Almost Tight Multi-User Security under Adaptive Corruptions & Leakages in the Standard Model

Shuai Han[1,2], Shengli Liu[1,2,3](✉), and Dawu Gu[1]

[1] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
{dalen17,slliu,dwgu}@sjtu.edu.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Westone Cryptologic Research Center, Beijing 100070, China

Abstract. In this paper, we consider tight multi-user security under adaptive corruptions, where the adversary can adaptively corrupt some users and obtain their secret keys. We propose generic constructions for a bunch of primitives, and the instantiations from the matrix decisional Diffie-Hellman (MDDH) assumptions yield the following schemes:

(1) the first digital signature (SIG) scheme achieving almost tight *strong* EUF-CMA security in the multi-user setting with adaptive corruptions in the standard model;
(2) the first public-key encryption (PKE) scheme achieving almost tight IND-CCA security in the multi-user multi-challenge setting with adaptive corruptions in the standard model;
(3) the first signcryption (SC) scheme achieving almost tight privacy and authenticity under CCA attacks in the multi-user multi-challenge setting with adaptive corruptions in the standard model.

As byproducts, our SIG and SC naturally derive the first strongly secure message authentication code (MAC) and the first authenticated encryption (AE) schemes achieving almost tight multi-user security under adaptive corruptions in the standard model. We further optimize constructions of SC, MAC and AE to admit better efficiency.

Furthermore, we consider key leakages besides corruptions, as a natural strengthening of tight multi-user security under adaptive corruptions. This security considers a more natural and more complete "all-or-*part*-or-nothing" setting, where secret keys of users are either fully exposed to adversary ("all"), or completely hidden to adversary ("nothing"), or *partially* leaked to adversary ("part"), and it protects the uncorrupted users even with bounded key leakages. All our schemes additionally support bounded key leakages and enjoy full compactness. This yields the first SIG, PKE, SC, MAC, AE schemes achieving almost tight multi-user security under both adaptive corruptions and leakages.

1 Introduction

Cryptography aims to provide two fundamental security guarantees: privacy and authenticity. Centered around privacy and authenticity, a variety of cryptographic primitives are developed, including public-key encryption (PKE), symmetric encryption (SE), digital signature (SIG), message authentication code

(MAC), signcryption (SC), authenticated encryption (AE), etc. To rigorously define security notions for these primitives, proper security models have to be set up according to their working environments and the adversaries' attacking abilities. Along the path of proving security, PKE and SE are defined with indistinguishability under chosen plaintext/ciphertext attacks (IND-CPA/CCA), SIG and MAC are defined with existential unforgeability under chosen message attacks (EUF-CMA), and SC and AE with both privacy (Priv) and authenticity (Auth). To prove a specific primitive construction achieves the security goals, the most important technique is security reduction. Roughly speaking, a security reduction establishes a link from an adversary $\mathcal{A}$ against the security of a primitive to another adversary $\mathcal{B}$ solving a well-studied computationally hard problem, such as the decisional Diffie-Hellman (DDH) and learning with errors (LWE) problems, with approximately the same running time. The ratio of $\mathcal{A}$'s advantage $\epsilon_{\mathcal{A}}$ to $\mathcal{B}$'s advantage $\epsilon_{\mathcal{B}}$ is defined as the loss factor $\ell := \epsilon_{\mathcal{A}}/\epsilon_{\mathcal{B}}$, which measures the quality of the security reduction.[1] If ℓ is a small constant, we call the reduction *tight*. Tight security is more desirable than non-tight one, since it enables a theoretically-sound instantiation without the need to compensate a security loss by increasing key lengths or group sizes, and allows universal key-length recommendations for applications. Many works (e.g., [12, 18, 19, 24, 28, 21]) also consider the tightness notion called *almost tight*, where ℓ depends at most linearly (or even better, logarithmically) on the security parameter λ. For ease of exposition, we will use the term "tight" to denote "(almost) tight" as conventionally did [18, 19, 24, 28, 21], but we will detail the security loss in the security theorems and scheme comparisons to reflect almost tightness.

Tight Multi-User Security under Adaptive Corruptions (MU$^{\mathbf{c}}$). Cryptographic primitives are usually deployed in multi-user settings. But most of the security models for the primitives only consider single user. This is acceptable, since single-user security generally implies multi-user security via a security reduction called hybrid argument. But the price is a large loss factor ℓ at least nQ, where n is the number of users and Q the number of instances per user [6]. Considering billions of users and trillions of running instances over Internet, the security loss ℓ can be as large as 2^{60}. Such a large loss factor does hurt and has to be taken into account in the security parameter configuration during the deployment of primitives over Internet. To avoid a large loss factor that varies with the number of users and/or the number of target instances, many works [23, 18, 19] (to name a few) focus on primitive design with tight multi-user security.

Compared with a single-user setting, a multi-user environment becomes more involved and leaves more opportunities to adversaries implementing new attacks. An important attack is *key corruption* in that the adversary takes full control of some users and of course their keys. This happens since some adversary may

[1] Strictly speaking, the loss factor is defined as $\ell := (\epsilon_{\mathcal{A}}/\epsilon_{\mathcal{B}}) \cdot (\mathbf{T}(\mathcal{B})/\mathbf{T}(\mathcal{A}))$, where $\mathbf{T}(\mathcal{A})$ and $\mathbf{T}(\mathcal{B})$ denote the running time of $\mathcal{A}$ and $\mathcal{B}$, respectively. For reductions where $\mathbf{T}(\mathcal{A})$ and $\mathbf{T}(\mathcal{B})$ are approximately the same (as in many related works and also in this work), the loss factor can be simplified to $\epsilon_{\mathcal{A}}/\epsilon_{\mathcal{B}}$.

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 Multi-User security under corruptions, which we denote MU^{c} or more precisely MU^{c}-XX with notion XX depending on the primitive.[2] The existing works on MU^{c} indicates that pursuing tight MU^{c} security is not easy, as shown below.

Technical Difficulties in Achieving Tight MU^{c} Security. As pointed out in [14, 21], there is a seemingly paradoxical technical problem needing to be addressed for proving tight MU^{c}-CMA security of SIG. On the one hand, the security reduction algorithm has to know the signing keys of *all* users so that it can successfully answer adversary's adaptive corruption query without resorting to a guessing strategy. On the other hand, the reduction algorithm should also be able to extract an answer to the underlying computationally hard problem from the adversary's forged signature. However, if the reduction knows all the signing keys, it should be able to forge a signature by itself without the adversary.

There exist similar technical problems in achieving tight MU^{c} security for other primitives. For example, to achieve tight MU^{c}-$\mathsf{CPA}/\mathsf{CCA}$ security for PKE, the security reduction algorithm has to know the secret keys of *all* users to avoid the loss factor incurred by a guessing strategy. On the other hand, it should also be able to extract an answer from the adversary's guessing of challenge bit. This seems to lead to a similar paradox since the reduction can decrypt the challenge ciphertexts to learn the challenge bit by itself if it knows all the secret keys.

Impossibility Results on Tight MU^{c} Security. In fact, there is a line of research which showed impossibility results on tight MU^{c} security for a class of PKE, SIG, MAC and AE schemes that meet certain conditions.

- **PKE.** Bader et al. [5] proved that there exists no tight security reduction from MU^{c}-$\mathsf{CPA}/\mathsf{CCA}$ security of PKE to non-interactive assumptions, if the relation between public key and secret key is "unique" or "re-randomizable".
- **SIG.** The above impossibility result for PKE also applies to MU^{c}-CMA security of SIG, except that the relation is defined for the verification key and signing key [5]. Alternatively, if the signing algorithm is a deterministic one, there exists no tight security reduction from MU^{c}-CMA security of SIG to bounded-round assumptions [32].
- **MAC.** Morgan et al. [32] showed that if MAC is a deterministic one, then there exists no tight security reduction from MU^{c}-CMA security of MAC to bounded-round assumptions.
- **AE.** Jager et al. [25] proved that if AE satisfies a minimal key uniqueness, any reasonable reduction from MU^{c} to single-user security is not tight.

These impossibility results indicate that it is not an easy job to obtain tight MU^{c} Security. However, it does not eliminate all hopes as long as we can find ways bypassing the conditions leading to the impossibility results.

[2] For primitives like PKE, SC, AE, we also consider Multi-User Multi-Challenge security under corruptions to capture multiple challenge ciphertexts, denoted by $\mathsf{MUMC}^{\mathsf{c}}$.

Possibility Results on Tight MU^c Security. There are very few constructions in the literature proved to have tight MU^c security, even in the Random Oracle (RO) model.

- **PKE.** To the best of our knowledge, only one PKE scheme in [29] is proved to be tightly multi-user multi-challenge CCA secure under adaptive corruptions (MUMC^c-CCA). Its security proof relies on the RO model.
- **SIG.** Gjøsteen and Jager [20] and Pan and Wagner [35] proposed tightly MU^c-CMA secure SIG schemes in the RO model. Bader et al. [4] constructed a tightly MU^c-CMA secure SIG scheme in the standard model. Its tree-based component makes the signature non-compact. Recently, Han et al. [21] designed a new MU^c-CMA secure SIG in the standard model. Their scheme enjoys compact signature while having non-compact public parameters (consisting of over a thousand group elements).

  It is more desirable to pursue *strong* MU^c-CMA security of SIG, which even guarantees the hardness for adversary to forge a new signature for an already signed message, thus additionally ensuring "non-malleablility" of signatures. Strongly MU^c-CMA secure SIG has important applications in building more complex primitives such as SC [3] and authenticated key exchange (AKE) [14], where it can help SC to achieve ciphertext integrity (authenticity) [7] and AKE to achieve strong notion of "matching conversations" security [8] (see more discussions in [14]). One may want to resort to the Generalized Boneh-Shen-Waters (GBSW) transform [38] to convert a (non-strongly) secure SIG scheme to a strongly secure one, with the help of chameleon hash functions. However, the GBSW transform was originally proposed in the single-user setting, and was recently extended to the multi-user setting in [30], but without the consideration of corruptions. As noted in [30], it seems difficult to show that the GBSW transform also works under corruptions and preserves the tightness, i.e., converting a tightly MU^c-CMA secure SIG scheme to a tightly and strongly MU^c-CMA secure one. The reason is, the resulting SIG scheme contains the trapdoor of chameleon hash in its secret key, thus corruption of secret key means revealing of trapdoor, which is not supported by the security of chameleon hash [30].

  Up to now, only one SIG scheme in a recent work [14] is proved to have tight strong MU^c-CMA security, based on the RO model.
- **SignCryption(SC).** In [9], Bellare and Stepanovs defined multi-user security for SC to cover both insider and outsider security. Their security notions are essentially multi-user CCA security under adaptive corruptions which considers both privacy (MUMC^c-Priv) and authenticity (MUMC^c-Auth). They also designed a SC scheme with security proved in the RO model.
- **MAC and AE.** Note that SIG naturally implies a MAC scheme and SC implies an AE scheme. As far as we know there is no approach to tight MU^c-CMA security other than derived from SIG. Similar statement holds for AE.

Up to now, there exists no PKE scheme achieving tight MUMC^c-CCA security, no SIG and MAC achieving tight strong MU^c-CMA security, and no SC and AE achieving tight MUMC^c-Priv&Auth in the standard model. The challenges are:

Can we fill the aforementioned blanks on tight MU^{c} *security in the standard model? Can we step even forward by considering tight multi-user security under not only adaptive corruptions but also key leakages?*

1.1 Our Contributions

We propose generic constructions for a bunch of primitives and prove their tight multi-user security under adaptive corruptions and key leakages.

- We propose generic constructions of SIG, PKE, SC, MAC, AE and prove their MU^{c} security with tight security reductions. The instantiations yield the following concrete schemes from the matrix DDH (MDDH) assumptions [17] (which corresponds to the standard DDH, k-Linear assumptions under different parameters) over asymmetric pairing groups *in the standard model*:
  - the first PKE scheme achieving almost tight $\mathsf{MUMC}^{\mathsf{c}}$-$\mathsf{CCA}$ security in the standard model;
  - the first SIG scheme achieving almost tight *strong* MU^{c}-CMA security in the standard model;
  - the first SC scheme achieving almost tight $\mathsf{MUMC}^{\mathsf{c}}$-$\mathsf{Priv}\&\mathsf{Auth}$ security in the standard model;
  - the first MAC scheme achieving almost tight *strong* MU^{c}-CMVA security in the standard model;
  - the first AE scheme achieving almost tight $\mathsf{MUMC}^{\mathsf{c}}$-$\mathsf{Priv}\&\mathsf{Auth}$ security in the standard model.

  Moreover, all our schemes are *fully compact*, i.e., all the parameters, keys, signatures, ciphertexts consist of only a constant number of group elements.
- We formalize stronger multi-user security notions for the primitives under not only adaptive corruptions but also *key leakages*, denoted by $\mathsf{MU}^{\mathsf{c\&l}}$. In addition to MU^{c}, the $\mathsf{MU}^{\mathsf{c\&l}}$ security protects the uncorrupted users even if adversary also obtains bounded leakage information on their secret keys.

  Key leakage [2, 34] is closely related to corruption, especially in the multi-user setting, and $\mathsf{MU}^{\mathsf{c\&l}}$ is a natural strengthening of MU^{c}. The reason is as follows. Existing MU^{c} security considers an "all-or-nothing" setting, where secret keys of users are either fully exposed to adversary ("all") or completely hidden to adversary ("nothing"), and it protects the uncorrupted users. In realistic environments, there would naturally be users whose secret keys are only partially leaked to adversary ("part"). These users sit in a situation that is neither "all" nor "nothing". The new $\mathsf{MU}^{\mathsf{c\&l}}$ security additionally takes into account the security of these users. Hence the new $\mathsf{MU}^{\mathsf{c\&l}}$ security considers a more natural and more complete setting of "all-or-*part*-or-nothing".

  Thanks to the leakage resilience property of the building blocks, the almost tight MU^{c} security of all our SIG, PKE, SC, MAC, AE schemes can be further strengthened to support key leakage, thus achieving almost tight $\mathsf{MU}^{\mathsf{c\&l}}$ security.

- At the heart of our constructions is new technical tool called *Publicly-Verifiable Quasi-Adaptive Hash Proof System* and a set of new properties for it. These, together with our novel tight proof strategies for handling corruptions, help us circumvent the seemingly paradoxical technical problems.

We refer to Table 1 and Table 2 for comparisons of our SIG and PKE with known schemes, respectively. We also show the size of parameters and compare the compactness of the schemes in Table 5 and Table 6 in Appendix A.

In summary, our work shows that almost tight MU^{c} security (and even together with full compactness) for SIG, PKE, SC, MAC and AE are achievable in the *standard model*. Moreover, our MDDH-based schemes support bounded key leakages as well, thus our work also provides the *first* schemes achieving almost tight $\mathsf{MU}^{\mathsf{c\&l}}$ security, no matter in the standard model or RO model.

Table 1. Comparison of signature (SIG) schemes that have (almost) tight MU-CMA security under adaptive corruptions (MU^{c}-CMA). The column **Standard Model** shows whether the security is proved in the standard model. The column **Strong Security** shows whether the scheme is proved *strongly* existentially unforgeable. The column **Corruption?** asks whether the security is proved in the presence of adaptive corruptions. The column **Leakage?** asks whether the security is proved additionally in the presence of key leakages, and if so, a *leakage rate* (defined as the ratio of leakage amount to secret key size) is presented. The column **Full Compactness** shows whether the scheme is fully compact (i.e., all the public parameters pp, verification key vk, signing key sk and signature σ consist of only a constant number of group elements or lattice vectors), and if not, the non-compact part is presented. The column **Security Loss** shows the security loss factor of the reductions, where λ denotes the security parameter. The column **Assumption** shows the computational assumption on which the security is based.

SIG Scheme	Standard Model	Strong Security	Corruption?	Leakage?	Full Compactness	Security Loss	Assumption
BHJKL [4, 23]	✓	–	✓	–	× (non-compact σ)	$O(1)$	MDDH
GJ [20]	×	–	✓	–	✓	$O(1)$	DDH
DGJL [14]	×	✓	✓	–	✓	$O(1)$	DDH or ϕ-Hiding
HJKLPRS [21]	✓	×	✓	–	× (non-compact pp)	$O(\lambda)$	MDDH
PW [35]	×	–	✓	–	× (non-compact vk)	$O(1)$	LWE
Our $\mathsf{SIG}_{\mathsf{MDDH}}$	✓	✓	✓	✓ $(\frac{1}{6} - o(1))$	✓	$O(\log \lambda)$	MDDH

Table 2. Comparison of public-key encryption (PKE) schemes that have (almost) tight MUMC-CCA security under adaptive corruptions ($\mathsf{MUMC}^{\mathsf{c}}$-$\mathsf{CCA}$) *or* key leakages. The columns have similar meanings as those in Table 1.

PKE Scheme	Standard Model	Corruption?	Leakage?	Full Compactness	Security Loss	Assumption
HLLG [22]	✓	–	✓ $(\frac{1}{18} - o(1))$	✓	$O(\log \lambda)$	MDDH
LLP [29]	×	✓	–	✓	$O(1)$	CDH
Our $\mathsf{PKE}_{\mathsf{MDDH}}$	✓	✓	✓ $(\frac{1}{3} - o(1))$	✓	$O(\log \lambda)$	MDDH

2 Technical Overview

In this section, we provide a technical overview of our results. We show the main ideas in our generic constructions of SIG and PKE, and give a high-level overview of their tight MU^{c} security proofs in Subsect. 2.1 and Subsect. 2.2, respectively. We describe our SC, MAC and AE constructions and how to optimize them in Subsect. 2.3. Then in Subsect. 2.4, we explain the instantiations from the MDDH assumptions and explain why our aforementioned constructions support key leakage and achieve tight $\mathsf{MU}^{\mathsf{c\&l}}$ security. Finally, in Subsect. 2.5, we compare our technique with existing techniques for tight MU^{c} security.

2.1 Our SIG: Technical Overview

Our starting point is a useful tool called Quasi-Adaptive Hash Proof System (QA-HPS), which was proposed by Han et al. [22] for achieving tight leakage resilient security of PKE. QA-HPS generalizes HPS [13] with a collection $\mathscr{L} = \{\mathcal{L}_\rho\}_\rho$ of NP-languages ($\mathcal{L}_\rho \subseteq \mathcal{X}$) and a family of projection functions $\alpha_{(\cdot)}$. The projection key is determined by $pk := \alpha_\rho(sk)$, hence depends on language $\mathcal{L}_\rho$. Meanwhile, QA-HPS has two ways of computing the hash value $\Lambda_{sk}(x)$: the public evaluation $\mathsf{Pub}(pk, x, w)$ for the instance $x \in \mathcal{L}_\rho$ with witness w, and the private evaluation $\mathsf{Priv}(sk, x)$ for $x \in \mathcal{X}$. Its correctness requires $\mathsf{Pub}(pk, x, w) = \mathsf{Priv}(sk, x) = \Lambda_{sk}(x)$ for $x \in \mathcal{L}_\rho$. Moreover, the subset membership problem (SMP) asks the computational indistinguishability of $x \leftarrow_\$ \mathcal{L}_\rho$ and $x \leftarrow_\$ \mathcal{X}$.

Another technical tool is Quasi-Adaptive Non-Interactive Zero-Knowledge argument (QA-NIZK) proposed by Jutla and Roy [26], where the common reference string crs depends on language $\mathcal{L}_\rho$. For tag-based QA-NIZK [27], there are two ways of generating a proof π for $x \in \mathcal{L}_\rho$ w.r.t. tag τ: $\mathsf{Prove}(\mathsf{crs}, \tau, x, w)$ using a witness w for $x \in \mathcal{L}_\rho$, and the simulator $\mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$ using a trapdoor $\mathsf{td}_{\mathsf{crs}}$. With $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi)$, one can verify whether π is a valid proof. Perfect zero-knowledge requires that the proofs generated by Prove and Sim are identically distributed. Besides, unbounded simulation-soundness (USS) [37, 24, 1] stipulates that a PPT adversary cannot prove a false statement $x \notin \mathcal{L}_\rho$, even if it can obtain multiple simulated proofs for instances not necessarily in $\mathcal{L}_\rho$.

QA-HPS and HPS have found wide applications in designing PKE [13], MAC [15], etc. However, there are rarely applications in building SIG schemes, mainly because the designated-verifier style inherent in (QA)HPS is insufficient to support public verification of SIG. To fill the gap, we propose a new tool.

Publicly-Verifiable QA-HPS. The core technical tool underlying our SIG construction is a *Publicly-Verifiable* variant of QA-HPS, or PV-QA-HPS in short, which enables public verification of hash values with an extra verification key. We introduce a verification key generation function $\nu(\cdot)$ to compute verification key $vk := \nu(sk)$, and a verification algorithm $\mathsf{Vrfy}_{\mathsf{HPS}}(vk, x, hv)$ to check whether an element hv equals the hash value $\Lambda_{sk}(x)$ of x with the help of vk.

We also define two important properties for PV-QA-HPS, which play essential roles in the tight security reduction of our SIG.

- **Verification soundness**. It is a computational property requiring that, given all secret/verification key pairs $\{(sk_i, vk_i)\}_{i\in[n]}$, it is hard for any PPT adversary to come up with an index $i^* \in [n]$, an instance $x^* \in \mathcal{X}$ and a hash value hv^* which is false but passes the verification w.r.t. key pair (sk_{i^*}, vk_{i^*}), i.e., $hv^* \neq \Lambda_{sk_{i^*}}(x^*)$ but $\mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, x^*, hv^*) = 1$.
- $\langle\mathscr{L}_0, \mathscr{L}\rangle$**-One-Time(OT)-extracting.** It is a statistical property parameterized by two language collections $\mathscr{L}_0 = \{\mathcal{L}_{\rho_0}\}_{\rho_0}$ and $\mathscr{L} = \{\mathcal{L}_\rho\}_\rho$. It demands that the hash value $\Lambda_{sk}(x^*)$ for any $x^* \in \mathcal{L}_\rho \in \mathscr{L}$ retains a large enough min-entropy, even conditioned on the verification key $vk = \nu(sk)$ and the projection key $pk_{\rho_0} = \alpha_{\rho_0}(sk)$ w.r.t. language $\mathcal{L}_{\rho_0} \in \mathscr{L}_0$. This min-entropy makes sure that any (unbounded) adversary is unable to guess the correct hash value $\Lambda_{sk}(x^*)$, except with a negligible probability.

Our SIG from PV-QA-HPS and QA-NIZK. The building blocks for our SIG construction consists of a PV-QA-HPS scheme $\mathsf{PVQAHPS} = (\alpha_{(\cdot)}, \nu(\cdot), \mathsf{Pub}, \mathsf{Priv}, \mathsf{Vrfy}_{\mathsf{HPS}})$ for both language $\mathcal{L}_\rho \in \mathscr{L}$ and language $\mathcal{L}_{\rho_0} \in \mathscr{L}_0$[3], a tag-based $\mathsf{QANIZK} = (\mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{Sim})$ for $\mathcal{L}_\rho$ and a collision-resistant hash function H. The signing and verification keys of SIG are just the secret key sk and verification key $vk = \nu(sk)$ of $\mathsf{PVQAHPS}$. The signature for message m is

$$\sigma := (\ x \leftarrow_{\$} \mathcal{L}_\rho,\ d := \mathsf{Priv}(sk, x),\ \pi := \mathsf{Prove}(\mathsf{crs}, \tau, x, w)\ )^{[4]},\ \text{with } \tau := H(vk, m).$$

The verification of SIG checks $\mathsf{Vrfy}_{\mathsf{HPS}}(vk, x, d) = 1$ and $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$.

In the strong $\mathsf{MU^c\text{-}CMA}$ security model, adversary $\mathcal{A}$ adaptively issues user-message pairs (i, m) to the signing oracle and obtains valid signatures σ. It can also issue corruption queries and get the corresponding signing keys. $\mathcal{A}$ tries to output a fresh and valid forgery $(i^*, m^*, \sigma^*) \notin \{(i, m, \sigma)\}$ for an uncorrupted user i^*.

Our tight strong $\mathsf{MU^c\text{-}CMA}$ security proof goes with three steps. See also Fig. 1 for a graphical high-level overview.

Step 1. Switch language from $\mathcal{L}_\rho$ to $\mathcal{L}_{\rho_0}$ for signing queries. Through signing queries, $\mathcal{A}$ obtains a bunch of tuples $(i, m, \sigma = (x, d, \pi))$, where σ is a valid signature of m under sk_i.

- According to the perfect zero-knowledge of QANIZK, the computation of π by Prove can be replaced by Sim without any witness of $x \in \mathcal{L}_\rho$.
- By the hardness of (multi-fold) SMP, the samplings of all x can be changed from $x \leftarrow_{\$} \mathcal{L}_\rho$ to $x \leftarrow_{\$} \mathcal{L}_{\rho_0}$.
- For $x \in \mathcal{L}_{\rho_0}$ with witness w, $d := \mathsf{Priv}(sk_i, x) = \mathsf{Pub}(\alpha_{\rho_0}(sk_i), x, w)$. So

$$\sigma = \big(\ x \leftarrow_{\$} \mathcal{L}_{\rho_0},\ d := \mathsf{Pub}(\alpha_{\rho_0}(sk_i), x, w), \pi := \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)\ \big).$$

[3] This means that $\mathsf{PVQAHPS}$ works correctly both for $x \in \mathcal{L}_\rho$ with $pk = \alpha_\rho(sk)$ and $x \in \mathcal{L}_{\rho_0}$ with $pk = \alpha_{\rho_0}(sk)$.

[4] Here ρ is part of the public parameters of SIG and is chosen from the language collection $\mathscr{L}$ by the setup algorithm of SIG, while w is a witness for $x \in \mathcal{L}_\rho$ and is picked along with $x \leftarrow_{\$} \mathcal{L}_\rho$ by the signing algorithm of SIG.

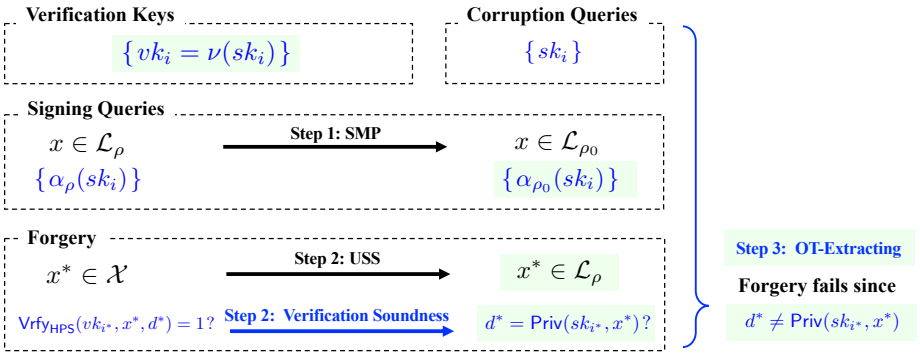

Fig. 1. The high-level overview of our proof strategy for tight strong MU^{c}-CMA security of SIG. The black arrows illustrate language switches, and the blue arrows as well as the blue brace show the applications of quasi-adaptive properties.

Now $\alpha_{\rho_0}(sk_i)$ (out of the whole sk_i) suffices for generating σ.

Step 2. Restrict language from $\mathcal{X}$ to $\mathcal{L}_\rho$ in the forgery. $\mathcal{A}$'s forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ is successful if it is fresh and passes the validity check $\mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, x^*, d^*) = 1 \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1$ with $\tau^* := H(vk_{i^*}, m^*)$.

- By the verification soundness of $\mathsf{PVQAHPS}$, the check of $\mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, x^*, d^*) = 1$ can be replaced by $d^* = \mathsf{Priv}(sk_{i^*}, x^*)$.
- The USS property of QANIZK makes sure that $x^* \in \mathcal{L}_\rho$ in the forgery, except with a negligible probability.

Strategy for corruptions in reductions. Note that in the above two steps, when reducing to SMP or QANIZK, the reduction algorithms can choose all users' signing keys themselves. As for the verification soundness of $\mathsf{PVQAHPS}$, the reduction algorithm gets all users' signing keys from its own challenger. Therefore, all of them are able to handle $\mathcal{A}$'s adaptive corruption queries.

Step 3. $\mathcal{A}$'s forgery fails due to the $\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting property. Now all information about sk_{i^*} that $\mathcal{A}$ learns from the signing queries is limited to the projection key $\alpha_{\rho_0}(sk_{i^*})$ on language $\mathcal{L}_{\rho_0}$. On the other hand, x^* in $\mathcal{A}$'s forgery is restricted in $\mathcal{L}_\rho$ and $\mathcal{A}$ wins only if $d^* = \mathsf{Priv}(sk_{i^*}, x^*)$. By the $\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting property of $\mathsf{PVQAHPS}$, $\mathcal{A}$ hardly succeeds.

How we circumvent the seemingly paradoxical technical problem. Now we conclude how we circumvent the paradoxical technical problem for achieving tight strong MU^{c}-CMA security of SIG: our proof goes with a constant number of computationally indistinguishable changes to arrive at a final game where the technical problem has turned into a statistical one.

(1) All the reduction algorithms to computational properties or problems possess the signing keys of all users to handle adaptive corruption queries.

(2) After arriving at a statistical problem ($\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting property), it is hard for the adversary to forge valid signature information-theoretically.

How we circumvent the existing impossibility results. Below we explain how we circumvent the impossibility results on tight MU^{c} security. Recall that the impossibility results apply to a SIG scheme when the relation between the verification key and the signing key is "unique" or "re-randomizable" [5], or the signing algorithm is a deterministic one [32].

Firstly, the signing algorithm of our SIG is not a deterministic one since it samples a random element x from $\mathcal{L}_\rho$ with witness w.

Next, we show that the relation between the verification key $vk = \nu(sk)$ and the signing key sk of our SIG is neither "unique" nor "re-randomizable", by the properties we defined for PV-QA-HPS.

- The relation is not "unique" due to the statistical $\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting property of PV-QA-HPS. Suppose, towards a contradiction, that the relation is unique, then an (unbounded) adversary can uniquely determine sk from $\nu(sk)$, and thus break the property easily by computing $hv^* = \Lambda_{sk}(x^*)$ for any $x^* \in \mathcal{L}_\rho$.
- The relation is not "re-randomizable" due to the verification soundness property of PV-QA-HPS. Suppose, towards a contradiction, that the relation is re-randomizable, then for any user $i^* \in [n]$, an adversary can resample another sk'_{i^*} from vk_{i^*} and sk_{i^*}, such that $vk_{i^*} = \nu(sk_{i^*}) = \nu(sk'_{i^*})$. Then the adversary picks x^* from $\mathcal{X}$ uniformly, computes $hv^* = \Lambda_{sk'_{i^*}}(x^*)$ using sk'_{i^*}, and outputs (i^*, x^*, hv^*). On the one hand, since vk_{i^*} is also the verification key of sk'_{i^*}, i.e., $vk_{i^*} = \nu(sk'_{i^*})$, hv^* passes the verification w.r.t. vk_{i^*}, i.e., $\mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, x^*, hv^*) = 1$. On the other hand, we have $sk'_{i^*} \neq sk_{i^*}$ with high probability ($\geq 1/2$, by the fact that the relation between vk and sk is not unique, as shown above), thus $hv^* = \Lambda_{sk'_{i^*}}(x^*) \neq \Lambda_{sk_{i^*}}(x^*)$ with high probability. Consequently, the adversary breaks the verification soundness with high probability.

Of course, being neither "unique" nor "re-randomizable" nor "deterministic" is only a necessary condition for tight MU^{c} security. To achieve tight MU^{c} security, the cooperation of PV-QA-HPS and QA-NIZK in the design of our SIG as well as the nice properties of PV-QA-HPS play the most important roles.

2.2 Our PKE: Technical Overview

Our PKE is built upon the recent work [22], where the concept of QA-HPS was proposed to construct PKE with tight leakage resilient security. That tight security heavily relies on two statistical properties of QA-HPS: key-switching and universal. Intuitively, $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-*key-switching* requires that conditioned on a projection key $\alpha_\rho(sk)$ w.r.t. language $\mathcal{L}_\rho \in \mathscr{L}$, the projection key $\alpha_{\rho_0}(sk)$ w.r.t. language $\mathcal{L}_{\rho_0} \in \mathscr{L}_0$ can be switched to $\alpha_{\rho_0}(sk')$ for an independent key sk'.

The PKE in [22] makes use of three QA-HPS schemes, one for masking the message and the other two for proving the well-formedness of ciphertext. As far as we understand, it is hard to prove the tight security of their PKE under adaptive corruptions, since their proof strategy that increases the entropy in secret keys gradually does not work in the presence of corruptions.

To support corruptions in the tight security, (1) we define *new properties* for QA-HPS, (2) we use *another approach*: QA-HPS with new properties to mask the message and QA-NIZK to prove the well-formedness of ciphertext, and (3) we develop a *new proof strategy* to achieve tight $\mathsf{MUMC^c\text{-}CCA}$ security.

QA-HPS with New Properties. We define two new properties for QA-HPS.

- **Multi-language multi-fold SMP**. This new type of SMP asks the computational indistinguishability of $(x_{i,j} \leftarrow_\$ \mathcal{L}_\rho)_{i\in[n],j\in[Q]}$ and $(x_{i,j} \leftarrow_\$ \mathcal{L}_{\rho_0^{(i)}})_{i\in[n],j\in[Q]}$, where $\mathcal{L}_\rho \in \mathscr{L}$, and $\mathcal{L}_{\rho_0^{(1)}}, ..., \mathcal{L}_{\rho_0^{(n)}} \in \mathscr{L}_0$ are n independent languages chosen from $\mathscr{L}_0$. Jumping ahead, this new SMP enables us to switch the language $\mathcal{L}_\rho$ to different languages $\{\mathcal{L}_{\rho_0^{(i)}}\}_{i\in[n]}$ for different users in our tight proof.
- **$\mathscr{L}_0$-Multi-key multi-extracting.** It demands the pseudorandomness of multiple hash values $\{\Lambda_{sk_i}(x_j)\}_{i\in[n],j\in[Q]}$ of multiple instances $x_1, ..., x_Q \in \mathcal{L}_{\rho_0}$ under uniformly and independently chosen keys $sk_1, ..., sk_n$.

Our PKE from QA-HPS with New Properties and QA-NIZK. The secret and public keys of PKE are just the secret key sk and projection key $pk = \alpha_\rho(sk)$ of QA-HPS for language $\mathcal{L}_\rho$. The ciphertext for plaintext m is

$$c := (x \leftarrow_\$ \mathcal{L}_\rho,\ d := \mathsf{Pub}(pk, x, w) + m,\ \pi := \mathsf{Prove}(\mathsf{crs}, \tau, x, w)),\quad \text{with } \tau := H(pk, d).$$

The decryption of $c = (x, d, \pi)$ checks whether $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$ and recovers $m := d - \mathsf{Priv}(sk, x)$ after a successful check.

It is interesting to note that our PKE shares a similar design with our SIG. However, their tight proofs are quite different.

In the $\mathsf{MUMC^c\text{-}CCA}$ security model, adversary $\mathcal{A}$ adaptively issues encryption queries (i^*, m_0, m_1) to encryption oracle and obtains challenge ciphertexts $c^* = (x^*, d^*, \pi^*)$ that encrypts m_β under pk_{i^*}, where $\beta \leftarrow_\$ \{0, 1\}$ is the challenge bit. It can issue corruption queries and get the corresponding secret keys, and issue decryption queries $(i, c = (x, d, \pi))$ and obtain the decryption of c under sk_i. Finally $\mathcal{A}$ outputs a guessing bit β' and wins if $\beta' = \beta$.

Our tight $\mathsf{MUMC^c\text{-}CCA}$ security proof goes with five steps. See also Fig. 2 for a graphical high-level overview.

Step 1. Switch language from $\mathcal{L}_\rho$ to $\{\mathcal{L}_{\rho_0^{(i^*)}}\}_{i^*\in[n]}$ for encryption queries.
Through encryption queries (i^*, m_0, m_1), $\mathcal{A}$ obtains multiple challenge ciphertexts $c^* = (x^*, d^*, \pi^*)$.

- According to the perfect zero-knowledge of QANIZK, the computation of π^* by Prove can be replaced by Sim without any witness of $x^* \in \mathcal{L}_\rho$.
- By the correctness of QAHPS, the computation of d^* by Pub can be replaced by $d^* := \mathsf{Priv}(sk_{i^*}, x^*) + m_\beta$, without any witness of $x^* \in \mathcal{L}_\rho$.
- By the new multi-language multi-fold SMP, for each user i^*, the samplings of all x^* can be changed from $x^* \leftarrow_\$ \mathcal{L}_\rho$ to $x^* \leftarrow_\$ \mathcal{L}_{\rho_0^{(i^*)}}$.
- For each user i^*, since $x^* \in \mathcal{L}_{\rho_0^{(i^*)}}$ with witness w^*, we have $d^* := \mathsf{Priv}(sk_{i^*}, x^*) + m_\beta = \mathsf{Pub}(\alpha_{\rho_0^{(i^*)}}(sk_{i^*}), x^*, w^*) + m_\beta$. Hence

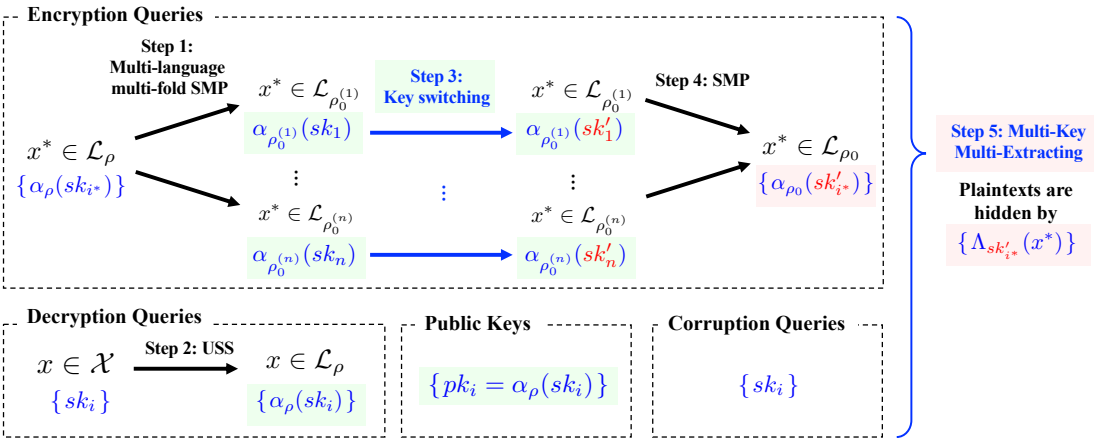

Fig. 2. The high-level overview of our proof strategy for tight $\mathsf{MUMC}^{\mathsf{c}}$-$\mathsf{CCA}$ security of PKE. The black arrows illustrate language switches, and the blue arrows as well as the blue brace show the applications of quasi-adaptive properties.

$$c^* := \left(x^* \leftarrow_{\$} \mathcal{L}_{\rho_0^{(i^*)}},\ d^* := \mathsf{Pub}(\alpha_{\rho_0^{(i^*)}}(sk_{i^*}), x^*, w^*) + m_\beta,\ \pi^* := \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*, x^*) \right).$$

Now $\{\alpha_{\rho_0^{(i^*)}}(sk_{i^*})\}_{i^*\in[n]}$ (out of whole $\{sk_{i^*}\}_{i^*\in[n]}$) suffices for generating c^*.

Step 2. Restrict language from $\mathcal{X}$ to $\mathcal{L}_\rho$ for decryption queries. For query $(i, c = (x, d, \pi))$, $\mathcal{A}$ obtains $m := d - \mathsf{Priv}(sk_i, x)$ if $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$.

- The USS property of QANIZK makes sure that $\mathcal{A}$ obtains m only if $x \in \mathcal{L}_\rho$ in the decryption query, except with a negligible probability.

Hence $\mathcal{A}$ learns only $\{\alpha_\rho(sk_i)\}_{i\in[n]}$ (out of $\{sk_i\}_{i\in[n]}$) from decryption queries.

Step 3. Switch $\{sk_{i^*}\}_{i^*\in[n]}$ to new keys $\{sk'_{i^*}\}_{i^*\in[n]}$ for encryption queries. Note that to avoid trivial attacks, $\mathcal{A}$ is not allowed to corrupt those users i^* for which $\mathcal{A}$ issues encryption queries. Thus for such users i^*, after the first two steps, $\mathcal{A}$'s information about sk_{i^*} can be summarized by $\alpha_\rho(sk_{i^*})$ (involved in public keys and decryption oracle) and $\alpha_{\rho_0^{(i^*)}}(sk_{i^*})$ (involved in encryption oracle).

- According to the $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching property of QAHPS, $\alpha_{\rho_0^{(i^*)}}(sk_{i^*})$ can be switched to $\alpha_{\rho_0^{(i^*)}}(sk'_{i^*})$ to compute d^* for encryption queries, with sk'_{i^*} uniformly and independently chosen.

Though there are n switches, it does not lead to a loose security reduction, since key-switching is a statistical property of QAHPS.

As a result, new independent secret keys $\{sk'_{i^*}\}_{i^*\in[n]}$ are split from the original $\{sk_{i^*}\}_{i^*\in[n]}$, and are only used for answering encryption queries.

Step 4. Switch languages $\{\mathcal{L}_{\rho_0^{(i^*)}}\}_{i^*\in[n]}$ to $\mathcal{L}_{\rho_0}$ for encryption queries. The argument is similar to step 1. As a result, the computation of $d^* := \mathsf{Pub}(\alpha_{\rho_0^{(i^*)}}(sk'_{i^*}), x^*, w^*) + m_\beta$ is changed to $d^* := \mathsf{Pub}(\alpha_{\rho_0}(sk'_{i^*}), x^*, w^*) + m_\beta$, which is equivalent to $d^* := \Lambda_{sk'_{i^*}}(x^*) + m_\beta$.

Step 5. Plaintexts m_β are perfectly hidden due to the $\mathscr{L}_0$-multi-key-multi-extracting property. Note that the new keys $\{sk'_{i^*}\}_{i^*\in[n]}$ are uniform and only used for computing $d^* := \Lambda_{sk'_{i^*}}(x^*) + m_\beta$.

- By the $\mathscr{L}_0$-multi-key-multi-extracting of QAHPS, the hash values $\Lambda_{sk'_{i^*}}(x^*)$ are pseudorandom, so all the d^*'s can be replaced by random elements.

Hence d^* perfectly hides m_β, and $\mathcal{A}$ has no advantage in guessing β.

Strategy for corruptions in reductions. Similar to the security reductions for SIG, the reduction algorithms in steps 1, 2, 4, 5 can handle $\mathcal{A}$'s adaptive corruption queries by choosing all users' secret keys themselves.

In particular, in step 5, new keys $\{sk'_{i^*}\}_{i^*\in[n]}$ (for answering encryption queries) have been split from $\{sk_{i^*}\}_{i^*\in[n]}$ (for answering adaptive corruptions, decryption queries and generation of public keys). Thus the reduction algorithm to the $\mathscr{L}_0$-multi-key-multi-extracting property of QAHPS is able to implicitly set $\{sk'_{i^*}\}_{i^*\in[n]}$ as the keys chosen by its own challenger, but choose $\{sk_{i^*}\}_{i^*\in[n]}$ itself to deal with $\mathcal{A}$'s adaptive corruption queries.

How we circumvent the seemingly paradoxical technical problem. Now we conclude how we circumvent the paradoxical technical problem for achieving tight $\mathsf{MUMC^c\text{-}CCA}$ security of PKE: our proof goes with a constant number of computationally indistinguishable changes, as well as n statistical changes, to arrive at a final game where the challenge ciphertexts are no longer generated by the users' real secret keys.

(1) All the reduction algorithms to computational properties or problems possess the secret keys of all users to handle adaptive corruption queries.
(2) With n statistical changes ($\langle\mathscr{L},\mathscr{L}_0\rangle$-key-switching), new and independent secret keys (for generating challenge ciphertexts) have been split from real secret keys (for corruption and other queries), ready for the final game.
(3) In the final game, the reduction algorithm (for $\mathscr{L}_0$-multi-key-multi-extracting) can embed its challenge instances in the new secret keys to randomize challenge ciphertexts, and sample the real secret keys itself to handle adaptive corruption queries from the adversary.

How we circumvent the existing impossibility results. Recall that the impossibility results apply to a PKE scheme when the relation between the public key and the secret key is "unique" or "re-randomizable" [5]. For reasons similar to our SIG (as shown in Subsect. 2.1), we can show that the relation between the public key $pk = \alpha_\rho(sk)$ and the secret key sk of our PKE is neither "unique" nor "re-randomizable", by the new properties we defined for QA-HPS.

2.3 Our SC, MAC and AE: Technical Overview

Our SC. There are a variety of constructions for building SignCryption (SC) from SIG and PKE, encompassing "Encrypt-then-Sign", "Sign-then-Encrypt", "Encrypt-and-Sign", etc. [3, 9]. However, there is no SC available with tight

$\mathsf{MUMC^c\text{-}Priv\&Auth}$ (multi-user multi-challenge CCA privacy and authenticity under corruptions) in the standard model. As far as we see, this is mainly due to the missing of tightly *strongly* $\mathsf{MU^c}$ secure SIG and tightly $\mathsf{MU^c}$ secure PKE.

Our SIG and PKE constructions fill the blank and immediately lead to tightly $\mathsf{MUMC^c\text{-}Priv\&Auth}$ secure SC.

