}_{mix} + e'_{mix} \bmod q.
\end{aligned}$$

Therefore, in the real proof (a, z), $\mathsf{a} = (\mathbf{c}_0, vk_0, d_0)$ is completely determined by $\mathsf{z} = (\mathbf{s}_{mix}, \mathbf{e}_{mix}, \mathbf{R}_{mix}, \mathbf{k}_{mix}, e'_{mix})$, $\mathsf{crs} = (\mathbf{A}, \mathbf{X})$, $x = (\mathbf{c}, vk, d)$ and ch, and in the simulated proof $(\tilde{\mathsf{a}}, \tilde{\mathsf{z}})$, $\tilde{\mathsf{a}} = (\tilde{\mathbf{c}}_0, \widetilde{vk}_0, \tilde{d}_0)$ is completely determined by $\tilde{\mathsf{z}} = (\tilde{\mathbf{s}}_{mix}, \tilde{\mathbf{e}}_{mix}, \tilde{\mathbf{R}}_{mix}, \tilde{\mathbf{k}}_{mix}, \tilde{e}'_{mix})$, $\mathsf{crs} = (\mathbf{A}, \mathbf{X})$, $x = (\mathbf{c}, vk, d)$ and ch in the same way as the real proof.

So the difference between the real proof and the simulated proof lies in the distribution of z and $\tilde{\mathsf{z}}$, where $\mathsf{z} = (\mathbf{s}_{mix}, \mathbf{e}_{mix}, \mathbf{R}_{mix}, \mathbf{k}_{mix}, e'_{mix})$ is generated by first sampling

$$\mathbf{s}_0 \leftarrow_{\$} \mathbb{Z}_q^n, \ \mathbf{e}_0 \leftarrow_{\$} D_{\mathbb{Z}^m,\zeta}, \ \mathbf{R}_0 \leftarrow_{\$} D_{\mathbb{Z}^{m \times m},\zeta}, \ \mathbf{k}_0 \leftarrow_{\$} D_{\mathbb{Z}^m,\zeta}, \ e'_0 \leftarrow_{\$} D_{\mathbb{Z},\zeta'},$$

then computing

$$\begin{aligned}
&\mathbf{s}_{mix} := \mathbf{s}_0 + \mathsf{ch} \cdot \mathbf{s} \bmod q, \quad \mathbf{e}_{mix} := \mathbf{e}_0 + \mathsf{ch} \cdot \mathbf{e} \bmod q, \\
&\mathbf{R}_{mix} := \mathbf{R}_0 + \mathsf{ch} \cdot \mathbf{R} \bmod q^2, \quad \mathbf{k}_{mix} := \mathbf{k}_0 + \mathsf{ch} \cdot \mathbf{k} \bmod q, \\
&e'_{mix} := e'_0 + \mathsf{ch} \cdot e' \bmod q,
\end{aligned}$$

while $\tilde{\mathsf{z}} = (\tilde{\mathbf{s}}_{mix}, \tilde{\mathbf{e}}_{mix}, \tilde{\mathbf{R}}_{mix}, \tilde{\mathbf{k}}_{mix}, \tilde{e}'_{mix})$ is sampled directly via

$$\tilde{\mathbf{s}}_{mix} \leftarrow_{\$} \mathbb{Z}_q^n, \ \tilde{\mathbf{e}}_{mix} \leftarrow_{\$} D_{\mathbb{Z}^m,\zeta}, \ \tilde{\mathbf{R}}_{mix} \leftarrow_{\$} D_{\mathbb{Z}^{m \times m},\zeta}, \ \tilde{\mathbf{k}}_{mix} \leftarrow_{\$} D_{\mathbb{Z}^m,\zeta}, \ \tilde{e}'_{mix} \leftarrow_{\$} D_{\mathbb{Z},\zeta'}.$$

Moreover, recall that the real proof (a, z) is outputted with probability θ, while the simulated proof $(\tilde{\mathsf{a}}, \tilde{\mathsf{z}})$ is outputted with probability $\tilde{\theta}$. Our analysis is as follows.