Moreover, we can optimize the SC construction by taking advantage of the similar structures and compatible underlying building blocks of our SIG and PKE. In our optimized construction of SC, we integrate the ciphertext of PKE and signature of SIG in a more efficient way of reusing the instance $x \in \mathcal{L}_\rho$ and the proof π of QANIZK, and the signcryption of message m is now given by

$$c := (x \leftarrow_{\$} \mathcal{L}_\rho,\ d := \mathsf{Pub}(pk_r, x, w) + m,\ \widetilde{d} := \mathsf{Priv}(\widetilde{sk}_s, x),\ \pi := \mathsf{Prove}(\mathsf{crs}, \tau, x, w)),$$

where $\tau := H(\widetilde{vk}_s, pk_r, d, \widetilde{d})$, pk_r is receiver's public (encryption) key and $\widetilde{sk}_s$ sender's secret (signing) key. The tight $\mathsf{MUMC^c\text{-}Priv\&Auth}$ security of our SC can be proved similar to the tight $\mathsf{MU^c}$ security of PKE and SIG.

Our MAC and AE. A SIG scheme is itself a MAC scheme and a SC scheme is an AE scheme, when taking the secret key as the symmetric key. Therefore, our SIG and SC constructions immediately lead to a strongly $\mathsf{MU^c\text{-}CMA}$ secure MAC and $\mathsf{MUMC^c\text{-}Priv\&Auth}$ secure AE. However, we can do more about MAC since it does not need public verification. We provide a more efficient MAC following our SIG construction but replacing the building block $\mathsf{PVQAHPS}$ by QAHPS with new properties. Furthermore, the security of MAC can also be improved to an even stronger notion, namely strong $\mathsf{MU^c\text{-}CMVA}$ security, which considers *chosen verification attacks* as well [15] in addition to strong $\mathsf{MU^c\text{-}CMA}$.

2.4 Instantiations from MDDH Assumptions and Leakage Resilience

Instantiations. We instantiate PV-QA-HPS and QA-HPS with new properties from the MDDH assumptions. The associated language collections $\mathscr{L}$ and $\mathscr{L}_0$ are independently generated linear subspaces [27]. The instantiations stem from the DDH-based HPS proposed by Cramer and Shoup [13], and rely on pairing groups to accomplish public verifiability of PV-QA-HPS, inspired by [27]. We provide tight security proofs for the properties of PV-QA-HPS and QA-HPS based on MDDH. Below we give a high-level overview of our PV-QA-HPS instantiation. We rely on an asymmetric pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ of prime order p with $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$. We use implicit representation of group elements [17], namely, using $[\cdot]_1, [\cdot]_2, [\cdot]_T$ to denote component-wise exponentiations in respective groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$.

- Let us start with the Cramer-Shoup HPS [13]. We describe the MDDH-based generalized version with $k \geq 1$ the MDDH parameter ($k = 1$ corresponds to the original DDH-based version). The hashing key is $sk = \mathbf{K} \in \mathbb{Z}_p^{(k+1)\times(2k+1)}$ and the projection key is $pk = [\mathbf{KA}]_1$ on a linear subspace language $\mathcal{L}_\rho = \mathsf{Span}([\mathbf{A}]_1) = \left\{ [\mathbf{c}]_1 \middle| \exists\ \mathbf{w} \in \mathbb{Z}_p^k, \text{ s.t. } [\mathbf{c}]_1 = [\mathbf{Aw}]_1 \right\}$ with

$\rho = [\mathbf{A}]_1 \in \mathbb{G}_1^{(2k+1)\times k}$. For an instance $[\mathbf{c}]_1 = [\mathbf{A}\mathbf{w}]_1 \in \mathcal{L}_\rho$, the HPS hash value is given by $[\boldsymbol{hv}]_1 =$

$$\text{(private evaluation)} \quad \mathbf{K} \cdot [\mathbf{c}]_1 = [\mathbf{K}\mathbf{A}]_1 \cdot \mathbf{w} \quad \text{(public evaluation)}.$$

- To support public verification, we resort to pairing technique, inspired by the Kiltz-Wee QA-NIZK [27]. We use $vk = [\mathbf{K}^\top \mathbf{B}]_2$ as the verification key with matrix $[\mathbf{B}]_2 \in \mathbb{G}_2^{(k+1)\times k}$ defined by the MDDH assumption. Then, the correctness of hash value $[\boldsymbol{hv}]_1 \stackrel{?}{=} [\mathbf{K}\mathbf{c}]_1$ can be verified publicly via pairing:

$$e([\boldsymbol{hv}^\top]_1, [\mathbf{B}]_2) \stackrel{?}{=} e([\mathbf{c}^\top]_1, [\mathbf{K}^\top \mathbf{B}]_2) \quad (= [(\mathbf{K}\mathbf{c})^\top \mathbf{B}]_T).$$

Verification soundness. This is tightly implied by the Kernal Matrix DH (KerMDH) assumption [33], which in turn is implied by the MDDH assumption [33]. If the adversary is able to produce an incorrect hash value $[\boldsymbol{hv}]_1 \neq [\mathbf{K}\mathbf{c}]_1$ but passes the public verification $e([\boldsymbol{hv}^\top]_1, [\mathbf{B}]_2) = e([\mathbf{c}^\top]_1, [\mathbf{K}^\top \mathbf{B}]_2)$, then $[\boldsymbol{hv} - \mathbf{K}\mathbf{c}]_1$ is a non-zero element such that $e([(\boldsymbol{hv} - \mathbf{K}\mathbf{c})^\top]_1, [\mathbf{B}]_2) = [\mathbf{0}]_T$, resulting in a solution to the KerMDH problem defined by $[\mathbf{B}]_2$.

$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting. This holds information-theoretically, where $\mathcal{L}_{\rho_0} = \mathsf{Span}([\mathbf{A}_0]_1) \in \mathscr{L}_0$ and $\mathcal{L}_\rho = \mathsf{Span}([\mathbf{A}]_1) \in \mathscr{L}$ with $\rho_0 = [\mathbf{A}_0]_1 \in \mathbb{G}_1^{(2k+1)\times k}$ chosen independently of $\rho = [\mathbf{A}]_1$. Note that $\mathbf{A}_0$ is $(2k+1)$ by k, $\mathbf{B}$ is $(k+1)$ by k, and $sk = \mathbf{K}$ is $(k+1)$ by $(2k+1)$ matrices. Given the projection key $pk_{\rho_0} = [\mathbf{K}\mathbf{A}_0]_1$ w.r.t. $\mathcal{L}_{\rho_0}$ and $vk = [\mathbf{K}^\top \mathbf{B}]_2$, the hashing key $sk = \mathbf{K}$ reserves entropy in its projection on the kernel of $\mathbf{A}_0$ and $\mathbf{B}$. Then for any (non-zero) instance $[\mathbf{c}]_1 \in \mathcal{L}_\rho = \mathsf{Span}([\mathbf{A}]_1)$, $[\mathbf{c}]_1$ is outside $\mathcal{L}_{\rho_0} = \mathsf{Span}([\mathbf{A}_0]_1)$, thus the reserved entropy of $sk = \mathbf{K}$ is transmitted to the hash value $[\mathbf{K}\mathbf{c}]_1$ so that the adversary can hardly guess $[\mathbf{K}\mathbf{c}]_1$ correctly. This holds even if some extra (bounded) information of $sk = \mathbf{K}$ is leaked to the adversary.

The instantiation of tag-based QA-NIZK is adapted from the QA-NIZK scheme proposed by Abe et al. [1], which has tight USS based on MDDH.

According to our generic constructions, the instantiations of PV-QA-HPS, QA-HPS and tag-based QA-NIZK result in concrete SIG, PKE, SC, MAC, AE schemes with tight MU^{c} security from MDDH in the standard model.

Leakage resilience. Note that HPS is intrinsically leakage resilient [34]. The leakage resilience can naturally extend to QA-HPS [22], and also to PV-QA-HPS. More precisely, we define leakage-resilient-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting property for PV-QA-HPS (cf. Sect. 4) and adopt the leakage-resilient-$\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching for QA-HPS defined in [22], which are met by our MDDH-based instantiations. This shows that all our SIG, PKE, SC, MAC, AE schemes not only have tight MU^{c} security but also support key leakage, thus achieving tight $\mathsf{MU}^{\mathsf{c\&l}}$ security.

The tight $\mathsf{MU}^{\mathsf{c\&l}}$ security protects our schemes from key leakages on the uncorrupted users besides adaptive corruptions. When used in the construction of more advanced protocols, the applications of our tightly $\mathsf{MU}^{\mathsf{c\&l}}$ secure primitives may also improve the security of the protocols to be leakage resilient ones. For

instance, we can always make a drop-in replacement of the tightly MU^{c} secure SIG with our tightly $\mathsf{MU}^{\mathsf{c\&l}}$ secure SIG in the construction of tightly secure authenticated key exchange (AKE) protocols [4, 31, 21] where the signing key of SIG serves as the long-term secret key of AKE, and the resulting AKEs readily augment their tight security with leakage-resilience.

Moreover, our tightly $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ secure PKE scheme has essential improvements in terms of leakage resilience beyond corruptions, compared with the tightly leakage-resilient CCA-secure PKE scheme in [22]. See Table 2. Concretely, (1) our leakage rate is $\frac{1}{3} - o(1)$ while theirs is $\frac{1}{18} - o(1)$; (2) our multi-user leakage model is stronger than theirs, since their model [22, Appendix A.1] does not allow any leakage queries to any user after the very first encryption query to *any* user, while our model allows leakage queries for any particular user until the first encryption query to *that* user (cf. Def. 16 and Remark 3 in Subsect. 6.1). Informally speaking, our PKE achieves the stronger multi-user leakage resilience mainly due to the introduction of *multi-language multi-fold SMP*, which helps to switch $\mathcal{L}_\rho$ to different and *independently chosen* languages $\{\mathcal{L}_{\rho_0^{(i)}}\}$ for different users, thus the leakages w.r.t. different users can be handled independently.

2.5 Comparison with Existing Techniques for Tight MU^{c} Security

Most existing works on tight MU^{c} security [4, 20, 29, 14] designed their schemes in a *"double encryption/signing"* fashion (the only exception is [21]), and the secret key of their schemes consists of only one key (say sk_0) out of two possible keys (say sk_0, sk_1). For example, in [4, 29], their PKE encrypts plaintext by running a "sub-encryption procedure" twice (possibly in a correlated way), resulting in a ciphertext containing two "sub-ciphertexts" of the plaintext, and there are two decryption ways according to which possible key (sk_0 or sk_1) is used. In their tight MU^{c} security proofs, the reduction algorithms always possess the real secret keys (sk_0) of all users, while embed the challenges in the other possible keys (sk_1). With this strategy, their reductions can handle adaptive corruptions.

In contrast, all our constructions are different from the "double encryption/signing" design. For example, it is hard to split the ciphertext of our PKE to two "sub-ciphertexts". So the proof strategy in [4, 20, 29, 14] does not apply.

We develop two different novel proof strategies for tight strong MU^{c}-CMA security of SIG and tight $\mathsf{MUMC}^{\mathsf{c}}$-$\mathsf{CCA}$ security of PKE (cf. Fig. 1 and Fig. 2), respectively. At a high level, we do not "double" the secret key by construction, but "split" the key during our tight proofs, which can be summarized as first "switch the languages for different oracles" then "apply quasi-adaptive properties" (such as $\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting, $\langle \mathscr{L}, \mathscr{L}_0 \rangle$-Key-switching, $\mathscr{L}_0$-Multi-key multi-extracting).

3 Preliminaries

Notations. Let $\lambda \in \mathbb{N}$ denote the security parameter throughout the paper, and all algorithms, distributions, functions and adversaries take 1^λ as an implicit

input. 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 $n \in \mathbb{N}$, define $[n] := \{1, 2, ..., n\}$. For a set $\mathcal{X}$, denote by $x \leftarrow_{\$} \mathcal{X}$ the procedure of sampling x from $\mathcal{X}$ uniformly at random. If $\mathcal{D}$ is distribution, $x \leftarrow_{\$} \mathcal{D}$ means that x is sampled according to $\mathcal{D}$. All our algorithms are probabilistic unless stated otherwise. We use $y \leftarrow_{\$} \mathcal{A}(x)$ to define the random variable y obtained by executing algorithm $\mathcal{A}$ on input x. 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. By $\Pr_i[\cdot]$ we denote the probability of a particular event occurring in game G_i.

The syntax of digital signature (SIG), public-key encryption (PKE) and the definition of collision-resistant hash functions are presented in Appendix B.

3.1 Language Distribution

We formalize a collection of NP-languages as a language distribution.

Definition 1 (Language Distribution). *A language distribution $\mathscr{L}$ is a probability distribution that outputs a language parameter ρ as well as a trapdoor td in polynomial time. The language parameter ρ publicly defines an NP-language $\mathcal{L}_\rho \subseteq \mathcal{X}_\rho$. For simplicity, we assume that the universe $\mathcal{X}_\rho$ is the same for all parameters ρ output by all distributions $\mathscr{L}$, and denoted by $\mathcal{X}$. The trapdoor td is required to contain enough information for efficiently deciding whether an instance $x \in \mathcal{X}$ is in $\mathcal{L}_\rho$. We require that there are PPT algorithms for sampling $x \leftarrow_{\$} \mathcal{L}_\rho$ uniformly together with a witness w and sampling $x \leftarrow_{\$} \mathcal{X}$ uniformly.*

A language distribution is associated with a subset membership problem (SMP), which asks whether an element is uniformly 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 a language distribution $\mathscr{L}$ 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 the probability is over $(\rho, td) \leftarrow_{\$} \mathscr{L}$, $x \leftarrow_{\$} \mathcal{L}_\rho$ and $x' \leftarrow_{\$} \mathcal{X}$.*

Definition 3 (Multi-fold SMP). *The multi-fold SMP related to a language distribution $\mathscr{L}$ 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 $(\rho, td) \leftarrow_{\$} \mathscr{L}$, $x_1, ..., x_Q \leftarrow_{\$} \mathcal{L}_\rho$ and $x'_1, ..., x'_Q \leftarrow_{\$} \mathcal{X}$.*

3.2 Quasi-Adaptive Hash Proof System

Hash proof system (HPS) was proposed by Cramer and Shoup [13], and turned out to be a powerful tool in a wide range of applications. Han et al. [22] 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. We give the definition of QA-HPS according to [22].

Definition 4 (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 public evaluation algorithm outputs a hash value* $hv = \Lambda_{sk}(x) \in \mathcal{HV}$*.*
- $hv \leftarrow \mathsf{Priv}(sk, x)$*: Taking as input a hashing key* sk *and an instance* $x \in \mathcal{X}$*, the private evaluation algorithm outputs a hash value* $hv = \Lambda_{sk}(x) \in \mathcal{HV}$*.*

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

We can naturally define QA-HPS for two language distributions $\mathscr{L}$ and $\mathscr{L}_0$, by requiring correctness to hold not only for language parameters ρ output by $\mathscr{L}$, but also for language parameters ρ_0 output by $\mathscr{L}_0$.

We recall a statistical property of QA-HPS from [22], parameterized by $\kappa \in \mathbb{N}$ and two language distributions $\mathscr{L}$, $\mathscr{L}_0$, called *κ-leakage-resilient(LR)-$\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching*. Informally speaking, it stipulates that in the presence of a projection key $\alpha_\rho(sk)$ w.r.t. a language parameter ρ output by $\mathscr{L}$ and given κ bits leakage information about sk, 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 5 (κ-LR-$\langle \mathscr{L}, \mathscr{L}_0 \rangle$-Key-Switching of QA-HPS). *Let* $\kappa = \kappa(\lambda) \in \mathbb{N}$*, and let* $\mathscr{L}$ *and* $\mathscr{L}_0$ *be a pair of language distributions. A QA-HPS scheme* QAHPS *for* $\mathscr{L}$ *supports* κ*-LR-*$\langle \mathscr{L}, \mathscr{L}_0 \rangle$*-key-switching, if for any (possibly unbounded) adversary* $\mathcal{A}$*, it holds that* $\boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS}, \mathcal{A}, \kappa}(\lambda) := \left| \Pr[\mathsf{Exp}^{\mathsf{lr}\text{-}\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS}, \mathcal{A}, \kappa} \Rightarrow 1] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{lr}\text{-}\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS}, \mathcal{A}, \kappa}$ *is specified in Fig. 3.*

Remark 1. In the experiment $\mathsf{Exp}^{\mathsf{lr}\text{-}\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS}, \mathcal{A}, \kappa}$ shown in Fig. 3, $\mathcal{A}$ is allowed to obtain at most κ bits leakage information about sk through oracle $\mathcal{O}_{\text{LEAK}}$: each time $\mathcal{A}$ submits a function L and obtains $L(sk)$ which outputs at least one bit.

Note that $\mathcal{A}$ is not allowed to query $\mathcal{O}_{\text{LEAK}}$ anymore after receiving the challenge $(\rho_0, ...)$ from $\mathcal{O}_{\text{CHAL}}$ (guaranteed by the variable chal). This is necessary to exclude the trivial attacks that after seeing ρ_0, $\mathcal{A}$ can set $L(\cdot)$ to be the first few bits of $\alpha_{\rho_0}(\cdot)$, thus trivially distinguish $\alpha_{\rho_0}(sk)$ from $\alpha_{\rho_0}(sk')$.

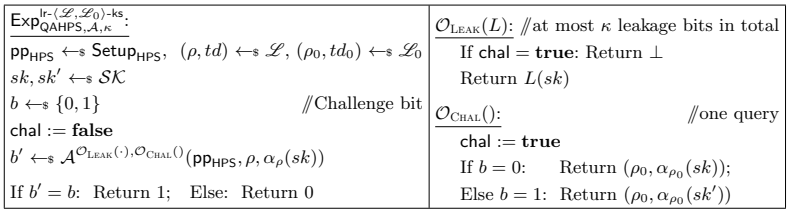

Fig. 3. The κ-LR-$\langle\mathscr{L},\mathscr{L}_0\rangle$-Key-Switching experiment $\mathsf{Exp}^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{A},\kappa}$ for QAHPS.

3.3 Tag-based Quasi-Adaptive Non-Interactive Zero-Knowledge

Quasi-Adaptive Non-Interactive Zero-Knowledge argument (QA-NIZK) was proposed by Jutla and Roy [26], where the common reference string (CRS) may depend on the specific language $\mathcal{L}_\rho$ for which proofs are generated. We present the formal definition of QA-NIZK in its *tag-based* variant following [27].

Definition 6 (Tag-based QA-NIZK). *A tag-based quasi-adaptive non-interactive zero-knowledge scheme* $\mathsf{QANIZK} = (\mathsf{Setup}_{\mathsf{NIZK}}, \mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{Sim})$ *for a language distribution* $\mathscr{L}$ *with tag space* $\mathcal{T}$ *consists of five PPT algorithms:*

- $\mathsf{pp}_{\mathsf{NIZK}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{NIZK}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp}_{\mathsf{NIZK}}$*, which serves as an implicit input of other algorithms.*
- $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow\!\!\$\ \mathsf{CRSGen}(\rho)$*: Taking as input a language parameter* ρ*, the CRS generation algorithm outputs a common reference string (CRS)* crs *and a simulation trapdoor* $\mathsf{td}_{\mathsf{crs}}$*.*
- $\pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$*: 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.*
- $\pi \leftarrow\!\!\$\ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$*: 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*.*

Perfect completeness requires that for all $(\rho, td) \in \mathscr{L}$*,* $\mathsf{pp}_{\mathsf{NIZK}} \in \mathsf{Setup}_{\mathsf{NIZK}}$*,* $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \in \mathsf{CRSGen}(\rho)$*,* $\tau \in \mathcal{T}$*,* $x \in \mathcal{L}_\rho$ *with witness* w*,* $\pi \in \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$*, it holds that* $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$*.*

Perfect zero-knowledge requires that for all $(\rho, td) \in \mathscr{L}$*,* $\mathsf{pp}_{\mathsf{NIZK}} \in \mathsf{Setup}_{\mathsf{NIZK}}$*,* $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \in \mathsf{CRSGen}(\rho)$*,* $\tau \in \mathcal{T}$*,* $x \in \mathcal{L}_\rho$ *with witness* w*, the outputs of* $\mathsf{Prove}(\mathsf{crs}, \tau, x, w)$ *and* $\mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$ *are identically distributed, where the probability is over the inner coin tosses of* Prove *and* Sim*.*

Below we define *Unbounded Simulation-Soundness* (USS) according to [24, 1].

Definition 7 (USS of Tag-based QA-NIZK). *A tag-based QA-NIZK scheme* QANIZK *for* $\mathscr{L}$ *has unbounded simulation-soundness (USS), if for any PPT adversary* $\mathcal{A}$*, it holds that* $\mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}$ *is defined in Fig. 4.*

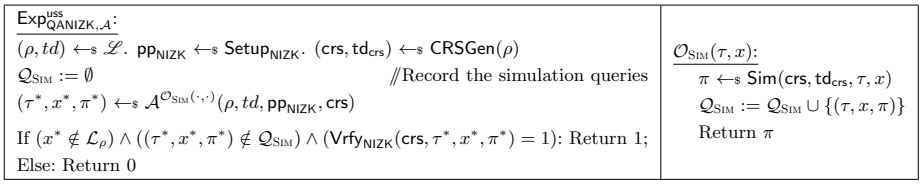

Fig. 4. The Unbounded Simulation-Soundness experiment $\mathsf{Exp}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{A}}$ for QANIZK.

We note that the above USS definition for tag-based QA-NIZK is stronger than the usual one in [27, 18] in two aspects.

- Firstly, $\mathcal{A}$ is given the trapdoor td of the language parameter ρ. Recall that td contains enough information for efficiently deciding whether or not an instance x is in $\mathcal{L}_\rho$. This is stronger than the usual USS, but weaker than the *USS for witness-sampleable distributions* defined in [24, 1], where $\mathcal{A}$ essentially samples (ρ, td) itself and provides (ρ, td) to the experiment.
- Secondly, $\mathcal{A}$ is allowed to output a forgery with a reused tag.

In [1], Abe et al. proposed a QA-NIZK scheme with tight USS for witness-sampleable distributions based on the MDDH assumptions. As noted in [1, Subsect. 3.2], their scheme can be easily extended to a tag-based QA-NIZK scheme with tight USS, by using collision-resistant hash functions. For completeness, we present the tag-based QA-NIZK scheme with tight USS in Appendix I.5.

4 Publicly-Verifiable QA-HPS and New Properties

In this section, we propose a new variant of QA-HPS, called *Publicly-Verifiable QA-HPS* (PV-QA-HPS), which additionally enables public verification of hash values with an extra verification key. Then we formalize a set of computational and statistical properties for PV-QA-HPS and QA-HPS serving different applications in subsequent sections.

- For PV-QA-HPS, we define a computational *verification soundness* and statistical properties including *leakage-resilient one-time-extracting (LR-OT-extracting)* and *verification key diversity (VK-diversity)*. PV-QA-HPS will be an important building block for SIG in Sect. 5 and these properties help SIG to achieve tight multi-user security under corruptions and leakages.
- For QA-HPS, we define a computational *multi-key-multi-extracting* and a statistical *projection key diversity (PK-diversity)*. We also define a *multi-language multi-fold SMP* for language distributions. QA-HPS will be an important building block for PKE in Sect. 6, and these new properties help PKE to achieve tight multi-user security under corruptions and leakages.

Jumping ahead, we will give instantiations of PV-QA-HPS and QA-HPS based on the matrix DDH (MDDH) assumptions in Sect. 7 and Appendix I.

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

Definition 8 (PV-QA-HPS). *A publicly-verifiable QA-HPS (PV-QA-HPS) scheme* $\mathsf{PVQAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \nu, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Vrfy}_{\mathsf{HPS}})$ *for a language distribution* $\mathscr{L}$ *consists of six PPT algorithms:*

- $(\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{Pub}, \mathsf{Priv})$ *is a QA-HPS scheme for* $\mathscr{L}$ *as per Definition 4.*
- $\mathsf{pp}_{\mathsf{HPS}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{HPS}}$*: It outputs a public parameter* $\mathsf{pp}_{\mathsf{HPS}}$*, which also defines a verification key space* $\mathcal{VK}$ *besides* $(\mathcal{SK}, \mathcal{HV}, \Lambda_{(\cdot)})$ *as per Definition 4.*
- $vk \leftarrow \nu(sk)$*: Taking as input a hashing key* $sk \in \mathcal{SK}$*, the verification key generation algorithm outputs a verification key* $vk \in \mathcal{VK}$*.*
- $0/1 \leftarrow \mathsf{Vrfy}_{\mathsf{HPS}}(vk, x, hv)$*: Taking as input a verification key* $vk = \nu(sk) \in \mathcal{VK}$*, an instance* $x \in \mathcal{X}$ *and a hash value* $hv \in \mathcal{HV}$*, the deterministic verification algorithm outputs a bit indicating whether* $hv = \Lambda_{sk}(x)$ *or not.*

Verification completeness requires that for all $(\rho, td) \in \mathscr{L}$*,* $\mathsf{pp}_{\mathsf{HPS}} \in \mathsf{Setup}_{\mathsf{HPS}}$*,* $sk \in \mathcal{SK}$*,* $x \in \mathcal{X}$*,* $vk := \nu(sk)$ *and* $hv := \Lambda_{sk}(x)$*, it holds* $\mathsf{Vrfy}_{\mathsf{HPS}}(vk, x, hv) = 1$*.*

Remark 2 (Relations between PV-QA-HPS and QA-NIZK). PV-QA-HPS can be viewed as a special kind of Designated-Prover (DP) QA-NIZK [1], but with different properties. The pk_ρ of PV-QA-HPS can be viewed as the proving key of DP-QA-NIZK, sk as the simulation trapdoor and vk as the common reference string (used for verification). With pk_ρ, the prover can prove $x \in \mathcal{L}_\rho$ with the help of a witness w via $hv \leftarrow \mathsf{Pub}(pk_\rho, x, w)$, where the hash value hv can be viewed as a proof for $x \in \mathcal{L}_\rho$. With vk, the verifier can check whether hv is a valid proof for $x \in \mathcal{L}_\rho$ via $\mathsf{Vrfy}_{\mathsf{HPS}}(vk, x, hv)$. Moreover, with sk, the simulator can generate a proof for x without knowing a witness via $hv \leftarrow \mathsf{Priv}(sk, x)$.

Verification completeness of PV-QA-HPS corresponds to the perfect completeness of DP-QA-NIZK. Correctness of (PV-)QA-HPS guarantees $\mathsf{Pub}(pk_\rho, x, w) = \mathsf{Priv}(sk, x)$ for all $x \in \mathcal{L}_\rho$ with witness w, thus corresponding to the perfect zero-knowledge of DP-QA-NIZK.

On the other hand, PV-QA-HPS has its own features. Firstly, it has a projection function $\alpha_\rho(\cdot)$ (which is inherent to HPS) and a verification key generation function $\nu(\cdot)$. Secondly, a set of properties of PV-QA-HPS and QA-HPS are built upon functions $\alpha_\rho(\cdot)$ and/or $\nu(\cdot)$. For instance, the κ-LR-$\langle \mathscr{L}, \mathscr{L}_0 \rangle$-Key-Switching (cf. Def. 5 in Subsect. 3.2) is closely associated with $\alpha_\rho(\cdot)$.

Next we define a computational *verification soundness* for PV-QA-HPS in the setting of multiple keys. Intuitively, it requires that for any (sk, vk) among the multiple key pairs, a PPT adversary cannot find a tuple $(x^* \in \mathcal{X}, hv^*)$ such that $hv^* \neq \Lambda_{sk}(x^*)$ but $\mathsf{Vrfy}_{\mathsf{HPS}}(vk, x^*, hv^*) = 1$, even given all the key pairs.

Definition 9 (Verification Soundness of PV-QA-HPS). *A PV-QA-HPS scheme* $\mathsf{PVQAHPS}$ *for* $\mathscr{L}$ *has verification soundness, if for any PPT adversary* $\mathcal{A}$ *and any polynomial* $n = \mathsf{poly}(\lambda)$*, it holds that* $\mathsf{Adv}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{A},n}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{A},n} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*, where* $\mathsf{Exp}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{A},n}$ *is defined in Fig. 5.*

$\underline{\mathsf{Exp}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{A},n}}$:

$\mathsf{pp}_{\mathsf{HPS}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{HPS}}$. For $i \in [n]$: $sk_i \leftarrow\!\!\$\ \mathcal{SK}$, $vk_i := \nu(sk_i)$

$(i^* \in [n], x^* \in \mathcal{X}, hv^*) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, (sk_i, vk_i)_{i\in[n]})$

If $(hv^* \neq \Lambda_{sk_{i^*}}(x^*)) \wedge (\mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, x^*, hv^*) = 1)$: Return 1; Else: Return 0

Fig. 5. Verification Soundness experiment $\mathsf{Exp}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{A},n}$ for PVQAHPS.

We formalize a statistical extracting property for (PV-)QA-HPS, parameterized by $\kappa \in \mathbb{N}$ and two language distributions $\mathscr{L}_0$, $\mathscr{L}$, called *κ-leakage-resilient(LR)-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-one-time(OT)-extracting*. 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 $\mathscr{L}_0$ and given κ bits leakage information about sk. For PV-QA-HPS, it requires the property to hold even in the presence of the verification key $\nu(sk)$.

Definition 10 (κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-Extracting of QA-HPS and PV-QA-HPS). *Let $\kappa = \kappa(\lambda) \in \mathbb{N}$, and let $\mathscr{L}_0$ and $\mathscr{L}$ be a pair of language distributions. A (PV-)QA-HPS scheme* (PV)QAHPS *for $\mathscr{L}$ supports κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting, if for any (unbounded) adversary $\mathcal{A}$, it holds that $\epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{(PV)QAHPS},\mathcal{A},\kappa}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{(PV)QAHPS},\mathcal{A},\kappa} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$, where $\mathsf{Exp}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{(PV)QAHPS},\mathcal{A},\kappa}$ is defined in Fig. 6.*

$\underline{\mathsf{Exp}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{(PV)QAHPS},\mathcal{A},\kappa}}$:

$\mathsf{pp}_{\mathsf{HPS}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{HPS}}$. $(\rho_0, td_0) \leftarrow\!\!\$\ \mathscr{L}_0$, $(\rho, td) \leftarrow\!\!\$\ \mathscr{L}$. $sk \leftarrow\!\!\$\ \mathcal{SK}$

$(x^*, hv^*) \leftarrow\!\!\$\ \mathcal{A}^{\mathcal{O}_{\text{LEAK}}(\cdot)}(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \rho, \alpha_{\rho_0}(sk), \boxed{\nu(sk)})$

If $(x^* \in \mathcal{L}_\rho) \wedge (hv^* = \Lambda_{sk}(x^*))$: Return 1; Else: Return 0

$\underline{\mathcal{O}_{\text{LEAK}}(L)}$: //at most κ leakage //bits in total

Return $L(sk)$

Fig. 6. The κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-Extracting experiment $\mathsf{Exp}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{(PV)QAHPS},\mathcal{A},\kappa}$ for QAHPS (without gray part) and Publicly-Verifiable PVQAHPS (with gray part).

Han et al. [22] proposed a computational property for QA-HPS, called *$\mathscr{L}_0$-multi-extracting*, which demands the pseudorandomness of $\Lambda_{sk}(x_j)$ for multiple instances $x_j \in \mathcal{L}_{\rho_0}$ $(j \in [Q])$ with ρ_0 output by $\mathscr{L}_0$, when sk is uniformly chosen from $\mathcal{SK}$. We extend this property in the multi-key setting as follows.

Definition 11 ($\mathscr{L}_0$-Multi-Key-Multi-Extracting of QA-HPS). *A QA-HPS scheme* QAHPS *for $\mathscr{L}$ supports $\mathscr{L}_0$-multi-key-multi-extracting, if for any PPT $\mathcal{A}$, any polynomial $n = \mathsf{poly}(\lambda)$ and any polynomial $Q = \mathsf{poly}(\lambda)$, it holds*

$$\mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk}\text{-}\mathsf{mext}}_{\mathsf{QAHPS},\mathcal{A},n,Q}(\lambda) := |\Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \{x_j, \boxed{\{\Lambda_{sk_i}(x_j)\}_{i\in[n]}}\}_{j\in[Q]}) = 1] - \Pr[\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \{x_j, \boxed{\{hv_{i,j}\}_{i\in[n]}}\}_{j\in[Q]}) = 1]| \leq \mathsf{negl}(\lambda),$$

where $\mathsf{pp}_{\mathsf{HPS}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{HPS}}$, $(\rho_0, td_0) \leftarrow\!\!\$\ \mathscr{L}_0$, $sk_1, ..., sk_n \leftarrow\!\!\$\ \mathcal{SK}$, $x_1, ..., x_Q \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$ and $hv_{1,1}, ..., hv_{n,Q} \leftarrow\!\!\$\ \mathcal{HV}$.

We formalize two statistical properties, called *projection key diversity (PK-diversity)* and *verification key diversity (VK-diversity)*, for QA-HPS and PV-QA-HPS respectively. Intuitively, PK-diversity (resp. VK-diversity) expresses statistical collision resistance of projection keys (resp. verification keys) under different hashing keys.

Definition 12 (PK-Diversity of QA-HPS). *A QA-HPS scheme* QAHPS *for* $\mathscr{L}$ *has projection key diversity (PK-diversity), if* $\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) := \Pr[\alpha_\rho(sk) = \alpha_\rho(sk')] \leq \mathsf{negl}(\lambda)$*, where* $(\rho, td) \leftarrow_\$ \mathscr{L}$*,* $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_\$ \mathsf{Setup}_{\mathsf{HPS}}$ *and* $sk, sk' \leftarrow_\$ \mathcal{SK}$*.*

Definition 13 (VK-Diversity of PV-QA-HPS). *A PV-QA-HPS scheme* $\mathsf{PVQAHPS}$ *for* $\mathscr{L}$ *has verification key diversity (VK-diversity), if* $\boldsymbol{\epsilon}^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}}(\lambda) := \Pr[\nu(sk) = \nu(sk')] \leq \mathsf{negl}(\lambda)$*, where* $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_\$ \mathsf{Setup}_{\mathsf{HPS}}$ *and* $sk, sk' \leftarrow_\$ \mathcal{SK}$*.*

Finally, we define a *multi-language* multi-fold SMP for language distributions.

Definition 14 (Multi-Language Multi-fold SMP). *The multi-language multi-fold SMP related to* $\mathscr{L}$ *is hard, if for any PPT adversary* $\mathcal{A}$*, any polynomial* $n = \mathsf{poly}(\lambda)$ *and any polynomial* $Q = \mathsf{poly}(\lambda)$*, it holds that* $\mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L},\mathcal{A},n,Q}(\lambda) := |\Pr[\mathcal{A}(\{\rho^{(i)}, \{x_j^{(i)}\}_{j\in[Q]}\}_{i\in[n]}) = 1] - \Pr[\mathcal{A}(\{\rho^{(i)}, \{x_j'^{(i)}\}_{j\in[Q]}\}_{i\in[n]}) = 1]| \leq \mathsf{negl}(\lambda)$*, where for each* $i \in [n]$*,* $(\rho^{(i)}, td^{(i)}) \leftarrow_\$ \mathscr{L}$*,* $x_1^{(i)}, ..., x_Q^{(i)} \leftarrow_\$ \mathcal{L}_{\rho^{(i)}}$*,* $x_1'^{(i)}, ..., x_Q'^{(i)} \leftarrow_\$ \mathcal{X}$*.*

Multi-language multi-fold SMP can generally be reduced to SMP with a security loss of nQ with n the number of languages and Q the number of folds per language. For some language distributions, such as those for linear subspaces based on the matrix DDH (MDDH) assumptions (cf. Appendix I.2), the hardness of multi-language multi-fold SMP can be tightly reduced to that of SMP.

5 SIG with Tight Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA Security

In this section, we present digital signature (SIG) schemes with tight strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA security, by using Publicly-Verifiable QA-HPS (PV-QA-NIZK) formalized in Sect. 4 as a central building block.

In Subsect. 5.1, we define the strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA security of SIG. Then in Subsect. 5.2, we present our generic construction of SIG.

5.1 Definition of Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA Security

In [4], 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 (MU^{c}-CMA). Here we extend it to $\mathsf{MU}^{\mathsf{c\&l}}$-CMA, which considers existential unforgeability under not only chosen-message attacks and adaptive corruptions but also key leakages in the multi-user setting. Moreover, *strong* $\mathsf{MU}^{\mathsf{c\&l}}$-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 $\mathsf{MU}^{\mathsf{c\&l}}$-CMA and the non-strong version can be easily adapted accordingly.

Definition 15 (Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA Security for SIG). *Let $\kappa = \kappa(\lambda) \in \mathbb{N}$. A signature scheme $\mathsf{SIG} = (\mathsf{Setup}_{\mathsf{SIG}}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy}_{\mathsf{SIG}})$ is strongly $\mathsf{MU}^{\mathsf{c\&l}}$-CMA secure under κ bits leakage per user, if for any PPT adversary $\mathcal{A}$ and any polynomial n, it holds that $\mathsf{Adv}^{\mathsf{s\text{-}cma\text{-}c\&l}}_{\mathsf{SIG},\mathcal{A},n,\kappa}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{s\text{-}cma\text{-}c\&l}}_{\mathsf{SIG},\mathcal{A},n,\kappa} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$, where the experiment $\mathsf{Exp}^{\mathsf{s\text{-}cma\text{-}c\&l}}_{\mathsf{SIG},\mathcal{A},n,\kappa}$ is defined in Fig. 7.*

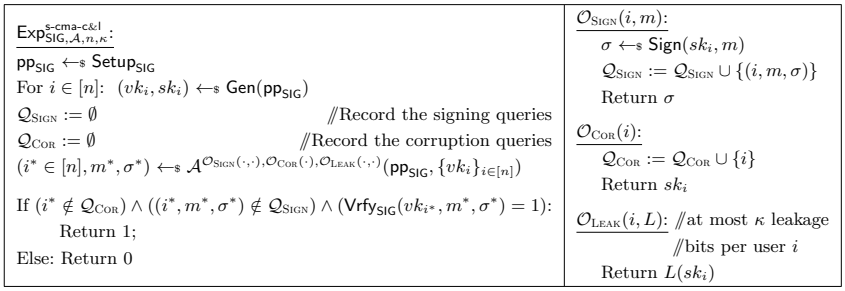

Fig. 7. The strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA security experiment $\mathsf{Exp}^{\mathsf{s\text{-}cma\text{-}c\&l}}_{\mathsf{SIG},\mathcal{A},n,\kappa}$ for SIG.

5.2 Generic Construction of SIG from PV-QA-HPS and QA-NIZK

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

- Two language distributions $\mathscr{L}$ and $\mathscr{L}_0$, both of which have hard SMPs.
- A publicly-verifiable $\mathsf{PVQAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \nu, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Vrfy}_{\mathsf{HPS}})$ for both $\mathscr{L}$ and $\mathscr{L}_0$, with hashing key space $\mathcal{SK}$ and verification key space $\mathcal{VK}$.
- A tag-based $\mathsf{QANIZK} = (\mathsf{Setup}_{\mathsf{NIZK}}, \mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{Sim})$ for $\mathscr{L}$, whose tag space is $\mathcal{T}$.
- A family of collision-resistant hash functions $\mathcal{H} = \{H : \mathcal{VK} \times \mathcal{M} \longrightarrow \mathcal{T}\}$.

Our generic construction of $\mathsf{SIG} = (\mathsf{Setup}_{\mathsf{SIG}}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy}_{\mathsf{SIG}})$ is shown in Fig. 8. In Appendix E.1, we also discuss some alternative ways of setup.

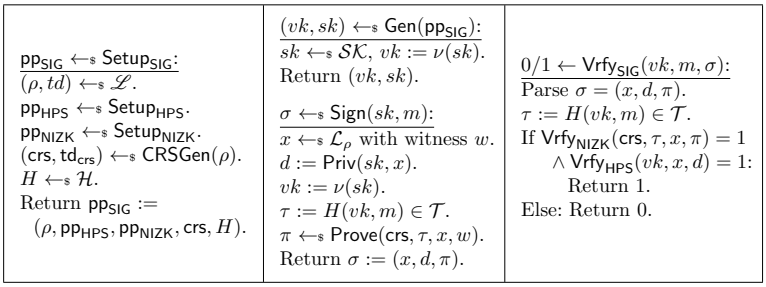

Fig. 8. Generic construction of $\mathsf{SIG} = (\mathsf{Setup}_{\mathsf{SIG}}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy}_{\mathsf{SIG}})$ from PVQAHPS, tag-based QANIZK and $\mathcal{H}$. The message space is $\mathcal{M}$.

Correctness of SIG follows directly from the verification completeness of PVQAHPS and the perfect completeness of QANIZK.

Next, we show its strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA security. We stress that the projection key $pk_\rho = \alpha_\rho(sk)$ is not published as part of SIG's verification key, and this is crucial to the security of SIG since otherwise one can publicly generate valid signatures for any message via the Pub algorithm of PVQAHPS by using pk_ρ.

Theorem 1 (Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA Security of SIG). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii) PVQAHPS is a publicly-verifiable QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having verification soundness, VK-diversity, and supporting κ-LR-$\langle\mathscr{L}_0,\mathscr{L}\rangle$-OT-extracting, (iii) QANIZK is a tag-based QA-NIZK for $\mathscr{L}$, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed SIG scheme in Fig. 8 is strongly $\mathsf{MU}^{\mathsf{c\&l}}$-CMA secure under κ bits leakage per user.*

Concretely, for any number n of users and any adversary $\mathcal{A}$ who makes at most Q_s times of $\mathcal{O}_{\mathrm{SIGN}}$ queries, there exist adversaries $\mathcal{B}_1,\cdots,\mathcal{B}_6$, such that $\mathbf{T}(\mathcal{B}_1)\approx\cdots\approx\mathbf{T}(\mathcal{B}_5)\approx\mathbf{T}(\mathcal{A})+(n+Q_s)\cdot\mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{s\text{-}cma\text{-}c\&l}}_{\mathsf{SIG},\mathcal{A},n,\kappa}(\lambda) \le\ & \mathsf{Adv}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{B}_1,n}(\lambda)+\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)+\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda)+\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda)\\ &+\mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)+\tfrac{n(n-1)}{2}\cdot\boldsymbol{\epsilon}^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}}(\lambda)+n\cdot\boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}(\lambda).\end{aligned}$$

We stress that only factors of computational reductions count when evaluating tight security reductions, while statistical losses are not taken into account [4, 18, 19, 20, 29, 14, 21]. See Remark 4 in Appendix C for more discussions.

We refer to Subsect. 2.1 and Fig. 1 therein for an overview of the proof of Theorem 1. The formal proof is postponed to Appendix C. Here we provide the game sequence G_0–G_6 used in the formal proof in Table 3.

6 PKE with Tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA Security

In this section, we present public-key encryption (PKE) schemes with tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA security, by using QA-HPS with new properties formalized in Sect. 4 as a central building block.

In Subsect. 6.1, we define the $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA security of PKE. Then in Subsect. 6.2, we present our generic construction of PKE.

6.1 Definition of $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA Security

In [29], 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 [29] and is denoted by $\mathsf{MUMC}^{\mathsf{c}}$-CCA in this paper). Here we extend it to $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA, which also takes key leakages into account. Below we present the formal definition.

Table 3. Brief Description of Games G_0-G_6 for the strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMA}$ 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. Columns "$\mathcal{O}_{\text{COR}}$" and "$\mathcal{O}_{\text{LEAK}}$" show the output returned by $\mathcal{O}_{\text{COR}}$ and $\mathcal{O}_{\text{LEAK}}$ respectively. 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 $i^* \notin \mathcal{Q}_{\text{COR}} \wedge (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{SIGN}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \wedge \mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, x^*, d^*) = 1$, where $\tau^* := H(vk_{i^*}, m^*)$.

	$\mathcal{O}_{\text{SIGN}}(i,m)$			$\mathcal{O}_{\text{COR}}(i)$	$\mathcal{O}_{\text{LEAK}}(i,L)$	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	$L(sk_i)$		The strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMA}$ experiment
G_1	$\mathcal{L}_\rho$	sk_i	Prove	sk_i	$L(sk_i)$	$d^* = \mathsf{Priv}(sk_{i^*}, x^*)$	By *verification soundness* of $\mathsf{PVQAHPS}$
G_2	$\mathcal{L}_\rho$	sk_i	Prove	sk_i	$L(sk_i)$	$d^* = \mathsf{Priv}(sk_{i^*}, x^*)$	Abort if verification keys collide: by *VK-diversity* of $\mathsf{PVQAHPS}$
G_3	$\mathcal{L}_\rho$	sk_i	Sim	sk_i	$L(sk_i)$	$d^* = \mathsf{Priv}(sk_{i^*}, x^*)$	By *perfect zero-knowledge* of QANIZK
G_4	$\mathcal{L}_\rho$	sk_i	Sim	sk_i	$L(sk_i)$	$d^* = \mathsf{Priv}(sk_{i^*}, x^*)$, $(\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}}$	By *collision-resistance* of $\mathcal{H}$
G_5	$\mathcal{L}_{\rho_0}$	sk_i	Sim	sk_i	$L(sk_i)$	$d^* = \mathsf{Priv}(sk_{i^*}, x^*)$, $(\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}}$	By *multi-fold SMP* of $\mathscr{L}$ and $\mathscr{L}_0$
G_6	$\mathcal{L}_{\rho_0}$	sk_i	Sim	sk_i	$L(sk_i)$	$d^* = \mathsf{Priv}(sk_{i^*}, x^*)$, $(\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}}$, $x^* \in \mathcal{L}_\rho$	By *USS* of QANIZK $\Pr[\mathsf{Win}] = \mathsf{negl}$ in G_6: by κ-*LR*-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-*OT-extracting* of $\mathsf{PVQAHPS}$

Definition 16 ($\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ **Security for PKE).** *Let $\kappa = \kappa(\lambda) \in \mathbb{N}$. A PKE scheme $\mathsf{PKE} = (\mathsf{Setup}_{\mathsf{PKE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ secure under κ bits leakage per user, if for any PPT adversary $\mathcal{A}$ and any polynomial n, it holds that $\mathsf{Adv}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}(\lambda) := \left| \Pr[\mathsf{Exp}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa} \Rightarrow 1] - \frac{1}{2} \right| \leq \mathsf{negl}(\lambda)$, where the experiment $\mathsf{Exp}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}$ is defined in Fig. 9.*

$\underline{\mathsf{Exp}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}}$:
$\mathsf{pp}_{\mathsf{PKE}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{PKE}}$
For $i \in [n]$: $(pk_i, sk_i) \leftarrow\!\!\$\ \mathsf{Gen}(\mathsf{pp}_{\mathsf{PKE}})$
$\mathcal{Q}_{\text{ENC}} := \emptyset$ //Record the encryption queries
$\mathcal{Q}_{\text{COR}} := \emptyset$ //Record the corruption queries
For $i \in [n]$: $\mathsf{chal}_i := \mathbf{false}$
$\beta \leftarrow\!\!\$\ \{0,1\}$ //Single challenge bit
$\beta' \leftarrow\!\!\$\ \mathcal{A}^{\mathcal{O}_{\text{ENC}}(\cdot,\cdot,\cdot), \mathcal{O}_{\text{DEC}}(\cdot,\cdot), \mathcal{O}_{\text{COR}}(\cdot), \mathcal{O}_{\text{LEAK}}(\cdot,\cdot)}(\mathsf{pp}_{\mathsf{PKE}}, \{pk_i\}_{i \in [n]})$
If $\beta' = \beta$: Return 1; Else: Return 0

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

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

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

$\underline{\mathcal{O}_{\text{LEAK}}(i, L)}$: //at most κ leakage //bits per user i
If $\mathsf{chal}_i = \mathbf{true}$: Return $\perp$
Return $L(sk_i)$

Fig. 9. The $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ security experiment $\mathsf{Exp}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}$ for PKE.

Remark 3. In the experiment $\mathsf{Exp}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}$, to avoid trivial attacks, adversary $\mathcal{A}$ is not allowed to submit a same user index i to both $\mathcal{O}_{\text{ENC}}$ and $\mathcal{O}_{\text{COR}}$.

Moreover, for any user $i \in [n]$, $\mathcal{A}$ is not allowed to submit i to $\mathcal{O}_{\text{LEAK}}$ anymore after receiving a challenge ciphertext c^* w.r.t. user i (guaranteed by the variable chal_i in $\mathsf{Exp}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}$), since otherwise $\mathcal{A}$ could trivially win by setting $L(\cdot)$ to be the first few bits of $\mathsf{Dec}(\cdot, c^*)$ and submitting (i, L) to $\mathcal{O}_{\text{LEAK}}$. We note this is stronger than the multi-user leakage model defined in [22, Appendix A.1], where

$\mathcal{A}$ is not allowed to submit *any* leakage query w.r.t *any* user after receiving the very first challenge ciphertext (no matter w.r.t. which user).

6.2 Generic Construction of PKE from QA-HPS and QA-NIZK

In this subsection, we present a generic construction of $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA secure PKE. The underlying building blocks are as follows.

- Two language distributions $\mathscr{L}$ and $\mathscr{L}_0$, both of which have hard SMPs.
- A $\mathsf{QAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{Pub}, \mathsf{Priv})$ for both $\mathscr{L}$ and $\mathscr{L}_0$, whose hashing key space is $\mathcal{SK}$, projection key space is $\mathcal{PK}$ and hash value space is $\mathcal{HV}$. We require $\mathcal{HV}$ to be an (additive) group. We stress that QAHPS is not required to be publicly-verifiable.
- A tag-based $\mathsf{QANIZK} = (\mathsf{Setup}_{\mathsf{NIZK}}, \mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{Sim})$ for $\mathscr{L}$, whose tag space is $\mathcal{T}$.
- A family of collision-resistant hash functions $\mathcal{H} = \{H : \mathcal{PK} \times \mathcal{HV} \longrightarrow \mathcal{T}\}$.

Our generic construction of $\mathsf{PKE} = (\mathsf{Setup}_{\mathsf{PKE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is shown in Fig. 10.

$\underline{\mathsf{pp}_{\mathsf{PKE}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{PKE}}}$: $(\rho, td) \leftarrow_{\$} \mathscr{L}$. $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$. $\mathsf{pp}_{\mathsf{NIZK}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{NIZK}}$. $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{CRSGen}(\rho)$. $H \leftarrow_{\$} \mathcal{H}$. Return $\mathsf{pp}_{\mathsf{PKE}} :=$ $(\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$.	$\underline{(pk, sk) \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp}_{\mathsf{PKE}})}$: $sk \leftarrow_{\$} \mathcal{SK}$, $pk := \alpha_\rho(sk)$. Return (pk, sk). $\underline{c \leftarrow_{\$} \mathsf{Enc}(pk, m \in \mathcal{HV})}$: $x \leftarrow_{\$} \mathcal{L}_\rho$ with witness w. $d := \mathsf{Pub}(pk, x, w) + m \in \mathcal{HV}$. $\tau := H(pk, d) \in \mathcal{T}$. $\pi \leftarrow_{\$} \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$. Return $c := (x, d, \pi)$.	$\underline{m/\perp \leftarrow \mathsf{Dec}(sk, c)}$: Parse $c = (x, d, \pi)$. $pk := \alpha_\rho(sk)$. $\tau := H(pk, d) \in \mathcal{T}$. If $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$: $m := d - \mathsf{Priv}(sk, x) \in \mathcal{HV}$. Return m. Else: Return $\perp$.

Fig. 10. Generic construction of $\mathsf{PKE} = (\mathsf{Setup}_{\mathsf{PKE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ from QAHPS, tag-based QANIZK and $\mathcal{H}$. The message space is $\mathcal{M} := \mathcal{HV}$.

Correctness of PKE follows directly from the correctness of QAHPS and the perfect completeness of QANIZK. Next, we show its $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA security.

Theorem 2 ($\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA Security of PKE). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii) QAHPS is a QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having PK-diversity, and supporting both κ-LR-$\langle\mathscr{L}, \mathscr{L}_0\rangle$-key-switching and $\mathscr{L}_0$-multi-key-multi-extracting, (iii) QANIZK is a tag-based QA-NIZK for $\mathscr{L}$, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed PKE scheme in Fig. 10 is $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA secure under κ bits leakage per user.*

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

$$\mathsf{Adv}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}(\lambda) \le \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda)$$
$$+ \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{QAHPS},\mathcal{B}_6,n,Q_e}(\lambda) + \frac{n(n-1)}{2} \cdot \epsilon^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) + 2n \cdot \epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda).$$

We refer to Subsect. 2.2 and Fig. 2 therein for an overview of the proof of Theorem 2. The formal proof is postponed to Appendix D. Here we provide the game sequence G_0–G_8 used in the formal proof in Table 4.

Table 4. Brief Description of Games G_0-G_8 for the $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ security proof of PKE. Here column "$\mathcal{O}_{\mathrm{ENC}}$" 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 "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 "$\mathcal{O}_{\mathrm{DEC}}$'s additional check" describes the additional check made by $\mathcal{O}_{\mathrm{DEC}}$ upon a decryption query $(i, c = (x, d, \pi))$, besides the routine check $(i, c) \notin \mathcal{Q}_{\mathrm{ENC}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$, where $\tau := H(pk_i, d)$; $\mathcal{O}_{\mathrm{DEC}}$ outputs $\perp$ if the check fails. Columns "$\mathcal{O}_{\mathrm{COR}}$" and "$\mathcal{O}_{\mathrm{LEAK}}$" show the output returned by $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{LEAK}}$ respectively. Recall that it is not allowed to query $\mathcal{O}_{\mathrm{ENC}}$ and $\mathcal{O}_{\mathrm{COR}}$ for a same user index i.

	$\mathcal{O}_{\mathrm{ENC}}(i^*, m_0, m_1)$			$\mathcal{O}_{\mathrm{DEC}}(i, c)$'s additional check	$\mathcal{O}_{\mathrm{COR}}(i)$	$\mathcal{O}_{\mathrm{LEAK}}(i, L)$	Remark/Assumption
	x^* from	d^* using	π^* via				
G_0	$\mathcal{L}_\rho$	pk_{i^*}	Prove		sk_i	$L(sk_i)$	The $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ security experiment
G_1	$\mathcal{L}_\rho$	pk_{i^*}	Prove		sk_i	$L(sk_i)$	Abort if public keys collide: by *PK-diversity* of QAHPS
G_2	$\mathcal{L}_\rho$	sk_{i^*}	Sim		sk_i	$L(sk_i)$	By *correctness* of QAHPS & by *perfect zero-knowledge* of QANIZK
G_3	$\mathcal{L}_\rho$	sk_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\mathrm{SIM}}$	sk_i	$L(sk_i)$	By *collision-resistance* of $\mathcal{H}$
G_4	$\mathcal{L}_{\rho_0^{(i^*)}}$	sk_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\mathrm{SIM}}$	sk_i	$L(sk_i)$	By *multi-fold SMP* of $\mathscr{L}$ & by *multi-language multi-fold SMP* of $\mathscr{L}_0$
G_5	$\mathcal{L}_{\rho_0^{(i^*)}}$	sk_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\mathrm{SIM}}$, $x \in \mathcal{L}_\rho$	sk_i	$L(sk_i)$	By *USS* of QANIZK
$\{\mathsf{G}_{6.\eta}\}_{\eta \in [n]}$	$\mathcal{L}_{\rho_0^{(i^*)}}$	$\begin{cases} sk'_{i^*}, & \text{if } i^* \le \eta \\ sk_{i^*}, & \text{if } i^* > \eta \end{cases}$	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\mathrm{SIM}}, x \in \mathcal{L}_\rho$	sk_i	$L(sk_i)$	By κ-*LR*-$\langle \mathscr{L}, \mathscr{L}_0 \rangle$-*key-switching* of QAHPS
$\mathsf{G}_{6.n}$	$\mathcal{L}_{\rho_0^{(i^*)}}$	sk'_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\mathrm{SIM}}, x \in \mathcal{L}_\rho$	sk_i	$L(sk_i)$	–
G_7	$\mathcal{L}_{\rho_0}$	sk'_{i^*}	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\mathrm{SIM}}, x \in \mathcal{L}_\rho$	sk_i	$L(sk_i)$	By *multi-language multi-fold SMP* of $\mathscr{L}_0$ & by *multi-fold SMP* of $\mathscr{L}_0$
G_8	$\mathcal{L}_{\rho_0}$	$= \mathsf{rand}$	Sim	$(\tau, x, \pi) \notin \mathcal{Q}_{\mathrm{SIM}}, x \in \mathcal{L}_\rho$	sk_i	$L(sk_i)$	By $\mathscr{L}_0$-*multi-key-multi-extracting* of QAHPS
							$\Pr[\mathsf{Win}] = \frac{1}{2}$ in G_8

In Appendix E.2, we also discuss some potential variants of our PKE.

7 More Primitives and Instantiations from MDDH

Tightly $\mathsf{MU}^{\mathsf{c\&l}}$ secure SC, MAC and AE. Our SIG and PKE immediately lead to direct constructions of tightly $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{Priv\&Auth}$ secure SC [3, 9]. By fully exploiting the similar and composable components of our SIG and PKE, we can obtain a more efficient SC construction, which is shown in Appendix F. Since SIG naturally implies MAC and SC implies AE, we can also obtain the constructions of tightly secure MAC and AE. We also give optimized MAC and AE constructions in Appendix G and Appendix H, where $\mathsf{PVQAHPS}$ is replaced with QAHPS. Our MAC achieves tight strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMVA}$ security, which also considers chosen verification attacks [15] in addition to strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMA}$.

Instantiations from MDDH. We give instantiations of SIG and PKE from the matrix DDH (MDDH) assumptions over asymmetric pairing groups. Our SC, MAC and AE can be similarly instantiated.

Firstly, we instantiate the building blocks needed in our generic constructions (cf. Appendix I). More precisely, we give concrete instantiations of Publicly-Verifiable QA-HPS in Appendix I.3 (with an overview in Subsect. 2.4) and QA-HPS in Appendix I.4, built upon the MDDH-based QA-HPS schemes proposed in [22], which are in turn generalizations of the well-known DDH-based HPS scheme proposed by Cramer and Shoup in [13]. Then we instantiate tag-based QA-NIZK with a tag-base variant of the QA-NIZK scheme proposed in [1] that has tight USS based on MDDH, which is recalled in Appendix I.5 for completeness.

Next we instantiate the generic SIG construction in Sect. 5 with the above building blocks. Let $x \cdot \mathbb{G}$ denote x elements in $\mathbb{G}$. Under MDDH parameters $\ell, k \in \mathbb{N}$ where $\ell \geq 2k+1$, the MDDH-based SIG scheme $\mathsf{SIG}_{\mathsf{MDDH}}$ has public parameter $\mathsf{pp}_{\mathsf{SIG}} : (5k^2+3k+\ell k) \cdot \mathbb{G}_1 + (5k^2+4k+1+2\ell k) \cdot \mathbb{G}_2$, verification key $vk : (\ell k) \cdot \mathbb{G}_2$, signing key $sk : \ell(k+1) \cdot \mathbb{Z}_p$, and signature $\sigma : (4k^2 + 4k + 2 + \ell) \cdot \mathbb{G}_1 + (2k^2 + 3k + 1) \cdot \mathbb{G}_2$. By plugging the theorems regarding the tight security of the MDDH-based PV-QA-HPS and QA-NIZK schemes in Appendix I into Theorem 1, we have the following corollary showing the tight strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMA}$ security of $\mathsf{SIG}_{\mathsf{MDDH}}$ based on the MDDH assumptions (as well as the collision-resistance of hash functions).

Corollary 1 (Tight Strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMA}$ Security of $\mathsf{SIG}_{\mathsf{MDDH}}$). *Let $\ell \geq 2k+1$ and $\kappa \leq \log p - \Omega(\lambda)$. For any number n of users and any adversary $\mathcal{A}$ who makes at most Q_s times of $\mathcal{O}_{\mathrm{SIGN}}$ queries, there exist adversaries $\mathcal{B}_1, \mathcal{B}_2$ and $\mathcal{B}_3$, such that $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{B}_3) \approx \mathbf{T}(\mathcal{A}) + (n+Q_s) \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{s\text{-}cma\text{-}c\&l}}_{\mathsf{SIG}_{\mathsf{MDDH}},\mathcal{A},n,\kappa}(\lambda) \leq\ & 2 \cdot \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + (4k\lceil \log Q_s \rceil + \ell - k + 6) \cdot \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_1,\mathcal{B}_2}(\lambda) \\ & + (2\lceil \log Q_s \rceil + 3) \cdot \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_3}(\lambda) + \tfrac{n+2\lceil \log Q_s \rceil Q_s}{p-1} + \tfrac{n(n-1)}{2} \cdot \tfrac{1}{p^{k\ell}}.\end{aligned}$$

Since $Q_s = \mathsf{poly}(\lambda)$ for PPT adversaries, the security loss is in fact $O(\log Q_s) = O(\log \lambda)$, which is lower than $O(\lambda)$. For $k = 1$ and $\ell = 3$, we get a fully compact SIG scheme with $\mathsf{pp}_{\mathsf{SIG}} : 11 \cdot \mathbb{G}_1 + 16 \cdot \mathbb{G}_2$, $vk : 3 \cdot \mathbb{G}_2$, $sk : 6 \cdot \mathbb{Z}_p$ and $\sigma : 13 \cdot \mathbb{G}_1 + 6 \cdot \mathbb{G}_2$. The resulting SIG scheme has tight strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMA}$ security based on the SXDH assumption (which requires the DDH assumption to hold both in $\mathbb{G}_1$ and $\mathbb{G}_2$), and supports $\kappa = \log p - \Omega(\lambda)$ bits leakage per user. The leakage rate (i.e., κ/ bit-length of sk) is $\frac{\log p - \Omega(\lambda)}{6 \log p} = \frac{1}{6} - o(1)$ asymptotically as p grows.

We also instantiate the generic PKE construction in Sect. 6. Under MDDH parameters $\ell, k \in \mathbb{N}$ where $\ell \geq 2k+1$, the MDDH-based PKE scheme $\mathsf{PKE}_{\mathsf{MDDH}}$ has public parameter $\mathsf{pp}_{\mathsf{PKE}} : (5k^2+3k+\ell k) \cdot \mathbb{G}_1 + (4k^2+3k+1+2\ell k) \cdot \mathbb{G}_2$, public key $pk : k \cdot \mathbb{G}_1$, secret key $sk : \ell \cdot \mathbb{Z}_p$, and ciphertext $c : (4k^2 + 3k + 2 + \ell) \cdot \mathbb{G}_1 + (2k^2 + 3k + 1) \cdot \mathbb{G}_2$. By plugging the theorems regarding the tight security of the MDDH-based QA-HPS and QA-NIZK schemes in Appendix I into Theorem 2, we have the following corollary showing the tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ security of $\mathsf{PKE}_{\mathsf{MDDH}}$ based on the MDDH assumptions (as well as the collision-resistance of hash functions).

Corollary 2 (Tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ Security of $\mathsf{PKE}_{\mathsf{MDDH}}$). *Let $\ell \geq 2k+1$ and $\kappa \leq \log p - \Omega(\lambda)$. 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 $\mathcal{B}_1, \mathcal{B}_2$ and $\mathcal{B}_3$, such that $\mathbf{T}(\mathcal{B}_1) \approx \mathbf{T}(\mathcal{B}_2) \approx \mathbf{T}(\mathcal{B}_3) \approx \mathbf{T}(\mathcal{A}) + (n + Q_e + Q_d) \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{cca\text{-}c\&l}}_{\mathsf{PKE}_{\mathsf{MDDH}},\mathcal{A},n,\kappa}(\lambda) \leq\ & 2 \cdot \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + (4k\lceil \log Q_e \rceil + \ell - k + 9) \cdot \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_1,\mathcal{B}_2}(\lambda) \\ & + (2\lceil \log Q_e \rceil + 2) \cdot \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_3}(\lambda) + \tfrac{2n+2\lceil \log Q_e \rceil Q_e}{p-1} + \tfrac{n(n-1)}{2} \cdot \tfrac{1}{p^k}.\end{aligned}$$

For $k = 1$ and $\ell = 3$, we get a fully compact PKE scheme with $\mathsf{pp}_{\mathsf{PKE}} : 11 \cdot \mathbb{G}_1 + 14 \cdot \mathbb{G}_2$, $pk : 1 \cdot \mathbb{G}_1$, $sk : 3 \cdot \mathbb{Z}_p$ and $c : 12 \cdot \mathbb{G}_1 + 6 \cdot \mathbb{G}_2$. The resulting PKE scheme has tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ security based on the SXDH assumption, and supports $\kappa = \log p - \Omega(\lambda)$ bits leakage per user. The leakage rate is $\frac{\log p - \Omega(\lambda)}{3 \log p} = \frac{1}{3} - o(1)$ asymptotically as p grows.

For an overview and comparison with other schemes, we refer to Table 1 and Table 2 in the introduction, and Table 5 and Table 6 in Appendix A.

On tightness of our MDDH-based schemes. Our MDDH-based schemes are the first ones achieving almost tight $\mathsf{MU}^{\mathsf{c}}/\mathsf{MU}^{\mathsf{c\&l}}$ security in the standard model, and the security loss factor is $O(\log \lambda)$. Note that the security loss factor $O(\log \lambda)$ depends only logarithmically on the security parameter λ, which is close to full tightness $O(1)$ compared to linear security loss factor $O(\lambda)$. As an example, in the security level of 256 bit, we have $\log \lambda = 8$.

We stress that all our generic constructions are *fully tightness-preserving*, i.e., the $\mathsf{MU}^{\mathsf{c}}/\mathsf{MU}^{\mathsf{c\&l}}$ securities of the resulting SIG, PKE, SC, MAC, AE schemes are tightly reduced to the security properties of the building blocks PV-QA-HPS, QA-HPS and tag-based QA-NIZK, with constant security loss factors. Moreover, our instantiations of PV-QA-HPS and QA-HPS have fully tight securities, and only the tag-based QA-NIZK instantiation has security loss factor $O(\log \lambda)$. Therefore, our fully tightness-preserving generic constructions leave spaces for even tighter (fully tight) $\mathsf{MU}^{\mathsf{c}}/\mathsf{MU}^{\mathsf{c\&l}}$ security, as long as we can find instantiations of tag-based QA-NIZK with tighter security.

On efficiency of our MDDH-based schemes. Note that all our schemes enjoy *full compactness* (i.e., all the parameters, keys, signatures and ciphertexts consist of only a constant number of group elements). Though not as efficient as those constructed in the RO model [20, 14, 29], our schemes are the first ones achieving almost tight MU^{c} security in the standard model, getting rid of the ideal object RO that may not result in secure implementations in practice [10]. We believe our fully compact schemes are good starts for almost tight MU^{c} security in the standard model and follow-up work might improve efficiency even further.

Moreover, all our schemes additionally support bounded key leakages and achieve almost tight $\mathsf{MU}^{\mathsf{c\&l}}$ security, which is not known to hold for the aforementioned RO schemes. This suggests that our schemes are the first ones achieving almost tight $\mathsf{MU}^{\mathsf{c\&l}}$ security, no matter in the standard model or RO model.

Acknowledgments. We would like to thank the reviewers for their helpful comments. Shuai Han and Shengli Liu were partially supported by National

Natural Science Foundation of China (Grant Nos. 62002223, 61925207), Guangdong Major Project of Basic and Applied Basic Research (2019B030302008), the National Key R&D Program of China under Grant 2022YFB2701500, Shanghai Sailing Program (20YF1421100), Young Elite Scientists Sponsorship Program by China Association for Science and Technology (YESS20200185), and Ant Group through CCF-Ant Research Fund (CCF-AFSG RF20220224). Dawu Gu was partially supported by the National Key R&D Program of China under Grant 2020YFA0712302.

References

[1] Abe, M., Jutla, C.S., Ohkubo, M., Pan, J., Roy, A., Wang, Y.: Shorter QA-NIZK and SPS with tighter security. In: ASIACRYPT 2019, pp. 669–699
[2] Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: TCC 2009, pp. 474–495
[3] An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: EUROCRYPT 2002, pp. 83–107
[4] Bader, C., Hofheinz, D., Jager, T., Kiltz, E., Li, Y.: Tightly-secure authenticated key exchange. In: TCC 2015, pp. 629–658
[5] Bader, C., Jager, T., Li, Y., Schäge, S.: On the impossibility of tight cryptographic reductions. In: EUROCRYPT 2016, pp. 273–304
[6] Bellare, M., Boldyreva, A., Micali, S.: Public-key encryption in a multi-user setting: Security proofs and improvements. In: EUROCRYPT 2000, pp. 259–274
[7] Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: ASIACRYPT 2000, pp. 531–545
[8] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: CRYPTO 1993, pp. 232–249
[9] Bellare, M., Stepanovs, I.: Security under message-derived keys: Signcryption in iMessage. In: EUROCRYPT 2020, pp. 507–537
[10] Canetti, R., Goldreich, O., Halevi, S.: The random oracle methodology, revisited. In: STOC 1998, pp. 209–218
[11] Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: CRYPTO 2003, pp. 565–582
[12] Chen, J., Wee, H.: Fully, (almost) tightly secure IBE and dual system groups. In: CRYPTO 2013, pp. 435–460
[13] Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: EUROCRYPT 2002, pp. 45–64
[14] Diemert, D., Gellert, K., Jager, T., Lyu, L.: More efficient digital signatures with tight multi-user security. In: PKC 2021, pp. 1–31
[15] Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: EUROCRYPT 2012, pp. 355–374
[16] 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)
[17] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: CRYPTO 2013, pp. 129–147
[18] Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly CCA-secure encryption without pairings. In: EUROCRYPT 2016, pp. 1–27

[19] Gay, R., Hofheinz, D., Kohl, L.: Kurosawa-Desmedt meets tight security. In: CRYPTO 2017, pp. 133–160
[20] Gjøsteen, K., Jager, T.: Practical and tightly-secure digital signatures and authenticated key exchange. In: CRYPTO 2018, pp. 95–125
[21] 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
[22] 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, https://eprint.iacr.org/2019/512
[23] Hofheinz, D., Jager, T.: Tightly secure signatures and public-key encryption. In: CRYPTO 2012, pp. 590–607
[24] Hofheinz, D., Jia, D., Pan, J.: Identity-based encryption tightly secure under chosen-ciphertext attacks. In: ASIACRYPT 2018, pp. 190–220
[25] Jager, T., Stam, M., Stanley-Oakes, R., Warinschi, B.: Multi-key authenticated encryption with corruptions: Reductions are lossy. In: TCC 2017, pp. 409–441
[26] Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: ASIACRYPT 2013, pp. 1–20
[27] Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: EUROCRYPT 2015, pp. 101–128
[28] Langrehr, R., Pan, J.: Tightly secure hierarchical identity-based encryption. In: PKC 2019, pp. 436–465
[29] 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)
[30] Liu, X., Liu, S., Gu, D.: Tightly secure chameleon hash functions in the multi-user setting and their applications. In: ACISP 2020, pp. 664–673, https://eprint.iacr.org/2022/1258
[31] Liu, X., Liu, S., Gu, D., Weng, J.: Two-pass authenticated key exchange with explicit authentication and tight security. In: ASIACRYPT 2020, pp. 785–814
[32] Morgan, A., Pass, R., Shi, E.: On the adaptive security of MACs and PRFs. In: ASIACRYPT 2020, pp. 724–753
[33] Morillo, P., Ràfols, C., Villar, J.L.: The kernel matrix Diffie-Hellman assumption. In: ASIACRYPT 2016, pp. 729–758
[34] Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: CRYPTO 2009, pp. 18–35
[35] Pan, J., Wagner, B.: Lattice-based signatures with tight adaptive corruptions and more. In: PKC 2022. pp. 347–378
[36] Rogaway, P.: Authenticated-encryption with associated-data. In: ACM CCS 2002, pp. 98–107
[37] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In: FOCS 1999, pp. 543–553
[38] Steinfeld, R., Pieprzyk, J., Wang, H.: How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. In: CT-RSA 2007, pp. 357–371

Appendix

A Full Comparison Tables on the Full Compactness

Table 5. Comparison of signature (SIG) schemes that have (almost) tight MU-CMA security under adaptive corruptions (MU^{c}-CMA). The column **Standard Model** shows whether the security is proved in the standard model. The column **Strong Security** shows whether the scheme is proved *strongly* existentially unforgeable. The column **Corruption?** asks whether the security is proved in the presence of adaptive corruptions. The column **Leakage?** asks whether the security is proved additionally in the presence of key leakages, and if so, a *leakage rate* (defined as the ratio of leakage amount to secret key size) is presented. The column **Full Compactness** shows whether the scheme is fully compact (i.e., all the public parameters pp, verification key vk, signing key sk and signature σ consist of only a constant number of group elements or lattice vectors), and if not, the non-compact part is presented. The columns $|\mathsf{pp}|$, $|\boldsymbol{vk}|$, $|\boldsymbol{sk}|$ and $|\boldsymbol{\sigma}|$ count the size of public parameters pp, verification key vk, signing key sk and signature σ, respectively, in terms of numbers of group elements, lattice vectors, exponents and bit-strings, where λ denotes the security parameter. The column **Security Loss** shows the security loss factor of the reductions. The column **Assumption** shows the computational assumption on which the security is based.

SIG Scheme	Standard Model	Strong Security	Corruption?	Leakage?	Full Compactness	$\lvert\mathsf{pp}\rvert$	$\lvert\boldsymbol{vk}\rvert$	$\lvert\boldsymbol{sk}\rvert$	$\lvert\boldsymbol{\sigma}\rvert$	Security Loss	Assumption
BHJKL [4, 23]	✓	–	✓	–	× (non-compact σ)	$O(1)$	$O(1)$	$O(1)$	$O(\lambda)$	$O(1)$	MDDH
GJ [20]	×	–	✓	–	✓	1	2	1	7	$O(1)$	DDH
DGJL [14]	×	✓	✓	–	✓	1	4	1	3	$O(1)$	DDH or ϕ-Hiding
HJKLPRS [21]	✓	×	✓	–	× (non-compact pp)	$10\lambda + 5$	1	2	5	$O(\lambda)$	MDDH
PW [35]	×	–	✓	–	× (non-compact vk)	$O(1)$	$O(\lambda)$	$O(\lambda)$	6	$O(1)$	LWE
Our $\mathsf{SIG}_{\mathsf{MDDH}}$	✓	✓	✓	✓ $(\frac{1}{6} - o(1))$	✓	27	3	6	19	$O(\log \lambda)$	MDDH

Table 6. Comparison of public-key encryption (PKE) schemes that have (almost) tight MUMC-CCA security under adaptive corruptions ($\mathsf{MUMC}^{\mathsf{c}}$-CCA) *or* key leakages. The columns have similar meanings as those in Table 5, with pp the public parameters, pk the public key, sk the secret key and c the ciphertext of PKE. .

PKE Scheme	Standard Model	Corruption?	Leakage?	Full Compact.	$\lvert\mathsf{pp}\rvert$	$\lvert pk\rvert$	$\lvert sk\rvert$	$\lvert c\rvert$	Security Loss	Assumption
HLLG [22]	✓	–	✓ $(\frac{1}{18} - o(1))$	✓	6	4	18	8	$O(\log \lambda)$	MDDH
LLP [29]	×	✓	–	✓	1	4	2	4	$O(1)$	CDH
Our $\mathsf{PKE}_{\mathsf{MDDH}}$	✓	✓	✓ $(\frac{1}{3} - o(1))$	✓	25	1	3	18	$O(\log \lambda)$	MDDH

B Additional Preliminaries

Definition 17 (SIG). *A signature (SIG) scheme* $\mathsf{SIG} = (\mathsf{Setup}_{\mathsf{SIG}}, \mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy}_{\mathsf{SIG}})$ *with message space* $\mathcal{M}$ *consists of four PPT algorithms:*

- $\mathsf{pp}_{\mathsf{SIG}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{SIG}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp}_{\mathsf{SIG}}$*, which serves as an implicit input of other algorithms.*
- $(vk, sk) \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp}_{\mathsf{SIG}})$*: Taking* $\mathsf{pp}_{\mathsf{SIG}}$ *as input, the key generation algorithm outputs a pair of verification key and signing key* (vk, sk)*.*
- $\sigma \leftarrow_{\$} \mathsf{Sign}(sk, m)$*: Taking as input a signing key* sk *and a message* $m \in \mathcal{M}$*, the signing algorithm outputs a signature* σ*.*
- $0/1 \leftarrow \mathsf{Vrfy}_{\mathsf{SIG}}(vk, m, \sigma)$*: 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 $\mathsf{pp}_{\mathsf{SIG}} \in \mathsf{Setup}_{\mathsf{SIG}}$*,* $(vk, sk) \in \mathsf{Gen}(\mathsf{pp}_{\mathsf{SIG}})$*,* $m \in \mathcal{M}$*,* $\sigma \in \mathsf{Sign}(sk, m)$*, it holds that* $\mathsf{Vrfy}_{\mathsf{SIG}}(vk, m, \sigma) = 1$*.*

Definition 18 (PKE). *A public-key encryption (PKE) scheme* $\mathsf{PKE} = (\mathsf{Setup}_{\mathsf{PKE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}$ *consists of four PPT algorithms:*

- $\mathsf{pp}_{\mathsf{PKE}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{PKE}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp}_{\mathsf{PKE}}$*, which serves as an implicit input of other algorithms.*
- $(pk, sk) \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp}_{\mathsf{PKE}})$*: Taking* $\mathsf{pp}_{\mathsf{PKE}}$ *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 $\mathsf{pp}_{\mathsf{PKE}} \in \mathsf{Setup}_{\mathsf{PKE}}$*,* $(pk, sk) \in \mathsf{Gen}(\mathsf{pp}_{\mathsf{PKE}})$*,* $m \in \mathcal{M}$*,* $c \in \mathsf{Enc}(pk, m)$*, it holds that* $\mathsf{Dec}(sk, c) = m$*.*

Definition 19 (Collision-resistant hash functions). *A family of hash functions* $\mathcal{H}$ *is collision-resistant, if for any PPT adversary* $\mathcal{A}$*, it holds that*

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

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

Theorem 1 (Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA Security of SIG) *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii)* PVQAHPS *is a publicly-verifiable QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having verification soundness, VK-diversity, and supporting κ-LR-$\langle\mathscr{L}_0,\mathscr{L}\rangle$-OT-extracting, (iii)* QANIZK *is a tag-based QA-NIZK for $\mathscr{L}$, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed* SIG *scheme in Fig. 8 is strongly* $\mathsf{MU}^{\mathsf{c\&l}}$-CMA *secure under κ bits leakage per user.*

Concretely, for any number n of users and any adversary $\mathcal{A}$ who makes at most Q_s times of $\mathcal{O}_{\mathrm{Sign}}$ queries, there exist adversaries $\mathcal{B}_1,\cdots,\mathcal{B}_6$, such that $\mathbf{T}(\mathcal{B}_1)\approx\cdots\approx\mathbf{T}(\mathcal{B}_5)\approx\mathbf{T}(\mathcal{A})+(n+Q_s)\cdot\mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{s\text{-}cma\text{-}c\&l}}_{\mathsf{SIG},\mathcal{A},n,\kappa}(\lambda)\le{}&\mathsf{Adv}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{B}_1,n}(\lambda)+\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)+\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda)+\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda)\\&+\mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)+\tfrac{n(n-1)}{2}\cdot\epsilon^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}}(\lambda)+n\cdot\epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}(\lambda).\end{aligned}$$

Remark 4 (On the Tightness of SIG's Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA security). According to Theorem 1, SIG has tight strong $\mathsf{MU}^{\mathsf{c\&l}}$-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 MDDH assumptions), PVQAHPS has a tight verification soundness and QANIZK has a tight USS. We stress that only factors of computational reductions count when evaluating whether the security is tight or not.

We also note that the statistical loss $\frac{n(n-1)}{2}\cdot\epsilon^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}}(\lambda)$ is quadratic in the number n of users. This statistical loss exactly reflects the collision probability of verification keys among n users. Such a quadratic statistical loss is inherent to the $\mathsf{MU}^{\mathsf{c\&l}}$-CMA security (even to MU^{c}) of all signature schemes, since if two verification keys collide, an adversary can simply corrupt one user and use the signing key to forge a signature for another user.

Proof of Theorem 1. We prove the theorem by defining a sequence of games G_0–G_6 and showing adjacent games indistinguishable. A brief description of differences between adjacent games is summarized in Table 3 in Subsect. 5.2.

Game G_0: This is the $\mathsf{Exp}^{\mathsf{s\text{-}cma\text{-}c\&l}}_{\mathsf{SIG},\mathcal{A},n,\kappa}$ experiment (cf. Fig. 7).

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

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

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

where $\tau^* := H(vk_{i^*}, m^*)$. By definition, $\mathsf{Adv}^{\text{s-cma-c\&l}}_{\mathsf{SIG},\mathcal{A},n,\kappa}(\lambda) = \Pr_0[\mathsf{Win}]$.

Game G_1: It is the same as G_0, except that, the event Win is now defined as

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

Claim 1. $\left|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\text{vrfy-snd}}_{\mathsf{PVQAHPS},\mathcal{B}_1,n}(\lambda)$.

Proof. By $\mathsf{InConsis}$ denote the event that $\mathcal{A}$'s forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ satisfying $\mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, x^*, d^*) = 1 \ \wedge \ d^* \neq \mathsf{Priv}(sk_{i^*}, x^*)$.

Clearly, G_1 is identical to G_0 unless $\mathsf{InConsis}$ occurs, thus $\left|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\right| \leq \Pr_1[\mathsf{InConsis}]$. It is straightforward to construct an adversary $\mathcal{B}_1$ against the verification soundness of $\mathsf{PVQAHPS}$, s.t. $\Pr_1[\mathsf{InConsis}] \leq \mathsf{Adv}^{\text{vrfy-snd}}_{\mathsf{PVQAHPS},\mathcal{B}_1,n}(\lambda)$. (Since $\mathcal{B}_1$ is given the signing keys of all users, it can simulate G_1 honestly for $\mathcal{A}$, output the (i^*, x^*, d^*) contained in $\mathcal{A}$'s forgery and succeed as long as $\mathsf{InConsis}$ occurs.) We also provide a full description of $\mathcal{B}_1$ in Appendix C.1. ∎

Game G_2: It is the same as G_1, except that, after generating n pairs of verification/signing keys $\{(vk_i, sk_i)\}_{i\in[n]}$, the challenger aborts immediately if there are two verification keys that collide, i.e., $\exists 1 \leq i < j \leq n$, s.t. $vk_i = vk_j$.

Since sk_i and sk_j are independently and uniformly chosen from $\mathcal{SK}$, by the VK-diversity of $\mathsf{PVQAHPS}$ and by a union bound, it follows that $\left|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]\right| \leq \sum_{1\leq i<j\leq n} \Pr[\nu(sk_i) = \nu(sk_j)] \leq \frac{n(n-1)}{2} \cdot \epsilon^{\text{vk-div}}_{\mathsf{PVQAHPS}}(\lambda)$.

Game G_3: It is the same as G_2, except that, when answering $\mathcal{O}_{\text{SIGN}}(i, m)$, the challenger computes π via the Sim algorithm of QANIZK by using the simulation trapdoor $\mathsf{td}_{\mathsf{crs}}$:

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

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

Since x is chosen from $\mathcal{L}_\rho$ with witness w, by the perfect zero-knowledge of QANIZK, the π in G_3 is identically distributed as that in G_2. Consequently, the change is just conceptual and $\Pr_2[\mathsf{Win}] = \Pr_3[\mathsf{Win}]$.

Game G_4: It is the same as G_3, except that, when answering $\mathcal{O}_{\text{SIGN}}(i, m)$, the challenger also puts (τ, x, π) to a set $\mathcal{Q}_{\text{SIM}}$, and for the forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ output by $\mathcal{A}$, the event Win is now defined as

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

Claim 2. $\left|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)$.

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

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

where $\tau^* := H(vk_{i^*}, m^*)$ and $\tau := H(vk_i, m)$. Clearly, G_3 and G_4 are the same until Bad occurs, thus $\left|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\right| \leq \Pr_4[\mathsf{Bad}]$.

To bound $\Pr_4[\mathsf{Bad}]$, we divide Bad into the following two cases:

- **Case 1:** $(i^*, m^*) = (i, m)$. Together with $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*)) \neq (i, m, \sigma = (x, d, \pi)) \wedge (\tau^*, x^*, \pi^*) = (\tau, x, \pi)$, it follows that $d^* \neq d$. However, this contradicts $d^* = \mathsf{Priv}(sk_{i^*}, x^*) = \mathsf{Priv}(sk_i, x) = d$. Therefore, this case can never occur.
- **Case 2:** $(i^*, m^*) \neq (i, m)$. Since there are no verification key collisions (due to the game change in G_2), $(i^*, m^*) \neq (i, m)$ implies $(vk_{i^*}, m^*) \neq (vk_i, m)$. Together with $\tau^* = H(vk_{i^*}, m^*) = H(vk_i, m) = \tau$, this case suggests a collision of H. It is straightforward to construct an adversary $\mathcal{B}_2$ so that $\Pr_4[\mathsf{Bad}] \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)$. ($\mathcal{B}_2$ can sample all signing keys itself, simulate G_4 honestly for $\mathcal{A}$, and successfully find a collision as long as Bad happens.)

Overall, $\left|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\right| \leq \Pr_4[\mathsf{Bad}] \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)$. ▮

Game G_5: It is the same as G_4, except that, at the beginning of the game, the challenger picks $(\rho_0, td_0) \leftarrow\!\!\$\ \mathscr{L}_0$ besides $(\rho, td) \leftarrow\!\!\$\ \mathscr{L}$, and for all the $\mathcal{O}_{\text{SIGN}}$ queries, the challenger samples $x \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$ instead of $x \leftarrow\!\!\$\ \mathcal{L}_\rho$.

Claim 3. $\left|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda)$.

Proof. We introduce an intermediate **Game** $\mathsf{G}_{4.5}$ between G_4 and G_5, where the challenger samples $x \leftarrow\!\!\$\ \mathcal{X}$ for all the $\mathcal{O}_{\text{SIGN}}$ queries.

Since witness w for x is not used at all in G_4, $\mathsf{G}_{4.5}$ and G_5 (due to the game change in G_3), 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. $\left|\Pr_4[\mathsf{Win}] - \Pr_{4.5}[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda)$ and $\left|\Pr_{4.5}[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda)$. The full description of $\mathcal{B}_3$ and $\mathcal{B}_4$ can be found in Appendix C.2. ($\mathcal{B}_3$ and $\mathcal{B}_4$ can sample all signing keys themselves, simulate $\mathsf{G}_4/\mathsf{G}_{4.5}/\mathsf{G}_5$ 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_6: It is the same as G_5, except that, the event Win is now defined as

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

Claim 4. $\left|\Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$.

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

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

G_5 and G_6 are the same unless Forge occurs, so $\left|\Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}]\right| \leq \Pr_6[\mathsf{Forge}]$. Note that Forge implies $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \wedge (\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}} \wedge x^* \notin$

$\mathcal{L}_\rho$. Thus by the USS of tag-based QANIZK, we can build an adversary $\mathcal{B}_5$ such that $\Pr_6[\mathsf{Forge}] \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$. $\mathcal{B}_5$ can sample all signing keys itself, simulate G_6 honestly for $\mathcal{A}$ (using its own oracle $\mathcal{O}_{\mathrm{SIM}}$ defined in Fig. 4 to generate simulated proofs π when answering $\mathcal{O}_{\mathrm{SIGN}}$ queries for $\mathcal{A}$), output the (τ^*, x^*, π^*) 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 C.3. ∎

Finally, we have the following claim regarding $\Pr_6[\mathsf{Win}]$.

Claim 5. $\Pr_6[\mathsf{Win}] \leq n \cdot \boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}(\lambda)$.

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_6.

- Firstly, the verification keys contain $vk_{i^*} = \nu(sk_{i^*})$.
- In $\mathcal{O}_{\mathrm{SIGN}}(i^*, m)$, since $x \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$, the behavior of $\mathcal{O}_{\mathrm{SIGN}}$ for user i^* is completely determined by $\alpha_{\rho_0}(sk_{i^*})$.
- Since $i^* \notin \mathcal{Q}_{\mathrm{COR}}$, $\mathcal{A}$ never queries $\mathcal{O}_{\mathrm{COR}}(i^*)$.
- From $\mathcal{O}_{\mathrm{LEAK}}(i^*, L)$, $\mathcal{A}$ obtains at most κ bits information about sk_{i^*}.

Overall, the information about sk_{i^*} that $\mathcal{A}$ learns in G_6 is limited in $\nu(sk_{i^*})$, $\alpha_{\rho_0}(sk_{i^*})$ and at most κ bits leakage information.

Then we analyze the probability $\Pr_6[\mathsf{Win}]$. For $\mathcal{A}$'s forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$, Win will not occur unless $x^* \in \mathcal{L}_\rho \wedge d^* = \mathsf{Priv}(sk_{i^*}, x^*)$. Intuitively, by the κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property of publicly-verifiable PVQAHPS (cf. Def. 10), we know that $x^* \in \mathcal{L}_\rho \ \wedge\ d^* = \mathsf{Priv}(sk_{i^*}, x^*)$ holds with only a negligible probability, even in the presence of $\nu(sk_{i^*})$, $\alpha_{\rho_0}(sk_{i^*})$ and at most κ bits leakage about sk_{i^*}. Hence Win hardly happens in G_6.