- Firstly, $\mathbf{s}_{mix} = \mathbf{s}_0 + \mathsf{ch} \cdot \mathbf{s}$ in z and $\tilde{\mathbf{s}}_{mix}$ in $\tilde{\mathsf{z}}$ are both uniform over $\mathbb{Z}_q^n$.
- Secondly, note that $\|\mathsf{ch} \cdot \mathbf{e}\| \leq \sqrt{m}B$ and $\zeta = \sqrt{m}B \cdot \omega(\sqrt{\log m})$, Lemma 14 shows that $\mathbf{e}_{mix} = \mathbf{e}_0 + \mathsf{ch} \cdot \mathbf{e}$ in z – when output with probability $\min\left(\frac{D_{\mathbb{Z}^m,\zeta}(\mathbf{e}_{mix})}{M \cdot D_{\mathbb{Z}^m,\zeta,\mathsf{ch}\cdot\mathbf{e}}(\mathbf{e}_{mix})}, 1\right)$ – and $\tilde{\mathbf{e}}_{mix}$ in $\tilde{\mathsf{z}}$ – when output with probability $1/M$ – have statistical distance at most $2^{-100}/M$.
- Similarly, note that $\|\mathsf{ch} \cdot \mathbf{R}\| \leq \sqrt{m}$, $\|\mathsf{ch} \cdot \mathbf{k}\| \leq \sqrt{m}$, $|\mathsf{ch} \cdot e'| \leq B'$, $\zeta = \sqrt{m}B \cdot \omega(\sqrt{\log m})$ and $\zeta' = \omega(B')$, Lemma 14 shows that $\mathbf{R}_{mix}$ (resp., $\mathbf{k}_{mix}$, resp., e'_{mix}) in z – when output with probability $\min\left(\frac{D_{\mathbb{Z}^{m \times m},\zeta}(\mathbf{R}_{mix})}{M \cdot D_{\mathbb{Z}^{m \times m},\zeta,\mathsf{ch}\cdot\mathbf{R}}(\mathbf{R}_{mix})}, 1\right)$ (resp., $\min\left(\frac{D_{\mathbb{Z}^m,\zeta}(\mathbf{k}_{mix})}{M \cdot D_{\mathbb{Z}^m,\zeta,\mathsf{ch}\cdot\mathbf{k}}(\mathbf{k}_{mix})}, 1\right)$, resp., $\min\left(\frac{D_{\mathbb{Z},\zeta'}(e'_{mix})}{M' \cdot D_{\mathbb{Z},\zeta',ch\cdot e'}(e'_{mix})}, 1\right)$) – and $\tilde{\mathbf{R}}_{mix}$ (resp., $\tilde{\mathbf{k}}_{mix}$, resp., $\tilde{e}'_{mix}$) in $\tilde{\mathsf{z}}$ – when output with probability $1/M$ (resp., $1/M$, resp., $1/M'$) – have statistical distance at most $2^{-100}/M$ (resp., $2^{-100}/M$, resp., $2^{-100}/M'$).

Overall, $\mathsf{z} = (\mathbf{s}_{mix}, \mathbf{e}_{mix}, \mathbf{R}_{mix}, \mathbf{k}_{mix}, e'_{mix})$ – when output with probability $\theta = \min\left(\frac{D_{\mathbb{Z}^m,\zeta}(\mathbf{e}_{mix})}{M \cdot D_{\mathbb{Z}^m,\zeta,\mathsf{ch}\cdot\mathbf{e}}(\mathbf{e}_{mix})} \cdot \frac{D_{\mathbb{Z}^{m\times m},\zeta}(\mathbf{R}_{mix})}{M \cdot D_{\mathbb{Z}^{m\times m},\zeta,\mathsf{ch}\cdot\mathbf{R}}(\mathbf{R}_{mix})} \cdot \frac{D_{\mathbb{Z}^m,\zeta}(\mathbf{k}_{mix})}{M \cdot D_{\mathbb{Z}^m,\zeta,\mathsf{ch}\cdot\mathbf{k}}(\mathbf{k}_{mix})} \cdot \frac{D_{\mathbb{Z},\zeta'}(e'_{mix})}{M' \cdot D_{\mathbb{Z},\zeta',ch\cdot e'}(e'_{mix})}, 1\right)$ – and $\tilde{\mathsf{z}} = (\tilde{\mathbf{s}}_{mix}, \tilde{\mathbf{e}}_{mix}, \tilde{\mathbf{R}}_{mix}, \tilde{\mathbf{k}}_{mix}, \tilde{e}'_{mix})$ – when output with probability $\tilde{\theta} = 1/(M^3 M')$ – have statistical distance at most $2^{-100} \cdot (3/M + 1/M')$.