Formally, we build an (unbounded) adversary $\mathcal{B}_6$ against the κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property of publicly-verifiable PVQAHPS. $\mathcal{B}_6$ is given $(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \rho, \alpha_{\rho_0}(sk), \nu(sk))$, where $sk \leftarrow\!\!\$\ \mathcal{SK}$ is chosen by its own challenger, and has access to the oracle $\mathcal{O}_{\mathrm{LEAK}}$ defined in Fig. 6, which is denoted by $\mathcal{O}^{(\mathsf{PVQAHPS})}_{\mathrm{LEAK}}$ below to avoid confusing with the $\mathcal{O}_{\mathrm{LEAK}}$ in SIG's experiment. $\mathcal{B}_6$ will simulate G_6 for $\mathcal{A}$. Intuitively, $\mathcal{B}_6$ will first guess the user index i^* for which $\mathcal{A}$ forges a signature (with a security loss n), implicitly set the signing key of user i^* as the sk chosen by its own challenger and explicitly define the verification key of user i^* as the $\nu(sk)$ contained in its input. For the remaining $n-1$ users, $\mathcal{B}_6$ samples signing keys itself, thus can honestly answer $\mathcal{O}_{\mathrm{SIGN}}$, $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{LEAK}}$ queries made by $\mathcal{A}$ for these users. For user i^*, $\mathcal{B}_6$ can answer $\mathcal{O}_{\mathrm{SIGN}}$ queries using the projection key $\alpha_{\rho_0}(sk)$ contained in its own input (since $x \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$), answer $\mathcal{O}_{\mathrm{LEAK}}$ queries via its own $\mathcal{O}^{(\mathsf{PVQAHPS})}_{\mathrm{LEAK}}$ oracle, and abort immediately if $\mathcal{A}$ corrupts i^*. Finally, $\mathcal{B}_6$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$, and returns (x^*, d^*) to its own challenger. Overall, $\mathcal{B}_6$ succeeds (i.e., $x^* \in \mathcal{L}_\rho \ \wedge\ d^* = \mathsf{Priv}(sk, x^*)$) as long as i^* is correctly guessed and Win occurs, thus $\boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}(\lambda) \geq \frac{1}{n} \cdot \Pr_6[\mathsf{Win}]$.

We also provide a full description of $\mathcal{B}_6$ in Appendix C.4. ▌

Taking all things together, Theorem 1 follows. □

C.1 Full Description of Reduction $\mathcal{B}_1$ for Claim 1

To bound $\Pr_1[\mathsf{InConsis}]$, we construct an adversary $\mathcal{B}_1$ against the verification soundness of $\mathsf{PVQAHPS}$ (cf. Def. 9). The full description of $\mathcal{B}_1$ is as follows. $\mathcal{B}_1$ is given $(\mathsf{pp}_{\mathsf{HPS}}, (sk_i, vk_i)_{i\in[n]})$, where $sk_i \leftarrow_{\$} \mathcal{SK}$ $(i \in [n])$ are chosen by its own challenger and $vk_i := \nu(sk_i)$. $\mathcal{B}_1$ simulates G_1 for $\mathcal{A}$ as follows.

- Firstly, $\mathcal{B}_1$ invokes $(\rho, td) \leftarrow_{\$} \mathscr{L}$, $\mathsf{pp}_{\mathsf{NIZK}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{NIZK}}$, $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{CRSGen}(\rho)$, $H \leftarrow_{\$} \mathcal{H}$, sets $\mathsf{pp}_{\mathsf{SIG}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$, and sends $(\mathsf{pp}_{\mathsf{SIG}}, \{vk_i\}_{i\in[n]})$ to $\mathcal{A}$.
- $\mathcal{B}_1$ has the signing keys $\{sk_i\}_{i\in[n]}$ of all users, thus can honestly answer $\mathcal{O}_{\text{SIGN}}$ queries, $\mathcal{O}_{\text{COR}}$ queries and $\mathcal{O}_{\text{LEAK}}$ queries made by $\mathcal{A}$, the same way as G_1.

  Concretely, for an $\mathcal{O}_{\text{SIGN}}$ query (i, m) made by $\mathcal{A}$, $\mathcal{B}_1$ samples $x \leftarrow_{\$} \mathcal{L}_\rho$ with witness w, computes $d := \mathsf{Priv}(sk_i, x)$, $\tau := H(vk_i, m)$, and invokes $\pi \leftarrow_{\$} \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$. Then, $\mathcal{B}_1$ 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, $\mathcal{B}_1$ returns sk_i to $\mathcal{A}$ and puts i to set $\mathcal{Q}_{\text{COR}}$. For an $\mathcal{O}_{\text{LEAK}}$ query (i, L), $\mathcal{B}_1$ returns $L(sk_i)$ to $\mathcal{A}$.
- Finally, $\mathcal{B}_1$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$. $\mathcal{B}_1$ outputs (i^*, x^*, d^*) to its own challenger.

It is clear to see that $\mathcal{B}_1$ simulates G_1 perfectly for $\mathcal{A}$, and $\mathcal{B}_1$'s output (i^*, x^*, d^*) succeeds in the $\mathsf{Exp}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{B}_1,n}$ experiment (cf. Fig. 5) so that $d^* \neq \mathsf{Priv}(sk_{i^*}, x^*) \wedge \mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, x^*, d^*) = 1$ as long as $\mathsf{InConsis}$ occurs. Thus, $\Pr_1[\mathsf{InConsis}] \leq \mathsf{Adv}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{B}_1,n}(\lambda)$ and Claim 1 follows. ▌

C.2 Full Description of Reductions $\mathcal{B}_3$ and $\mathcal{B}_4$ for Claim 3

We introduce an intermediate game $\mathsf{G}_{4.5}$ between G_4 and G_5:

– **Game $\mathsf{G}_{4.5}$:** It is the same as game G_4, 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_4, $\mathsf{G}_{4.5}$ and G_5 (due to the game change in G_3).

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. $\left|\Pr_4[\mathsf{Win}] - \Pr_{4.5}[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda)$ and $\left|\Pr_{4.5}[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda)$.

We first provide the full description of $\mathcal{B}_3$ for solving the multi-fold SMP related to $\mathscr{L}$ (cf. Def. 3). $\mathcal{B}_3$ is given $(\rho, \{x_j\}_{j\in[Q_s]})$, where $(\rho, td) \leftarrow_{\$} \mathscr{L}$, 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_4 or $\mathsf{G}_{4.5}$ for $\mathcal{A}$, depending on the input that $\mathcal{B}_3$ receives.

- Firstly, $\mathcal{B}_3$ invokes $\mathsf{pp}_{\mathsf{HPS}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{HPS}}$, $\mathsf{pp}_{\mathsf{NIZK}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{NIZK}}$, $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow\!\!\$\ \mathsf{CRSGen}(\rho)$, $H \leftarrow\!\!\$\ \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{SIG}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$. Then for each user $i \in [n]$, $\mathcal{B}_3$ samples signing key $sk_i \leftarrow\!\!\$\ \mathcal{SK}$ itself and computes the corresponding verification key $vk_i := \nu(sk_i)$. $\mathcal{B}_3$ 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 $d := \mathsf{Priv}(sk_i, x)$, $\tau := H(vk_i, m)$ and $\pi \leftarrow\!\!\$\ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$, 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 (τ, x, π) 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_4 for $\mathcal{A}$; in the case that $x = x_j$ is uniformly chosen from $\mathcal{X}$, $\mathcal{B}_3$ perfectly simulates $\mathsf{G}_{4.5}$ for $\mathcal{A}$.
- $\mathcal{B}_3$ uses $\{sk_i\}_{i\in[n]}$ to answer $\mathcal{O}_{\text{COR}}$ and $\mathcal{O}_{\text{LEAK}}$ queries for $\mathcal{A}$, the same way as G_4 and $\mathsf{G}_{4.5}$.
- Finally, $\mathcal{B}_3$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$. $\mathcal{B}_3$ uses the signing keys $\{sk_i\}_{i\in[n]}$ to decide whether the event Win defined in G_4 (which is the same as that defined in $\mathsf{G}_{4.5}$ and G_5) occurs, i.e.,

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

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

Overall, $\mathcal{B}_3$ simulates G_4 for $\mathcal{A}$ in the case $x_1, ..., x_{Q_s} \leftarrow\!\!\$\ \mathcal{L}_\rho$ and simulates $\mathsf{G}_{4.5}$ 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_4 differs non-negligibly from that in $\mathsf{G}_{4.5}$. Consequently, we have $\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_s}(\lambda) \geq \big|\Pr_4[\mathsf{Win}] - \Pr_{4.5}[\mathsf{Win}]\big|$.

Next, we provide the description of $\mathcal{B}_4$ for solving the multi-fold SMP related to $\mathscr{L}_0$ (cf. Def. 3). $\mathcal{B}_4$ is given $(\rho_0, \{x_j\}_{j\in[Q_s]})$, where $(\rho_0, td_0) \leftarrow\!\!\$\ \mathscr{L}_0$, 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 $(\rho, td) \leftarrow\!\!\$\ \mathscr{L}$ 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_5 for $\mathcal{A}$; in the case that $x = x_j$ is uniformly chosen from $\mathcal{X}$, $\mathcal{B}_4$ perfectly simulates $\mathsf{G}_{4.5}$ 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_5 differs non-negligibly from that in $\mathsf{G}_{4.5}$. Consequently, we have $\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_s}(\lambda) \geq \big|\Pr_{4.5}[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\big|$.

This completes the proof of Claim 3. ∎

C.3 Full Description of Reduction $\mathcal{B}_5$ for Claim 4

To bound $\Pr_6[\mathsf{Forge}]$, we construct an adversary $\mathcal{B}_5$ against the USS of tag-based QANIZK (cf. Def. 7). The full description of $\mathcal{B}_5$ is as follows. $\mathcal{B}_5$ is given $(\rho, td, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs})$ and has access to the oracle $\mathcal{O}_{\text{SIM}}$ defined in Fig. 4. $\mathcal{B}_5$ simulates G_6 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 $\mathsf{pp}_{\mathsf{SIG}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$. $\mathcal{B}_5$ also invokes $(\rho_0, td_0) \leftarrow\!\!\$\ \mathscr{L}_0$. Then for each user $i \in [n]$, $\mathcal{B}_5$ samples signing key $sk_i \leftarrow\!\!\$\ \mathcal{SK}$ itself and computes the corresponding verification key $vk_i := \nu(sk_i)$. $\mathcal{B}_5$ 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 $d := \mathsf{Priv}(sk_i, x)$ and $\tau := H(vk_i, m)$. 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 $\sigma := (x, d, \pi)$ to $\mathcal{A}$, puts (i, m, σ) to $\mathcal{Q}_{\text{SIGN}}$ and puts (τ, x, π) to $\mathcal{Q}_{\text{SIM}}$.
- $\mathcal{B}_5$ uses $\{sk_i\}_{i\in[n]}$ to answer $\mathcal{O}_{\text{COR}}$ and $\mathcal{O}_{\text{LEAK}}$ queries for $\mathcal{A}$, the same as G_6.
- Finally, $\mathcal{B}_5$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$. $\mathcal{B}_5$ computes $\tau^* := H(vk_{i^*}, m^*)$, and outputs (τ^*, x^*, π^*) to its own challenger.

It is clear to see that $\mathcal{B}_5$ simulates G_6 perfectly for $\mathcal{A}$, and $\mathcal{B}_5$ outputs a successful forgery (τ^*, x^*, π^*) to its own challenger so that $x^* \notin \mathcal{L}_\rho \ \wedge\ (\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}} \ \wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1$ as long as Forge occurs. Thus, $\Pr_6[\mathsf{Forge}] \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$ and Claim 4 follows. ∎

C.4 Full Description of Reduction $\mathcal{B}_6$ for Claim 5

To bound $\Pr_6[\mathsf{Win}]$, we construct an (unbounded) adversary $\mathcal{B}_6$ against the κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property of $\mathsf{PVQAHPS}$ (cf. Def. 10). The full description of $\mathcal{B}_6$ is as follows. $\mathcal{B}_6$ is given $(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \rho, \alpha_{\rho_0}(sk), \nu(sk))$, where $sk \leftarrow\!\!\$\ \mathcal{SK}$ is chosen by its own challenger, and has access to the oracle $\mathcal{O}_{\text{LEAK}}$ defined in Fig. 6, which is denoted by $\mathcal{O}^{(\mathsf{PVQAHPS})}_{\text{LEAK}}$ below to avoid confusing with the $\mathcal{O}_{\text{LEAK}}$ in SIG's experiment. $\mathcal{B}_6$ simulates G_6 for $\mathcal{A}$ as follows.

- Firstly, $\mathcal{B}_6$ invokes $\mathsf{pp}_{\mathsf{NIZK}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{NIZK}}$, $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow\!\!\$\ \mathsf{CRSGen}(\rho)$, samples $H \leftarrow\!\!\$\ \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{SIG}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$.

  $\mathcal{B}_6$ samples an index $\widehat{i} \leftarrow\!\!\$\ [n]$ uniformly, sets $sk_{\widehat{i}} := sk$ implicitly and defines $vk_{\widehat{i}} := \nu(sk)$ explicitly for user $\widehat{i}$, where sk is the hashing key chosen by $\mathcal{B}_6$'s own challenger and $\nu(sk)$ is part of $\mathcal{B}_6$'s own input. For all other users $i \in [n] \setminus \{\widehat{i}\}$, $\mathcal{B}_6$ samples signing key $sk_i \leftarrow\!\!\$\ \mathcal{SK}$ itself and computes $vk_i := \nu(sk_i)$. $\mathcal{B}_6$ sends $(\mathsf{pp}_{\mathsf{SIG}}, \{vk_i\}_{i\in[n]})$ to $\mathcal{A}$.
- For an $\mathcal{O}_{\text{SIGN}}$ query (i, m) made by $\mathcal{A}$, $\mathcal{B}_6$ computes a signature σ as follows.

  $\mathcal{B}_6$ first samples $x \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$ with witness w. If $i \neq \widehat{i}$, $\mathcal{B}_6$ computes $d := \mathsf{Priv}(sk_i, x)$ using sk_i, the same as G_6; if $i = \widehat{i}$, $\mathcal{B}_6$ computes $d := \mathsf{Pub}(\alpha_{\rho_0}(sk), x, w)$ using the projection key $\alpha_{\rho_0}(sk)$ contained in its own input, which is also the same as G_6 by the correctness of $\mathsf{PVQAHPS}$. Then, $\mathcal{B}_6$ computes $\tau := H(vk_i, m)$, invokes $\pi \leftarrow\!\!\$\ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$ and sets $\sigma := (x, d, \pi)$.

  $\mathcal{B}_6$ returns σ to $\mathcal{A}$, puts (i, m, σ) to $\mathcal{Q}_{\text{SIGN}}$ and puts (τ, x, π) to $\mathcal{Q}_{\text{SIM}}$.
- For an $\mathcal{O}_{\text{COR}}$ query i made by $\mathcal{A}$, if $i \neq \widehat{i}$, $\mathcal{B}_6$ returns sk_i to $\mathcal{A}$; if $i = \widehat{i}$, $\mathcal{B}_6$ aborts immediately.
- For an $\mathcal{O}_{\text{LEAK}}$ query (i, L) made by $\mathcal{A}$, if $i \neq \widehat{i}$, $\mathcal{B}_6$ returns $L(sk_i)$ to $\mathcal{A}$; if $i = \widehat{i}$, $\mathcal{B}_6$ submits L to its own $\mathcal{O}^{(\mathsf{PVQAHPS})}_{\text{LEAK}}$ oracle, obtains $L(sk)$ from $\mathcal{O}^{(\mathsf{PVQAHPS})}_{\text{LEAK}}$, and returns $L(sk)$ to $\mathcal{A}$.

- Finally, $\mathcal{B}_6$ receives a forgery $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ from $\mathcal{A}$. If $i^* = \widehat{i}$, $\mathcal{B}_6$ outputs (x^*, d^*) to its own challenger; if $i^* \neq \widehat{i}$, $\mathcal{B}_6$ 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^*, $\mathcal{B}_6$ simulates G_6 perfectly for $\mathcal{A}$, and $\mathcal{B}_6$'s output (x^*, d^*) succeeds in the $\mathsf{Exp}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}$ experiment so that $x^* \in \mathcal{L}_\rho \ \wedge \ d^* = \mathsf{Priv}(sk, x^*)$ as long as Win occurs. Thus, $\boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}(\lambda) \geq \frac{1}{n} \cdot \Pr_6[\mathsf{Win}]$ and Claim 5 follows. ▌

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

Theorem 2 (MUMC$^{\mathsf{c\&l}}$-CCA **Security of** PKE) *Assume that (i)* $\mathscr{L}$ *and* $\mathscr{L}_0$ *have hard SMPs, (ii)* QAHPS *is a QA-HPS for both* $\mathscr{L}$ *and* $\mathscr{L}_0$*, having PK-diversity, and supporting both* κ*-LR-*$\langle\mathscr{L},\mathscr{L}_0\rangle$*-key-switching and* $\mathscr{L}_0$*-multi-key-multi-extracting, (iii)* QANIZK *is a tag-based QA-NIZK for* $\mathscr{L}$*, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (iv)* $\mathcal{H}$ *is collision-resistant. Then the proposed* PKE *scheme in Fig. 10 is* MUMC$^{\mathsf{c\&l}}$-CCA *secure under* κ *bits leakage per user.*

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*

$$\mathsf{Adv}^{\mathsf{cca}\text{-}\mathsf{c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}(\lambda) \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{ml}\text{-}\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda)$$
$$+ \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk}\text{-}\mathsf{mext}}_{\mathsf{QAHPS},\mathcal{B}_6,n,Q_e}(\lambda) + \tfrac{n(n-1)}{2} \cdot \boldsymbol{\epsilon}^{\mathsf{pk}\text{-}\mathsf{div}}_{\mathsf{QAHPS}}(\lambda) + 2n \cdot \boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda).$$

Remark 5 (On the Tightness of PKE**'s** MUMC$^{\mathsf{c\&l}}$**-CCA security).** According to Theorem 2, PKE has tight MUMC$^{\mathsf{c\&l}}$-CCA security as long as both the multi-fold SMP related to $\mathscr{L}$ and the multi-language multi-fold SMP related to $\mathscr{L}_0$ have tight reductions, QAHPS has a tight $\mathscr{L}_0$-multi-key-multi-extracting property and QANIZK has a tight USS.

Moreover, similar to Remark 4, the statistical loss $\frac{n(n-1)}{2} \cdot \boldsymbol{\epsilon}^{\mathsf{pk}\text{-}\mathsf{div}}_{\mathsf{QAHPS}}(\lambda)$ exactly reflects the collision probability of public keys among n users, and is inherent to the MUMC$^{\mathsf{c\&l}}$-CCA security (even to MU$^{\mathsf{c}}$-CPA) of all PKE schemes, since if two public keys collide, an adversary can simply corrupt one user and use the secret key to decrypt a challenge ciphertext for another user. Nevertheless, the statistical loss does not affect the security tightness of PKE.

Proof of Theorem 2. We prove Theorem 2 by defining a sequence of games G_0–G_8 and showing adjacent games indistinguishable. A brief description of differences between adjacent games is summarized in Table 4 in Subsect. 6.2.

Game G_0**:** This is the $\mathsf{Exp}^{\mathsf{cca}\text{-}\mathsf{c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}$ experiment (cf. Fig. 9). Let Win denote the event that $\beta' = \beta$. By definition, $\mathsf{Adv}^{\mathsf{cca}\text{-}\mathsf{c\&l}}_{\mathsf{PKE},\mathcal{A},n,\kappa}(\lambda) = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

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 $d^* := \mathsf{Pub}(pk_{i^*}, x^*, w^*) + m_\beta$, $\tau^* := H(pk_{i^*}, d^*)$ 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 $\mathcal{Q}_{\text{ENC}}$. Upon an $\mathcal{O}_{\text{DEC}}$ query $(i, c = (x, d, \pi))$, the challenger computes $\tau := H(pk_i, d)$, returns $m := d - \mathsf{Priv}(sk_i, x)$ to $\mathcal{A}$ if $(i, c) \notin \mathcal{Q}_{\text{ENC}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1$ holds, and returns $\perp$ otherwise. For an $\mathcal{O}_{\text{COR}}$ query i, the challenger returns sk_i to $\mathcal{A}$ and puts i to set $\mathcal{Q}_{\text{COR}}$. For an $\mathcal{O}_{\text{LEAK}}$ query (i, L), the challenger returns $L(sk_i)$ to $\mathcal{A}$.

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., $\exists 1 \le i < j \le n$, s.t. $pk_i = pk_j$.

By the PK-diversity of QAHPS, $\left|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\right| \le \frac{n(n-1)}{2} \cdot \epsilon^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda)$.

Game G_2: It is the same as G_1, except that, when answering $\mathcal{O}_{\text{ENC}}(i^*, m_0, m_1)$, the challenger computes d^* and π^* without using the witness w^* for $x^* \in \mathcal{L}_\rho$:

- $d^* := \mathsf{Priv}(sk_{i^*}, x^*) + m_\beta$,
- $\pi^* \leftarrow_\$ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*, x^*)$.

Since x^* is chosen from $\mathcal{L}_\rho$ with witness w^*, by the correctness of QAHPS and by the perfect zero-knowledge of QANIZK, we have $\Pr_1[\mathsf{Win}] = \Pr_2[\mathsf{Win}]$.

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

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

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

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

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 $\left|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]\right| \le \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda)$.

Game G_4: It is the same as G_3, except that, at the beginning of the game, the challenger picks $(\rho_0^{(i)}, td_0^{(i)}) \leftarrow_\$ \mathscr{L}_0$ independently for each user $i \in [n]$ besides $(\rho, td) \leftarrow_\$ \mathscr{L}$, and when answering $\mathcal{O}_{\text{ENC}}(i^*, m_0, m_1)$, the challenger samples x^* from the i^*-th language $\mathcal{L}_{\rho_0^{(i^*)}}$, i.e., $x^* \leftarrow_\$ \mathcal{L}_{\rho_0^{(i^*)}}$, instead of $x^* \leftarrow_\$ \mathcal{L}_\rho$.

By the multi-fold SMP related to $\mathscr{L}$ and by the multi-language multi-fold SMP related to $\mathscr{L}_0$ (cf. Def. 14), 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 $\mathcal{O}_{\text{ENC}}$ queries, then further change $\mathsf{G}_{3.5}$ to G_4. Overall, we have the following claim.

Claim 6. $\left|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda)$.

We provide a proof for Claim 6 in Appendix D.1.

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

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

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

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

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

Claim 7. $\left|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$.

We provide a proof for Claim 7 in Appendix D.2. A subtlety is that $\mathcal{B}_5$ obtains the language trapdoor td from its own challenger, thus can use td to efficiently decide the membership of $\mathcal{L}_\rho$ when answering $\mathcal{O}_{\mathrm{DEC}}$ queries for $\mathcal{A}$.

Game $\mathsf{G}_{6.\eta}$, $0 \leq \eta \leq n$: It is the same as G_5, 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 $i^* \leq \eta$, the challenger switches sk_{i^*} to the new secret key sk'_{i^*} in computing d^*:

- $d^* := \mathsf{Priv}(sk'_{i^*}, x^*) + m_\beta = \mathsf{Pub}(\alpha_{\rho_0^{(i^*)}}(sk'_{i^*}), x^*, w^*) + m_\beta$,

where w^* is a witness of $x^* \in \mathcal{L}_{\rho_0^{(i^*)}}$. 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 $\mathcal{O}_{\mathrm{DEC}}$, $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{LEAK}}$ queries for all users $i \in [n]$.

It is clearly that $\mathsf{G}_{6.0}$ is identical to G_5, thus $\Pr_5[\mathsf{Win}] = \Pr_{6.0}[\mathsf{Win}]$.

For each $\eta \in [n]$, note that the only difference between $\mathsf{G}_{6.\eta-1}$ and $\mathsf{G}_{6.\eta}$ lies in the $\mathcal{O}_{\mathrm{ENC}}$ oracle for user η: in $\mathsf{G}_{6.\eta-1}$, $\mathcal{O}_{\mathrm{ENC}}$ computes $d^* := \mathsf{Priv}(sk_\eta, x^*) + m_\beta = \mathsf{Pub}(\alpha_{\rho_0^{(\eta)}}(sk_\eta), x^*, w^*) + m_\beta$ using sk_η, while in $\mathsf{G}_{6.\eta}$, $\mathcal{O}_{\mathrm{ENC}}$ computes $d^* = \mathsf{Priv}(sk'_\eta, x^*) + m_\beta = \mathsf{Pub}(\alpha_{\rho_0^{(\eta)}}(sk'_\eta), x^*, w^*) + m_\beta$ using sk'_η. Since $x^* \in \mathcal{L}_{\rho_0^{(\eta)}}$ with $\rho_0^{(\eta)}$ output by $\mathscr{L}_0$, by the κ-LR-$\langle\mathscr{L},\mathscr{L}_0\rangle$-key-switching property of QAHPS (cf. Def. 5), 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 8. For each $\eta \in [n]$, $|\Pr_{6.\eta-1}[\mathsf{Win}] - \Pr_{6.\eta}[\mathsf{Win}]| \leq 2 \cdot \epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda)$.

We provide a proof for Claim 8 in Appendix D.3.

Game G_7: It is the same as $\mathsf{G}_{6.n}$, except that, at the beginning of the game, the challenger picks $(\rho_0, td_0) \leftarrow_\$ \mathscr{L}_0$ besides $(\rho, td) \leftarrow_\$ \mathscr{L}$ and $(\rho_0^{(i)}, td_0^{(i)}) \leftarrow_\$ \mathscr{L}_0$

for each $i \in [n]$, and when answering $\mathcal{O}_{\text{ENC}}(i^*, m_0, m_1)$, the challenger always samples $x^* \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$ independently of i^*, instead of $x^* \leftarrow\!\!\$\ \mathcal{L}_{\rho_0^{(i^*)}}$.

By the multi-language multi-fold SMP related to $\mathscr{L}_0$ (cf. Def. 14) and by the multi-fold SMP related to $\mathscr{L}_0$, we can first change $\mathsf{G}_{6.n}$ to an intermediate game where the challenger samples $x^* \leftarrow\!\!\$\ \mathcal{X}$ for all the $\mathcal{O}_{\text{ENC}}$ queries, then change to G_7. Overall, we have $\left|\Pr_{6.n}[\mathsf{Win}] - \Pr_7[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda)$. The proof is similar to that of Claim 6. A subtlety is that $\mathcal{B}_3$ and $\mathcal{B}_4$ sample $(\rho, td) \leftarrow\!\!\$\ \mathscr{L}$ themselves, thus can always use td to decide the membership of $\mathcal{L}_\rho$ when answering $\mathcal{O}_{\text{DEC}}$ queries for $\mathcal{A}$.

Game G_8: It is the same as G_7, except that, for all the $\mathcal{O}_{\text{ENC}}$ queries, the challenger samples $d^* \leftarrow\!\!\$\ \mathcal{HV}$ uniformly, instead of computing using $\{sk'_i\}_{i\in[n]}$.

Note that the only place that G_7 differs from G_8 lies in the computations of d^* in the $\mathcal{O}_{\text{ENC}}$ oracle for all users $i^* \in [n]$, where $d^* := \mathsf{Priv}(sk'_{i^*}, x^*) + m_\beta$ in G_7 while $d^* \leftarrow\!\!\$\ \mathcal{HV}$ in G_8. Since $\{sk'_i\}_{i\in[n]}$ is used only in the computations of d^* 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 QAHPS (cf. Def. 11), we have the following claim.

Claim 9. $\left|\Pr_7[\mathsf{Win}] - \Pr_8[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{QAHPS},\mathcal{B}_6,n,Q_e}(\lambda)$.

We provide a proof for Claim 9 in Appendix D.4.

Finally in G_8, d^* is uniformly chosen from $\mathcal{HV}$ regardless of the value of β, thus the challenge bit β is completely hidden to $\mathcal{A}$. Then $\Pr_8[\mathsf{Win}] = \frac{1}{2}$.

Taking all things together, Theorem 2 follows. □

D.1 Proof of Claim 6

Claim 6. $\left|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda)$.

Proof. 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 $\mathcal{O}_{\text{ENC}}$ queries, the challenger samples $x^* \leftarrow\!\!\$\ \mathcal{X}$.

Since 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), we can directly construct two adversaries $\mathcal{B}_2$ and $\mathcal{B}_3$ for solving the multi-fold SMP related to $\mathscr{L}$ and the multi-language multi-fold SMP related to $\mathscr{L}_0$ respectively, so that $\left|\Pr_3[\mathsf{Win}] - \Pr_{3.5}[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda)$ and $\left|\Pr_{3.5}[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda)$. ($\mathcal{B}_2$ and $\mathcal{B}_3$ can sample all secret keys themselves, simulate $\mathsf{G}_3/\mathsf{G}_{3.5}/\mathsf{G}_4$ honestly for $\mathcal{A}$ depending on the challenges that $\mathcal{B}_2$ and $\mathcal{B}_3$ receive, and succeed as long as $\mathcal{A}$ distinguishes these games.)

Here we provide the full description of $\mathcal{B}_3$ for solving the multi-language multi-fold SMP related to $\mathscr{L}_0$ (cf. Def. 14), and $\mathcal{B}_2$ can be similarly described. $\mathcal{B}_3$ is given $(\{\rho_0^{(i)}, \{x_j^{(i)}\}_{j\in[Q_e]}\}_{i\in[n]})$, where $(\rho_0^{(i)}, td_0^{(i)}) \leftarrow\!\!\$\ \mathscr{L}_0$, and $\mathcal{B}_3$ aims to

decide whether $x_1^{(i)}, ..., x_{Q_e}^{(i)} \leftarrow\!\!\$\ \mathcal{L}_{\rho_0^{(i)}}$ (say $b = 0$) or $x_1^{(i)}, ..., x_{Q_e}^{(i)} \leftarrow\!\!\$\ \mathcal{X}$ (say $b = 1$), for each $i \in [n]$. $\mathcal{B}_3$ will simulate $\mathsf{G}_{3.5}$ or G_4 for $\mathcal{A}$, depending on the value of b.

- Firstly, $\mathcal{B}_3$ invokes $(\rho, td) \leftarrow\!\!\$\ \mathscr{L}$, $\mathsf{pp}_{\mathsf{HPS}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{HPS}}$, $\mathsf{pp}_{\mathsf{NIZK}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{NIZK}}$, $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow\!\!\$\ \mathsf{CRSGen}(\rho)$, samples $H \leftarrow\!\!\$\ \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{PKE}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$. Then for each user $i \in [n]$, $\mathcal{B}_3$ samples secret key $sk_i \leftarrow\!\!\$\ \mathcal{SK}$ itself and computes the corresponding public key $pk_i := \alpha_\rho(sk_i)$. $\mathcal{B}_3$ sends $(\mathsf{pp}_{\mathsf{PKE}}, \{pk_i\}_{i\in[n]})$ to $\mathcal{A}$. $\mathcal{B}_3$ also picks a challenge bit $\beta \leftarrow\!\!\$\ \{0, 1\}$ for $\mathcal{A}$.
- $\mathcal{B}_3$ has the secret keys sk_i of all users, thus can honestly answer $\mathcal{O}_{\textsc{Dec}}$ queries, $\mathcal{O}_{\textsc{Cor}}$ queries and $\mathcal{O}_{\textsc{Leak}}$ queries made by $\mathcal{A}$, the same way as $\mathsf{G}_{3.5}$ and G_4.
- As for $\mathcal{O}_{\textsc{Enc}}$ queries, when answering the j-th ($j \in [Q_e]$) $\mathcal{O}_{\textsc{Enc}}$ query (i^*, m_0, m_1), $\mathcal{B}_3$ sets x^* as the $x_j^{(i^*)}$ in its own input, and computes $d^* := \mathsf{Priv}(sk_{i^*}, x^*) + m_\beta$, $\tau^* := H(pk_{i^*}, d^*)$ and $\pi^* \leftarrow\!\!\$\ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*, x^*)$, without knowing a witness of x^*. $\mathcal{B}_3$ returns $c^* := (x^*, d^*, \pi^*)$ to $\mathcal{A}$, puts (i^*, c^*) to $\mathcal{Q}_{\textsc{Enc}}$ and puts (τ^*, x^*, π^*) to $\mathcal{Q}_{\textsc{Sim}}$.

  In the case $b = 0$, $x^* = x_j^{(i^*)}$ is uniformly random over $\mathcal{L}_{\rho_0^{(i^*)}}$, thus $\mathcal{B}_3$ perfectly simulates G_4 for $\mathcal{A}$; in the case $b = 1$, $x^* = x_j^{(i^*)}$ is uniformly random over $\mathcal{X}$, thus $\mathcal{B}_3$ perfectly simulates $\mathsf{G}_{3.5}$ for $\mathcal{A}$.
- Finally, $\mathcal{B}_3$ receives a bit β' from $\mathcal{A}$ and returns 1 to its own challenger if and only if $\beta' = \beta$.

Overall, $\mathcal{B}_3$ simulates G_4 for $\mathcal{A}$ in the case $b = 0$ and simulates $\mathsf{G}_{3.5}$ for $\mathcal{A}$ in the case $b = 1$, thus $\mathcal{B}_3$ successfully distinguishes $b = 0$ from $b = 1$ as long as the probability that $\beta' = \beta$ in G_4 differs non-negligibly from that in $\mathsf{G}_{3.5}$. Consequently, we have $\mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda) \geq \big|\Pr_{3.5}[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\big|$.

This completes the proof of Claim 6. ∎

D.2 Proof of Claim 7

Claim 7. $\big|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$.

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

$$(i, c = (x, d, \pi)) \notin \mathcal{Q}_{\textsc{Enc}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\textsc{Sim}} \wedge x \notin \mathcal{L}_\rho.$$

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

To bound $\Pr_5[\mathsf{Forge}]$, we construct an adversary $\mathcal{B}_5$ against the USS of tag-based QANIZK as follows. $\mathcal{B}_5$ is given $(\rho, td, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs})$ and has access to the oracle $\mathcal{O}_{\textsc{Sim}}$ defined in Fig. 4. $\mathcal{B}_5$ 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 $\mathsf{pp}_{\mathsf{PKE}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$. Then for each user $i \in [n]$, $\mathcal{B}_5$ picks $(\rho_0^{(i)}, td_0^{(i)}) \leftarrow\!\!\$\ \mathscr{L}_0$, samples secret key $sk_i \leftarrow\!\!\$\ \mathcal{SK}$ itself and computes the corresponding public key $pk_i := \alpha_\rho(sk_i)$. $\mathcal{B}_5$ 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 $x^* \leftarrow_{\$} \mathcal{L}_{\rho_0^{(i^*)}}$, computes $d^* := \mathsf{Priv}(sk_{i^*}, x^*) + m_\beta$ 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 to further check whether $x \in \mathcal{L}_\rho$. If $x \notin \mathcal{L}_\rho$, $\mathcal{B}_5$ returns $\perp$ to $\mathcal{A}$, the same as G_5, and sends (τ, x, π) to its own challenger as its forgery. If $x \in \mathcal{L}_\rho$, $\mathcal{B}_5$ returns $m := d - \mathsf{Priv}(sk_i, x)$ to $\mathcal{A}$, the same as G_5.
- $\mathcal{B}_5$ uses $\{sk_i\}_{i \in [n]}$ to answer $\mathcal{O}_{\text{COR}}$ and $\mathcal{O}_{\text{LEAK}}$ 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 $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge (\tau, x, \pi) \notin \mathcal{Q}_{\text{SIM}} \wedge x \notin \mathcal{L}_\rho$ as long as Forge occurs. Therefore, $\Pr_5[\mathsf{Forge}] \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK}, \mathcal{B}_5}(\lambda)$ and Claim 7 follows. ∎

D.3 Proof of Claim 8

Claim 8. For each $\eta \in [n]$, $|\Pr_{6.\eta-1}[\mathsf{Win}] - \Pr_{6.\eta}[\mathsf{Win}]| \leq 2 \cdot \epsilon^{\mathsf{lr}\text{-}\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS}, \mathcal{B}_7, \kappa}(\lambda)$.

Proof. Note that the only difference between $\mathsf{G}_{6.\eta-1}$ and $\mathsf{G}_{6.\eta}$ lies in the $\mathcal{O}_{\text{ENC}}$ oracle for user η: in $\mathsf{G}_{6.\eta-1}$, $\mathcal{O}_{\text{ENC}}$ computes $d^* := \mathsf{Priv}(sk_\eta, x^*) + m_\beta$ using sk_η, while in $\mathsf{G}_{6.\eta}$, $\mathcal{O}_{\text{ENC}}$ computes $d^* := \mathsf{Priv}(sk'_\eta, x^*) + m_\beta$ using sk'_η.

Let Cor_η denote the event that $\mathcal{A}$ corrupts user η, i.e., $\mathcal{A}$ ever queries $\mathcal{O}_{\text{COR}}(\eta)$ when $(\eta, \cdot) \notin \mathcal{Q}_{\text{ENC}}$ and obtains sk_η. In the case that Cor_η occurs, η is appended to $\mathcal{Q}_{\text{COR}}$, thus $\mathcal{A}$ is not allowed to query $\mathcal{O}_{\text{ENC}}(\eta, m_0, m_1)$ for user η, and $\mathsf{G}_{6.\eta-1}$ is identical to $\mathsf{G}_{6.\eta}$. Consequently,

$$|\Pr_{6.\eta-1}[\mathsf{Win}] - \Pr_{6.\eta}[\mathsf{Win}]| = |\Pr_{6.\eta-1}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta] - \Pr_{6.\eta}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta]|. \quad (1)$$

To bound (1), we first analyze the information about sk_η (resp. sk_η and sk'_η) that $\mathcal{A}$ may obtain in $\mathsf{G}_{6.\eta-1}$ (resp. $\mathsf{G}_{6.\eta}$) in the case that $\neg\mathsf{Cor}_\eta$ occurs.

- Firstly, the public keys contain $pk_\eta = \alpha_\rho(sk_\eta)$.
- In $\mathcal{O}_{\text{ENC}}(\eta, m_0, m_1)$, since $x^* \leftarrow_{\$} \mathcal{L}_{\rho_0^{(\eta)}}$, the behavior of $\mathcal{O}_{\text{ENC}}$ for user η is completely determined by $\alpha_{\rho_0^{(\eta)}}(sk_\eta)$ (resp. $\alpha_{\rho_0^{(\eta)}}(sk'_\eta)$).
- In $\mathcal{O}_{\text{DEC}}(\eta, c)$, the challenger will not output m unless $x \in \mathcal{L}_\rho$ (due to the new rejection rule added in G_5), thus the behavior of $\mathcal{O}_{\text{DEC}}$ for user η is completely determined by $\alpha_\rho(sk_\eta)$.
- In the case that $\neg\mathsf{Cor}_\eta$, $\mathcal{A}$ never queries $\mathcal{O}_{\text{COR}}(\eta)$.
- From $\mathcal{O}_{\text{LEAK}}(\eta, L)$, $\mathcal{A}$ can obtain at most κ bits information about sk_η before it issues the first encryption query $\mathcal{O}_{\text{ENC}}(\eta, \cdot, \cdot)$ w.r.t user η.

Overall, the information about sk_η (resp. sk_η and sk'_η) that $\mathcal{A}$ learns in $\mathsf{G}_{6.\eta-1}$ (resp. $\mathsf{G}_{6.\eta}$) is limited in $\alpha_\rho(sk_\eta)$, $\alpha_{\rho_0^{(\eta)}}(sk_\eta)$ (resp. $\alpha_{\rho_0^{(\eta)}}(sk'_\eta)$) and at most κ bits leakage information of sk_η.

Then we analyze (1). Intuitively, by the κ-LR-$\langle\mathscr{L},\mathscr{L}_0\rangle$-key-switching property of QAHPS (cf. Def. 5), $\alpha_{\rho_0^{(\eta)}}(sk_\eta)$ is statistically close to $\alpha_{\rho_0^{(\eta)}}(sk'_\eta)$, even in the presence of $\alpha_\rho(sk_\eta)$ and at most κ bits leakage about sk_η. Thus, the $\mathcal{O}_{\text{ENC}}$ for user η in $\mathsf{G}_{6.\eta-1}$ (using sk_η) is statistically close to that in $\mathsf{G}_{6.\eta}$ (using sk'_η).

Formally, we build an (unbounded) adversary $\mathcal{B}_7$ against the κ-LR-$\langle\mathscr{L},\mathscr{L}_0\rangle$-key-switching property of QAHPS. $\mathcal{B}_7$ is given $(\mathsf{pp}_{\mathsf{HPS}},\rho,\alpha_\rho(\widetilde{sk}))$, where $\widetilde{sk}\leftarrow_{\$}\mathcal{SK}$ is chosen by its own challenger, and has access to the oracles $\mathcal{O}_{\text{LEAK}}$ and $\mathcal{O}_{\text{CHAL}}$ defined in Fig. 3, which are denoted by $\mathcal{O}_{\text{LEAK}}^{(\mathsf{QAHPS})}$ and $\mathcal{O}_{\text{CHAL}}^{(\mathsf{QAHPS})}$ below to avoid confusing with the oracles in PKE's experiment. Recall that $\mathcal{B}_7$ is not allowed to access $\mathcal{O}_{\text{LEAK}}^{(\mathsf{QAHPS})}$ anymore once it queries $\mathcal{O}_{\text{CHAL}}^{(\mathsf{QAHPS})}$ (cf. Remark 1). $\mathcal{B}_7$ will simulate $\mathsf{G}_{6.\eta-1}$ (or $\mathsf{G}_{6.\eta}$) for $\mathcal{A}$. $\mathcal{B}_7$ picks a challenge bit $\beta\leftarrow_{\$}\{0,1\}$. Intuitively, $\mathcal{B}_7$ will implicitly set the secret key of user η as the $\widetilde{sk}$ chosen by its own challenger and explicitly define the public key of user η as the $\alpha_\rho(\widetilde{sk})$ 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 and the language parameters $(\rho_0^{(i)}, td_0^{(i)})\leftarrow_{\$}\mathscr{L}_0$ itself, thus can honestly answer $\mathcal{O}_{\text{ENC}}$ queries (sampling x^* from $\mathcal{L}_{\rho_0^{(i)}}$), $\mathcal{O}_{\text{DEC}}$ queries (using brute force to decide the membership of $\mathcal{L}_\rho$), $\mathcal{O}_{\text{COR}}$ queries and $\mathcal{O}_{\text{LEAK}}$ queries made by $\mathcal{A}$ for these users. For user η, $\mathcal{B}_7$ can answer $\mathcal{O}_{\text{DEC}}$ queries using the projection key $\alpha_\rho(\widetilde{sk})$ contained in its own input (since $\mathcal{O}_{\text{DEC}}$ will output $\perp$ unless $x\in\mathcal{L}_\rho$ & $\mathcal{B}_7$ can decide the membership of $\mathcal{L}_\rho$ using brute force), answer $\mathcal{O}_{\text{LEAK}}$ queries via its own $\mathcal{O}_{\text{LEAK}}^{(\mathsf{QAHPS})}$ oracle, and aborts immediately if $\mathcal{A}$ corrupts η. As for the *first* $\mathcal{O}_{\text{ENC}}$ query w.r.t. user η made by $\mathcal{A}$, $\mathcal{B}_7$ queries its own $\mathcal{O}_{\text{CHAL}}^{(\mathsf{QAHPS})}$ oracle, and obtains a challenge $(\rho_0,\alpha_{\rho_0}(\widetilde{sk}))$ if $b=0$ or $(\rho_0,\alpha_{\rho_0}(\widetilde{sk}'))$ if $b=1$, where $(\rho_0,td_0)\leftarrow_{\$}\mathscr{L}_0$, $\widetilde{sk}'\leftarrow_{\$}\mathcal{SK}$ and $b\leftarrow_{\$}\{0,1\}$ (the challenge bit) are chosen by $\mathcal{B}_7$'s own challenger. $\mathcal{B}_7$ will explicitly define the η-th language parameter $\rho_0^{(\eta)}$ as ρ_0 and implicitly set sk'_η as $\widetilde{sk}'$ for user 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 $\alpha_{\rho_0}(\widetilde{sk})$ (or $\alpha_{\rho_0}(\widetilde{sk}')$) contained in its own challenge to compute d^*. After the first $\mathcal{O}_{\text{ENC}}$ query w.r.t. user η, $\mathcal{A}$ is not allowed to query $\mathcal{O}_{\text{LEAK}}$ for user η (cf. Remark 3), so $\mathcal{B}$ does not need (and is in fact not allowed) to query its own $\mathcal{O}_{\text{LEAK}}^{(\mathsf{QAHPS})}$ oracle anymore. 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 $\mathsf{G}_{6.\eta-1}$ perfectly for $\mathcal{A}$ if $b=0$ and $\neg\mathsf{Cor}_\eta$ occurs, and simulates $\mathsf{G}_{6.\eta}$ 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 $\mathsf{G}_{6.\eta-1}$ differs

non-negligibly from that in $\mathsf{G}_{6.\eta}$ in the case $\neg\mathsf{Cor}_\eta$, and consequently, we have $\epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda) \geq \frac{1}{2}\cdot|\Pr_{6.\eta-1}[\mathsf{Win}\wedge\neg\mathsf{Cor}_\eta]-\Pr_{6.\eta}[\mathsf{Win}\wedge\neg\mathsf{Cor}_\eta]|$. Here the scalar $\frac{1}{2}$ arises due to the definition style of the advantage function $\epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda)$.

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

- Given $(\mathsf{pp}_{\mathsf{HPS}},\rho,\alpha_\rho(\widetilde{sk}))$, $\mathcal{B}_7$ first invokes $\mathsf{pp}_{\mathsf{NIZK}}\leftarrow_{\$}\mathsf{Setup}_{\mathsf{NIZK}}$, $(\mathsf{crs},\mathsf{td}_{\mathsf{crs}})\leftarrow_{\$}\mathsf{CRSGen}(\rho)$, samples $H\leftarrow_{\$}\mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{PKE}}:=(\rho,\mathsf{pp}_{\mathsf{HPS}},\mathsf{pp}_{\mathsf{NIZK}},\mathsf{crs},H)$. $\mathcal{B}_7$ also samples a challenge bit $\beta\leftarrow_{\$}\{0,1\}$ for $\mathcal{A}$.

  For user η, $\mathcal{B}_7$ sets $sk_\eta:=\widetilde{sk}$ implicitly and defines $pk_\eta:=\alpha_\rho(\widetilde{sk})$ explicitly, where $\widetilde{sk}$ is the hashing key chosen by $\mathcal{B}_7$'s own challenger and $\alpha_\rho(\widetilde{sk})$ is part of $\mathcal{B}_7$'s own input. For all other users $i\in[n]\setminus\{\eta\}$, $\mathcal{B}_7$ samples $(\rho_0^{(i)},td_0^{(i)})\leftarrow_{\$}\mathscr{L}_0$, 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 $\mathcal{O}_{\mathrm{ENC}}$ 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 $\mathsf{G}_{6.\eta-1}$ and $\mathsf{G}_{6.\eta}$.

  More precisely, $\mathcal{B}_7$ samples $x^*\leftarrow_{\$}\mathcal{L}_{\rho_0^{(i^*)}}$, computes $d^*:=\mathsf{Priv}(sk'_{i^*},x^*)+m_\beta$ using sk'_{i^*} if $i^*<\eta$ and computes $d^*:=\mathsf{Priv}(sk_{i^*},x^*)+m_\beta$ using sk_{i^*} if $i^*>\eta$. Then $\mathcal{B}_7$ computes $\tau^*:=H(pk_{i^*},d^*)$, 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 $\mathcal{Q}_{\mathrm{ENC}}$ and puts (τ^*,x^*,π^*) to $\mathcal{Q}_{\mathrm{SIM}}$.
- When answering an $\mathcal{O}_{\mathrm{ENC}}$ query (η,m_0,m_1) for user η made by $\mathcal{A}$, $\mathcal{B}_7$ computes a challenge ciphertext c^* as follows.

  If this is the *first* $\mathcal{O}_{\mathrm{ENC}}$ query w.r.t. user η made by $\mathcal{A}$, $\mathcal{B}_7$ queries its own $\mathcal{O}_{\mathrm{CHAL}}^{(\mathsf{QAHPS})}$ oracle, and obtains a challenge $(\rho_0,\widetilde{pk}_b)$, where $(\rho_0,td_0)\leftarrow_{\$}\mathscr{L}_0$, $\widetilde{pk}_0:=\alpha_{\rho_0}(\widetilde{sk})$, $\widetilde{pk}_1:=\alpha_{\rho_0}(\widetilde{sk'})$ with $\widetilde{sk'}\leftarrow_{\$}\mathcal{SK}$, $b\leftarrow_{\$}\{0,1\}$ are chosen by $\mathcal{B}_7$'s own challenger, and b is the challenge bit that $\mathcal{B}_7$ aims to guess. $\mathcal{B}_7$ defines $\rho_0^{(\eta)}:=\rho_0$ explicitly and sets $sk'_\eta:=\widetilde{sk'}$ implicitly, where ρ_0 is the language parameter contained in $\mathcal{B}_7$'s challenge and $\widetilde{sk'}$ is the hashing key chosen by $\mathcal{B}_7$'s own challenger to compute $\widetilde{pk}_1$.

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

  $\mathcal{B}_7$ returns c^* to $\mathcal{A}$, puts (η,c^*) to $\mathcal{Q}_{\mathrm{ENC}}$ and puts (τ^*,x^*,π^*) to $\mathcal{Q}_{\mathrm{SIM}}$.
- For an $\mathcal{O}_{\mathrm{DEC}}$ query $(i,c=(x,d,\pi))$ made by $\mathcal{A}$, $\mathcal{B}_7$ decrypts as follows.

  $\mathcal{B}_7$ computes $\tau:=H(pk_i,d)$, checks whether $(i,c)\notin\mathcal{Q}_{\mathrm{ENC}}\wedge\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs},\tau,x,\pi)=1\wedge(\tau,x,\pi)\notin\mathcal{Q}_{\mathrm{SIM}}$, and returns $\perp$ to $\mathcal{A}$ if the check fails. Then $\mathcal{B}_7$ uses brute force to further decide whether $x\in\mathcal{L}_\rho$. If $x\notin\mathcal{L}_\rho$, $\mathcal{B}_7$ returns $\perp$ to $\mathcal{A}$. If $x\in\mathcal{L}_\rho$, $\mathcal{B}_7$ uses brute force to find a witness w for $x\in\mathcal{L}_\rho$, computes $m:=d-\mathsf{Priv}(sk_i,x)$ using sk_i if $i\neq\eta$ and computes $m:=d-\mathsf{Pub}(pk_\eta,x,w)$ using pk_η and witness w if $i=\eta$, and returns m to $\mathcal{A}$.

  By the correctness of QAHPS, $\mathcal{B}_7$'s simulation of $\mathcal{O}_{\mathrm{DEC}}$ oracle is the same as $\mathsf{G}_{6.\eta-1}$ and $\mathsf{G}_{6.\eta}$.

- 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.
- For an $\mathcal{O}_{\text{LEAK}}$ query (i, L) made by $\mathcal{A}$, if $i \neq \eta$, $\mathcal{B}_7$ returns $L(sk_i)$ to $\mathcal{A}$; if $i = \eta$, $\mathcal{B}_7$ submits L to its own $\mathcal{O}_{\text{LEAK}}^{(\mathsf{QAHPS})}$ oracle, obtains $L(\widetilde{sk})$ from $\mathcal{O}_{\text{LEAK}}^{(\mathsf{QAHPS})}$, and returns $L(\widetilde{sk})$ to $\mathcal{A}$.

  Note that after $\mathcal{A}$'s first $\mathcal{O}_{\text{ENC}}$ query w.r.t. user η, $\mathcal{A}$ is not allowed to query $\mathcal{O}_{\text{LEAK}}$ for user η (cf. Remark 3). This perfectly matches the $\mathcal{B}_7$'s situation in which $\mathcal{B}_7$ is not allowed to query its own $\mathcal{O}_{\text{LEAK}}^{(\mathsf{QAHPS})}$ oracle after $\mathcal{B}_7$ queries $\mathcal{O}_{\text{CHAL}}^{(\mathsf{QAHPS})}$ (cf. Remark 1). Thus, $\mathcal{B}_7$ simulates $\mathcal{O}_{\text{LEAK}}$ perfectly for $\mathcal{A}$.
- 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 $i^* \neq \eta$, $\mathcal{O}_{\text{DEC}}$ and $\mathcal{O}_{\text{LEAK}}$ perfectly for $\mathcal{A}$, and simulates oracle $\mathcal{O}_{\text{COR}}$ perfectly for $\mathcal{A}$ as well in the case of $\neg\mathsf{Cor}_\eta$. Next, we analyze $\mathcal{B}_7$'s simulation of oracle $\mathcal{O}_{\text{ENC}}$ w.r.t. user η.

– If $b = 0$, $\mathcal{B}_7$'s challenge is $(\rho_0, \widetilde{pk}_0 = \alpha_{\rho_0}(\widetilde{sk}))$. Since $\mathcal{B}_7$ implicitly sets $sk_\eta := \widetilde{sk}$, by the correctness of QAHPS and by the fact that $x^* \in \mathcal{L}_{\rho_0}$ with witness w^*, it follows that $d^* := \mathsf{Pub}(\widetilde{pk}_0, x^*, w^*) + m_\beta = \mathsf{Priv}(\widetilde{sk}, x^*) + m_\beta = \mathsf{Priv}(sk_\eta, x^*) + m_\beta$, the same as $\mathsf{G}_{6.\eta-1}$.

– If $b = 1$, $\mathcal{B}_7$'s challenge is $(\rho_0, \widetilde{pk}_1 = \alpha_{\rho_0}(\widetilde{sk}'))$. Since $\mathcal{B}_7$ implicitly sets $sk'_\eta := \widetilde{sk}'$, by the correctness of QAHPS and by the fact that $x^* \in \mathcal{L}_{\rho_0}$ with witness w^*, it follows that $d^* := \mathsf{Pub}(\widetilde{pk}_1, x^*, w^*) + m_\beta = \mathsf{Priv}(\widetilde{sk}', x^*) + m_\beta = \mathsf{Priv}(sk'_\eta, x^*) + m_\beta$, the same as $\mathsf{G}_{6.\eta}$.

Overall, $\mathcal{B}_7$ simulates $\mathsf{G}_{6.\eta-1}$ perfectly for $\mathcal{A}$ in the case $b = 0$ and $\neg\mathsf{Cor}_\eta$, and simulates $\mathsf{G}_{6.\eta}$ perfectly for $\mathcal{A}$ in the case $b = 1$ and $\neg\mathsf{Cor}_\eta$. Therefore, we have

$$\begin{aligned}\boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda) &= |\Pr[\mathcal{B}_7 \Rightarrow b] - \tfrac{1}{2}| = \tfrac{1}{2} \cdot |\Pr[\mathcal{B}_7 \Rightarrow 1|b=0] - \Pr[\mathcal{B}_7 \Rightarrow 1|b=1]| \\ &= \tfrac{1}{2} \cdot |\Pr[\beta' = \beta \ \wedge \ \neg\mathsf{Cor}_\eta | b=0] - \Pr[\beta' = \beta \ \wedge \ \neg\mathsf{Cor}_\eta | b=1]| \\ &= \tfrac{1}{2} \cdot |\Pr_{6.\eta-1}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta] - \Pr_{6.\eta}[\mathsf{Win} \wedge \neg\mathsf{Cor}_\eta]|. \end{aligned} \tag{2}$$

Taking (1) and (2) together, Claim 8 follows. ∎

D.4 Proof of Claim 9

Claim 9. $\big|\Pr_7[\mathsf{Win}] - \Pr_8[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk}\text{-}\mathsf{mext}}_{\mathsf{QAHPS},\mathcal{B}_6,n,Q_e}(\lambda)$.

Proof. The only place that G_7 differs from G_8 lies in $\mathcal{O}_{\text{ENC}}$. For an $\mathcal{O}_{\text{ENC}}(i^*, m_0, m_1)$ query, the challenger samples $x^* \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$ (due to the change in G_7), and computes $d^* := \mathsf{Priv}(sk'_{i^*}, x^*) + m_\beta$ in G_7 while samples $d^* \leftarrow\!\!\$\ \mathcal{HV}$ in G_8.

Let us fix some notations. Let i^*_j, x^*_j, d^*_j, $m_{\beta,j}$ denote the i^*, x^*, d^*, m_β in the j-th $\mathcal{O}_{\text{ENC}}$ query, respectively, where $j \in [Q_e]$. The difference between G_7 and G_8 can be characterized by the following two distributions:

- G_7: $\left(x_j^* \leftarrow\!\!\$\ \mathcal{L}_{\rho_0},\ d_j^* := \mathsf{Priv}(sk'_{i_j^*}, x_j^*) + m_{\beta,j} \in \mathcal{HV} \right)_{j\in[Q_e]}$,
- G_8: $\left(x_j^* \leftarrow\!\!\$\ \mathcal{L}_{\rho_0},\ d_j^* \leftarrow\!\!\$\ \mathcal{HV} \right)_{j\in[Q_e]}$.

Since $\{sk'_i\}_{i\in[n]}$ is used only in the computations of $\{d_j^*\}_{j\in[Q_e]}$ 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 QAHPS (cf. Def. 11), 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 QAHPS by invoking $\mathcal{A}$. $\mathcal{B}_6$ is given $(\mathsf{pp}_{\mathsf{HPS}}, \rho_0, \{x_j, \{hv_{i,j}\}_{i\in[n]}\}_{j\in[Q_e]})$, where $(\rho_0, td_0) \leftarrow\!\!\$\ \mathscr{L}_0$, $sk'_1, ..., sk'_n \leftarrow\!\!\$\ \mathcal{SK}$, and $x_1, ..., x_{Q_e} \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$. $\mathcal{B}_6$ aims to decide whether $hv_{i,j} = \Lambda_{sk'_i}(x_j)$ 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_7 or G_8 for $\mathcal{A}$, depending on the value of b.

- Firstly, $\mathcal{B}_6$ invokes $(\rho, td) \leftarrow\!\!\$\ \mathscr{L}$, $\mathsf{pp}_{\mathsf{NIZK}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{NIZK}}$, $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow\!\!\$\ \mathsf{CRSGen}(\rho)$, samples $H \leftarrow\!\!\$\ \mathcal{H}$, and sets $\mathsf{pp}_{\mathsf{PKE}} := (\rho, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$. 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 to decide the membership of $\mathcal{L}_\rho$), $\mathcal{O}_{\text{COR}}$ queries and $\mathcal{O}_{\text{LEAK}}$ queries made by $\mathcal{A}$, the same way as G_7 and G_8.
- 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_j in its own input, and computes $d_j^* := hv_{i_j^*,j} + m_{\beta,j}$ using the $hv_{i_j^*,j}$ in its input. Then $\mathcal{B}_6$ computes $\tau_j^* := H(pk_{i_j^*}, d_j^*)$ 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 $b = 0$, $hv_{i_j^*,j} = \Lambda_{sk'_{i_j^*}}(x_j) = \Lambda_{sk'_{i_j^*}}(x_j^*) = \mathsf{Priv}(sk'_{i_j^*}, x_j^*)$, thus $\mathcal{B}_6$ perfectly simulates G_7 for $\mathcal{A}$; in the case $b = 1$, $hv_{i_j^*,j}$ is uniformly random over $\mathcal{HV}$, and so is d_j^*, thus $\mathcal{B}_6$ perfectly simulates G_8 for $\mathcal{A}$.
- 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_7 for $\mathcal{A}$ in the case $b = 0$ and simulates G_8 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_7 differs non-negligibly from that in G_8. Consequently, we have $\mathsf{Adv}^{\mathscr{L}_0\text{-mk-mext}}_{\mathsf{QAHPS},\mathcal{B}_6,n,Q_e}(\lambda) \geq \left| \Pr_7[\mathsf{Win}] - \Pr_8[\mathsf{Win}] \right|$.

This completes the proof of Claim 9. ∎

E Discussions on Potential Variants of Our Constructions

E.1 Discussions on Setup of Our SIG and PKE

In our SIG constructed in Fig. 8 and our PKE constructed in Fig. 10, the setup algorithms $\mathsf{Setup}_{\mathsf{SIG}}$ and $\mathsf{Setup}_{\mathsf{PKE}}$ produce ρ and crs along with trapdoors td and

$\mathsf{td}_{\mathsf{crs}}$, via $(\rho, td) \leftarrow_{\$} \mathscr{L}$ and $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{CRSGen}(\rho)$. Note that the trapdoors are not used at all in the other algorithms of SIG and PKE. The reason why we make the trapdoors explicit in our constructions is just to ease the security proofs (of Theorem 1 and Theorem 2). Note that the related works (e.g. [4, 21]) also produce trapdoors in their setup algorithms (though implicitly), so that their security reductions can use trapdoors for simulation.

Alternatively, we can define variants of $\mathscr{L}$ and CRSGen, say $\widetilde{\mathscr{L}}$ and $\widetilde{\mathsf{CRSGen}}$, that only output ρ and crs, i.e., $\rho \leftarrow_{\$} \widetilde{\mathscr{L}}$ and $\mathsf{crs} \leftarrow_{\$} \widetilde{\mathsf{CRSGen}}(\rho)$, respectively. We require the indistinguishability of the ρ output by $\mathscr{L}$ and $\widetilde{\mathscr{L}}$, and the indistinguishability of the crs output by CRSGen and $\widetilde{\mathsf{CRSGen}}$. A trivial implementation of $\widetilde{\mathscr{L}}$ is just to invoke $(\rho, td) \leftarrow_{\$} \mathscr{L}$ and output ρ, and similarly, $\widetilde{\mathsf{CRSGen}}$ simply invokes $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \widetilde{\mathsf{CRSGen}}(\rho)$ and outputs crs. There might exist other non-trivial ways of implementing $\widetilde{\mathscr{L}}$ and $\widetilde{\mathsf{CRSGen}}$ depending on the concrete instantiations.[5] Consequently, by invoking $\widetilde{\mathscr{L}}$ and $\widetilde{\mathsf{CRSGen}}$ instead of $\mathscr{L}$ and CRSGen, we obtain SIG and PKE whose setup algorithms $\mathsf{Setup}_{\mathsf{SIG}}$ and $\mathsf{Setup}_{\mathsf{PKE}}$ do not produce trapdoors explicitly. During the security proofs, we can then switch to $\mathscr{L}$ and CRSGen that produce trapdoors used for simulation, assuming the indistinguishability of those variants.

E.2 Discussions on Potential Variants of Our PKE

Here we discuss some potential variants of our PKE constructed in Fig. 10. One potential variant might be moving the sampling of $(\rho, td) \leftarrow_{\$} \mathscr{L}$ from $\mathsf{Setup}_{\mathsf{PKE}}$ to Gen, and putting the language parameter ρ into the public key of PKE. For this PKE variant, each user has its own ρ in the public key, and consequently, the $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ security proof will start with many languages (one per user). One might think that this would nullify the need of the new property "multi-language multi-fold SMP" (defined in Def. 14) by using the "multi-fold SMP" (defined in Def. 3) directly, thus simplifying the analysis.

However, as far as we understand, the "multi-fold SMP" is not sufficient to prove the tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{CCA}$ security of this PKE variant, and it seems that the new property "multi-language multi-fold SMP" is still a natural requirement. The reason is as follows.

– **Necessity of Multiple Language Changes.** Suppose there are n users, and the language parameter for user i is $\rho^{(i)}$, $i \in [n]$. According to our tight security proof strategy (cf. a high-level overview in Subsect. 2.2 and Fig. 2 and a formal proof in Appendix D), we need to separate the languages

[5] As an example, for our MDDH-based instantiation in Appendix I, the language distribution $\mathscr{L}_{\mathcal{D}_{\ell,k}}$ in Appendix I.2 samples $\mathbf{A} \leftarrow_{\$} \mathcal{D}_{\ell,k}$ and outputs $(\rho := [\mathbf{A}]_1 \in \mathbb{G}_1^{\ell \times k}, td := \mathbf{A} \in \mathbb{Z}_p^{\ell \times k})$, where td is the discrete logarithm of ρ. The variant $\widetilde{\mathscr{L}}_{\mathcal{D}_{\ell,k}}$ could sample ρ from $\mathbb{G}_1^{\ell \times k}$ directly according to appropriate distribution, without knowing its discrete logarithm. For instance, in the case that $\mathcal{D}_{\ell,k}$ is the uniform distribution $\mathcal{U}_{\ell,k}$ over $\mathbb{Z}_p^{\ell \times k}$, $\widetilde{\mathscr{L}}_{\mathcal{D}_{\ell,k}}$ can simply sample ρ uniformly from $\mathbb{G}_1^{\ell \times k}$.

involved in encryption ($\{\mathcal{L}_{\rho^{(i)}}\}_{i\in[n]}$) from the languages involved in decryption ($\{\mathcal{L}_{\rho^{(i)}}\}_{i\in[n]}$), so that the secret keys used for encryption can then be switched to independent ones to randomize the challenge ciphertexts via the key-switching property of QAHPS. As a result, we need to change the multiple languages ($\{\mathcal{L}_{\rho^{(i)}}\}_{i\in[n]}$) involved in encryption to some newly sampled multiple languages (say $\{\mathcal{L}_{\rho'^{(i)}}\}_{i\in[n]}$), in a tight way.

- **Insufficiency of "Multi-fold SMP".** "Multi-fold SMP" can only deal with a single language, hence it is hard for us to rely only on "multi-fold SMP" to change $\{\mathcal{L}_{\rho^{(i)}}\}_{i\in[n]}$ to $\{\mathcal{L}_{\rho'^{(i)}}\}_{i\in[n]}$ tightly and generically, without the loss of the number of languages.
- **Sufficiency of "Multi-language Multi-fold SMP".** Our new property "multi-language multi-fold SMP" is a natural choice for the multiple language changes. By "multi-language multi-fold SMP", we can first change the multiple languages involved in encryption ($\{\mathcal{L}_{\rho^{(i)}}\}_{i\in[n]}$) to $\mathcal{X}$, then change to the newly sampled multiple languages ($\{\mathcal{L}_{\rho'^{(i)}}\}_{i\in[n]}$).
- **Other Possible Approach.** It might be also possible to accomplish the tight security proof by considering other PKE variants and requiring some other properties from the language distribution, instead of the "multi-language multi-fold SMP". For example, another potential PKE variant would be setting the language parameters $\{\rho^{(i)}\}_{i\in[n]}$ of different users as some "re-randomized versions" of a same ρ. To define this PKE variant formally, one need to formalize an additional algorithm to support "language re-randomization" along with some new properties to support tight reduction. As far as we see, contrary to the expected simplification of the analysis, this approach seems to make the PKE construction and its security analysis more complicated.

Besides, moving the sampling of $(\rho, td) \leftarrow_{\$} \mathscr{L}$ from $\mathsf{Setup}_{\mathsf{PKE}}$ to Gen might lead to another subtlety. Note that in QANIZK, crs might depend on ρ, i.e., crs is generated by $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow_{\$} \mathsf{CRSGen}(\rho)$, so the generation of crs has to be moved from $\mathsf{Setup}_{\mathsf{PKE}}$ to Gen along with ρ for the PKE variant. Consequently, each user has its own crs in addition to ρ, and the multi-user security proof involves not only many languages but also many CRSs. To prove the tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA security for this PKE variant, a stronger requirement like "multi-CRS version of USS" would be needed from the building block QANIZK.

Lastly, moving the sampling of $(\rho, td) \leftarrow_{\$} \mathscr{L}$ from $\mathsf{Setup}_{\mathsf{PKE}}$ to Gen might lead to a less efficient signcryption (SC) construction, when combining the PKE variant with our SIG proposed in Sect. 5. Jumping a bit ahead, we propose an optimized SC construction in Appendix F by taking advantage of the similar structures and compatible underlying building blocks of our SIG and PKE (see also Subsect. 2.3 for an overview). In particular, we reuse the language parameter ρ in the Setup algorithms ($\mathsf{Setup}_{\mathsf{SIG}}$ and $\mathsf{Setup}_{\mathsf{PKE}}$), and based on this, we further integrate the ciphertext of PKE and signature of SIG by reusing the instance $x \in \mathcal{L}_\rho$ and the proof π of QANIZK (see Fig. 13 in Appendix F.2). If using the above PKE variant instead, it seems hard for us to reuse the language parameter

ρ, since our SIG's ρ is generated in Setup while the above PKE variant's ρ is moved to Gen, unless one moves the sampling of ρ from $\mathsf{Setup}_{\mathsf{SIG}}$ to Gen for the SIG as well, which will encounter similar subtleties as described above. Consequently, the inconsistent ways of generating ρ will lead to less compatible structures in PKE and SIG and might bring more obstacles for the optimization of SC.

F Signcryption with Tight MUMC$^{\mathsf{c\&l}}$-Priv&Auth Security

In this section, we present signcryption (SC) schemes with tight *privacy* (MUMC$^{\mathsf{c\&l}}$-Priv) and *authenticity* (MUMC$^{\mathsf{c\&l}}$-Auth), in the multi-user multi-challenge (MUMC) setting under CCA attacks, adaptive corruptions and key leakages.

In Appendix F.1, we define the syntax of SC and its MUMC$^{\mathsf{c\&l}}$-Priv and MUMC$^{\mathsf{c\&l}}$-Auth security. Then in Appendix F.2, we present our generic construction of SC from PV-QA-HPS, QA-HPS and QA-NIZK.

F.1 Signcryption and Its MUMC$^{\mathsf{c\&l}}$-Priv&Auth Security

Definition 20 (SC). *A signcryption (SC) scheme* $\mathsf{SC} = (\mathsf{Setup}_{\mathsf{SC}}, \mathsf{Gen}, \mathsf{SignEnc}, \mathsf{VrfyDec})$ *with message space* $\mathcal{M}$ *consists of four PPT algorithms:*

- $\mathsf{pp}_{\mathsf{SC}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{SC}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp}_{\mathsf{SC}}$*, which serves as an implicit input of other algorithms.*
- $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(\mathsf{pp}_{\mathsf{SC}})$*: Taking* $\mathsf{pp}_{\mathsf{SC}}$ *as input, the key generation algorithm outputs a pair of public key and secret key* $(\mathsf{pk}, \mathsf{sk})$*.*
- $c \leftarrow\!\!\$\ \mathsf{SignEnc}(\mathsf{sk}_s, \mathsf{pk}_r, m)$*: Taking as input a sender's secret key* sk_s*, a receiver's public key* pk_r *and a message* $m \in \mathcal{M}$*, the signcryption algorithm outputs a ciphertext* c*.*
- $m/\perp \leftarrow \mathsf{VrfyDec}(\mathsf{pk}_s, \mathsf{sk}_r, c)$*: Taking as input a sender's public key* pk_s*, a receiver's secret key* sk_r *and a ciphertext* c*, the deterministic unsigncryption algorithm outputs either a message* $m \in \mathcal{M}$ *or a special rejection symbol* $\perp$*.*

Correctness requires that for all $\mathsf{pp}_{\mathsf{SC}} \in \mathsf{Setup}_{\mathsf{SC}}$*,* $(\mathsf{pk}_s, \mathsf{sk}_s), (\mathsf{pk}_r, \mathsf{sk}_r) \in \mathsf{Gen}(\mathsf{pp}_{\mathsf{SC}})$*,* $m \in \mathcal{M}$*,* $c \in \mathsf{SignEnc}(\mathsf{sk}_s, \mathsf{pk}_r, m)$*, it holds that* $\mathsf{VrfyDec}(\mathsf{pk}_s, \mathsf{sk}_r, c) = m$*.*

Remark 6. Note that our syntax of SC above is slightly simplified. More precisely, we do not explicitly take identities of users (e.g., Alice, Bob) into account, while implicitly use the indices $i \in [n]$ of users (e.g., s, r) to represent their identities. Jumping a bit ahead, our generic construction of SC later in Appendix F.2 can naturally handle explicit user identities, and even support *associated data* as well, by simply putting them inside the collision-resistant hash function H when computing τ. To better illustrate our core ideas and techniques, we choose to use a slightly simplified syntax to present our SC construction later.

Bellare and Stepanovs [9] defined the syntax of SC with a multi-receiver version, since they focused on security proof of SC schemes used in Apple iMessage system. Meanwhile, they formalized privacy and authenticity for SC in

a multi-user and multi-challenge (MUMC) setting, capturing both insider and outsider security by allowing user corruptions. Their formalization unifies different security notions via so-called relaxing relations, hence covering attacks like CCA/gCCA2[3]/RCCA[11] attacks and ciphertext/plaintext integrity [7]. Here we define the syntax of SC in the common case of single receiver. We choose the strongest security notion that considers *CCA attacks* and *ciphertext integrity*, and formalize them as $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv and $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth, which asks ciphertext indistinguishability and ciphertext integrity, respectively, under CCA attacks, adaptive corruptions as well as *key leakages*, in the MUMC setting. Below we present the formal definition.

Definition 21 ($\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv and $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth for SC). *Let $\kappa = \kappa(\lambda) \in \mathbb{N}$. A signcryption scheme SC is $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv secure (resp. $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth secure) under κ bits leakage per user, if for any PPT adversary $\mathcal{A}$ and any polynomial n, it holds that $\mathsf{Adv}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa} \Rightarrow 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$ (resp. $\mathsf{Adv}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$), where the experiments $\mathsf{Exp}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}$ and $\mathsf{Exp}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}$ are defined in Fig. 11 and Fig. 12 respectively.*

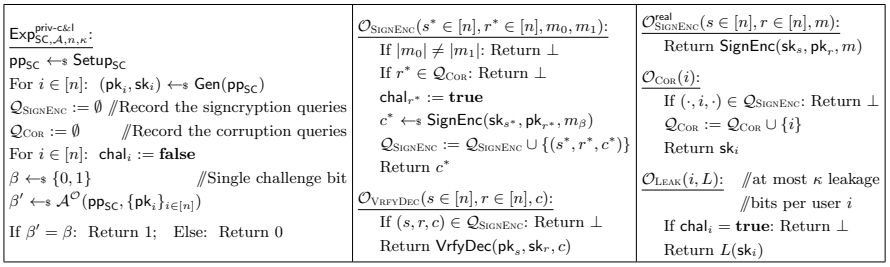

Fig. 11. The $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv security experiment $\mathsf{Exp}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}$ for SC, where the adversary $\mathcal{A}$ has access to oracles $\mathcal{O} = \{\mathcal{O}_{\textsc{SignEnc}}(\cdot,\cdot,\cdot,\cdot), \mathcal{O}^{\mathsf{real}}_{\textsc{SignEnc}}(\cdot,\cdot,\cdot), \mathcal{O}_{\textsc{VrfyDec}}(\cdot,\cdot,\cdot), \mathcal{O}_{\textsc{Cor}}(\cdot), \mathcal{O}_{\textsc{Leak}}(\cdot,\cdot)\}$. We note that besides challenge ciphertexts return by $\mathcal{O}_{\textsc{SignEnc}}$, $\mathcal{A}$ can also obtain honestly generated ciphertexts via $\mathcal{O}^{\mathsf{real}}_{\textsc{SignEnc}}$.

F.2 Generic Construction of Signcryption

We present a generic construction of $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv&Auth secure signcryption (SC). The underlying building blocks are as follows.

- Two language distributions $\mathscr{L}$ and $\mathscr{L}_0$, both of which have hard SMPs.
- A $\mathsf{QAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{Pub}, \mathsf{Priv})$ for both $\mathscr{L}$ and $\mathscr{L}_0$, whose hashing key space is $\mathcal{SK}$, projection key space is $\mathcal{PK}$ and hash value space is $\mathcal{HV}$. We require $\mathcal{HV}$ to be an (additive) group.

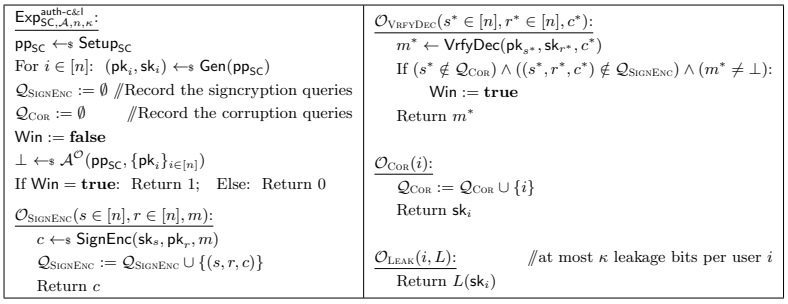

Fig. 12. The $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth security experiment $\mathsf{Exp}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}$ for SC, where the adversary $\mathcal{A}$ has access to oracles $\mathcal{O} = \{\mathcal{O}_{\text{SIGNENC}}(\cdot,\cdot,\cdot), \mathcal{O}_{\text{VRFYDEC}}(\cdot,\cdot,\cdot), \mathcal{O}_{\text{COR}}(\cdot), \mathcal{O}_{\text{LEAK}}(\cdot,\cdot)\}$.

- A publicly-verifiable $\mathsf{PVQAHPS} = (\widetilde{\mathsf{Setup}_{\mathsf{HPS}}}, \widetilde{\alpha}_{(\cdot)}, \widetilde{\nu}, \widetilde{\mathsf{Pub}}, \widetilde{\mathsf{Priv}}, \widetilde{\mathsf{Vrfy}_{\mathsf{HPS}}})$ for both $\mathscr{L}$ and $\mathscr{L}_0$, whose hashing key space is $\widetilde{\mathcal{SK}}$, verification key space is $\widetilde{\mathcal{VK}}$ and hash value space is $\widetilde{\mathcal{HV}}$.
- A tag-based $\mathsf{QANIZK} = (\mathsf{Setup}_{\mathsf{NIZK}}, \mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{Sim})$ for $\mathscr{L}$, whose tag space is $\mathcal{T}$.
- A family of collision-resistant hash functions $\mathcal{H} = \{H : \widetilde{\mathcal{VK}} \times \mathcal{PK} \times \mathcal{HV} \times \widetilde{\mathcal{HV}} \longrightarrow \mathcal{T}\}$.

Our generic construction of $\mathsf{SC} = (\mathsf{Setup}_{\mathsf{SC}}, \mathsf{Gen}, \mathsf{SignEnc}, \mathsf{VrfyDec})$ from QAHPS, PVQAHPS, QANIZK and $\mathcal{H}$ is shown in Fig. 13. The message space is $\mathcal{M} := \mathcal{HV}$.

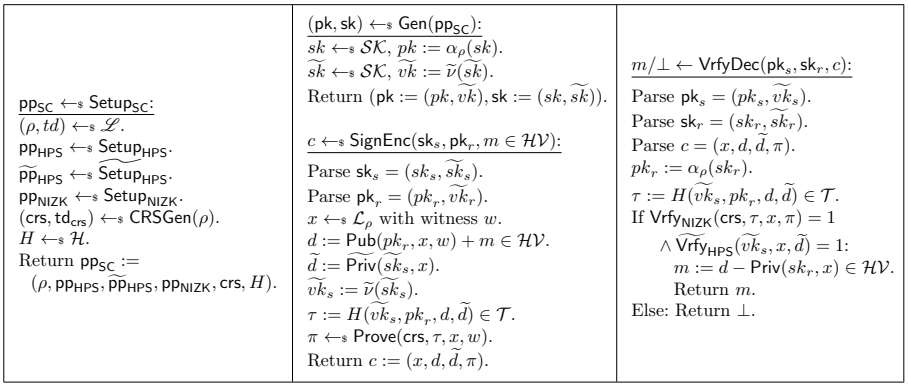

Fig. 13. Generic construction of $\mathsf{SC} = (\mathsf{Setup}_{\mathsf{SC}}, \mathsf{Gen}, \mathsf{SignEnc}, \mathsf{VrfyDec})$ from QAHPS, PVQAHPS, tag-based QANIZK and $\mathcal{H}$. The message space is $\mathcal{M} := \mathcal{HV}$.

Intuitively, our SC in Fig. 13 can be viewed as an integration of the SIG and PKE constructed in Subsect. 5.2 and Subsect. 6.2, but in a way different from the straightforward Encrypt-then-Sign method. We reuse the instance x and the QA-NIZK proof π, in order to obtain a more efficient construction. This is possible

thanks to the similar structures and compatible building blocks of our SIG and PKE. As a result, compared with SIG, our SC adds only one more component d to provide privacy; compared with PKE, our SC adds only one more component $\widetilde{d}$ to provide authenticity. Actually, our SC is more efficient than using existing generic constructions for building SC from SIG and PKE, such as "Encrypt-then-Sign", "Sign-then-Encrypt", "Encrypt-and-Sign", etc. [3, 9].

Correctness of SC follows directly from the correctness of QAHPS, the verification completeness of PVQAHPS and the perfect completeness of QANIZK.

Next, we show its $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv and $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth security. As noted above, our SC can be viewed as an integration of the SIG and PKE in Subsect. 5.2 and Subsect. 6.2, by reusing the instance x and the QA-NIZK proof π. Moreover, the PKE part (pk_r, d) and the SIG part $(\widetilde{vk}_s, \widetilde{d})$ are bound together via the collision-resistant hash function H, in order to avoid a trivial CCA attack. The $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv security of SC follows from a similar proof as the $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA security of PKE shown in Subsect. 6.2 and the $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth security of SC follows from a similar proof as the strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA security of SIG shown in Subsect. 5.2, with additional cares to the analysis of H's collision-resistance. Formally, we have two theorems below, followed by the proof sketches of them.

Theorem 3 ($\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv Security of SC). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii)* QAHPS *is a QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having PK-diversity, and supporting both κ-LR-$\langle\mathscr{L}, \mathscr{L}_0\rangle$-key-switching and $\mathscr{L}_0$-multi-key-multi-extracting, (iii)* PVQAHPS *is a publicly-verifiable QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having VK-diversity, (iv)* QANIZK *is a tag-based QA-NIZK for $\mathscr{L}$, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (v) $\mathcal{H}$ is collision-resistant. Then the proposed* SC *scheme in Fig. 13 is* $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv *secure under κ bits leakage per user.*

Concretely, for any number n of users and any adversary $\mathcal{A}$ who makes at most Q_e times of $\mathcal{O}_{\text{SignEnc}}$ queries and Q_d times of $\mathcal{O}_{\text{VrfyDec}}$ 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

$$\mathsf{Adv}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}(\lambda) \le \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda)$$
$$+ \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{QAHPS},\mathcal{B}_6,n,Q_e}(\lambda) + \tfrac{n(n-1)}{2} \cdot (\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) + \boldsymbol{\epsilon}^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}}(\lambda)) + 2n \cdot \boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda).$$

Theorem 4 ($\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth Security of SC). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii)* QAHPS *is a QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having PK-diversity, (iii)* PVQAHPS *is a publicly-verifiable QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having verification soundness, VK-diversity, and supporting κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting, (iv)* QANIZK *is a tag-based QA-NIZK for $\mathscr{L}$, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (v) $\mathcal{H}$ is collision-resistant. Then the proposed* SC *scheme in Fig. 13 is* $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth *secure under κ bits leakage per user.*

Concretely, for any number n of users and any adversary $\mathcal{A}$ who makes at most Q_e times of $\mathcal{O}_{\text{SignEnc}}$ queries and Q_d times of $\mathcal{O}_{\text{VrfyDec}}$ queries, there exist

adversaries $\mathcal{B}_1, \cdots, \mathcal{B}_6$*, such that* $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_5) \approx \mathbf{T}(\mathcal{A}) + (n + Q_e + Q_d) \cdot \mathsf{poly}(\lambda)$*, with* $\mathsf{poly}(\lambda)$ *independent of* $\mathbf{T}(\mathcal{A})$*, and*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}(\lambda) \leq\ & \mathsf{Adv}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{B}_1,n}(\lambda) + \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda) \\ & + \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \tfrac{n(n-1)}{2} \cdot (\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) + \boldsymbol{\epsilon}^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}}(\lambda)) + nQ_d \cdot \boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}(\lambda).\end{aligned}$$

Remark 7 (On the Tightness of SC's MUMC$^{\mathsf{c\&l}}$-Priv&Auth Security). According to Theorem 3 and Theorem 4, SC has both tight MUMC$^{\mathsf{c\&l}}$-Priv and tight MUMC$^{\mathsf{c\&l}}$-Auth security, as long as the multi-fold SMPs related to $\mathscr{L}$ and $\mathscr{L}_0$ and the multi-language multi-fold SMP related to $\mathscr{L}_0$ have tight reductions, QAHPS has a tight $\mathscr{L}_0$-multi-key-multi-extracting property, PVQAHPS has a tight verification soundness and QANIZK has a tight USS.

Proof Sketch of Theorem 3. The proof for the MUMC$^{\mathsf{c\&l}}$-Priv security of SC is essentially the same as that for the MUMC$^{\mathsf{c\&l}}$-CCA security of PKE shown in Subsect. 6.2, with additional cares to the analysis of H's collision-resistance.

The security proof goes with a sequence of games G_0–G_8, which are essentially the same as those defined in the proof of Theorem 2 for PKE.

Game G_0: This is the $\mathsf{Exp}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}$ experiment (cf. Fig. 11). Let Win denote the event that $\beta' = \beta$. By definition, $\mathsf{Adv}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}(\lambda) = |\Pr_0[\mathsf{Win}] - \frac{1}{2}|$.

Let $(\mathsf{pk}_i = (pk_i, \widetilde{vk}_i), \mathsf{sk}_i = (sk_i, \widetilde{sk}_i))$ denote the public/secret key pair of user $i \in [n]$. In this game, when answering an $\mathcal{O}_{\textsc{SignEnc}}$ query (s^*, r^*, m_0, m_1), the challenger samples $x^* \leftarrow\!\!\$\ \mathcal{L}_\rho$ with witness w^*, computes $d^* := \mathsf{Pub}(pk_{r^*}, x^*, w^*) + m_\beta$, $\widetilde{d}^* := \widetilde{\mathsf{Priv}}(\widetilde{sk}_{s^*}, x^*)$, $\tau^* := H(\widetilde{vk}_{s^*}, pk_{r^*}, d^*, \widetilde{d}^*)$, and invokes $\pi^* \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, \tau^*, x^*, w^*)$. Then, the challenger returns the challenge ciphertext $c^* := (x^*, d^*, \widetilde{d}^*, \pi^*)$ to $\mathcal{A}$ and puts (s^*, r^*, c^*) to set $\mathcal{Q}_{\textsc{SignEnc}}$.

When answering an $\mathcal{O}^{\mathsf{real}}_{\textsc{SignEnc}}$ query (s, r, m), the challenger samples $x \leftarrow\!\!\$\ \mathcal{L}_\rho$ with witness w, computes $d := \mathsf{Pub}(pk_r, x, w) + m$, $\widetilde{d} := \widetilde{\mathsf{Priv}}(\widetilde{sk}_s, x)$, $\tau := H(\widetilde{vk}_s, pk_r, d, \widetilde{d})$, and invokes $\pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$. Then, the challenger returns the honestly generated ciphertext $c := (x, d, \widetilde{d}, \pi)$ to $\mathcal{A}$.

Upon an $\mathcal{O}_{\textsc{VrfyDec}}$ query $(s, r, c = (x, d, \widetilde{d}, \pi))$, the challenger computes $\tau := H(\widetilde{vk}_s, pk_r, d, \widetilde{d})$, returns $m := d - \mathsf{Priv}(sk_r, x)$ to $\mathcal{A}$ if $(s, r, c) \notin \mathcal{Q}_{\textsc{SignEnc}} \wedge \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \wedge \widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_s, x, \widetilde{d}) = 1$ holds, and returns $\perp$ otherwise. For an $\mathcal{O}_{\textsc{Cor}}$ query i, the challenger returns $\mathsf{sk}_i = (sk_i, \widetilde{sk}_i)$ to $\mathcal{A}$ and puts i to set $\mathcal{Q}_{\textsc{Cor}}$. For an $\mathcal{O}_{\textsc{Leak}}$ query (i, L), the challenger returns $L(\mathsf{sk}_i) = L(sk_i, \widetilde{sk}_i)$ to $\mathcal{A}$.

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

By the PK-diversity of QAHPS and by the VK-diversity of PVQAHPS, we have that $\big|\Pr_0[\mathsf{Win}] - \Pr_1[\mathsf{Win}]\big| \leq \frac{n(n-1)}{2} \cdot (\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) + \boldsymbol{\epsilon}^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}}(\lambda))$.

Game G_2: It is the same as G_1, except that, when answering $\mathcal{O}_{\textsc{SignEnc}}(s^*, r^*, m_0, m_1)$, the challenger computes d^* and π^* without using the witness for $x^* \in \mathcal{L}_\rho$:

- $d^* := \mathsf{Priv}(sk_{r^*}, x^*) + m_\beta$,
- $\pi^* \leftarrow_\$ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau^*, x^*)$.

Since x^* is chosen from $\mathcal{L}_\rho$ with witness w^*, by the correctness of QAHPS and by the perfect zero-knowledge of QANIZK, we have $\mathrm{Pr}_1[\mathsf{Win}] = \mathrm{Pr}_2[\mathsf{Win}]$.

Game G_3: It is the same as G_2, except that, when answering $\mathcal{O}_{\textsc{SignEnc}}(s^*, r^*, m_0, m_1)$, the challenger also puts (τ^*, x^*, π^*) to a set $\mathcal{Q}_{\textsc{Sim}}$, and when answering $\mathcal{O}_{\textsc{VrfyDec}}(s, r, c = (x, d, \widetilde{d}, \pi))$, the challenger adds a new rejection rule:

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

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

$$\begin{aligned}&\exists\ (s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*)) \in \mathcal{Q}_{\textsc{SignEnc}},\ \text{s.t.}\\ &\quad (s, r, c = (x, d, \widetilde{d}, \pi)) \neq (s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*)) \ \wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1\\ &\wedge\ \widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_s, x, \widetilde{d}) = 1 \ \wedge\ (\tau, x, \pi) = (\tau^*, x^*, \pi^*) \in \mathcal{Q}_{\textsc{Sim}},\end{aligned}$$

where $\tau := H(\widetilde{vk}_s, pk_r, d, \widetilde{d})$ and $\tau^* := H(\widetilde{vk}_{s^*}, pk_{r^*}, d^*, \widetilde{d}^*)$.

Note that by $(s, r, c = (x, d, \widetilde{d}, \pi)) \neq (s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ and $(\tau, x, \pi) = (\tau^*, x^*, \pi^*)$, it follows that $(s, r, d, \widetilde{d}) \neq (s^*, r^*, d^*, \widetilde{d}^*)$ and $\tau = H(\widetilde{vk}_s, pk_r, d, \widetilde{d}) = H(\widetilde{vk}_{s^*}, pk_{r^*}, d^*, \widetilde{d}^*) = \tau^*$. Since there are no collisions among $\{pk_i\}_{i\in[n]}$ and no collisions among $\{\widetilde{vk}_i\}_{i\in[n]}$ (due to the game change in G_1), $(s, r, d, \widetilde{d}) \neq (s^*, r^*, d^*, \widetilde{d}^*)$ implies $(\widetilde{vk}_s, pk_r, d, \widetilde{d}) \neq (\widetilde{vk}_{s^*}, pk_{r^*}, d^*, \widetilde{d}^*)$. Thus the above event suggests a collision of H, and we have $\big|\mathrm{Pr}_2[\mathsf{Win}] - \mathrm{Pr}_3[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda)$.