This completes the proof of special zero-knowledge.

<u>*Perfect CRS Indistinguishability.*</u> On input of language parameter $\rho' = (\mathbf{A}, \mathbf{X})$, both $\mathsf{crs} \leftarrow_{\$} \Sigma.\mathsf{CRSGen}(\rho')$ and $(\mathsf{crs}, \mathsf{td}_{\Sigma}) \leftarrow_{\$} \Sigma.\mathsf{TrapGen}(\rho', td_{\rho'})$ simply set $\mathsf{crs} := (\mathbf{A}, \mathbf{X})$. So perfect CRS indistinguishability trivially holds. □

The Trapdoor Σ-protocol for $\mathcal{GL}_\rho$. For the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ defined in Subsect. 6.1 and specified by (55) and (56), the LWE-based trapdoor Σ-protocol is just a simplified version of that for $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})}$, since it only needs to prove the instance $\mathbf{c}$ satisfies $\mathbf{c}^\top = \mathbf{s}^\top \cdot \mathbf{A} + \mathbf{e}^\top$ with witness $(\mathbf{s}, \mathbf{e})$. As a result, $\mathsf{crs} := \mathbf{A}$; $\Sigma.\mathsf{Prove}_1$ outputs $\mathsf{a} := \mathbf{c}_0$ and $\mathsf{st} := (\mathbf{s}_0, \mathbf{e}_0)$; $\Sigma.\mathsf{Prove}_2$ outputs $\mathsf{z} := (\mathbf{s}_{mix}, \mathbf{e}_{mix})$ with probability $\theta = \min\left(\frac{D_{\mathbb{Z}^m,\zeta}(\mathbf{e}_{mix})}{M \cdot D_{\mathbb{Z}^m,\zeta,\mathsf{ch}\cdot\mathbf{e}}(\mathbf{e}_{mix})}, 1\right)$; $\Sigma.\mathsf{Vrfy}$ only checks $\|\mathbf{e}_{mix}\|_\infty \le \tilde{B}/2$ and $\mathbf{c}_0^\top + \mathsf{ch} \cdot \mathbf{c}^\top = \mathbf{s}_{mix}^\top \cdot \mathbf{A} + \mathbf{e}_{mix}^\top \bmod q$; $\Sigma.\mathsf{Sim}$ outputs $\tilde{\mathsf{a}} := \tilde{\mathbf{c}}_0$ and $\tilde{\mathsf{z}} = (\tilde{\mathbf{s}}_{mix}, \tilde{\mathbf{e}}_{mix})$ with probability $\tilde{\theta} = 1/M$.

Similarly, we have the following theorem. Its proof is a simplified version of that for Theorem 15, thus we omit it.

Theorem 16 (Trapdoor Σ-protocol for $\mathcal{GL}_\rho$). *Let $\zeta = \sqrt{m}B \cdot \omega(\sqrt{\log m})$, $\tilde{B} = 2 \cdot (\zeta\sqrt{m} \cdot \omega(\sqrt{\log \lambda}) + B)$ and $q \ge 5m\tilde{B}$. Then the above construction is a trapdoor Σ-protocol for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ defined in Subsect. 6.1 and specified by (55) and (56).*

G.4 Generic QA-NIZK Transformation and QA-NIZK from LWE

In this subsection, we will use the generic QA-NIZK transformation in [34, Subsect. 4.2] to convert the LWE-based trapdoor Σ-protocols proposed in the previous subsection (Appendix G.3) to LWE-based QA-NIZK schemes, which in turn serve as building blocks for our SIG and PKE constructions in Sect. 4.

To this end, we will first recall the generic QA-NIZK transformation in [34, Subsect. 4.2] for completeness, then describe how to compile our LWE-based trapdoor Σ-protocols proposed in the previous subsection (Appendix G.3) into LWE-based QA-NIZK schemes via the generic transformation.

The Generic QA-NIZK Transformation in [34, Subsect. 4.2]. The generic transformation proposed by Libert et al. in [34, Subsect. 4.2] is able to compile any trapdoor Σ-protocol for gap language into tag-based QA-NIZK for the same gap language, with the help of correlation intractable (CI) hash function and lossy PKE. Moreover, the transformation is tightness-preserving, i.e., the resulting tag-based QA-NIZK has tight zero-knowledge and tight USS as long as the building blocks are tightly secure.

We recall the generic transformation for completeness. To construct a tag-based QA-NIZK scheme for a gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$, the underlying building blocks are as follows.

- A trapdoor Σ-protocol $\Sigma = (\Sigma.\mathsf{CRSGen}, \Sigma.\mathsf{Prove}_1, \Sigma.\mathsf{Prove}_2, \Sigma.\mathsf{Vrfy}, \Sigma.\mathsf{Sim}, \Sigma.\mathsf{TrapGen}, \Sigma.\mathsf{BadChallenge})$ for the same gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$.
- A pseudorandom function $\mathsf{PRF} : \mathcal{K} \times \{0,1\}^\ell \rightarrow \{0,1\}^\lambda$ with key space $\mathcal{K}$ and input space $\{0,1\}^\ell$. PRF define a relation $R_{\mathsf{PRF}} : \mathcal{K} \times \{0,1\}^\ell \times \{0,1\}^\lambda \rightarrow \{0,1\}$ with $\mathcal{K} = \{0,1\}^\lambda$, where $R_{\mathsf{PRF}}(K, t_a, t_c) = 1$ iff $t_c \neq \mathsf{PRF}(K, t_a)$. (The syntax, security requirements and specific construction of PRF are recalled in Definition 28.)
- An R_{PRF}-lossy PKE scheme R_{PRF}-$\mathsf{LPKE} = (\mathsf{LPKE.Gen}, \mathsf{LPKE.LGen}, \mathsf{LPKE.Enc}, \mathsf{LPKE.Dec}, \mathsf{LPKE.Opener}, \mathsf{LPKE.LOpener})$ for the relation R_{PRF} with tag space $\mathcal{T} = \{0,1\}^\lambda \times \{0,1\}^\ell$, randomness space $\mathcal{R}_{\mathsf{LPKE}}$, message space $\mathcal{M}$, ciphertext space $\mathcal{CT}$ and randomness distribution $D_{\mathcal{R}_{\mathsf{LPKE}}}$ over $\mathcal{R}_{\mathsf{LPKE}}$. (The syntax, security requirements and specific construction of lossy PKE are recalled in Definition 27.)
- A somewhere correlation intractable (CI) hash $\mathsf{CIH} = (\mathsf{CIH.Gen}, \mathsf{CIH.StGen})$ which is associate with efficiently computable keyed hash family $\mathcal{H} = \{h : \mathcal{K}' \times \mathcal{X} \times \mathcal{CT}^\lambda \times \{0,1\}^\lambda \times \{0,1\}^\ell \rightarrow \{0,1\}^\lambda\}$. (The syntax, security requirements and specific construction of CI hash are recalled in Definition 26.)
- A one-time signature scheme $\mathsf{OTS} = (\mathsf{OTS.Setup}, \mathsf{OTS.Gen}, \mathsf{OTS.Sign}, \mathsf{OTS.Vrfy})$ with verification key space $\mathcal{VK} = \{0,1\}^\ell$. (The syntax is the same as signature, see Definition 13. The security requirements and specific construction of one-time signature are recalled in Definition 29)

The generic construction of tag-based QA-NIZK scheme $\mathsf{QANIZK} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{SimGen}, \mathsf{Sim})$ for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ proposed in [34, Subsect. 4.2] is presented in Fig. 10.