Game G_4: It is the same as G_3, except that, at the beginning of the game, the challenger picks $(\rho_0^{(i)}, td_0^{(i)}) \leftarrow_\$ \mathscr{L}_0$ independently for each user $i \in [n]$ besides $(\rho, td) \leftarrow_\$ \mathscr{L}$, and when answering $\mathcal{O}_{\textsc{SignEnc}}(s^*, r^*, m_0, m_1)$, the challenger samples x^* from the r^*-th language $\mathcal{L}_{\rho_0^{(r^*)}}$, i.e., $x^* \leftarrow_\$ \mathcal{L}_{\rho_0^{(r^*)}}$, instead of $x^* \leftarrow_\$ \mathcal{L}_\rho$.

By the multi-fold SMP related to $\mathscr{L}$ and by the multi-language multi-fold SMP related to $\mathscr{L}_0$, $\big|\mathrm{Pr}_3[\mathsf{Win}] - \mathrm{Pr}_4[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda)$.

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

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

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

$$\begin{aligned}&(s, r, c = (x, d, \widetilde{d}, \pi)) \notin \mathcal{Q}_{\textsc{Enc}} \ \wedge\ \widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_s, x, \widetilde{d}) = 1\\ &\wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \ \wedge\ (\tau, x, \pi) \notin \mathcal{Q}_{\textsc{Sim}} \ \wedge\ x \notin \mathcal{L}_\rho.\end{aligned}$$

This event implies $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, x, \pi) = 1 \ \wedge\ (\tau, x, \pi) \notin \mathcal{Q}_{\textsc{Sim}} \ \wedge\ x \notin \mathcal{L}_\rho$. Thus by the USS of QANIZK, we have that $\big|\mathrm{Pr}_4[\mathsf{Win}] - \mathrm{Pr}_5[\mathsf{Win}]\big| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$.

Game $\mathsf{G}_{6.\eta}$, $0 \le \eta \le n$: It is the same as G_5, 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 $\mathcal{O}_{\text{SIGNENC}}(s^*, r^*, m_0, m_1)$ for receivers $r^* \le \eta$, the challenger computes d^* using sk'_{r^*} instead of using sk_{r^*}:

- $d^* := \mathsf{Priv}(sk'_{r^*}, x^*) + m_\beta$.

The challenger still uses $\{sk_i\}_{i\in[n]}$ to compute the public keys for all users $i \in [n]$, to answer $\mathcal{O}_{\text{SIGNENC}}$ queries for receivers $r^* > \eta$, and to answer $\mathcal{O}^{\mathsf{real}}_{\text{SIGNENC}}$, $\mathcal{O}_{\text{VRFYDEC}}$, $\mathcal{O}_{\text{COR}}$ and $\mathcal{O}_{\text{LEAK}}$ queries for all users $i \in [n]$.

It is clearly that $\mathsf{G}_{6.0}$ is identical to G_5, thus $\Pr_5[\mathsf{Win}] = \Pr_{6.0}[\mathsf{Win}]$.

For each $\eta \in [n]$, by the κ-LR-$\langle\mathscr{L}, \mathscr{L}_0\rangle$-key-switching property of QAHPS, the challenger can safely switch sk_η to sk'_η when answering $\mathcal{O}_{\text{SIGNENC}}$ for receiver η. We have that $|\Pr_{6.\eta-1}[\mathsf{Win}] - \Pr_{6.\eta}[\mathsf{Win}]| \le 2 \cdot \epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda)$ following a similar reduction as that in the proof of Claim 8 for PKE (cf. Appendix D.3). Note that $\mathcal{B}_7$ can sample all hashing keys $\{\widetilde{sk}_i\}_{i\in[n]}$ of $\mathsf{PVQAHPS}$ and handle all parts related to $\mathsf{PVQAHPS}$ by itself. In particular, to answer an $\mathcal{O}_{\text{LEAK}}$ query (η, L) for user η made by $\mathcal{A}$, $\mathcal{B}_7$ sends $L(\cdot, \widetilde{sk}_\eta)$ to its own leakage oracle and returns the answer to $\mathcal{A}$.

Game G_7: It is the same as $\mathsf{G}_{6.n}$, except that, at the beginning of the game, the challenger picks $(\rho_0, td_0) \leftarrow_\$ \mathscr{L}_0$ besides $(\rho, td) \leftarrow_\$ \mathscr{L}$ and $(\rho_0^{(i)}, td_0^{(i)}) \leftarrow_\$ \mathscr{L}_0$ for each $i \in [n]$, and when answering $\mathcal{O}_{\text{SIGNENC}}(s^*, r^*, m_0, m_1)$, the challenger always samples $x^* \leftarrow_\$ \mathcal{L}_{\rho_0}$ independently of r^*, instead of $x^* \leftarrow_\$ \mathcal{L}_{\rho_0^{(r^*)}}$.

By the multi-language multi-fold SMP related to $\mathscr{L}_0$ and by the multi-fold SMP related to $\mathscr{L}_0$, $\big|\Pr_{6.n}[\mathsf{Win}] - \Pr_7[\mathsf{Win}]\big| \le \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda)$.

Game G_8: It is the same as G_7, except that, for all the $\mathcal{O}_{\text{SIGNENC}}$ queries, the challenger samples $d^* \leftarrow_\$ \mathcal{HV}$ uniformly, instead of computing d^* with $\{sk'_i\}_{i\in[n]}$.

Since $\{sk'_i\}_{i\in[n]}$ is used only in the computations of d^* in $\mathcal{O}_{\text{SIGNENC}}$, and x^* in $\mathcal{O}_{\text{SIGNENC}}$ are uniformly chosen from $\mathcal{L}_{\rho_0}$, by the $\mathscr{L}_0$-multi-key-multi-extracting property of QAHPS, we have $\big|\Pr_7[\mathsf{Win}] - \Pr_8[\mathsf{Win}]\big| \le \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{QAHPS},\mathcal{B}_6,n,Q_e}(\lambda)$.

Finally in G_8, d^* is uniformly chosen from $\mathcal{HV}$ regardless of the value of β, thus the challenge bit β is perfectly hidden from $\mathcal{A}$. Then $\Pr_8[\mathsf{Win}] = \frac{1}{2}$.

Taking all things together, Theorem 3 follows. □

Proof Sketch of Theorem 4. The proof for the $\mathsf{MUMC}^{\mathsf{c\&l}}$-$\mathsf{Auth}$ security of SC is similar to the proof for the strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMA}$ security of SIG shown in Subsect. 5.2, with additional cares to the analysis of H's collision-resistance.

The security proof goes with a sequence of games G_0–G_8, which are similar to the games defined in the proof of Theorem 1 for SIG.

Game G_0: This is the $\mathsf{Exp}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}$ experiment (cf. Fig. 12). Let Win denote the event that $\mathsf{Win} = \textbf{true}$. By definition, $\mathsf{Adv}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{SC},\mathcal{A},n,\kappa}(\lambda) = \Pr_0[\mathsf{Win}]$.

Let $(\mathsf{pk}_i = (pk_i, \widetilde{vk}_i), \mathsf{sk}_i = (sk_i, \widetilde{sk}_i))$ denote the public/secret key pair of user $i \in [n]$. In this game, when answering an $\mathcal{O}_{\text{SignEnc}}$ query (s, r, m), the challenger samples $x \leftarrow\!\!\$\ \mathcal{L}_\rho$ with witness w, computes $d := \mathsf{Pub}(pk_r, x, w) + m$, $\widetilde{d} := \widetilde{\mathsf{Priv}}(\widetilde{sk}_s, x)$, $\tau := H(\widetilde{vk}_s, pk_r, d, \widetilde{d})$ and $\pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$. Then, the challenger returns $c := (x, d, \widetilde{d}, \pi)$ to $\mathcal{A}$ and puts (s, r, c) to set $\mathcal{Q}_{\text{SignEnc}}$. For an $\mathcal{O}_{\text{Cor}}$ query i, the challenger returns $\mathsf{sk}_i = (sk_i, \widetilde{sk}_i)$ to $\mathcal{A}$ and puts i to set $\mathcal{Q}_{\text{Cor}}$. For an $\mathcal{O}_{\text{Leak}}$ query (i, L), the challenger returns $L(\mathsf{sk}_i) = L(sk_i, \widetilde{sk}_i)$ to $\mathcal{A}$.

Upon an $\mathcal{O}_{\text{VrfyDec}}$ query $(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$, the challenger computes $\tau^* := H(\widetilde{vk}_{s^*}, pk_{r^*}, d^*, \widetilde{d}^*)$, sets $m^* := d^* - \mathsf{Priv}(sk_{r^*}, x^*)$ if $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \wedge \widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_{s^*}, x^*, \widetilde{d}^*) = 1$ holds, and sets $m^* := \perp$ otherwise. Then, the challenger returns m^* to $\mathcal{A}$, and sets $\mathsf{Win} := \mathbf{true}$ if and only if

$$s^* \notin \mathcal{Q}_{\text{Cor}} \ \wedge\ (s^*, r^*, c^*) \notin \mathcal{Q}_{\text{SignEnc}} \ \wedge\ m^* \neq \perp. \tag{3}$$

For simplicity and without loss of generality, we assume that the challenger terminates the interactions with $\mathcal{A}$ immediately once Win is set to $\mathbf{true}$.

Game G_1: It is the same as G_0, except that, when answering $\mathcal{O}_{\text{SignEnc}}(s, r, m)$, the challenger also puts (s, r, m, c) to a set $\mathcal{Q}_{\text{SignEncWithMsg}}$, and when answering $\mathcal{O}_{\text{VrfyDec}}(s^*, r^*, c^*)$, the challenger adds the following new rule:

- If $(s^*, r^*, c^*) \in \mathcal{Q}_{\text{SignEnc}}$, find $(s^*, r^*, m, c^*) \in \mathcal{Q}_{\text{SignEncWithMsg}}$ and return $m^* := m$ directly (and keep $\mathsf{Win} = \mathbf{false}$).

By the correctness of SC, those $(s^*, r^*, c^*) \in \mathcal{Q}_{\text{SignEnc}}$ generated by $\mathcal{O}_{\text{SignEnc}}$ always unsigncrypt to the m contained in $(s^*, r^*, m, c^*) \in \mathcal{Q}_{\text{SignEncWithMsg}}$, thus $m^* = m$ holds both in G_0 and G_1 for such queries. The change is just conceptual and we have $\Pr_0[\mathsf{Win}] = \Pr_1[\mathsf{Win}]$.

Game G_2: It is the same as G_1, except that, when answering $\mathcal{O}_{\text{VrfyDec}}(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ where $(s^*, r^*, c^*) \notin \mathcal{Q}_{\text{SignEnc}}$, the challenger adds the following new rejection rule:

- If $\widetilde{d}^* \neq \widetilde{\mathsf{Priv}}(\widetilde{sk}_{s^*}, x^*)$, return $m^* := \perp$ directly (and keep $\mathsf{Win} = \mathbf{false}$).

Clearly, G_1 and G_2 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\text{VrfyDec}}(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ s.t.

$$\begin{aligned}(s^*, r^*, c^*) \notin \mathcal{Q}_{\text{SignEnc}} \ \wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \\ \wedge\ \widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_{s^*}, x^*, \widetilde{d}^*) = 1 \ \wedge\ \widetilde{d}^* \neq \widetilde{\mathsf{Priv}}(\widetilde{sk}_{s^*}, x^*).\end{aligned}$$

This event implies $\widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_{s^*}, x^*, \widetilde{d}^*) = 1 \ \wedge\ \widetilde{d}^* \neq \widetilde{\mathsf{Priv}}(\widetilde{sk}_{s^*}, x^*)$. Thus by the verification soundness of $\mathsf{PVQAHPS}$, $\left|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS}, \mathcal{B}_1, n}(\lambda)$.

Game G_3: It is the same as G_2, except that, the challenger aborts immediately if there are collisions in $\{pk_i\}_{i\in[n]}$ or collisions in $\{\widetilde{vk}_i\}_{i\in[n]}$, i.e., $\exists 1 \le i < j \le n$, s.t. $pk_i = pk_j \vee \widetilde{vk}_i = \widetilde{vk}_j$.

By the PK-diversity of QAHPS and by the VK-diversity of $\mathsf{PVQAHPS}$, we have that $\left|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]\right| \le \frac{n(n-1)}{2} \cdot (\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) + \boldsymbol{\epsilon}^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}}(\lambda))$.

Game G_4: It is the same as G_3, except that, when answering $\mathcal{O}_{\textsc{SignEnc}}(s, r, m)$, the challenger computes d and π without using the witness w for $x \in \mathcal{L}_\rho$:

- $d := \mathsf{Priv}(sk_r, x) + m$,
- $\pi \leftarrow_{\$} \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, x)$.

Since x is chosen from $\mathcal{L}_\rho$ with witness w, by the correctness of QAHPS and by the perfect zero-knowledge of QANIZK, we have $\Pr_3[\mathsf{Win}] = \Pr_4[\mathsf{Win}]$.

Game G_5: It is the same as G_4, except that, when answering $\mathcal{O}_{\textsc{SignEnc}}(s, r, m)$, the challenger also puts (τ, x, π) to a set $\mathcal{Q}_{\textsc{Sim}}$, and when answering $\mathcal{O}_{\textsc{VrfyDec}}(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ where $(s^*, r^*, c^*) \notin \mathcal{Q}_{\textsc{SignEnc}}$, the challenger adds a second new rejection rule:

- If $(\tau^*, x^*, \pi^*) \in \mathcal{Q}_{\textsc{Sim}}$, return $m^* := \perp$ directly (and keep $\mathsf{Win} = \textbf{false}$).

Clearly, G_4 and G_5 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\textsc{VrfyDec}}(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ s.t.

$$\begin{aligned}
&\exists\, (s, r, c = (x, d, \widetilde{d}, \pi)) \in \mathcal{Q}_{\textsc{SignEnc}},\ \text{s.t.}\\
&\quad (s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*)) \neq (s, r, c = (x, d, \widetilde{d}, \pi)) \ \wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1\\
&\wedge\ \widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_{s^*}, x^*, \widetilde{d}^*) = 1 \ \wedge\ \widetilde{d}^* = \widetilde{\mathsf{Priv}}(\widetilde{sk}_{s^*}, x^*) \ \wedge\ (\tau^*, x^*, \pi^*) = (\tau, x, \pi) \in \mathcal{Q}_{\textsc{Sim}},
\end{aligned}$$

where $\tau^* := H(\widetilde{vk}_{s^*}, pk_{r^*}, d^*, \widetilde{d}^*)$ and $\tau := H(\widetilde{vk}_s, pk_r, d, \widetilde{d})$.

Note that by $(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*)) \neq (s, r, c = (x, d, \widetilde{d}, \pi))$ and $(\tau^*, x^*, \pi^*) = (\tau, x, \pi)$, it follows that $(s^*, r^*, d^*, \widetilde{d}^*) \neq (s, r, d, \widetilde{d})$ and $\tau^* = H(\widetilde{vk}_{s^*}, pk_{r^*}, d^*, \widetilde{d}^*) = H(\widetilde{vk}_s, pk_r, d, \widetilde{d}) = \tau$. Since there are no collisions among $\{pk_i\}_{i\in[n]}$ and no collisions among $\{\widetilde{vk}_i\}_{i\in[n]}$ (due to the game change in G_3), $(s^*, r^*, d^*, \widetilde{d}^*) \neq (s, r, d, \widetilde{d})$ implies $(\widetilde{vk}_{s^*}, pk_{r^*}, d^*, \widetilde{d}^*) \neq (\widetilde{vk}_s, pk_r, d, \widetilde{d})$. Thus the above event suggests a collision of H, and we have $\left|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\right| \le \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda)$.

Game G_6: It is the same as G_5, except that, at the beginning of the game, the challenger picks $(\rho_0, td_0) \leftarrow_{\$} \mathscr{L}_0$ besides $(\rho, td) \leftarrow_{\$} \mathscr{L}$, and for all the $\mathcal{O}_{\textsc{SignEnc}}$ 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 have that $\left|\Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}]\right| \le \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_3,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda)$.

Game G_7: It is the same as G_6, except that, when answering $\mathcal{O}_{\textsc{VrfyDec}}(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ where $(s^*, r^*, c^*) \notin \mathcal{Q}_{\textsc{SignEnc}}$, the challenger adds a third new rejection rule:

- If $x^* \notin \mathcal{L}_\rho$, return $m^* := \perp$ directly (and keep $\mathsf{Win} = \textbf{false}$).

Clearly, G_6 and G_7 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\text{VRFYDEC}}(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ s.t.

$$(s^*, r^*, c^*) \notin \mathcal{Q}_{\text{SIGNENC}} \ \wedge \ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \ \wedge \ \widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_{s^*}, x^*, \widetilde{d}^*) = 1$$
$$\wedge \ \widetilde{d}^* = \widetilde{\mathsf{Priv}}(\widetilde{sk}_{s^*}, x^*) \ \wedge \ (\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}} \ \wedge \ x^* \notin \mathcal{L}_\rho.$$

This event implies $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \wedge (\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}} \wedge x^* \notin \mathcal{L}_\rho$. Thus by the USS of QANIZK, we have $\left|\Pr_6[\mathsf{Win}] - \Pr_7[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda)$.

Game G_8: It is the same as G_7, except that,when answering $\mathcal{O}_{\text{VRFYDEC}}(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ where $(s^*, r^*, c^*) \notin \mathcal{Q}_{\text{SIGNENC}}$, the challenger adds a fourth new rejection rule:

- If $s^* \notin \mathcal{Q}_{\text{COR}}$, return $m^* := \perp$ directly (and keep $\mathsf{Win} = \textbf{false}$).

Clearly, G_7 and G_8 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\text{VRFYDEC}}(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ s.t.

$$(s^*, r^*, c^*) \notin \mathcal{Q}_{\text{SIGNENC}} \ \wedge \ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \ \wedge \ \widetilde{\mathsf{Vrfy}}_{\mathsf{HPS}}(\widetilde{vk}_{s^*}, x^*, \widetilde{d}^*) = 1$$
$$\wedge \ \widetilde{d}^* = \widetilde{\mathsf{Priv}}(\widetilde{sk}_{s^*}, x^*) \ \wedge \ (\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\text{SIM}} \ \wedge \ x^* \in \mathcal{L}_\rho \ \wedge \ s^* \notin \mathcal{Q}_{\text{COR}}.$$

By the κ-LR-$\langle \mathscr{L}_0, \mathscr{L} \rangle$-OT-extracting property of $\mathsf{PVQAHPS}$, the event that $x^* \in \mathcal{L}_\rho \ \wedge \ \widetilde{d}^* = \widetilde{\mathsf{Priv}}(\widetilde{sk}_{s^*}, x^*) \ \wedge \ s^* \notin \mathcal{Q}_{\text{COR}}$ can happen with only a negligible probability. We have that $|\Pr_7[\mathsf{Win}] - \Pr_8[\mathsf{Win}]| \leq nQ_d \cdot \epsilon^{\mathsf{lr}\text{-}\langle \mathscr{L}_0, \mathscr{L} \rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}(\lambda)$ following a similar reduction as that in the proof of Claim 5 for SIG (cf. Appendix C.4). Intuitively, $\mathcal{B}_6$ will randomly guess the user index $s^* \in [n]$ to embed the hashing key chosen by its own challenger into $\widetilde{sk}_{s^*}$ of user s^*. Besides, $\mathcal{B}_6$ can sample all hashing keys $\{sk_i\}_{i \in [n]}$ of QAHPS and handle all parts related to QAHPS itself. In particular, to answer an $\mathcal{O}_{\text{LEAK}}$ query (s^*, L) for user s^* made by $\mathcal{A}$, $\mathcal{B}_6$ sends $L(sk_{s^*}, \cdot)$ to its own leakage oracle and returns the answer to $\mathcal{A}$. At the end of reduction, $\mathcal{B}_6$ will randomly pick an $\mathcal{O}_{\text{VRFYDEC}}$ query $(s^*, r^*, c^* = (x^*, d^*, \widetilde{d}^*, \pi^*))$ among all the Q_d queries made by $\mathcal{A}$, and return the extracted pair $(x^*, \widetilde{d}^*)$ to its own challenger. Overall, $\mathcal{B}_6$ succeeds as long as s^* is correctly guessed and the above event occurs exactly in the $\mathcal{O}_{\text{VRFYDEC}}$ query chosen by $\mathcal{B}_6$. Thus, the security loss to $\epsilon^{\mathsf{lr}\text{-}\langle \mathscr{L}_0, \mathscr{L} \rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS},\mathcal{B}_6,\kappa}(\lambda)$ is nQ_d.

Finally in G_8, for any $\mathcal{O}_{\text{VRFYDEC}}$ query, the challenger always sets and returns $m^* := \perp$ to $\mathcal{A}$ in the case of $s^* \notin \mathcal{Q}_{\text{COR}} \ \wedge \ (s^*, r^*, c^*) \notin \mathcal{Q}_{\text{SIGNENC}}$. Thus, by the definition of Win, cf. (3), Win can never be set to $\textbf{true}$ in G_8, and $\Pr_8[\mathsf{Win}] = 0$.

Taking all things together, Theorem 4 follows. □

G MAC with Tight Strong MU$^{\text{c\&l}}$-CMVA Security

In this section, we present *probabilistic* message authentication code (MAC) schemes with tight strong MU$^{\text{c\&l}}$-CMVA security. We note that the strongly

$\mathsf{MU}^{\mathsf{c\&l}}$-CMA secure SIG constructed in Subsect. 5.2 directly implies a strongly $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA secure MAC [15]. Here we provide a new generic construction of MAC from QA-HPS (which is not necessarily publicly-verifiable) and QA-NIZK, admitting more efficient instantiations from MDDH.

In Appendix G.1, we define the syntax of MAC and its strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA security. Then in Appendix G.2, we present our new generic construction of MAC from QA-HPS and QA-NIZK.

G.1 MAC and Its Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA Security

Definition 22 (MAC). *A message authentication code (MAC) scheme* $\mathsf{MAC} = (\mathsf{Setup}_{\mathsf{MAC}}, \mathsf{Gen}, \mathsf{Tag}, \mathsf{Vrfy}_{\mathsf{MAC}})$ *with message space* $\mathcal{M}$ *consists of four PPT algorithms:*

- $\mathsf{pp}_{\mathsf{MAC}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{MAC}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp}_{\mathsf{MAC}}$*, which serves as an implicit input of other algorithms.*
- $sk \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp}_{\mathsf{MAC}})$*: Taking* $\mathsf{pp}_{\mathsf{MAC}}$ *as input, the key generation algorithm outputs a symmetric key* sk*.*
- $\sigma \leftarrow_{\$} \mathsf{Tag}(sk, m)$*: Taking as input a key* sk *and a message* $m \in \mathcal{M}$*, the (possibly probabilistic) tag generation algorithm outputs a tag* σ*.*
- $0/1 \leftarrow \mathsf{Vrfy}_{\mathsf{MAC}}(sk, m, \sigma)$*: Taking as input a key* sk*, a message* $m \in \mathcal{M}$ *and a tag* σ*, the deterministic verification algorithm outputs a bit indicating whether* σ *is a valid tag for* m*.*

Correctness requires that for all $\mathsf{pp}_{\mathsf{MAC}} \in \mathsf{Setup}_{\mathsf{MAC}}$*,* $sk \in \mathsf{Gen}(\mathsf{pp}_{\mathsf{MAC}})$*,* $m \in \mathcal{M}$*,* $\sigma \in \mathsf{Tag}(sk, m)$*, it holds that* $\mathsf{Vrfy}_{\mathsf{MAC}}(sk, m, \sigma) = 1$*.*

The standard security notion for MAC is EUF-CMA (existential unforgeability under chosen message attacks). In [15], Dodis et al. formalized a stronger notion called *EUF-CMVA*, by additionally allowing *chosen verification attacks* (CVA). The CVA attacks enable the adversary to adaptively submit any polynomial number of pairs (m^*, σ^*) as forgery and learn the results whether they pass the verification of MAC. The adversary wins as long as there exists one fresh pair (m^*, σ^*) passing the verification. As noted in [15], there exist MAC schemes that are EUF-CMA secure but not EUF-CMVA secure, thus EUF-CMVA is strictly stronger than EUF-CMA.

Here we extend EUF-CMVA security to the Multi-User setting, and additionally allow adaptive corruptions and key leakages. We formalize such security as $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA. Moreover, *strong* $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA requires that the adversary cannot even forge a new tag for a message that it has ever queried. Below we present the definition of strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA and the non-strong version can be easily adapted accordingly.

Definition 23 (Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA Security for MAC). *Let* $\kappa = \kappa(\lambda) \in \mathbb{N}$*. A MAC scheme* MAC *is strongly* $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA *secure under* κ *bits leakage per user, if for any PPT adversary* $\mathcal{A}$ *and any polynomial* n*, it holds that* $\mathsf{Adv}^{\mathsf{s\text{-}cmva\text{-}c\&l}}_{\mathsf{MAC},\mathcal{A},n,\kappa}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{s\text{-}cmva\text{-}c\&l}}_{\mathsf{MAC},\mathcal{A},n,\kappa} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*, where the experiment* $\mathsf{Exp}^{\mathsf{s\text{-}cmva\text{-}c\&l}}_{\mathsf{MAC},\mathcal{A},n,\kappa}$ *is defined in Fig. 14.*

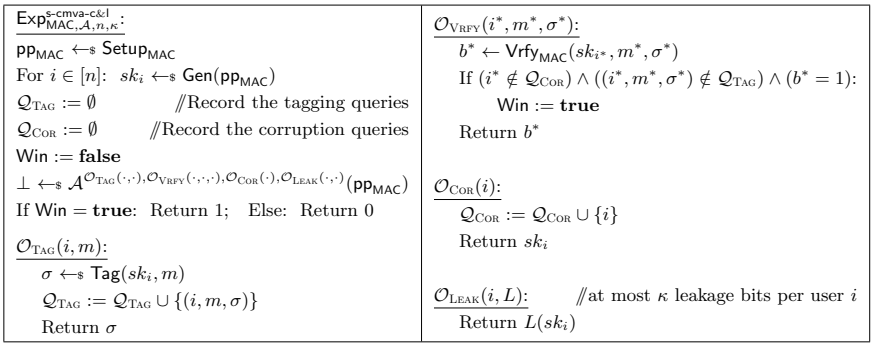

Fig. 14. The strong $\mathsf{MU}^{\text{c\&l}}$-CMVA security experiment $\mathsf{Exp}^{\text{s-cmva-c\&l}}_{\mathsf{MAC},\mathcal{A},n,\kappa}$ for MAC.

G.2 Generic Construction of MAC from QA-HPS and QA-NIZK

We present a generic construction of strongly $\mathsf{MU}^{\text{c\&l}}$-CMVA secure MAC. Let $\mathcal{M}$ be an arbitrary message space. The underlying building blocks are as follows.

- Two language distributions $\mathscr{L}$ and $\mathscr{L}_0$, both of which have hard SMPs.
- A $\mathsf{QAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{Pub}, \mathsf{Priv})$ for both $\mathscr{L}$ and $\mathscr{L}_0$, whose hashing key space is $\mathcal{SK}$ and projection key space is $\mathcal{PK}$. We stress that QAHPS is not required to be publicly-verifiable.
- A tag-based $\mathsf{QANIZK} = (\mathsf{Setup}_{\mathsf{NIZK}}, \mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{Sim})$ for $\mathscr{L}$, whose tag space is $\mathcal{T}$.
- A family of collision-resistant hash functions $\mathcal{H} = \{H : \mathcal{PK} \times \mathcal{M} \longrightarrow \mathcal{T}\}$.

Our generic construction of $\mathsf{MAC} = (\mathsf{Setup}_{\mathsf{MAC}}, \mathsf{Gen}, \mathsf{Tag}, \mathsf{Vrfy}_{\mathsf{MAC}})$ is shown in Fig. 15.

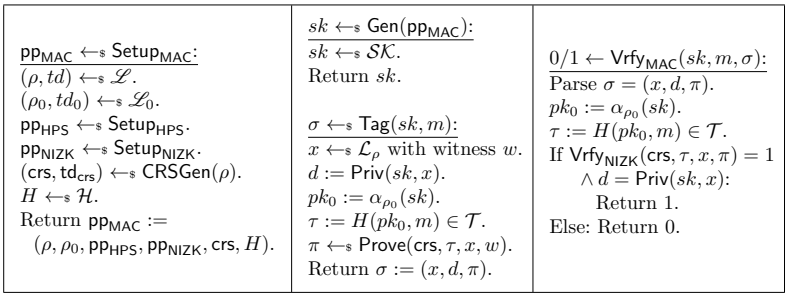

Fig. 15. Generic construction of $\mathsf{MAC} = (\mathsf{Setup}_{\mathsf{MAC}}, \mathsf{Gen}, \mathsf{Tag}, \mathsf{Vrfy}_{\mathsf{MAC}})$ from QAHPS, tag-based QANIZK and $\mathcal{H}$. The message space is $\mathcal{M}$.

The MAC construction resembles the SIG construction shown in Fig. 8 in Subsect. 5.2. From a high-level view, our MAC construction can be considered as replacing the building block PV-QA-HPS in SIG with a (not necessarily publicly-verifiable) QA-HPS. However, there are two main differences.

– Firstly, the $\mathsf{Vrfy}_{\mathsf{MAC}}$ algorithm of MAC checks $d = \mathsf{Priv}(sk, x)$ directly using sk, while the $\mathsf{Vrfy}_{\mathsf{SIG}}$ algorithm of SIG checks $\mathsf{Vrfy}_{\mathsf{HPS}}(vk, x, d) = 1$ publicly using $vk = \nu(sk)$.
– Secondly, unlike SIG, vk is not available any more in our MAC, so it cannot serve as part of input for the collision-resistant hash function H. Neither can $pk := \alpha_\rho(sk)$ since pk cannot be published or leaked for the sake of security of MAC (for similar reasons as the security of SIG, cf. the paragraph before Theorem 1 & the proof of Claim 5). We stress that as an input of H, $vk := \nu(sk)$ plays an important role in differentiating users by the VK-diversity (see the proof of Claim 2 in the proof of Theorem 1 for SIG). Without the help of vk for H, the security proof of MAC will be affected.

  To solve the problem of lacking vk in MAC, we exploit a new technical trick by taking advantage of the quasi-adaptive property of QAHPS again. More precisely, we project sk on another public language $\mathcal{L}_{\rho_0}$ with ρ_0 *independent* of ρ, and put the corresponding projection key $pk_0 := \alpha_{\rho_0}(sk)$ in the input of H. In the security proof of MAC, we can safely take care of the leakage on sk from $\alpha_{\rho_0}(sk)$ (similar to the proof of Claim 5 for SIG). Meanwhile, different users have different pk_0 (with overwhelming probability), then a security proof similar to that of Theorem 1 can go soundly for MAC.

Correctness of MAC follows directly from the perfect completeness of QANIZK. Next, we show its strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA security. According to the above discussion, the underlying building block QAHPS is additionally required to have PK-diversity for language parameters ρ_0 output by $\mathscr{L}_0$ (instead of ρ output by $\mathscr{L}$ as in Def. 12), i.e., $\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) := \Pr[\alpha_{\rho_0}(sk) = \alpha_{\rho_0}(sk')] \leq \mathsf{negl}(\lambda)$, where $(\rho_0, td_0) \leftarrow_{\$} \mathscr{L}_0$, $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$ and $sk, sk' \leftarrow_{\$} \mathcal{SK}$.

Theorem 5 (Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA Security of MAC). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii)* QAHPS *is a QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having PK-diversity for $\mathscr{L}_0$ and supporting κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting, (iii)* QANIZK *is a tag-based QA-NIZK for $\mathscr{L}$, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed* MAC *scheme in Fig. 15 is strongly* $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA *secure under κ bits leakage per user.*

Concretely, for any number n of users and any adversary $\mathcal{A}$ who makes at most Q_t times of $\mathcal{O}_{\mathrm{TAG}}$ queries and Q_v times of $\mathcal{O}_{\mathrm{VRFY}}$ queries, there exist adversaries $\mathcal{B}_1, \cdots, \mathcal{B}_5$, such that $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_4) \approx \mathbf{T}(\mathcal{A}) + (n + Q_t + Q_v) \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{s\text{-}cmva\text{-}c\&l}}_{\mathsf{MAC},\mathcal{A},n,\kappa}(\lambda) \leq\ & \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_t}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_3,Q_t}(\lambda) + \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_4}(\lambda) \\ & + \tfrac{n(n-1)}{2} \cdot \boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) + nQ_v \cdot \boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{QAHPS},\mathcal{B}_5,\kappa}(\lambda).\end{aligned}$$

Remark 8 (On the Tightness of MAC's Strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA security). According to Theorem 5, MAC has tight strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA security as long as both the multi-fold SMPs related to $\mathscr{L}$ and $\mathscr{L}_0$ have tight reductions and QANIZK has a tight USS.

Table 7. Brief Description of Games G_0-G_7 for the strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMVA}$ security proof of MAC. Here column "$\mathcal{O}_{\mathrm{TAG}}$" suggests how a tag $\sigma = (x, d, \pi)$ 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. Columns "$\mathcal{O}_{\mathrm{COR}}$" and "$\mathcal{O}_{\mathrm{LEAK}}$" show the output returned by $\mathcal{O}_{\mathrm{COR}}$ and $\mathcal{O}_{\mathrm{LEAK}}$ respectively. Column "$\mathcal{O}_{\mathrm{VRFY}}$'s additional check" describes the additional check made by $\mathcal{O}_{\mathrm{VRFY}}$ upon a verification query $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$, besides the routine check $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \wedge d^* = \mathsf{Priv}(sk_{i^*}, x^*)$, where $\tau^* := H(pk_{0,i^*}, m^*)$; $\mathcal{O}_{\mathrm{VRFY}}$ sets $b^* := 1$ if the check passes and sets $\mathsf{Win} := \mathbf{true}$ if $i^* \notin \mathcal{Q}_{\mathrm{COR}} \ \wedge \ (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\mathrm{TAG}} \ \wedge \ b^* = 1$.

	$\mathcal{O}_{\mathrm{TAG}}(i, m)$			$\mathcal{O}_{\mathrm{COR}}(i)$	$\mathcal{O}_{\mathrm{LEAK}}(i, L)$	$\mathcal{O}_{\mathrm{VRFY}}(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$'s additional check	Remark/Assumption
	x from	d using	π via				
G_0	$\mathcal{L}_\rho$	sk_i	Prove	sk_i	$L(sk_i)$		The strong $\mathsf{MU}^{\mathsf{c\&l}}$-$\mathsf{CMVA}$ experiment
G_1	$\mathcal{L}_\rho$	sk_i	Prove	sk_i	$L(sk_i)$		$\mathcal{O}_{\mathrm{VRFY}}$ sets $b^* := 1$ directly if $(i^*, m^*, \sigma^*) \in \mathcal{Q}_{\mathrm{TAG}}$: by correctness of MAC
G_2	$\mathcal{L}_\rho$	sk_i	Prove	sk_i	$L(sk_i)$		Abort if projection keys w.r.t. ρ_0 collide: by *PK-diversity for* $\mathscr{L}_0$ of QAHPS
G_3	$\mathcal{L}_\rho$	sk_i	Sim	sk_i	$L(sk_i)$		By *perfect zero-knowledge* of QANIZK
G_4	$\mathcal{L}_\rho$	sk_i	Sim	sk_i	$L(sk_i)$	$(\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}}$	By *collision-resistance* of $\mathcal{H}$
G_5	$\mathcal{L}_{\rho_0}$	sk_i	Sim	sk_i	$L(sk_i)$	$(\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}}$	By *multi-fold SMP* of $\mathscr{L}$ and $\mathscr{L}_0$
G_6	$\mathcal{L}_{\rho_0}$	sk_i	Sim	sk_i	$L(sk_i)$	$(\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}}$, $x^* \in \mathcal{L}_\rho$	By *USS* of QANIZK
G_7	$\mathcal{L}_{\rho_0}$	sk_i	Sim	sk_i	$L(sk_i)$	$(\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}}$, $x^* \in \mathcal{L}_\rho$, $i^* \in \mathcal{Q}_{\mathrm{COR}}$	By κ-*LR-*$(\mathscr{L}_0, \mathscr{L})$-*OT-extracting* of QAHPS
							$\Pr[\mathsf{Win}] = 0$ in G_7: since $b^* = 1$ contradicts $i^* \notin \mathcal{Q}_{\mathrm{COR}} \ \wedge \ (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\mathrm{TAG}}$

The proof of Theorem 5 mainly follows the proof of Theorem 1 for the SIG construction in Subsect. 5.2 and the proof of Theorem 4 for the SC construction in Appendix F.2. Here we provide a proof sketch.

Proof Sketch of Theorem 5. 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 7.

Game G_0: This is the $\mathsf{Exp}^{\mathsf{s\text{-}cmva\text{-}c\&l}}_{\mathsf{MAC},\mathcal{A},n,\kappa}$ experiment (cf. Fig. 14). Let Win denote the event that $\mathsf{Win} = \mathbf{true}$. By definition, $\mathsf{Adv}^{\mathsf{s\text{-}cmva\text{-}c\&l}}_{\mathsf{MAC},\mathcal{A},n,\kappa}(\lambda) = \Pr_0[\mathsf{Win}]$.

Let sk_i denote the key of user $i \in [n]$, and define by $pk_{0,i} := \alpha_{\rho_0}(sk_i)$ its corresponding projection key w.r.t. ρ_0, where ρ_0 is the language parameter contained in $\mathsf{pp}_{\mathsf{MAC}} = (\rho, \rho_0, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$.

In this game, when answering an $\mathcal{O}_{\mathrm{TAG}}$ query (i, m), the challenger samples $x \leftarrow_{\$} \mathcal{L}_\rho$ with witness w, computes $d := \mathsf{Priv}(sk_i, x)$, $\tau := H(pk_{0,i}, m)$, and invokes $\pi \leftarrow_{\$} \mathsf{Prove}(\mathsf{crs}, \tau, x, w)$. Then, the challenger returns $\sigma := (x, d, \pi)$ to $\mathcal{A}$ and puts (i, m, σ) to set $\mathcal{Q}_{\mathrm{TAG}}$. For an $\mathcal{O}_{\mathrm{COR}}$ query i, the challenger returns sk_i to $\mathcal{A}$ and puts i to set $\mathcal{Q}_{\mathrm{COR}}$. For an $\mathcal{O}_{\mathrm{LEAK}}$ query (i, L), the challenger returns $L(sk_i)$ to $\mathcal{A}$.

Upon an $\mathcal{O}_{\mathrm{VRFY}}$ query $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$, the challenger computes $\tau^* := H(pk_{0,i^*}, m^*)$, sets $b^* := 1$ if $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \wedge d^* = \mathsf{Priv}(sk_{i^*}, x^*)$ holds, and sets $b^* := 0$ otherwise. Then, the challenger returns b^* to $\mathcal{A}$, and

sets $\mathsf{Win} := \mathbf{true}$ if and only if

$$i^* \notin \mathcal{Q}_{\text{COR}} \ \wedge \ (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{TAG}} \ \wedge \ b^* = 1. \tag{4}$$

For simplicity and without loss of generality, we assume that the challenger terminates the interactions with $\mathcal{A}$ immediately once Win is set to $\mathbf{true}$.

Game G_1: It is the same as G_0, except that, when answering $\mathcal{O}_{\text{VRFY}}(i^*, m^*, \sigma^*)$, the challenger adds the following new rule:

- If $(i^*, m^*, \sigma^*) \in \mathcal{Q}_{\text{TAG}}$, return $b^* := 1$ directly (and keep $\mathsf{Win} = \mathbf{false}$).

By the correctness of MAC, those $(i^*, m^*, \sigma^*) \in \mathcal{Q}_{\text{TAG}}$ generated by $\mathcal{O}_{\text{TAG}}$ always pass the verification, thus $b^* = 1$ holds both in G_0 and G_1 for such queries. The change is just conceptual and we have $\Pr_0[\mathsf{Win}] = \Pr_1[\mathsf{Win}]$.

Game G_2: It is the same as G_1, except that, after generating n keys $\{sk_i\}_{i\in[n]}$, the challenger computes their corresponding projection keys $\{pk_{0,i} := \alpha_{\rho_0}(sk_i)\}_{i\in[n]}$ w.r.t. ρ_0, and aborts immediately if there are collisions in $\{pk_{0,i}\}_{i\in[n]}$, i.e., $\exists 1 \le i < j \le n$, s.t. $pk_{0,i} = pk_{0,j}$.

By the PK-diversity for $\mathscr{L}_0$ property of QAHPS, $\left|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]\right| \le \frac{n(n-1)}{2} \cdot \boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda)$.

Game G_3: It is the same as G_2, except that, when answering $\mathcal{O}_{\text{TAG}}(i, m)$, the challenger computes π without using the witness w for $x \in \mathcal{L}_\rho$:

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

Since x is chosen from $\mathcal{L}_\rho$ with witness w, by the perfect zero-knowledge of QANIZK, the change is just conceptual, and we have $\Pr_2[\mathsf{Win}] = \Pr_3[\mathsf{Win}]$.

Game G_4: It is the same as G_3, except that, when answering $\mathcal{O}_{\text{TAG}}(i, m)$, the challenger also puts (τ, x, π) to a set $\mathcal{Q}_{\text{SIM}}$, and when answering $\mathcal{O}_{\text{VRFY}}(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ where $(i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{TAG}}$, the challenger adds the following new rejection rule:

- If $(\tau^*, x^*, \pi^*) \in \mathcal{Q}_{\text{SIM}}$, return $b^* := 0$ directly (and keep $\mathsf{Win} = \mathbf{false}$).

Clearly, G_3 and G_4 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\text{VRFY}}(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ s.t.

$$\begin{aligned}
&\exists\ (i, m, \sigma = (x, d, \pi)) \in \mathcal{Q}_{\text{TAG}},\ \text{s.t.} \\
&\quad (i^*, m^*, \sigma^* = (x^*, d^*, \pi^*)) \neq (i, m, \sigma = (x, d, \pi)) \ \wedge \ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \\
&\wedge \ d^* = \mathsf{Priv}(sk_{i^*}, x^*) \ \wedge \ (\tau^*, x^*, \pi^*) = (\tau, x, \pi) \in \mathcal{Q}_{\text{SIM}},
\end{aligned}$$

where $\tau^* := H(pk_{0,i^*}, m^*)$ and $\tau := H(pk_{0,i}, m)$. There are two cases.

- **Case 1:** $(i^*, m^*) = (i, m)$. Together with $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*)) \neq (i, m, \sigma = (x, d, \pi)) \wedge (\tau^*, x^*, \pi^*) = (\tau, x, \pi)$, it follows that $d^* \neq d$. However, this contradicts $d^* = \mathsf{Priv}(sk_{i^*}, x^*) = \mathsf{Priv}(sk_i, x) = d$. Therefore, this case can never occur.
- **Case 2:** $(i^*, m^*) \neq (i, m)$. Since there are no collisions among $\{pk_{0,i}\}_{i\in[n]}$ (due to the game change in G_2), $(i^*, m^*) \neq (i, m)$ implies $(pk_{0,i^*}, m^*) \neq (pk_{0,i}, m)$. Together with $\tau^* = H(pk_{0,i^*}, m^*) = H(pk_{0,i}, m) = \tau$, this case suggests a collision of H.

Overall, we have $\left|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda)$.

Game G_5: It is the same as G_4, except that, for all the $\mathcal{O}_{\mathrm{TAG}}$ queries, the challenger samples $x \leftarrow\!\!\$\ \mathcal{L}_{\rho_0}$ instead of $x \leftarrow\!\!\$\ \mathcal{L}_\rho$, where ρ_0 is the language parameter contained in $\mathsf{pp}_{\mathsf{MAC}} = (\rho, \rho_0, \mathsf{pp}_{\mathsf{HPS}}, \mathsf{pp}_{\mathsf{NIZK}}, \mathsf{crs}, H)$.

By the multi-fold SMP related to $\mathscr{L}$ and by the multi-fold SMP related to $\mathscr{L}_0$, we have that $\left|\Pr_4[\mathsf{Win}] - \Pr_5[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_t}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_3,Q_t}(\lambda)$.

Game G_6: It is the same as G_5, except that, when answering $\mathcal{O}_{\mathrm{VRFY}}(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ where $(i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\mathrm{TAG}}$, the challenger adds a second new rejection rule:

- If $x^* \notin \mathcal{L}_\rho$, return $b^* := 0$ directly (and keep $\mathsf{Win} = \textbf{false}$).

Clearly, G_5 and G_6 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\mathrm{VRFY}}(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ s.t.

$$(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*)) \notin \mathcal{Q}_{\mathrm{TAG}} \ \wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1$$
$$\wedge\ d^* = \mathsf{Priv}(sk_{i^*}, x^*) \ \wedge\ (\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}} \ \wedge\ x^* \notin \mathcal{L}_\rho.$$

This event implies $\mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1 \wedge (\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}} \wedge x^* \notin \mathcal{L}_\rho$. Thus by the USS of QANIZK, we have $\left|\Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}]\right| \leq \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_4}(\lambda)$.

Game G_7: It is the same as G_6, except that, when answering $\mathcal{O}_{\mathrm{VRFY}}(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ where $(i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\mathrm{TAG}}$, the challenger adds a third new rejection rule:

- If $i^* \notin \mathcal{Q}_{\mathrm{COR}}$, return $b^* := 0$ directly (and keep $\mathsf{Win} = \textbf{false}$).