In [34], Libert et al. proved the tightness-preserving of the transformation, i.e., the resulting tag-based QA-NIZK scheme has tight zero-knowledge and tight USS as long as the building blocks are tightly secure. Formally, we recall the following theorem from [34].

Theorem 17 ([34]). *The ZK and USS of the generic tag-based QA-NIZK construction for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ proposed in Fig. 10 can be tightly reduced to the security and property of the underlying building blocks: (1) The security of the trapdoor Σ-protocol; (2) The pseudorandomness of* PRF*; (3) The security of* CIH*; (4) The key indistinguishability of R-*LPKE*; (5) The strong* MU-OT *security of* OTS*.*

Concretely, if the trapdoor Σ-protocol has special zero-knowledge with statistical distance at most $\varepsilon_{\mathsf{zk}}$, then the advantage of zero-knowledge for any (even all powerful) adversary $\mathcal{A}'$ is given by $\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{A}'}(\lambda) \leq \varepsilon_{\mathsf{zk}} + 2^{-\Omega(\lambda)}$. Meanwhile, the advantage of USS for any PPT adversary $\mathcal{A}$ is given by

$$\begin{aligned}\mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{str\text{-}ot}}_{\mathsf{OTS},\mathcal{B}_1,Q}(\lambda) + \mathsf{Adv}^{\mathsf{ind\text{-}1}}_{R\text{-}\mathsf{LPKE},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{ind}}_{\mathsf{CIH},\mathcal{B}_3}(\lambda)\\ + \mathsf{Adv}^{\mathsf{ind\text{-}2}}_{R\text{-}\mathsf{LPKE},\mathcal{B}_4}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{pse}}_{\mathsf{PRF},\mathcal{B}_5}(\lambda) + 2^{-\Omega(\lambda)},\end{aligned}$$

```
$\mathsf{crs} \leftarrow_{\$} \mathsf{CRSGen}(\rho)$:
$\mathsf{crs}' \leftarrow_{\$} \Sigma.\mathsf{CRSGen}(\rho)$, $(pk, sk, tk) \leftarrow_{\$} \mathsf{LPKE.LGen}(0^{\lambda})$.
$k \leftarrow_{\$} \mathsf{CIH.Gen}$, $\mathsf{pp}_{\mathsf{SIG}} \leftarrow_{\$} \mathsf{OTS.Setup}$.
Return $\mathsf{crs} := (\mathsf{crs}', pk, k, \mathsf{pp}_{\mathsf{SIG}})$.
```

```
$\pi \leftarrow_{\$} \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$:
Parse $\mathsf{crs} := (\mathsf{crs}', pk, k, \mathsf{pp}_{\mathsf{SIG}})$.
$(vk, sigk) \leftarrow_{\$} \mathsf{OTS.Gen}(\mathsf{pp}_{\mathsf{SIG}})$.
$t_c \leftarrow_{\$} \{0,1\}^{\lambda}$.
For all $i \in [\lambda]$:
    $(\mathsf{a}'_i, \mathsf{st}_i) \leftarrow_{\$} \Sigma.\mathsf{Prove}_1(\mathsf{crs}', x, w)$.
    $r_i \leftarrow_{\$} D_{\mathcal{R}_{\mathsf{LPKE}}}$.
    $\mathsf{a}_i \leftarrow_{\$} \mathsf{LPKE.Enc}(pk, (t_c, vk), \mathsf{a}'_i; r_i)$.
$\mathbf{a}' := (\mathsf{a}'_1, \cdots, \mathsf{a}'_{\lambda})$.
$\mathbf{r} := (r_1, \cdots, r_{\lambda})$.
$\mathbf{a} := (\mathsf{a}_1, \cdots, \mathsf{a}_{\lambda})$.
$\mathsf{ch} := h(k, (x, \mathbf{a}, t_c, vk)) \in \{0,1\}^{\lambda}$.
Parse $\mathsf{ch} = (\mathsf{ch}_1, \ldots, \mathsf{ch}_{\lambda}) \in \{0,1\}^{\lambda}$
For all $i \in [\lambda]$:
    $\mathsf{z}'_i \leftarrow_{\$} \Sigma.\mathsf{Prove}_2(\mathsf{st}_i, \mathsf{a}'_i, \mathsf{ch}_i)$.
$\mathbf{z}' := (\mathsf{z}'_1, \cdots, \mathsf{z}'_{\lambda})$.
$\mathbf{z} := (\mathbf{z}', \mathbf{a}', \mathbf{r})$.
$\sigma \leftarrow_{\$} \mathsf{OTS.Sign}(sigk, (x, t_c, \mathbf{a}, \mathbf{z}, \tau))$.
Return $\pi := ((t_c, vk), (\mathbf{a}, \mathbf{z}), \sigma)$.
```

```
$0/1 \leftarrow \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi)$:
Parse $\mathsf{crs} = (\mathsf{crs}', pk, k, \mathsf{pp}_{\mathsf{SIG}})$.
Parse $\pi = ((t_c, vk), (\mathbf{a}, \mathbf{z}), \sigma)$.
If $\mathsf{OTS.Vrfy}(vk, (x, t_c, \mathbf{a}, \mathbf{z}, \tau), \sigma) \neq 1$:
    Return 0.
$\mathsf{ch} := h(k, (x, \mathbf{a}, t_c, vk)) \in \{0,1\}^{\lambda}$.
Parse $\mathsf{ch} = (\mathsf{ch}_1, \ldots, \mathsf{ch}_{\lambda}) \in \{0,1\}^{\lambda}$.
Parse $\mathbf{a} = (\mathsf{a}_1, \cdots, \mathsf{a}_{\lambda})$.
Parse $\mathbf{z} = (\mathbf{z}', \mathbf{a}', \mathbf{r})$.
Parse $\mathbf{z}' = (\mathsf{z}'_1, \cdots, \mathsf{z}'_{\lambda})$.
Parse $\mathbf{a}' = (\mathsf{a}'_1, \cdots, \mathsf{a}'_{\lambda})$.
Parse $\mathbf{r} = (r_1, \cdots, r_{\lambda})$.
If for all $i \in [\lambda]$:
    $\mathsf{a}_i = \mathsf{LPKE.Enc}(pk, (t_c, vk), \mathsf{a}'_i; r_i)$
    and $\Sigma.\mathsf{Vrfy}(\mathsf{crs}', x, \mathsf{a}'_i, \mathsf{ch}_i, \mathsf{z}'_i) = 1$ :
        Return 1;
Else: Return 0.
```

```
$(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{SimGen}(\rho)$:
$\mathsf{crs}' \leftarrow_{\$} \Sigma.\mathsf{CRSGen}(\rho)$.
$(pk, sk, tk) \leftarrow_{\$} \mathsf{LPKE.LGen}(0^{\lambda})$.
$k \leftarrow_{\$} \mathsf{CIH.Gen}$.
$\mathsf{pp}_{\mathsf{SIG}} \leftarrow_{\$} \mathsf{OTS.Setup}$.
$\mathsf{crs} := (\mathsf{crs}', pk, k, \mathsf{pp}_{\mathsf{SIG}})$.
$\mathsf{td}_{\mathsf{crs}} := sk$.
Return $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}})$.
```

```
$\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}} = sk, \tau, x)$:
Parse $\mathsf{crs} = (\mathsf{crs}', pk, k, \mathsf{pp}_{\mathsf{SIG}})$.
$(vk, sigk) \leftarrow_{\$} \mathsf{OTS.Gen}(\mathsf{pp}_{\mathsf{SIG}})$.
$t_c \leftarrow_{\$} \{0,1\}^{\lambda}$.
For all $i \in [\lambda]$:
    $r_{i,0} \leftarrow_{\$} D_{\mathcal{R}_{\mathsf{LPKE}}}$.
    $\mathsf{a}_i \leftarrow_{\$} \mathsf{LPKE.Enc}(pk, (t_c, vk), 0; r_{i,0})$.
$\mathbf{a} := (\mathsf{a}_1, \cdots, \mathsf{a}_{\lambda})$.
$\mathsf{ch} := h(k, (x, \mathbf{a}, t_c, vk)) \in \{0,1\}^{\lambda}$.
Parse $\mathsf{ch} = (\mathsf{ch}_1, \ldots, \mathsf{ch}_{\lambda}) \in \{0,1\}^{\lambda}$.
For all $i \in [\lambda]$:
    $(\mathsf{a}'_i, \mathsf{z}'_i) \leftarrow_{\$} \Sigma.\mathsf{Sim}(\mathsf{crs}', x, \mathsf{ch}_i)$.
    $r_i \leftarrow_{\$} \mathsf{LPKE.LOpener}(sk, (t_c, vk), \mathsf{a}_i, \mathsf{a}'_i)$.
$\mathbf{a}' := (\mathsf{a}'_1, \cdots, \mathsf{a}'_{\lambda})$.
$\mathbf{z}' := (\mathsf{z}'_1, \cdots, \mathsf{z}'_{\lambda})$.
$\mathbf{r} := (r_1, \cdots, r_{\lambda})$.
$\mathbf{z} := (\mathbf{z}', \mathbf{a}', \mathbf{r})$.
$\sigma \leftarrow_{\$} \mathsf{OTS.Sign}(sigk, (x, t_c, \mathbf{a}, \mathbf{z}, \tau))$.
Return $\pi := ((t_c, vk), (\mathbf{a}, \mathbf{z}), \sigma)$.
```