Clearly, G_6 and G_7 are the same unless that $\mathcal{A}$ ever queries $\mathcal{O}_{\mathrm{VRFY}}(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ s.t.

$$(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*)) \notin \mathcal{Q}_{\mathrm{TAG}} \ \wedge\ \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau^*, x^*, \pi^*) = 1$$
$$\wedge\ d^* = \mathsf{Priv}(sk_{i^*}, x^*) \ \wedge\ (\tau^*, x^*, \pi^*) \notin \mathcal{Q}_{\mathrm{SIM}} \ \wedge\ x^* \in \mathcal{L}_\rho \ \wedge\ i^* \notin \mathcal{Q}_{\mathrm{COR}}.$$

By the κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property of QAHPS, the event that $x^* \in \mathcal{L}_\rho \ \wedge\ d^* = \mathsf{Priv}(sk_{i^*}, x^*) \ \wedge\ i^* \notin \mathcal{Q}_{\mathrm{COR}}$ can happen with only a negligible probability. We have that $|\Pr_6[\mathsf{Win}] - \Pr_7[\mathsf{Win}]| \leq nQ_v \cdot \epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{QAHPS},\mathcal{B}_5,\kappa}(\lambda)$ following a similar reduction as that in the proof of Claim 5 for SIG (cf. Appendix C.4).

Intuitively, $\mathcal{B}_5$ will randomly guess the user index $i^* \in [n]$ to embed the hashing key chosen by its own challenger into the key sk_{i^*} of user i^*. At the end of reduction, $\mathcal{B}_5$ will randomly pick an $\mathcal{O}_{\text{VRFY}}$ query $(i^*, m^*, \sigma^* = (x^*, d^*, \pi^*))$ among all the Q_v queries made by $\mathcal{A}$, and return the contained (x^*, d^*) to its own challenger. Overall, $\mathcal{B}_5$ succeeds as long as i^* is correctly guessed and the above event occurs exactly in the $\mathcal{O}_{\text{VRFY}}$ query chosen by $\mathcal{B}_5$. Therefore, the security loss to $\epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{QAHPS},\mathcal{B}_5,\kappa}(\lambda)$ is nQ_v.

Finally in G_7, for any $\mathcal{O}_{\text{VRFY}}$ query, the challenger always sets and returns $b^* := 0$ to $\mathcal{A}$ in the case of $i^* \notin \mathcal{Q}_{\text{COR}} \ \wedge \ (i^*, m^*, \sigma^*) \notin \mathcal{Q}_{\text{TAG}}$. Thus, by the definition of Win, cf. (4), Win can never be set to **true** in G_7, and $\Pr_7[\mathsf{Win}] = 0$.

Taking all things together, Theorem 5 follows. □

H AE with Tight MUMC$^{\mathsf{c\&l}}$-Priv&Auth Security

In this section, we present *probabilistic* authenticated encryption (AE) schemes with tight *privacy* (MUMC$^{\mathsf{c\&l}}$-Priv) and *integrity/authenticity* (MUMC$^{\mathsf{c\&l}}$-Auth), in the multi-user multi-challenge (MUMC) setting under CCA attacks, adaptive corruptions and key leakages.

Recall that in Subsect. 6.2 and in Appendix G.2, we proposed generic constructions of MUMC$^{\mathsf{c\&l}}$-CCA secure PKE (which can be used as an symmetric encryption (SE) scheme) and strongly MU$^{\mathsf{c\&l}}$-CMVA secure MAC. By the standard *Encrypt-then-MAC* method [7], we can immediately obtain an AE scheme that achieves both MUMC$^{\mathsf{c\&l}}$-Priv and MUMC$^{\mathsf{c\&l}}$-Auth security. A ciphertext of the resulting AE scheme consists of a PKE (SE) ciphertext $c = (x, d, \pi)$ and a MAC tag $\sigma = (\widetilde{x}, \widetilde{d}, \widetilde{\pi})$.

Here we provide a more efficient generic construction of AE, by optimizing the Encrypt-then-MAC method with similar ideas in our SC construction in Appendix F.2. Intuitively, thanks to the similar structures and the same building blocks of our PKE (SE) and MAC constructions, we can reuse the instance x and the QA-NIZK proof π, and the resulting ciphertext of our new AE consists of only $(x, d, \widetilde{d}, \pi)$, thus saving one instance $\widetilde{x}$ and one QA-NIZK proof $\widetilde{\pi}$.

As a result, compared with our MAC, the new AE scheme adds only one more component d to provide privacy; compared with our PKE (SE), the new AE scheme adds only one more component $\widetilde{d}$ to provide integrity/authenticity.

The rest of this section is organized as follows. In Appendix H.1, we define the syntax of AE and its MUMC$^{\mathsf{c\&l}}$-Priv and MUMC$^{\mathsf{c\&l}}$-Auth security. Then in Appendix H.2, we present our new generic construction of AE.

H.1 Authenticated Encryption and Its MUMC$^{\mathsf{c\&l}}$-Priv&Auth Security

Definition 24 (AE). *An authenticated encryption (AE) scheme* $\mathsf{AE} = (\mathsf{Setup}_{\mathsf{AE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}$ *consists of four PPT algorithms:*

- $\mathsf{pp}_{\mathsf{AE}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{AE}}$*: The setup algorithm outputs a public parameter* $\mathsf{pp}_{\mathsf{AE}}$*, which serves as an implicit input of other algorithms.*
- $\mathsf{sk} \leftarrow_{\$} \mathsf{Gen}(\mathsf{pp}_{\mathsf{AE}})$*: Taking* $\mathsf{pp}_{\mathsf{AE}}$ *as input, the key generation algorithm outputs a symmetric key* sk*.*
- $c \leftarrow_{\$} \mathsf{Enc}(\mathsf{sk}, m)$*: Taking as input a key* sk *and a message* $m \in \mathcal{M}$*, the (possibly probabilistic) encryption algorithm outputs a ciphertext* c*.*
- $m/\perp \leftarrow \mathsf{Dec}(\mathsf{sk}, c)$*: Taking as input a 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 $\mathsf{pp}_{\mathsf{AE}} \in \mathsf{Setup}_{\mathsf{AE}}$, $\mathsf{sk} \in \mathsf{Gen}(\mathsf{pp}_{\mathsf{AE}})$, $m \in \mathcal{M}$, $c \in \mathsf{Enc}(\mathsf{sk}, m)$*, it holds that* $\mathsf{Dec}(\mathsf{sk}, c) = m$*.*

Remark 9. Note that our syntax of AE above can be extended to AE with Associated Data (AEAD) [36], by allowing the Enc and Dec algorithms additionally take an *associated data* as input. Jumping a bit ahead, our generic construction of AE later in Appendix H.2 can naturally support associated data, by simply putting it inside the collision-resistant hash function H when computing τ. We choose to use a slightly simplified syntax to better illustrate the core ideas and techniques in our AE construction.

The primary security goals of AE include privacy and integrity/authenticity. In [25], Jager et al. formalized privacy and integrity/authenticity for AE in a multi-user and multi-challenge (MUMC) setting under adaptive corruptions, via a general definition framework. Here we choose a strong variant that considers *CCA attacks* and *ciphertext integrity*. Our formalization of privacy and integrity/authenticity, denoted by $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv and $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth, like those defined for SC in Appendix F.1, ask ciphertext indistinguishability and ciphertext integrity, respectively, under CCA attacks, adaptive corruptions as well as *key leakages*, in the MUMC setting. Below we present the formal definition.

Definition 25 ($\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv and $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth for AE). *Let* $\kappa = \kappa(\lambda) \in \mathbb{N}$*. An AE scheme* AE *is* $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv *secure (resp.* $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth *secure) under* κ *bits leakage per user, if for any PPT adversary* $\mathcal{A}$ *and any polynomial* n*, it holds that* $\mathsf{Adv}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa} \Rightarrow 1] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda)$ *(resp.* $\mathsf{Adv}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa}(\lambda) := \Pr[\mathsf{Exp}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa} \Rightarrow 1] \leq \mathsf{negl}(\lambda)$*), where the experiments* $\mathsf{Exp}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa}$ *and* $\mathsf{Exp}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa}$ *are defined in Fig. 16 and Fig. 17 respectively.*

H.2 Generic Construction of AE from QA-HPS and QA-NIZK

In this subsection, we present a generic construction of $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv&Auth secure AE. The underlying building blocks are as follows.

- Two language distributions $\mathscr{L}$ and $\mathscr{L}_0$, both of which have hard SMPs.
- A $\mathsf{QAHPS} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{Pub}, \mathsf{Priv})$ for both $\mathscr{L}$ and $\mathscr{L}_0$, whose hashing key space is $\mathcal{SK}$, projection key space is $\mathcal{PK}$ and hash value space is $\mathcal{HV}$. We require $\mathcal{HV}$ to be an (additive) group. We stress that QAHPS is not required to be publicly-verifiable.

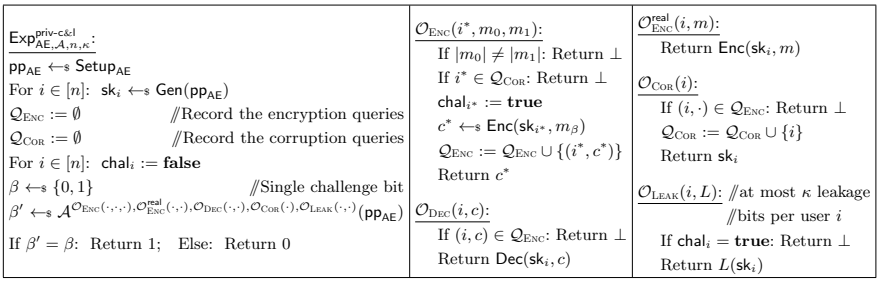

Fig. 16. The $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv security experiment $\mathsf{Exp}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa}$ for AE. We note that besides challenge ciphertexts return by $\mathcal{O}_{\mathrm{ENC}}$, $\mathcal{A}$ can also obtain honestly generated ciphertexts via $\mathcal{O}^{\mathsf{real}}_{\mathrm{ENC}}$.

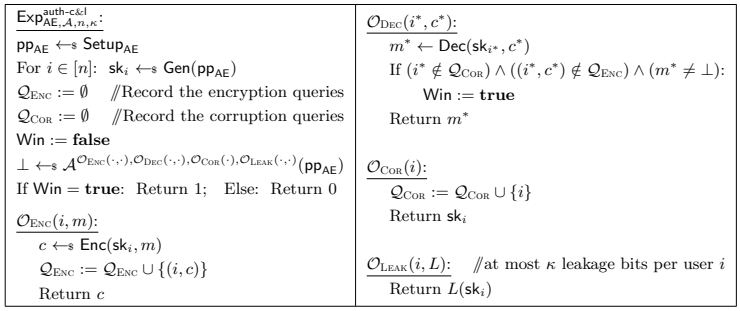

Fig. 17. The $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth security experiment $\mathsf{Exp}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa}$ for AE.

- A tag-based $\mathsf{QANIZK} = (\mathsf{Setup}_{\mathsf{NIZK}}, \mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{Sim})$ for $\mathscr{L}$, whose tag space is $\mathcal{T}$.
- A family of collision-resistant hash functions $\mathcal{H} = \{H : \mathcal{PK} \times \mathcal{HV} \longrightarrow \mathcal{T}\}$.

Our generic construction of $\mathsf{AE} = (\mathsf{Setup}_{\mathsf{AE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is shown in Fig. 18.

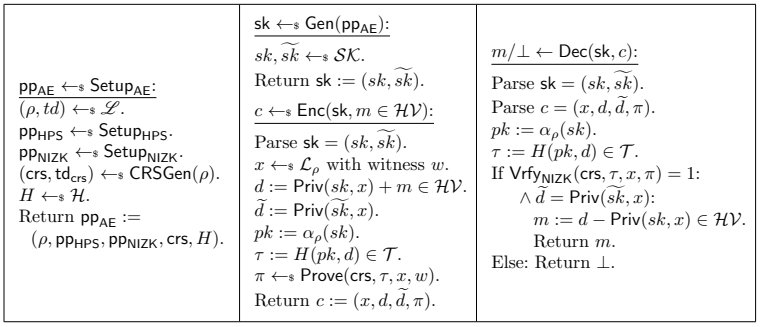

Fig. 18. Generic construction of $\mathsf{AE} = (\mathsf{Setup}_{\mathsf{AE}}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ from QAHPS, tag-based QANIZK and $\mathcal{H}$. The message space is $\mathcal{M} := \mathcal{HV}$.

Correctness of AE follows directly from the perfect completeness of QANIZK. Next, we show its $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv and $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth security.

Theorem 6 ($\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv **Security of** AE). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii)* QAHPS *is a QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having PK-diversity, and supporting both κ-LR-$\langle\mathscr{L}, \mathscr{L}_0\rangle$-key-switching and $\mathscr{L}_0$-multi-key-multi-extracting, (iii)* QANIZK *is a tag-based QA-NIZK for $\mathscr{L}$, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed* AE *scheme in Fig. 18 is* $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv *secure under κ bits leakage per user.*

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

$$\begin{aligned}\mathsf{Adv}^{\mathsf{priv\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa}(\lambda) \le\ & \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda) + 2 \cdot \mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{B}_3,n,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_4,Q_e}(\lambda) \\ & + \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_5}(\lambda) + \mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{QAHPS},\mathcal{B}_6,n,Q_e}(\lambda) + \tfrac{n(n-1)}{2} \cdot \boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) + 2n \cdot \boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L},\mathscr{L}_0\rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS},\mathcal{B}_7,\kappa}(\lambda).\end{aligned}$$

Theorem 7 ($\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth **Security of** AE). *Assume that (i) $\mathscr{L}$ and $\mathscr{L}_0$ have hard SMPs, (ii)* QAHPS *is a QA-HPS for both $\mathscr{L}$ and $\mathscr{L}_0$, having PK-diversity and supporting κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting, (iii)* QANIZK *is a tag-based QA-NIZK for $\mathscr{L}$, satisfying both perfect zero-knowledge and unbounded simulation-soundness, (iv) $\mathcal{H}$ is collision-resistant. Then the proposed* AE *scheme in Fig. 18 is* $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth *secure under κ bits leakage per user.*

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 $\mathcal{B}_1, \cdots, \mathcal{B}_5$, such that $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_4) \approx \mathbf{T}(\mathcal{A}) + (n + Q_e + Q_d) \cdot \mathsf{poly}(\lambda)$, with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{auth\text{-}c\&l}}_{\mathsf{AE},\mathcal{A},n,\kappa}(\lambda) \leq\ & \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{B}_2,Q_e}(\lambda) + \mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{B}_3,Q_e}(\lambda) \\ & + \mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK},\mathcal{B}_4}(\lambda) + \tfrac{n(n-1)}{2} \cdot \boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}}(\lambda) + nQ_d \cdot \boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{QAHPS},\mathcal{B}_5,\kappa}(\lambda).\end{aligned}$$

Remark 10 (On the Tightness of AE's $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv&Auth Security). According to Theorem 6 and Theorem 7, AE has both tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv and tight $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth security, as long as the multi-fold SMPs related to $\mathscr{L}$ and $\mathscr{L}_0$ and the multi-language multi-fold SMP related to $\mathscr{L}_0$ have tight reductions, QAHPS has a tight $\mathscr{L}_0$-multi-key-multi-extracting property and QANIZK has a tight USS.

The proofs of Theorem 6 and Theorem 7 for AE almost verbatim follow the proofs of Theorem 3 and Theorem 4 for SC shown in Appendix F.2. Hence we omit them.

I Instantiations of PV-QA-HPS, QA-HPS and QA-NIZK

In this section, we present the instantiations of the building blocks PV-QA-HPS, QA-HPS and QA-NIZK needed in our generic constructions from the matrix DDH (MDDH) assumptions over asymmetric pairing groups.

More precisely, in Appendix I.1, we recall the definitions of asymmetric pairing groups and MDDH assumptions. In Appendix I.2, we present the instantiations of the language distributions $\mathscr{L}$ and $\mathscr{L}_0$, whose (multi-language) multi-fold SMPs have tight reductions to the MDDH assumptions. Then we give concrete instantiations of Publicly-Verifiable QA-HPS in Appendix I.3 and QA-HPS in Appendix I.4 from the MDDH assumptions. Finally, in Appendix I.5, we instantiate tag-based QA-NIZK with a tag-base variant of the QA-NIZK scheme proposed in [1] that has tight USS based on the MDDH assumptions.

I.1 Pairing Groups and MDDH Assumptions

Let PGGen be a PPT algorithm outputting a description of pairing group $\mathsf{gpar} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, P_1, P_2, P_T)$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are additive cyclic groups of order p, p is a prime number of bit-length at least λ, $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$ is a non-degenerated bilinear pairing, and P_1, P_2, P_T are generators of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, respectively, with $P_T := e(P_1, P_2)$. We assume that the operations in $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ and the pairing e are efficiently computable. We consider *Type-III asymmetric pairing group*, where $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is no efficient homomorphism between them. We require gpar to be an implicit input of other algorithms.

We use implicit representation of group elements as in [17]. For $s \in \{1, 2, T\}$ and $a \in \mathbb{Z}_p$, denote by $[a]_s = aP_s \in \mathbb{G}_s$ as the implicit representation of a in

$\mathbb{G}_s$. Similarly, for a matrix $\mathbf{A} = (a_{i,j}) \in \mathbb{Z}_p^{n\times m}$ we define $[\mathbf{A}]_s$ as the implicit representation of $\mathbf{A}$ in $\mathbb{G}_s$. $\mathsf{Span}(\mathbf{A}) := \{\mathbf{Ar} | \mathbf{r} \in \mathbb{Z}_p^m\} \subseteq \mathbb{Z}_p^n$ denotes the linear span of $\mathbf{A}$, and similarly $\mathsf{Span}([\mathbf{A}]_s) := \{[\mathbf{Ar}]_s | \mathbf{r} \in \mathbb{Z}_p^m\} \subseteq \mathbb{G}_s^n$. $\mathsf{Ker}(\mathbf{A}) := \{\mathbf{a} | \mathbf{a}^\top \mathbf{A} = \mathbf{0}\} \subseteq \mathbb{Z}_p^n$ denotes the left kernal space of $\mathbf{A}$, and similarly $\mathsf{Ker}([\mathbf{A}]_s) := \{\mathbf{a} | \mathbf{a}^\top [\mathbf{A}]_s = [\mathbf{0}]_s\} \subseteq \mathbb{Z}_p^n$. Note that given $\mathbf{A}$, $[\mathbf{B}]_s$, $[\mathbf{C}]_s$ and $\mathbf{D}$ with matching dimensions, one can efficiently compute $[\mathbf{AB}]_s$, $[\mathbf{B}+\mathbf{C}]_s$, $[\mathbf{CD}]_s$, and given $[\mathbf{A}]_1$ and $[\mathbf{B}]_2$, one can efficiently compute $[\mathbf{AB}]_T := e([\mathbf{A}]_1, [\mathbf{B}]_2)$.

Let $\ell, k \in \mathbb{N}$ be integers with $\ell > k$. A probabilistic distribution $\mathcal{D}_{\ell,k}$ is called a *matrix distribution*, if it outputs matrices in $\mathbb{Z}_p^{\ell\times k}$ of full rank k in polynomial time. Without loss of generality, we assume that the first k rows of $\mathbf{A} \leftarrow_\$ \mathcal{D}_{\ell,k}$ form an invertible matrix. Let $\mathcal{D}_k := \mathcal{D}_{k+1,k}$. Denote by $\mathcal{U}_{\ell,k}$ the *uniform distribution* over all matrices in $\mathbb{Z}_p^{\ell\times k}$. Let $\mathcal{U}_k := \mathcal{U}_{k+1,k}$.

In the following, we recall the Matrix DDH (MDDH), Q-fold MDDH and Kernal Matrix DH (KerMDH) assumptions, parameterized by a matrix distribution $\mathcal{D}_{\ell,k}$, as well as the random self-reducibility of the MDDH assumptions. The advantage functions for an adversary $\mathcal{A}$ against the $\mathcal{D}_{\ell,k}$-MDDH, Q-fold $\mathcal{D}_{\ell,k}$-MDDH and $\mathcal{D}_{\ell,k}$-KerMDH assumptions over group $\mathbb{G}_s$ (which is $\mathbb{G}_1$ or $\mathbb{G}_2$) are denoted by $\mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda)$, $\mathsf{Adv}^{Q\text{-}\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda)$ and $\mathsf{Adv}^{\mathsf{kmdh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda)$, respectively.

MDDH Assumptions. The $\mathcal{D}_{\ell,k}$-Matrix DDH ($\mathcal{D}_{\ell,k}$-MDDH) problem over group $\mathbb{G}_s$ is to distinguish the two distributions $([\mathbf{A}]_s, [\mathbf{Aw}]_s)$ and $([\mathbf{A}]_s, [\mathbf{u}]_s)$, where $\mathbf{A} \leftarrow_\$ \mathcal{D}_{\ell,k}$, $\mathbf{w} \leftarrow_\$ \mathbb{Z}_p^k$ and $\mathbf{u} \leftarrow_\$ \mathbb{Z}_p^\ell$.

Definition 26 ($\mathcal{D}_{\ell,k}$-MDDH Assumption). *Let $s \in \{1,2\}$. The $\mathcal{D}_{\ell,k}$-MDDH assumption holds over group $\mathbb{G}_s$, if for any PPT adversary $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda) := \big|\Pr[\mathcal{A}([\mathbf{A}]_s, [\mathbf{Aw}]_s) = 1] - \Pr[\mathcal{A}([\mathbf{A}]_s, [\mathbf{u}]_s) = 1]\big| \leq \mathsf{negl}(\lambda)$, where the probability is over $\mathbf{A} \leftarrow_\$ \mathcal{D}_{\ell,k}$, $\mathbf{w} \leftarrow_\$ \mathbb{Z}_p^k$ and $\mathbf{u} \leftarrow_\$ \mathbb{Z}_p^\ell$.*

MDDH assumption covers many well-studied assumptions, such as the DDH and the k-Linear (k-LIN) assumptions, by specifying the matrix distribution as $\mathcal{LIN}_1$ and $\mathcal{LIN}_k$ respectively [17], where $\mathcal{LIN}_k$: $\mathbf{A} = \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_k \\ 1 & \cdots & 1 \end{pmatrix} \in \mathbb{Z}_p^{(k+1)\times k}$.

Several relations among MDDH assumptions parameterized by different matrix distributions were established in [17, 18].

Lemma 1 ($\mathcal{D}_{\ell,k}$-MDDH $\Rightarrow$ $\mathcal{U}_k$-MDDH [17] $\Rightarrow$ $\mathcal{U}_{\ell',k}$-MDDH [18]). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_k,\mathbb{G}_s,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{B}}(\lambda)$.*

For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_{\ell',k},\mathbb{G}_s,\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_k,\mathbb{G}_s,\mathcal{B}}(\lambda)$.

Consequently, for any $\ell > k$, $\mathcal{U}_{\ell,k}$-MDDH assumption is tightly implied by the k-LIN assumption (i.e., $\mathcal{LIN}_k$-MDDH).

For $Q \geq 1$, consider the Q-fold $\mathcal{D}_{\ell,k}$-MDDH problem over group $\mathbb{G}_s$, which is to distinguish two distributions $([\mathbf{A}]_s, [\mathbf{AW}]_s)$ and $([\mathbf{A}]_s, [\mathbf{U}]_s)$, where $\mathbf{A} \leftarrow_\$ \mathcal{D}_{\ell,k}$,

$\mathbf{W} \leftarrow_\$ \mathbb{Z}_p^{k\times Q}$ and $\mathbf{U} \leftarrow_\$ \mathbb{Z}_p^{\ell\times Q}$. The distinguishing advantage of an adversary $\mathcal{A}$ is denoted by $\mathsf{Adv}^{Q\text{-mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda) := |\Pr[\mathcal{A}([\mathbf{A}]_s, [\mathbf{AW}]_s) = 1] - \Pr[\mathcal{A}([\mathbf{A}]_s, [\mathbf{U}]_s) = 1]|$, where $\mathbf{A} \leftarrow_\$ \mathcal{D}_{\ell,k}$, $\mathbf{W} \leftarrow_\$ \mathbb{Z}_p^{k\times Q}$ and $\mathbf{U} \leftarrow_\$ \mathbb{Z}_p^{\ell\times Q}$. The Q-fold $\mathcal{D}_{\ell,k}$-MDDH assumption over $\mathbb{G}_s$ assumes that $\mathsf{Adv}^{Q\text{-mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda) \le \mathsf{negl}(\lambda)$ for any PPT $\mathcal{A}$.

$\mathcal{D}_{\ell,k}$-MDDH problem is random self-reducible [17], namely Q-fold and (1-fold) $\mathcal{D}_{\ell,k}$-MDDH problems can be tightly reduced to each other. In particular, for the uniform distribution $\mathcal{U}_{\ell,k}$, the reduction is even tighter.

Lemma 2 (Random Self-Reducibility of MDDH [17]). *Let $Q > \ell - k$. 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 $\mathsf{Adv}^{Q\text{-mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda) \le (\ell - k) \cdot \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{B}}(\lambda) + 1/(p-1)$.*

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 $\mathsf{Adv}^{Q\text{-mddh}}_{\mathcal{U}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_{\ell,k},\mathbb{G}_s,\mathcal{B}}(\lambda) + 1/(p-1)$.

We also define the $\mathcal{D}_{\ell,k}$-Kernal Matrix DH ($\mathcal{D}_{\ell,k}$-KerMDH) assumption according to [33] which is a natural search variant of the $\mathcal{D}_{\ell,k}$-MDDH assumption.

Definition 27 ($\mathcal{D}_{\ell,k}$-KerMDH Assumption). *Let $s \in \{1,2\}$. The $\mathcal{D}_{\ell,k}$-KerMDH assumption holds over group $\mathbb{G}_s$, if for any PPT adversary $\mathcal{A}$, it holds that $\mathsf{Adv}^{\mathsf{kmdh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda) := \Pr\left[[\mathbf{x}]_{3-s} \in \mathbb{G}_{3-s}^{\ell} \leftarrow_\$ \mathcal{A}([\mathbf{A}]_s) : \mathbf{x}^\top\mathbf{A} = \mathbf{0} \wedge \mathbf{x} \neq \mathbf{0}\right] \le \mathsf{negl}(\lambda)$, where the probability is over $\mathbf{A} \leftarrow_\$ \mathcal{D}_{\ell,k}$.*

The following lemma shows that the $\mathcal{D}_{\ell,k}$-KerMDH assumption is tightly implied by the $\mathcal{D}_{\ell,k}$-MDDH assumption, since one can use a non-zero $[\mathbf{x}]_{3-s}$ satisfying $\mathbf{x}^\top\mathbf{A} = \mathbf{0}$ to test membership in $\mathsf{Span}([\mathbf{A}]_s)$.

Lemma 3 ($\mathcal{D}_{\ell,k}$-MDDH $\Rightarrow$ $\mathcal{D}_{\ell,k}$-KerMDH [33]). *For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A})$ and $\mathsf{Adv}^{\mathsf{kmdh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}(\lambda) \le \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{B}}(\lambda) + 1/(p-1)$.*

I.2 Instantiations of Language Distribution for Linear Subspaces

Let $\mathsf{gpar} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, P_1, P_2, P_T)$ be a description of asymmetric pairing group. For any matrix distribution $\mathcal{D}_{\ell,k}$, it naturally gives rise to a language distribution $\mathscr{L}_{\mathcal{D}_{\ell,k}}$ for linear subspaces over group $\mathbb{G}_1$:

- $\mathscr{L}_{\mathcal{D}_{\ell,k}}$ invokes $\mathbf{A} \leftarrow_\$ \mathcal{D}_{\ell,k}$ and outputs $(\rho := [\mathbf{A}]_1, td := \mathbf{A})$.

The language parameter $\rho = [\mathbf{A}]_1 \in \mathbb{G}_1^{\ell\times k}$ defines a linear subspace language

$$\mathcal{L}_\rho := \mathsf{Span}([\mathbf{A}]_1)\backslash\{[\mathbf{0}]_1\} = \left\{ [\mathbf{c}]_1 \,\middle|\, \exists\, \mathbf{w} \in \mathbb{Z}_p^k \setminus \{\mathbf{0}\},\ \text{s.t. } [\mathbf{c}]_1 = [\mathbf{Aw}]_1 \right\}^6$$

with universe $\mathcal{X} := \mathbb{G}_1^\ell \setminus \{[\mathbf{0}]_1\}$. The trapdoor $td = \mathbf{A}$ can be used to decide whether an instance $[\mathbf{c}]_1$ is in $\mathcal{L}_\rho$ efficiently: one can first compute a basis of the left kernel space of $\mathbf{A}$, namely $\mathbf{A}^\perp \in \mathbb{Z}_p^{\ell\times(\ell-k)}$ satisfying $(\mathbf{A}^\perp)^\top \cdot \mathbf{A} = \mathbf{0}$, then check whether $(\mathbf{A}^\perp)^\top \cdot [\mathbf{c}]_1 = [\mathbf{0}]_1$ holds.

Instantiations of $\mathscr{L}$ and $\mathscr{L}_0$. Let $\ell \geq 2k+1$. Let $\mathcal{D}_{\ell,k}$ be an (arbitrary) matrix distribution and $\mathcal{U}_{\ell,k}$ the uniform distribution. We designate the language distributions $\mathscr{L}$ and $\mathscr{L}_0$ as follows.

- $\mathscr{L} := \mathscr{L}_{\mathcal{D}_{\ell,k}}$, which invokes $\mathbf{A} \leftarrow_\$ \mathcal{D}_{\ell,k}$ and outputs $(\rho = [\mathbf{A}]_1, td = \mathbf{A})$.
- $\mathscr{L}_0 := \mathscr{L}_{\mathcal{U}_{\ell,k}}$, which invokes $\mathbf{A}_0 \leftarrow_\$ \mathcal{U}_{\ell,k}$ and outputs $(\rho_0 = [\mathbf{A}_0]_1, td_0 = \mathbf{A}_0)$.

Clearly, the (multi-fold) SMP related to $\mathscr{L}$ (resp., $\mathscr{L}_0$) corresponds to the (multi-fold) $\mathcal{D}_{\ell,k}$-MDDH (resp., $\mathcal{U}_{\ell,k}$-MDDH) assumptions over $\mathbb{G}_1$. By the random self-reducibility of $\mathcal{D}_{\ell,k}$-MDDH and $\mathcal{U}_{\ell,k}$-MDDH (cf. Lemma 2), we have the following lemma.

Lemma 4 ($\mathcal{D}_{\ell,k}/\mathcal{U}_{\ell,k}$-MDDH $\Rightarrow$ Multi-fold SMP related to $\mathscr{L}/\mathscr{L}_0$). *Let $Q > \ell - k$. 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 $\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L},\mathcal{A},Q}(\lambda) \leq (\ell-k)\cdot\mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{\ell,k},\mathbb{G}_1,\mathcal{B}}(\lambda) + 2/(p-1)$. 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 $\mathsf{Adv}^{\mathsf{msmp}}_{\mathscr{L}_0,\mathcal{A},Q}(\lambda) \leq \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_{\ell,k},\mathbb{G}_1,\mathcal{B}}(\lambda) + 2/(p-1)$.*

Next, we show that the multi-language multi-fold SMP related to $\mathscr{L}_0$ can be tightly reduced to the $\mathcal{U}_k$-MDDH assumption.

Lemma 5 ($\mathcal{U}_k$-MDDH $\Rightarrow$ Multi-language multi-fold SMP related to $\mathscr{L}_0$). *Let $nQ > n\ell - k$. For any adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + nQ\cdot\mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and $\mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{A},n,Q}(\lambda) \leq \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_k,\mathbb{G}_1,\mathcal{B}}(\lambda) + 2/(p-1)$.*

Proof of Lemma 5. Firstly, we construct an adversary $\mathcal{B}'$ against the nQ-fold $\mathcal{U}_{n\ell,k}$-MDDH over $\mathbb{G}_1$, so that $\mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{A},n,Q}(\lambda) \leq \mathsf{Adv}^{nQ\text{-}\mathsf{mddh}}_{\mathcal{U}_{n\ell,k},\mathbb{G}_1,\mathcal{B}'}(\lambda)$. Then by the random self-reducibility of $\mathcal{U}_{n\ell,k}$-MDDH (i.e., Lemma 2) and $\mathcal{U}_k$-MDDH $\Rightarrow$ $\mathcal{U}_{n\ell,k}$-MDDH (i.e., Lemma 1), Lemma 5 follows.

Given a challenge $([\mathbf{B}]_1, [\mathbf{U}]_1)$, $\mathcal{B}'$ wants to distinguish $[\mathbf{U}]_1 = [\mathbf{BW}]_1$ from $[\mathbf{U}]_1 \leftarrow_\$ \mathbb{G}_1^{(n\ell)\times(nQ)}$, where $\mathbf{B} \leftarrow_\$ \mathcal{U}_{n\ell,k}$ and $\mathbf{W} \leftarrow_\$ \mathbb{Z}_p^{k\times(nQ)}$.

Let us fix some *notations* used in this proof. We parse $\mathbf{B} \in \mathbb{Z}_p^{(n\ell)\times k}$ as $\mathbf{B} = \begin{pmatrix} \mathbf{B}^{(1)} \\ \mathbf{B}^{(2)} \\ \vdots \\ \mathbf{B}^{(n)} \end{pmatrix}$ with each $\mathbf{B}^{(i)} \in \mathbb{Z}_p^{\ell\times k}$, parse $\mathbf{W} \in \mathbb{Z}_p^{k\times(nQ)}$ as $\mathbf{W} = \left(\mathbf{W}_{(1)}\ \mathbf{W}_{(2)} \cdots \mathbf{W}_{(n)}\right)$

[6] For technical reasons (more precisely, for the κ-LR-OT-extracting property of the QA-HPS schemes constructed later), the zero vector $[\mathbf{0}]_1$ must be excluded from $\mathcal{L}_\rho$ and $\mathcal{X}$. For the sake of simplicity, we forgo making this explicit in the sequel.

with each $\mathbf{W}_{(j)} \in \mathbb{Z}_p^{k\times Q}$, and parse $\mathbf{U} \in \mathbb{Z}_p^{(n\ell)\times(nQ)}$ as $\mathbf{U} = \begin{pmatrix} \mathbf{U}_{(1)}^{(1)} & \mathbf{U}_{(2)}^{(1)} & \cdots & \mathbf{U}_{(n)}^{(1)} \\ \mathbf{U}_{(1)}^{(2)} & \mathbf{U}_{(2)}^{(2)} & \cdots & \mathbf{U}_{(n)}^{(2)} \\ \vdots & \vdots & & \vdots \\ \mathbf{U}_{(1)}^{(n)} & \mathbf{U}_{(2)}^{(n)} & \cdots & \mathbf{U}_{(n)}^{(n)} \end{pmatrix}$ with each $\mathbf{U}_{(j)}^{(i)} \in \mathbb{Z}_p^{\ell\times Q}$. We also denote by $\mathbf{W}_{(j)|m} \in \mathbb{Z}_p^k$ the m-th column of $\mathbf{W}_{(j)}$ and by $\mathbf{U}_{(j)|m}^{(i)} \in \mathbb{Z}_p^\ell$ the m-th column of $\mathbf{U}_{(j)}^{(i)}$, where $m \in [Q]$.

$\mathcal{B}'$ is constructed by invoking $\mathcal{A}$ as follows. $\mathcal{B}'$ sets $\rho^{(i)} := [\mathbf{B}^{(i)}]_1 \in \mathbb{G}_1^{\ell\times k}$ for each $i \in [n]$, and sets $x_m^{(i)} := [\mathbf{U}_{(i)|m}^{(i)}]_1 \in \mathbb{G}_1^\ell$ for each $i \in [n]$ and $m \in [Q]$. Then $\mathcal{B}'$ invokes $\mathcal{A}(\{\rho^{(i)}, \{x_m^{(i)}\}_{m\in[Q]}\}_{i\in[n]})$, and returns to its own challenger whatever $\mathcal{A}$ outputs.

Next, we analyze the advantage of $\mathcal{B}'$.

- Since $\mathbf{B} \leftarrow_\$ \mathcal{U}_{n\ell,k}$, each $\mathbf{B}^{(i)}$ is independently and uniformly random over $\mathbb{Z}_p^{\ell\times k}$. Thus, each $\rho^{(i)} = [\mathbf{B}^{(i)}]_1$ exactly follows the language parameter distribution output by $\mathscr{L}_{\mathcal{U}_{\ell,k}} = \mathscr{L}_0$.
- In the case that $[\mathbf{U}]_1 = [\mathbf{BW}]_1$, for each $i \in [n]$, we have $[\mathbf{U}_{(i)}^{(i)}]_1 = [\mathbf{B}^{(i)}\mathbf{W}_{(i)}]_1 \in \mathbb{G}_1^{\ell\times Q}$, and for each $m \in [Q]$, we have $x_m^{(i)} = [\mathbf{U}_{(i)|m}^{(i)}]_1 = [\mathbf{B}^{(i)}\mathbf{W}_{(i)|m}]_1 \in \mathbb{G}_1^\ell$, which is independently and uniformly random over $\mathsf{Span}([\mathbf{B}^{(i)}]_1) = \mathcal{L}_{\rho^{(i)}}$, due to the randomness of $\mathbf{W}_{(i)|m}$.
- In the case that $[\mathbf{U}]_1 \leftarrow_\$ \mathbb{G}_1^{(n\ell)\times(nQ)}$, for each $i \in [n]$, $\mathbf{U}_{(i)}^{(i)}$ is uniformly random over $\mathbb{G}_1^{\ell\times Q}$, and for each $m \in [Q]$, $x_m^{(i)} = [\mathbf{U}_{(i)|m}^{(i)}]_1$ is uniformly random over $\mathbb{G}_1^\ell = \mathcal{X}$.

Therefore, $\mathcal{B}'$ successfully distinguishes $[\mathbf{U}]_1 = [\mathbf{BW}]_1$ from $[\mathbf{U}]_1 \leftarrow_\$ \mathbb{G}_1^{(n\ell)\times(nQ)}$, as long as $\mathcal{A}$ solves the multi-language multi-fold SMP related to $\mathscr{L}_0$. Consequently, we get $\mathsf{Adv}^{\mathsf{ml\text{-}msmp}}_{\mathscr{L}_0,\mathcal{A},n,Q}(\lambda) \le \mathsf{Adv}^{nQ\text{-}\mathsf{mddh}}_{\mathcal{U}_{n\ell,k},\mathbb{G}_1,\mathcal{B}'}(\lambda)$, as desired. This completes the proof of Lemma 5. □

I.3 Instantiation of PV-QA-HPS from MDDH

In this subsection, we present a PV-QA-HPS scheme $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$ based on the MDDH assumptions over asymmetric pairing groups.

Let $\mathscr{L} = \mathscr{L}_{\mathcal{D}_{\ell,k}}$ and $\mathscr{L}_0 = \mathscr{L}_{\mathcal{U}_{\ell,k}}$ be the language distributions specified in Appendix I.2. Formally, the MDDH-based $\mathsf{PVQAHPS}_{\mathsf{MDDH}} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \nu, \mathsf{Pub}, \mathsf{Priv}, \mathsf{Vrfy}_{\mathsf{HPS}})$ for $\mathscr{L}$ is presented in Fig. 19. It is straightforward to check the correctness of $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$ for both $\mathscr{L}$ and $\mathscr{L}_0$ and the verification completeness of $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$.

Through the following theorems, we prove the verification soundness, VK-diversity, and κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting of $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$, as needed for the strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMA security of our SIG in Subsect. 5.2 (cf. Theorem 1) and the $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv and $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth security of our SC in Appendix F.2 (cf. Theorem 3 and Theorem 4).

$\underline{\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}}$:
$\mathsf{gpar} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, P_1, P_2, P_T) \leftarrow_{\$} \mathsf{PGGen}$.
$\mathbf{B} \leftarrow_{\$} \mathcal{D}_k$, where $\mathbf{B} \in \mathbb{Z}_p^{(k+1)\times k}$.
Return $\mathsf{pp}_{\mathsf{HPS}} := (\mathsf{gpar}, [\mathbf{B}]_2)$, which implicitly defines
$(\mathcal{SK} := \mathbb{Z}_p^{(k+1)\times\ell},\ \mathcal{HV} := \mathbb{G}_1^{k+1},\ \mathcal{VK} := \mathbb{G}_2^{\ell\times k},\ \Lambda_{(\cdot)})$,
where $\Lambda_{sk}([\mathbf{c}]_1) := \mathbf{K} \cdot [\mathbf{c}]_1$ for $sk = \mathbf{K} \in \mathcal{SK}$ and $[\mathbf{c}]_1 \in \mathcal{X}$.

$\underline{pk_\rho \leftarrow \alpha_\rho(sk = \mathbf{K} \in \mathcal{SK}),\ \text{where } \rho = [\mathbf{A}]_1 \in \mathbb{G}_1^{\ell\times k}}$:
Return $pk_\rho := [\mathbf{P}]_1 := \mathbf{K} \cdot [\mathbf{A}]_1 \in \mathbb{G}_1^{(k+1)\times k}$.

$\underline{vk \leftarrow \nu(sk = \mathbf{K} \in \mathcal{SK})}$:
Return $vk := \mathbf{K}^\top \cdot [\mathbf{B}]_2 \in \mathbb{G}_2^{\ell\times k}$.

$\underline{[\boldsymbol{hv}]_1 \leftarrow \mathsf{Pub}(pk_\rho = [\mathbf{P}]_1, [\mathbf{c}]_1, \mathbf{w} \in \mathbb{Z}_p^k),\ \text{where } [\mathbf{c}]_1 \in \mathcal{L}_\rho \text{ for } \rho = [\mathbf{A}]}$:
Return $[\boldsymbol{hv}]_1 := [\mathbf{P}]_1 \cdot \mathbf{w} \in \mathbb{G}_1^{k+1}$.

$\underline{[\boldsymbol{hv}]_1 \leftarrow \mathsf{Priv}(sk = \mathbf{K} \in \mathcal{SK}, [\mathbf{c}]_1 \in \mathcal{X})}$:
Return $[\boldsymbol{hv}]_1 := \mathbf{K} \cdot [\mathbf{c}]_1 \in \mathbb{G}_1^{k+1}$.

$\underline{0/1 \leftarrow \mathsf{Vrfy}_{\mathsf{HPS}}(vk, [\mathbf{c}]_1 \in \mathcal{X}, [\boldsymbol{hv}]_1)}$:
Parse $vk = [\mathbf{K}^\top\mathbf{B}]_2 \in \mathbb{G}_2^{\ell\times k}$.
If $e([\mathbf{c}^\top]_1, [\mathbf{K}^\top\mathbf{B}]_2) = e([\boldsymbol{hv}^\top]_1, [\mathbf{B}]_2)$:
Return 1;
Else: Return 0.

Fig. 19. The MDDH-based publicly-verifiable QA-HPS scheme $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$.

Theorem 8 (Tight Verification Soundness of $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$). *If the $\mathcal{D}_k$-KerMDH assumption holds over $\mathbb{G}_2$, then the proposed $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$ in Fig. 19 has verification soundness. Concretely, for any adversary $\mathcal{A}$ and any n, there exists an adversary $\mathcal{B}$, such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + n \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and $\mathsf{Adv}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS}_{\mathsf{MDDH}},\mathcal{A},n}(\lambda) \le \mathsf{Adv}^{\mathsf{kmdh}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}}(\lambda)$.*

Proof of Theorem 8. We construct an adversary $\mathcal{B}$ against the $\mathcal{D}_k$-KerMDH over $\mathbb{G}_2$ by simulating the experiment $\mathsf{Exp}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{A},n}$ (defined in Fig. 5) for $\mathcal{A}$.

Given a challenge $[\mathbf{B}]_2$ where $\mathbf{B} \leftarrow_{\$} \mathcal{D}_k$, $\mathcal{B}$ is constructed as follows.

- $\mathcal{B}$ sets $\mathsf{pp}_{\mathsf{HPS}} := (\mathsf{gpar}, [\mathbf{B}]_2)$. For each $i \in [n]$, $\mathcal{B}$ samples $sk_i = \mathbf{K}_i \leftarrow_{\$} \mathbb{Z}_p^{(k+1)\times\ell}$ itself, and computes $vk_i := \mathbf{K}_i^\top \cdot [\mathbf{B}]_2$.
- Then $\mathcal{B}$ invokes $\mathcal{A}(\mathsf{pp}_{\mathsf{HPS}}, (sk_i, vk_i)_{i\in[n]})$ and obtains $(i^*, [\mathbf{c}^*]_1 \in \mathbb{G}_1^\ell, [\boldsymbol{hv}^*]_1 \in \mathbb{G}_1^{k+1})$ from $\mathcal{A}$.
- Finally, $\mathcal{B}$ computes $[\mathbf{x}]_1 := [\boldsymbol{hv}^* - \mathbf{K}_{i^*}\mathbf{c}^*]_1 \in \mathbb{G}_1^{k+1}$ from $[\boldsymbol{hv}^*]_1$, $\mathbf{K}_{i^*}$ and $[\mathbf{c}^*]_1$, and outputs $[\mathbf{x}]_1$ to its own challenger.

Clearly, $\mathcal{B}$ simulates the experiment $\mathsf{Exp}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{A},n}$ perfectly for $\mathcal{A}$.