Fig. 10. The generic construction of $\mathsf{QANIZK} = (\mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{SimGen}, \mathsf{Sim})$ for the gap language $\mathcal{GL}_{\rho} = (\mathcal{L}_{\rho}, \widetilde{\mathcal{L}}_{\rho})$ from trapdoor Σ-protocol, R_{PRF}-LPKE, CIH and OTS, proposed in [34, Subsect. 4.2].

where Q is the number of oracle queries by $\mathcal{A}$ and PPT algorithms $\mathcal{B}_1, \cdots, \mathcal{B}_5$ run in about the same time as $\mathcal{A}$.

QA-NIZK from LWE. Finally, by compiling the LWE-based trapdoor Σ-protocols proposed in Appendix G.3 with the help of the instantiations of other building blocks in Appendix G.1 via the generic transformation proposed by Libert et al. in [34, Subsect. 4.2], we are able to obtain a tag-based QA-NIZK scheme for the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ specified by (53) and (54) and a tag-based QA-NIZK scheme for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ specified by (55) and (56) based on the LWE assumptions, serving as building blocks for our SIG and PKE constructions. Formally, we have the following corollary.

Corollary 1 (Almost Tight Security of LWE-based QA-NIZK). *Given the instantiations of the building blocks in Appendix G.1 and the instantiations of trapdoor Σ-protocol in Appendix G.3, we obtain a specific tag-based QA-NIZK scheme for the gap language $\mathcal{GL}_{\rho'}^{(\mathsf{QANIZK})} = (\mathcal{L}_{\rho'}^{(\mathsf{QANIZK})}, \widetilde{\mathcal{L}}_{\rho'}^{(\mathsf{QANIZK})})$ specified by (53) and (54) and a specific tag-based QA-NIZK scheme for the gap language $\mathcal{GL}_\rho = (\mathcal{L}_\rho, \widetilde{\mathcal{L}}_\rho)$ specified by (55) and (56), both of which have almost tight zero-knowledge and USS based on the LWE assumption.*

Concretely, the advantage of zero-knowledge for any (even all powerful) adversary $\mathcal{A}'$ is given by $\mathsf{Adv}^{\mathsf{zk}}_{\mathsf{QANIZK},\mathcal{A}'}(\lambda) \leq 2^{-\Omega(\lambda)}$. Meanwhile, the advantage of USS for any PPT adversary $\mathcal{A}$ is given by

$$\mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SIS}}_{[n,q,m,\beta],\mathcal{B}_1}(\lambda) + 2\lambda^2 \cdot \mathsf{Adv}^{\mathsf{LWE}}_{[\lambda,q,\chi,m],\mathcal{B}_2}(\lambda) + 2^{-\Omega(\lambda)},$$

where PPT algorithms $\mathcal{B}_1$ and $\mathcal{B}_2$ run in about the same time as $\mathcal{A}$.

Table of Contents