Now we show that $\mathcal{B}$ succeeds, i.e., $\mathcal{B}$'s output $[\mathbf{x}]_1 := [\boldsymbol{hv}^* - \mathbf{K}_{i^*}\mathbf{c}^*]_1$ satisfies $\mathbf{x} \neq \mathbf{0}$ and $\mathbf{x}^\top\mathbf{B} = \mathbf{0}$, as long as $\mathcal{A}$ succeeds. Suppose that $\mathcal{A}$ succeeds in $\mathsf{Exp}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS},\mathcal{A},n}$, i.e., $[\boldsymbol{hv}^*]_1 \neq \Lambda_{sk_{i^*}}([\mathbf{c}^*]_1)$ but $\mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, [\mathbf{c}^*]_1, [\boldsymbol{hv}^*]_1) = 1$.

- The former $[\boldsymbol{hv}^*]_1 \neq \Lambda_{sk_{i^*}}([\mathbf{c}^*]_1) \iff [\boldsymbol{hv}^*]_1 \neq \mathbf{K}_{i^*}[\mathbf{c}^*]_1 \iff [\mathbf{x}]_1 = [\boldsymbol{hv}^* - \mathbf{K}_{i^*}\mathbf{c}^*]_1 \neq [\mathbf{0}]_1$, so $\mathbf{x}$ is non-zero.
- The latter $\mathsf{Vrfy}_{\mathsf{HPS}}(vk_{i^*}, [\mathbf{c}^*]_1, [\boldsymbol{hv}^*]_1) = 1 \iff e([\mathbf{c}^{*\top}]_1, [\mathbf{K}_{i^*}^\top\mathbf{B}]_2) = e([\boldsymbol{hv}^{*\top}]_1, [\mathbf{B}]_2) \iff e([\mathbf{x}^\top]_1, [\mathbf{B}]_2) = e([\boldsymbol{hv}^{*\top} - \mathbf{c}^{*\top}\mathbf{K}_{i^*}^\top]_1, [\mathbf{B}]_2) = [\mathbf{0}]_T$, so $\mathbf{x}^\top\mathbf{B} = \mathbf{0}$.

Thus, $\mathcal{B}$ successfully breaks the $\mathcal{D}_k$-KerMDH assumption. Consequently, we get $\mathsf{Adv}^{\mathsf{vrfy\text{-}snd}}_{\mathsf{PVQAHPS}_{\mathsf{MDDH}},\mathcal{A},n}(\lambda) \le \mathsf{Adv}^{\mathsf{kmdh}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}}(\lambda)$ and complete the proof. □

Theorem 9 (VK-Diversity of $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$). *The proposed $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$ in Fig. 19 has VK-diversity with $\epsilon^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}_{\mathsf{MDDH}}}(\lambda) = 1/p^{k\ell}$.*

Proof of Theorem 9. For $\mathbf{B} \leftarrow_{\$} \mathcal{D}_k$ chosen in $\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}$ and for $sk = \mathbf{K} \leftarrow_{\$} \mathbb{Z}_p^{(k+1)\times\ell}$ and $sk' = \mathbf{K}' \leftarrow_{\$} \mathbb{Z}_p^{(k+1)\times\ell}$, the event $\nu(sk) = \nu(sk') \iff \mathbf{K}^\top \cdot [\mathbf{B}]_2 = \mathbf{K}'^\top \cdot [\mathbf{B}]_2 \iff (\mathbf{K} - \mathbf{K}')^\top \cdot \mathbf{B} = \mathbf{0} \iff \forall i \in [\ell], \mathbf{K}_i - \mathbf{K}'_i \in \mathsf{Ker}(\mathbf{B})$, where $\mathbf{K}_i$ (resp. $\mathbf{K}'_i$) denotes the i-th column of $\mathbf{K}$ (resp. $\mathbf{K}'$). Since $\mathbf{B} \in \mathbb{Z}_p^{(k+1)\times k}$ output by $\mathcal{D}_k$ has rank k, the left kernal $\mathsf{Ker}(\mathbf{B})$ has rank 1. So the probability that each $\mathbf{K}_i - \mathbf{K}'_i \in \mathsf{Ker}(\mathbf{B})$ is $p^1/p^{k+1} = 1/p^k$, by the uniformity of $\mathbf{K}_i - \mathbf{K}'_i$ over $\mathbb{Z}_p^{k+1}$. Taking all $i \in [\ell]$ into account, this shows the VK-diversity of $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$ with $\epsilon^{\mathsf{vk\text{-}div}}_{\mathsf{PVQAHPS}_{\mathsf{MDDH}}}(\lambda) = 1/p^{k\ell}$. □

Before presenting the κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting of $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$, we define the *notations* of min-entropy and average min-entropy, and recall a useful lemma from [16]. Let X and Y be two random variables. The min-entropy of X is defined as $\mathbf{H}_\infty(X) := -\log(\max_x \Pr[X = x])$, and the average min-entropy of X conditioned on Y is defined as $\widetilde{\mathbf{H}}_\infty(X \,|\, Y) := -\log\big(\mathbb{E}_{y \leftarrow_{\$} Y}\big[\max_x \Pr[X = x \,|\, Y = y]\big]\big)$, where $\mathbb{E}$ denotes the mathematical expectation.

Lemma 6 ([16]). *Let X, Y, Z be three (possibly correlated) random variables. If Z has at most 2^κ possible values, then $\widetilde{\mathbf{H}}_\infty(X \,|\, Y, Z) \geq \widetilde{\mathbf{H}}_\infty(X \,|\, Y) - \kappa$.*

Theorem 10 (κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-Extracting of $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$). *Let $\ell \geq 2k + 1$ and $\kappa \leq \log p - \Omega(\lambda)$. The proposed $\mathsf{PVQAHPS}_{\mathsf{MDDH}}$ in Fig. 19 supports κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting with $\epsilon^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-}\mathsf{otext}}_{\mathsf{PVQAHPS}_{\mathsf{MDDH}},\mathcal{A},\kappa}(\lambda) \leq 2^{-\Omega(\lambda)}$ for any (unbounded) adversary $\mathcal{A}$.*

Proof of Theorem 10. Let $(\rho = [\mathbf{A}]_1 \in \mathbb{G}_1^{\ell\times k}, td) \leftarrow_{\$} \mathscr{L}$ and $(\rho_0 = [\mathbf{A}_0]_1 \in \mathbb{G}_1^{\ell\times k}, td_0) \leftarrow_{\$} \mathscr{L}_0$. With overwhelming probability $1 - 2^{-\Omega(\lambda)}$, the matrix $(\mathbf{A}, \mathbf{A}_0) \in \mathbb{Z}_p^{\ell\times 2k}$ is of full column rank, and in this case, $\mathsf{Span}([\mathbf{A}]_1) \cap \mathsf{Span}([\mathbf{A}_0]_1) = \{[\mathbf{0}]_1\}$. In the following analysis, we take it for granted.

Firstly, we prove the κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property in the case $\kappa = 0$, i.e., there is no leakage at all. For $sk = \mathbf{K} \leftarrow_{\$} \mathbb{Z}_p^{(k+1)\times\ell}$, we will show that in the presence of $\mathsf{pp}_{\mathsf{HPS}} = (\mathsf{gpar}, [\mathbf{B}]_2)$, $\rho_0 = [\mathbf{A}_0]_1$, $\rho = [\mathbf{A}]_1$, $\alpha_{\rho_0}(sk) = [\mathbf{K}\mathbf{A}_0]_1$ and $\nu(sk) = [\mathbf{K}^\top\mathbf{B}]_2$, the hash value $\Lambda_{sk}([\mathbf{c}^*]_1) = [\mathbf{K}\mathbf{c}^*]_1$ has entropy at least $\log p$ for any $[\mathbf{c}^*]_1 \in \mathcal{L}_\rho = \mathsf{Span}([\mathbf{A}]_1) \setminus \{[\mathbf{0}]_1\}$, i.e.,

$$\widetilde{\mathbf{H}}_\infty(\ [\mathbf{K}\mathbf{c}^*]_1 \mid \mathsf{gpar}, [\mathbf{B}]_2, [\mathbf{A}_0]_1, [\mathbf{A}]_1, [\mathbf{K}\mathbf{A}_0]_1, [\mathbf{K}^\top\mathbf{B}]_2\ ) \ \geq\ \log p. \tag{5}$$

Therefore, any (unbounded) adversary $\mathcal{A}$ is able to output $([\mathbf{c}^*]_1, [\boldsymbol{h}\boldsymbol{v}^*]_1)$ such that $[\mathbf{c}^*]_1 \in \mathcal{L}_\rho \ \wedge\ [\boldsymbol{h}\boldsymbol{v}^*]_1 = \Lambda_{sk}([\mathbf{c}^*]_1)$ holds with probability at most $1/p$.

We prove the high entropy of $\Lambda_{sk}([\mathbf{c}^*]_1) = [\mathbf{K}\mathbf{c}^*]_1$ (i.e., (5)) as follows. Let $\mathbf{a}_0^\perp \in \mathbb{Z}_p^\ell$ (resp. $\mathbf{b}^\perp \in \mathbb{Z}_p^{k+1}$) be an arbitrary non-zero vector in the left kernel space of $\mathbf{A}_0$ (resp. $\mathbf{B}$) such that $(\mathbf{a}_0^\perp)^\top\mathbf{A}_0 = \mathbf{0}$ (resp. $(\mathbf{b}^\perp)^\top\mathbf{B} = \mathbf{0}$) holds. For the convenience of our analysis, we sample $sk = \mathbf{K} \leftarrow_{\$} \mathbb{Z}_p^{(k+1)\times\ell}$ equivalently via

$$sk = \mathbf{K} := \widetilde{\mathbf{K}} + \mu \cdot \mathbf{b}^\perp \cdot (\mathbf{a}_0^\perp)^\top \in \mathbb{Z}_p^{(k+1)\times\ell},$$

where $\widetilde{\mathbf{K}} \leftarrow_{\$} \mathbb{Z}_p^{(k+1)\times\ell}$ and $\mu \leftarrow_{\$} \mathbb{Z}_p$. Consequently, we have

$$\alpha_{\rho_0}(sk) = [\mathbf{K}\mathbf{A}_0]_1 = [\widetilde{\mathbf{K}}\mathbf{A}_0]_1, \qquad \nu(sk) = [\mathbf{K}^\top\mathbf{B}]_2 = [\widetilde{\mathbf{K}}^\top\mathbf{B}]_2,$$

which may leak $\widetilde{\mathbf{K}}$, but the value of μ is completely hidden. Besides,

$$\Lambda_{sk}([\mathbf{c}^*]_1) = [\mathbf{K}\mathbf{c}^*]_1 = [\widetilde{\mathbf{K}}\mathbf{c}^*]_1 + \boxed{[\mu \cdot \mathbf{b}^\perp \cdot (\mathbf{a}_0^\perp)^\top \cdot \mathbf{c}^*]_1}.$$

By the facts $\mathsf{Span}([\mathbf{A}]_1) \cap \mathsf{Span}([\mathbf{A}_0]_1) = \{[\mathbf{0}]_1\}$ and $[\mathbf{c}^*]_1 \in \mathcal{L}_\rho = \mathsf{Span}([\mathbf{A}]_1) \setminus \{[\mathbf{0}]_1\}$, it follows that $[\mathbf{c}^*]_1 \notin \mathsf{Span}([\mathbf{A}_0]_1)$, and consequently, there always exists an $\mathbf{a}_0^\perp \in \mathbb{Z}_p^\ell$ such that $[(\mathbf{a}_0^\perp)^\top \cdot \mathbf{c}^*]_1 \neq [0]_1$ holds. As a result, conditioned on $\mathsf{pp}_{\mathsf{HPS}} = (\mathsf{gpar}, [\mathbf{B}]_2)$, $\rho_0 = [\mathbf{A}_0]_1$, $\rho = [\mathbf{A}]_1$, $\alpha_{\rho_0}(sk) = [\mathbf{K}\mathbf{A}_0]_1$ and $\nu(sk) = [\mathbf{K}^\top\mathbf{B}]_2$, the term $[\mu \cdot \mathbf{b}^\perp \cdot (\mathbf{a}_0^\perp)^\top \cdot \mathbf{c}^*]_1$ has entropy $\log p$ due to the randomness of μ, and so does $\Lambda_{sk}([\mathbf{c}^*]_1) = [\mathbf{K}\mathbf{c}^*]_1$. This finishes the proof of (5).

Next, we prove the κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting property for any $\kappa \leq \log p - \Omega(\lambda)$. In this case, the adversary can obtain at most κ bits leakage information about sk through oracle $\mathcal{O}_{\text{LEAK}}$. We note that $\mathcal{A}$ can query oracle $\mathcal{O}_{\text{LEAK}}$ adaptively and each time submit a leakage functions $L(\cdot)$ which may arbitrarily depend on all the information that $\mathcal{A}$ obtains. Nevertheless, we denote by $Leak^{\mathsf{all}}$ the overall leakage information, which is at most κ bits. By Lemma 6 and (5), it follows that even additionally in the presence of $Leak^{\mathsf{all}}$, the hash value $\Lambda_{sk}([\mathbf{c}^*]_1) = [\mathbf{K}\mathbf{c}^*]_1$ still has entropy at least $\log p - \kappa \geq \Omega(\lambda)$ for any $[\mathbf{c}^*]_1 \in \mathcal{L}_\rho = \mathsf{Span}([\mathbf{A}]_1) \setminus \{[\mathbf{0}]_1\}$, i.e.,

$$\begin{aligned}
&\widetilde{\mathbf{H}}_\infty(\ [\mathbf{K}\mathbf{c}^*]_1 \mid \mathsf{gpar}, [\mathbf{B}]_2, [\mathbf{A}_0]_1, [\mathbf{A}]_1, [\mathbf{K}\mathbf{A}_0]_1, [\mathbf{K}^\top\mathbf{B}]_2, \ Leak^{\mathsf{all}}\ ) \\
&\geq \widetilde{\mathbf{H}}_\infty(\ [\mathbf{K}\mathbf{c}^*]_1 \mid \mathsf{gpar}, [\mathbf{B}]_2, [\mathbf{A}_0]_1, [\mathbf{A}]_1, [\mathbf{K}\mathbf{A}_0]_1, [\mathbf{K}^\top\mathbf{B}]_2\ ) - \kappa \\
&\geq \log p - \kappa \geq \Omega(\lambda).
\end{aligned}$$

Therefore, $\mathcal{A}$ is able to output $([\mathbf{c}^*]_1, [\boldsymbol{h}\boldsymbol{v}^*]_1)$ such that $[\mathbf{c}^*]_1 \in \mathcal{L}_\rho \ \wedge \ [\boldsymbol{h}\boldsymbol{v}^*]_1 = \Lambda_{sk}([\mathbf{c}^*]_1)$ with probability at most $2^{-\Omega(\lambda)}$.

This completes the proof of κ-LR-$\langle\mathscr{L}_0, \mathscr{L}\rangle$-OT-extracting. □

I.4 Instantiation of QA-HPS from MDDH

In this subsection, we instantiate QA-HPS with (a slightly simplified variant of) the MDDH-based QA-HPS scheme $\mathsf{QAHPS}_{\mathsf{MDDH}}$ proposed in [22, Subsect. 5.3], which is in turn a generalization of the well-known DDH-based HPS scheme proposed by Cramer and Shoup in [13].

Let $\mathscr{L} = \mathscr{L}_{\mathcal{D}_{\ell,k}}$ and $\mathscr{L}_0 = \mathscr{L}_{\mathcal{U}_{\ell,k}}$ be the language distributions specified in Appendix I.2. Formally, we present the MDDH-based scheme $\mathsf{QAHPS}_{\mathsf{MDDH}} = (\mathsf{Setup}_{\mathsf{HPS}}, \alpha_{(\cdot)}, \mathsf{Pub}, \mathsf{Priv})$ for $\mathscr{L}$ in Fig. 20. It is straightforward to check the correctness of $\mathsf{QAHPS}_{\mathsf{MDDH}}$ for both $\mathscr{L}$ and $\mathscr{L}_0$.

Through the following theorems, we show the PK-diversity, κ-LR-$\langle\mathscr{L}, \mathscr{L}_0\rangle$-key-switching and $\mathscr{L}_0$-multi-key-multi-extracting of $\mathsf{QAHPS}_{\mathsf{MDDH}}$, as needed for the $\mathsf{MUMC}^{\mathsf{c\&l}}$-CCA security of our PKE in Subsect. 6.2 (cf. Theorem 2) and the $\mathsf{MUMC}^{\mathsf{c\&l}}$-Priv security of our AE in Appendix H.2 (cf. Theorem 6).

Theorem 11 (PK-Diversity of $\mathsf{QAHPS}_{\mathsf{MDDH}}$). *The proposed $\mathsf{QAHPS}_{\mathsf{MDDH}}$ in Fig. 20 has PK-diversity with $\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}_{\mathsf{MDDH}}}(\lambda) = 1/p^k$.*

$\underline{\mathsf{pp}_{\mathsf{HPS}} \leftarrow_{\$} \mathsf{Setup}_{\mathsf{HPS}}:}$
$\mathsf{gpar} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, P_1, P_2, P_T) \leftarrow_{\$} \mathsf{PGGen}$.
Return $\mathsf{pp}_{\mathsf{HPS}} := \mathsf{gpar}$, which implicitly defines
$(\mathcal{SK} := \mathbb{Z}_p^\ell,\ \mathcal{HV} := \mathbb{G}_1,\ \Lambda_{(\cdot)})$,
where $\Lambda_{sk}([\mathbf{c}]_1) := \mathbf{k}^\top \cdot [\mathbf{c}]_1 \in \mathbb{G}_1$ for $sk = \mathbf{k} \in \mathcal{SK}$ and $[\mathbf{c}]_1 \in \mathcal{X}$.

$\underline{pk_\rho \leftarrow \alpha_\rho(sk), \text{ where } \rho = [\mathbf{A}]_1 \in \mathbb{G}_1^{\ell \times k}:}$
Parse $sk = \mathbf{k} \in \mathbb{Z}_p^\ell$.
$[\mathbf{p}^\top]_1 := \mathbf{k}^\top \cdot [\mathbf{A}]_1 \in \mathbb{G}_1^{1 \times k}$.
Return $pk_\rho := [\mathbf{p}]_1$.

$\underline{[hv]_1 \leftarrow \mathsf{Pub}(pk_\rho, [\mathbf{c}]_1, \mathbf{w} \in \mathbb{Z}_p^k),}$
$\underline{\text{where } [\mathbf{c}]_1 \in \mathcal{L}_\rho \text{ for } \rho = [\mathbf{A}]:}$
Parse $pk_\rho = [\mathbf{p}]_1 \in \mathbb{G}_1^k$.
Return $[hv]_1 := [\mathbf{p}^\top]_1 \cdot \mathbf{w} \in \mathbb{G}_1$.

$\underline{[hv]_1 \leftarrow \mathsf{Priv}(sk, [\mathbf{c}]_1 \in \mathcal{X}):}$
Parse $sk = \mathbf{k} \in \mathbb{Z}_p^\ell$.
Return $[hv]_1 := \mathbf{k}^\top \cdot [\mathbf{c}]_1 \in \mathbb{G}_1$.

Fig. 20. The MDDH-based QA-HPS scheme $\mathsf{QAHPS}_{\mathsf{MDDH}}$.

Proof of Theorem 11. For $(\rho = [\mathbf{A}]_1, td = \mathbf{A}) \leftarrow_{\$} \mathscr{L}$ where $\mathbf{A} \leftarrow_{\$} \mathcal{D}_{\ell,k}$ and for $sk = \mathbf{k} \leftarrow_{\$} \mathbb{Z}_p^\ell$ and $sk' = \mathbf{k}' \leftarrow_{\$} \mathbb{Z}_p^\ell$, the event $\alpha_\rho(sk) = \alpha_\rho(sk') \iff \mathbf{k}^\top \cdot [\mathbf{A}]_1 = \mathbf{k}'^\top \cdot [\mathbf{A}]_1 \iff (\mathbf{k}-\mathbf{k}')^\top \cdot \mathbf{A} = \mathbf{0} \iff \mathbf{k}-\mathbf{k}' \in \mathsf{Ker}(\mathbf{A})$. Since $\mathbf{A} \in \mathbb{Z}_p^{\ell \times k}$ output by $\mathcal{D}_{\ell,k}$ has rank k, the left kernal $\mathsf{Ker}(\mathbf{A})$ has rank $\ell - k$. So the probability that $\mathbf{k} - \mathbf{k}' \in \mathsf{Ker}(\mathbf{A})$ is $p^{\ell-k}/p^\ell = 1/p^k$, by the uniformity of $\mathbf{k} - \mathbf{k}'$ over $\mathbb{Z}_p^\ell$. This shows the PK-diversity of $\mathsf{QAHPS}_{\mathsf{MDDH}}$ with $\boldsymbol{\epsilon}^{\mathsf{pk\text{-}div}}_{\mathsf{QAHPS}_{\mathsf{MDDH}}}(\lambda) = 1/p^k$. □

We note that the PK-diversity of $\mathsf{QAHPS}_{\mathsf{MDDH}}$ also holds for language parameters $\rho_0 = [\mathbf{A}_0]_1$ output by $\mathscr{L}_0 = \mathscr{L}_{\mathcal{U}_{\ell,k}}$, since $\mathbf{A}_0 \in \mathbb{Z}_p^{\ell \times k}$ output by $\mathcal{U}_{\ell,k}$ has rank k with overwhelming probability $1 - 2^{-\Omega(\lambda)}$.

The κ-LR-$\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching of $\mathsf{QAHPS}_{\mathsf{MDDH}}$ is shown in [22, Theorem 3]. Here we recall the result in the following theorem.

Theorem 12 (κ-LR-$\langle \mathscr{L}, \mathscr{L}_0 \rangle$-Key-Switching of $\mathsf{QAHPS}_{\mathsf{MDDH}}$ [22, Theorem 3]). *Let $\ell \geq 2k+1$ and $\kappa \leq \log p - \Omega(\lambda)$. The proposed $\mathsf{QAHPS}_{\mathsf{MDDH}}$ in Fig. 20 supports κ-LR-$\langle \mathscr{L}, \mathscr{L}_0 \rangle$-key-switching with $\boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle \mathscr{L}, \mathscr{L}_0 \rangle\text{-}\mathsf{ks}}_{\mathsf{QAHPS}_{\mathsf{MDDH}}, \mathcal{A}, \kappa}(\lambda) \leq 2^{-\Omega(\lambda)}$ for any (unbounded) adversary $\mathcal{A}$.*

In [22, Theorem 2], the $\mathscr{L}_0$-Multi-Extracting of $\mathsf{QAHPS}_{\mathsf{MDDH}}$ in a single-key setting is tightly reduced to the $\mathcal{U}_k$-MDDH assumption over $\mathbb{G}_1$. Due to the random self-reducibility of $\mathcal{U}_k$-MDDH, the $\mathscr{L}_0$-Multi-Key-Multi-Extracting of $\mathsf{QAHPS}_{\mathsf{MDDH}}$ also holds with a tight reduction to $\mathcal{U}_k$-MDDH, as shown in the following theorem.

Theorem 13 (Tight $\mathscr{L}_0$-Multi-Key-Multi-Extracting of $\mathsf{QAHPS}_{\mathsf{MDDH}}$). *If the $\mathcal{U}_k$-MDDH assumption holds over $\mathbb{G}_1$, then the proposed $\mathsf{QAHPS}_{\mathsf{MDDH}}$ in Fig. 20 supports $\mathscr{L}_0$-multi-key-multi-extracting. Concretely, for any adversary $\mathcal{A}$, any n and any Q, there exists an adversary $\mathcal{B}$, such that $\mathbf{T}(\mathcal{B}) \approx \mathbf{T}(\mathcal{A}) + nQ \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and $\mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{QAHPS}_{\mathsf{MDDH}}, \mathcal{A}, n, Q}(\lambda) \leq \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_k, \mathbb{G}_1, \mathcal{B}}(\lambda) + 1/(p-1)$.*

Proof of Theorem 13. Firstly, we construct an adversary $\mathcal{B}'$ against the Q-fold $\mathcal{U}_{k+n,k}$-MDDH over $\mathbb{G}_1$, so that $\mathsf{Adv}^{\mathscr{L}_0\text{-}\mathsf{mk\text{-}mext}}_{\mathsf{QAHPS}_{\mathsf{MDDH}}, \mathcal{A}, n, Q}(\lambda) \leq \mathsf{Adv}^{Q\text{-}\mathsf{mddh}}_{\mathcal{U}_{k+n,k}, \mathbb{G}_1, \mathcal{B}'}(\lambda)$.

Then by the random self-reducibility of $\mathcal{U}_{k+n,k}$-MDDH (i.e., Lemma 2) and $\mathcal{U}_k$-MDDH $\Rightarrow \mathcal{U}_{k+n,k}$-MDDH (i.e., Lemma 1), Theorem 13 follows.

Let us fix some *notations*. For any matrix $\mathbf{X}$ consisting $(k+n)$ rows, denote by $\overline{\mathbf{X}}$ the upper k rows of $\mathbf{X}$, $\underline{\mathbf{X}}$ the lower n rows of $\mathbf{X}$, and $\underline{\mathbf{X}}^{(i)}$ the i-th row of $\underline{\mathbf{X}}$ (which is also the the $(k+i)$-th row of $\mathbf{X}$) for any $i \in [n]$.

Given a challenge $([\mathbf{B}]_1, [\mathbf{U}]_1)$, $\mathcal{B}'$ wants to distinguish $[\mathbf{U}]_1 = [\mathbf{BW}]_1$ from $[\mathbf{U}]_1 \leftarrow_{\$} \mathbb{G}_1^{(k+n)\times Q}$, where $\mathbf{B} \leftarrow_{\$} \mathcal{U}_{k+n,k}$ and $\mathbf{W} \leftarrow_{\$} \mathbb{Z}_p^{k\times Q}$. For each $j \in [Q]$, denote by $[\mathbf{u}_j]_1 \in \mathbb{G}_1^{k+n}$ the j-th column of $[\mathbf{U}]_1$. $\mathcal{B}'$ is constructed as follows.

- $\mathcal{B}'$ chooses $\mathbf{V} \leftarrow_{\$} \mathbb{Z}_p^{\ell\times k}$ uniformly, computes $[\mathbf{A}_0]_1 := \mathbf{V}[\overline{\mathbf{B}}]_1 \in \mathbb{G}_1^{\ell\times k}$ and sets $\rho_0 := [\mathbf{A}_0]_1$ as the language parameter.
- For each $i \in [n]$, $\mathcal{B}'$ implicitly sets $sk_i = \mathbf{k}_i \leftarrow_{\$} \mathbb{Z}_p^{\ell}$ with $\mathbf{k}_i^\top \cdot [\mathbf{A}_0]_1 = [\underline{\mathbf{B}}^{(i)}]_1$.
- For each $j \in [Q]$, $\mathcal{B}'$ computes $[\mathbf{c}_j]_1 := \mathbf{V}[\overline{\mathbf{u}_j}]_1 \in \mathbb{G}_1^{\ell}$.
- For each $i \in [n]$ and each $j \in [Q]$, $\mathcal{B}'$ computes $[hv_{i,j}]_1 := [\underline{\mathbf{u}_j}^{(i)}]_1 \in \mathbb{G}_1$.
- Finally, $\mathcal{B}'$ submits $\left(\ \rho_0,\ \left\{[\mathbf{c}_j]_1,\ \{[hv_{i,j}]_1\}_{i\in[n]}\right\}_{j\in[Q]}\ \right)$ to $\mathcal{A}$, and outputs whatever $\mathcal{A}$ outputs.

Clearly, $[\mathbf{A}_0]_1$ is uniformly distributed over $\mathbb{G}_1^{\ell\times k}$, due to the randomness of $\mathbf{V}$. Thus $\mathcal{B}'$'s simulation of $\rho_0 = [\mathbf{A}_0]_1$ is perfect. Meanwhile, $\mathcal{B}'$'s implicit simulation of $sk_i = \mathbf{k}_i$ is also perfect for each $i \in [n]$, since $\mathbf{k}_i$ is uniformly random as long as $\underline{\mathbf{B}}^{(i)} \in \mathbb{Z}_p^{1\times k}$ is.

- If $[\mathbf{U}]_1 = [\mathbf{BW}]_1$, for each $j \in [Q]$, $[\mathbf{u}_j]_1 = [\mathbf{Bw}_j]_1$ with $\mathbf{w}_j \leftarrow_{\$} \mathbb{Z}_p^k$, and consequently:
  - $[\mathbf{c}_j]_1 := \mathbf{V}[\overline{\mathbf{u}_j}]_1 = [\mathbf{V}\overline{\mathbf{B}}\mathbf{w}_j]_1 = [\mathbf{A}_0\mathbf{w}_j]_1$, thus is uniformly distributed over $\mathsf{Span}([\mathbf{A}_0]_1) = \mathcal{L}_{\rho_0}$;
  - for each $i \in [n]$, $[hv_{i,j}]_1 := [\underline{\mathbf{u}_j}^{(i)}]_1 = [\underline{\mathbf{B}}^{(i)}\mathbf{w}_j]_1 = \mathbf{k}_i^\top \cdot [\mathbf{A}_0\mathbf{w}_j]_1 = \mathbf{k}_i^\top \cdot [\mathbf{c}_j]_1 = \Lambda_{sk_i}([\mathbf{c}_j]_1)$.
- If $[\mathbf{U}]_1 \leftarrow_{\$} \mathbb{G}_2^{(k+n)\times Q}$, for each $j \in [Q]$, $[\mathbf{u}_j]_1 \leftarrow_{\$} \mathbb{G}_2^{k+n}$, and consequently:
  - $[\mathbf{c}_j]_1 := \mathbf{V}[\overline{\mathbf{u}_j}]_1 = [\mathbf{V}\overline{\mathbf{B}}\overline{\mathbf{B}}^{-1}\overline{\mathbf{u}_j}]_1 = [\mathbf{A}_0\overline{\mathbf{B}}^{-1}\overline{\mathbf{u}_j}]_1$, thus is also uniformly distributed over $\mathsf{Span}([\mathbf{A}_0]_1) = \mathcal{L}_{\rho_0}$ with witness $\overline{\mathbf{B}}^{-1}\overline{\mathbf{u}_j}$;
  - for each $i \in [n]$, $[hv_{i,j}]_1 := [\underline{\mathbf{u}_j}^{(i)}]_1 \in \mathbb{G}_1$, which is uniformly distributed over $\mathcal{HV} = \mathbb{G}_1$ (and in particular, is independent of $[\mathbf{c}_j]_1$).

Consequently, we get $\mathsf{Adv}^{\mathscr{L}_0\text{-mk-mext}}_{\mathsf{QAHPS}_{\mathsf{MDDH}},\mathcal{A},n,Q}(\lambda) \le \mathsf{Adv}^{Q\text{-mddh}}_{\mathcal{U}_{k+n,k},\mathbb{G}_1,\mathcal{B}'}(\lambda)$, as desired. This completes the proof of Theorem 13. □

We also show the κ-LR-$\langle\mathscr{L}_0,\mathscr{L}\rangle$-OT-extracting of $\mathsf{QAHPS}_{\mathsf{MDDH}}$, which together with the PK-diversity (for $\mathscr{L}_0$) are needed for the strong $\mathsf{MU}^{\mathsf{c\&l}}$-CMVA security of our MAC in Appendix G.2 (cf. Theorem 5) and the $\mathsf{MUMC}^{\mathsf{c\&l}}$-Auth security of our AE in Appendix H.2 (cf. Theorem 7).

Theorem 14 (κ-LR-$\langle\mathscr{L}_0,\mathscr{L}\rangle$-OT-Extracting of $\mathsf{QAHPS}_{\mathsf{MDDH}}$). *Let $\ell \ge 2k+1$ and $\kappa \le \log p - \Omega(\lambda)$. The proposed $\mathsf{QAHPS}_{\mathsf{MDDH}}$ in Fig. 20 supports κ-LR-$\langle\mathscr{L}_0,\mathscr{L}\rangle$-OT-extracting with $\boldsymbol{\epsilon}^{\mathsf{lr}\text{-}\langle\mathscr{L}_0,\mathscr{L}\rangle\text{-otext}}_{\mathsf{QAHPS}_{\mathsf{MDDH}},\mathcal{A},\kappa}(\lambda) \le 2^{-\Omega(\lambda)}$ for any (unbounded) adversary $\mathcal{A}$.*

The proof of Theorem 14 is similar to that of Theorem 10, by simply ignoring the parts related to verification key. Hence we omit it.

I.5 Instantiation of Tag-Based QA-NIZK from MDDH

In this subsection, we instantiate tag-based QA-NIZK with a tag-based variant of the MDDH-based QA-NIZK scheme proposed in [1, Subsect. 3.2]. The original scheme in [1] is not tag-based. As noted by Abe et al. [1], their scheme can be easily adapted to tag-based QA-NIZK by putting the tag τ inside a collision-resistant hash function.

Let $\mathscr{L} = \mathscr{L}_{\mathcal{D}_{\ell,k}}$ be the language distribution specified in Appendix I.2. Let $\mathcal{T}$ be an arbitrary tag space. We present the tag-based variant of the MDDH-based QA-NIZK scheme in [1, Subsect. 3.2], $\mathsf{QANIZK}_{\mathsf{MDDH}} = (\mathsf{Setup}_{\mathsf{NIZK}}, \mathsf{CRSGen}, \mathsf{Prove}, \mathsf{Vrfy}_{\mathsf{NIZK}}, \mathsf{Sim})$ for $\mathscr{L}$, as follows. The scheme $\mathsf{QANIZK}_{\mathsf{MDDH}}$ is built upon a NIZK scheme $\Pi_{\mathsf{or}} = (\Pi_{\mathsf{or}}.\mathsf{CRSGen}, \Pi_{\mathsf{or}}.\mathsf{Prove}, \Pi_{\mathsf{or}}.\mathsf{Vrfy})$ for OR-languages as sub-procedures, and uses a family of collision-resistant hash functions $\mathcal{H}'$ with range $\mathbb{Z}_p$. We present $\mathsf{QANIZK}_{\mathsf{MDDH}}$ with tag space $\mathcal{T}$ and it sub-procedures Π_{or} in Fig. 21. It is straightforward to check the perfect completeness of $\mathsf{QANIZK}_{\mathsf{MDDH}}$.

$\underline{\mathsf{pp}_{\mathsf{NIZK}} \leftarrow\!\!\$\ \mathsf{Setup}_{\mathsf{NIZK}}:}$
$\mathsf{gpar} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, e, P_1, P_2, P_T) \leftarrow\!\!\$\ \mathsf{PGGen}.$
Return $\mathsf{pp}_{\mathsf{NIZK}} := \mathsf{gpar}.$

$\underline{(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}) \leftarrow\!\!\$\ \mathsf{CRSGen}(\rho = [\mathbf{A}]_1 \in \mathbb{G}_1^{\ell\times k}):}$
$\mathbf{A}_0, \mathbf{A}_1 \leftarrow \mathcal{D}_{2k,k}$, $\mathbf{B} \leftarrow \mathcal{D}_k$, $H' \leftarrow\!\!\$\ \mathcal{H}'$.
$\mathsf{crs}_{\mathsf{or}} \leftarrow \Pi_{\mathsf{or}}.\mathsf{Gen}([\mathbf{A}_0]_1, [\mathbf{A}_1]_1).$
$\mathbf{K} \leftarrow\!\!\$\ \mathbb{Z}_p^{2k\times(k+1)}$, $\mathbf{K}_0, \mathbf{K}_1 \leftarrow\!\!\$\ \mathbb{Z}_p^{\ell\times(k+1)}$.
$\mathbf{P} := \mathbf{A}_0^\top \mathbf{K} \in \mathbb{Z}_p^{k\times(k+1)}$.
$[\mathbf{P}_0]_1 := [\mathbf{A}^\top \mathbf{K}_0]_1$, $[\mathbf{P}_1]_1 := [\mathbf{A}^\top \mathbf{K}_1]_1$.
$\mathbf{C} := \mathbf{K}\mathbf{B}$, $\mathbf{C}_0 := \mathbf{K}_0\mathbf{B}$, $\mathbf{C}_1 := \mathbf{K}_1\mathbf{B}$.
$\mathsf{crs} := (\mathsf{crs}_{\mathsf{or}}, [\mathbf{A}_0]_1, [\mathbf{P}]_1, [\mathbf{P}_0]_1, [\mathbf{P}_1]_1, [\mathbf{B}]_2, [\mathbf{C}]_2, [\mathbf{C}_0]_2, [\mathbf{C}_1]_2, H').$
$\mathsf{td}_{\mathsf{crs}} := (\mathbf{K}_0, \mathbf{K}_1).$
Return $(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}})$.

$\underline{\pi \leftarrow\!\!\$\ \mathsf{Prove}(\mathsf{crs}, \tau, [\mathbf{c}]_1, \mathbf{w}):}$ //Prove $\mathbf{c} = \mathbf{A}\mathbf{w}$
$\mathbf{r} \leftarrow\!\!\$\ \mathbb{Z}_p^k$, $[\mathbf{t}]_1 := [\mathbf{A}_0]_1\mathbf{r}.$
$\pi_{\mathsf{or}} \leftarrow\!\!\$\ \Pi_{\mathsf{or}}.\mathsf{Prove}(\mathsf{crs}_{\mathsf{or}}, [\mathbf{t}]_1, \mathbf{r}).$
$t := H'([\mathbf{c}]_1, [\mathbf{t}]_1, \pi_{\mathsf{or}}, \tau) \in \mathbb{Z}_p.$
$[\mathbf{u}]_1 := \mathbf{w}^\top([\mathbf{P}_0]_1 + t[\mathbf{P}_1]_1) + \mathbf{r}^\top[\mathbf{P}]_1.$
Return $\pi := ([\mathbf{t}]_1, [\mathbf{u}]_1, \pi_{\mathsf{or}})$.

$\underline{0/1 \leftarrow \mathsf{Vrfy}_{\mathsf{NIZK}}(\mathsf{crs}, \tau, [\mathbf{c}]_1, \pi):}$
$t := H'([\mathbf{c}]_1, [\mathbf{t}]_1, \pi_{\mathsf{or}}, \tau) \in \mathbb{Z}_p.$
If $\Pi_{\mathsf{or}}.\mathsf{Vrfy}(\mathsf{crs}_{\mathsf{or}}, [\mathbf{t}]_1, \pi_{\mathsf{or}}) = 0$: Return 0.
If $e([\mathbf{u}]_1, [\mathbf{B}]_2) = e([\mathbf{c}^\top]_1, [\mathbf{C}_0 + t\mathbf{C}_1]_2)$
$\quad + e([\mathbf{t}^\top]_1, [\mathbf{C}]_2)$: Return 1;
Else: Return 0.

$\underline{\pi \leftarrow\!\!\$\ \mathsf{Sim}(\mathsf{crs}, \mathsf{td}_{\mathsf{crs}}, \tau, [\mathbf{c}]_1):}$
$\mathbf{r} \leftarrow\!\!\$\ \mathbb{Z}_p^k$, $[\mathbf{t}]_1 := [\mathbf{A}_0]_1\mathbf{r}.$
$\pi_{\mathsf{or}} \leftarrow\!\!\$\ \Pi_{\mathsf{or}}.\mathsf{Prove}(\mathsf{crs}_{\mathsf{or}}, [\mathbf{t}]_1, \mathbf{r}).$
$t := H'([\mathbf{c}]_1, [\mathbf{t}]_1, \pi_{\mathsf{or}}, \tau) \in \mathbb{Z}_p.$
$[\mathbf{u}]_1 := [\mathbf{c}^\top(\mathbf{K}_0 + t\mathbf{K}_1)]_1 + \mathbf{r}^\top[\mathbf{P}]_1.$
Return $\pi := ([\mathbf{t}]_1, [\mathbf{u}]_1, \pi_{\mathsf{or}})$.

Sub-procedures:
$\underline{\mathsf{crs}_{\mathsf{or}} \leftarrow\!\!\$\ \Pi_{\mathsf{or}}.\mathsf{CRSGen}([\mathbf{A}_0]_1, [\mathbf{A}_1]_1):}$
$\mathbf{D} \leftarrow \mathcal{D}_k$, $\mathbf{z} \leftarrow\!\!\$\ \mathbb{Z}_p^{k+1} \setminus \mathsf{Span}(\mathbf{D})$.
Return $\mathsf{crs}_{\mathsf{or}} := ([\mathbf{D}]_2, [\mathbf{z}]_2)$.

$\underline{\pi_{\mathsf{or}} \leftarrow\!\!\$\ \Pi_{\mathsf{or}}.\mathsf{Prove}(\mathsf{crs}_{\mathsf{or}}, [\mathbf{t}]_1, \mathbf{r}):}$
Let $j \in \{0,1\}$ s.t. $[\mathbf{t}]_1 = [\mathbf{A}_j]_1\mathbf{r}$.
$\mathbf{v} \leftarrow\!\!\$\ \mathbb{Z}_p^k$.
$[\mathbf{z}_{1-j}]_2 := [\mathbf{D}]_2\mathbf{v}$, $[\mathbf{z}_j]_2 := [\mathbf{z}]_2 - [\mathbf{z}_{1-j}]_2$.
$\mathbf{S}_0, \mathbf{S}_1 \leftarrow\!\!\$\ \mathbb{Z}_p^{k\times k}$.
$[\mathbf{G}_j]_2 := \mathbf{S}_j[\mathbf{D}]_2^\top + \mathbf{r}[\mathbf{z}_j]_2^\top.$
$[\Pi_j]_1 := [\mathbf{A}_j]_1\mathbf{S}_j.$
$[\mathbf{G}_{1-j}]_2 := \mathbf{S}_{1-j}[\mathbf{D}]_2^\top.$
$[\Pi_{1-j}]_1 := [\mathbf{A}_{1-j}]_1\mathbf{S}_{1-j} - [\mathbf{t}]_1\mathbf{v}^\top.$
Return $\pi_{\mathsf{or}} := ([\mathbf{z}_0]_2, ([\mathbf{G}_i]_2, [\Pi_i]_1)_{i\in\{0,1\}})$.

$\underline{0/1 \leftarrow \Pi_{\mathsf{or}}.\mathsf{Vrfy}(\mathsf{crs}_{\mathsf{or}}, [\mathbf{t}]_1, \pi_{\mathsf{or}}):}$
$[\mathbf{z}_1]_2 = [\mathbf{z}]_2 - [\mathbf{z}_0]_2.$
If for all $i \in \{0,1\}$, it holds that
$\quad e([\mathbf{A}_i]_1, [\mathbf{G}_i]_2) = e([\Pi_i]_1, [\mathbf{D}^\top]_2)$
$\quad\quad + e([\mathbf{t}]_1, [\mathbf{z}_i^\top]_2)$: Return 1;
Else: Return 0.

Fig. 21. The MDDH-based tag-based QA-NIZK scheme $\mathsf{QANIZK}_{\mathsf{MDDH}}$ in [1].

In [1], Abe et al. proved the tight USS for witness-sampleable distributions from MDDH assumptions. Formally, we recall the following theorem from [1].

Theorem 15 (Tight USS of $\mathsf{QANIZK}_{\mathsf{MDDH}}$ [1, Theorem 1, Theorem 2]). *If the $\mathcal{D}_{2k,k}$-MDDH assumption holds in $\mathbb{G}_1$, $\mathcal{D}_k$-MDDH assumption holds in $\mathbb{G}_2$, $\mathcal{D}_k$-KerMDH assumption holds in $\mathbb{G}_1$ and $\mathcal{H}'$ is a collision-resistant hash family, then the proposed $\mathsf{QANIZK}_{\mathsf{MDDH}}$ in Fig. 21 has perfect zero-knowledge and unbounded simulation-soundness.*

Concretely, for any adversary $\mathcal{A}$ who makes at most Q_s times of $\mathcal{O}_{\mathrm{SIM}}$ queries, there exist adversaries $\mathcal{B}_1, \cdots, \mathcal{B}_4$ such that $\mathbf{T}(\mathcal{B}_1) \approx \cdots \approx \mathbf{T}(\mathcal{B}_4) \approx \mathbf{T}(\mathcal{A}) + Q_s \cdot \mathsf{poly}(\lambda)$ with $\mathsf{poly}(\lambda)$ independent of $\mathbf{T}(\mathcal{A})$, and

$$\begin{aligned}\mathsf{Adv}^{\mathsf{uss}}_{\mathsf{QANIZK}_{\mathsf{MDDH}},\mathcal{A}}(\lambda) \leq\ & \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H}',\mathcal{B}_1}(\lambda) + (4k\lceil \log Q_s \rceil + 2) \cdot \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_{2k,k},\mathbb{G}_1,\mathcal{B}_2}(\lambda) \\ & + (2\lceil \log Q_s \rceil + 2) \cdot \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_3}(\lambda) + \mathsf{Adv}^{\mathsf{kmdh}}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}_4}(\lambda) + 2^{-\Omega(\lambda)}.\end{aligned}$$

By Lemma 3, $\mathcal{D}_k$-MDDH $\Rightarrow$ $\mathcal{D}_k$-KerMDH. Hence, Theorem 15 indicates that the $\mathsf{QANIZK}_{\mathsf{MDDH}}$ scheme in Fig. 21 has a tight USS based on the MDDH assumptions, with a security loss $O(\log Q_s)$. Since $Q_s = \mathsf{poly}(\lambda)$ for PPT adversaries, the security loss is in fact $O(\log Q_s) = O(\log \lambda)$, which is lower than $O(\lambda)$.

Table of Contents