# More Efficient Public-Key Cryptography with Leakage and Tamper Resilience

Shuai Han<sup>1,2</sup>, Shengli Liu<sup>2,3</sup>, and Dawu Gu<sup>1,3</sup>

 <sup>1</sup> School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China {dalen17,dwgu}@sjtu.edu.cn
 <sup>2</sup> State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China <sup>3</sup> Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China slliu@sjtu.edu.cn

Abstract. In this paper, we study the design of efficient signature and public-key encryption (PKE) schemes in the presence of both leakage and tampering attacks. Firstly, we formalize the strong leakage and tamperresilient (sLTR) security model for signature, which provides strong existential unforgeability, and deals with bounded leakage and restricted tampering attacks, as a counterpart to the sLTR security introduced by Sun et al. (ACNS 2019) for PKE. Then, we present direct constructions of signature and chosen-ciphertext attack (CCA) secure PKE schemes in the sLTR model, based on the matrix decisional Diffie-Hellman (MDDH) assumptions (which covers the standard symmetric external DH (SXDH) and k-Linear assumptions) over asymmetric pairing groups. Our schemes avoid the use of heavy building blocks such as the true-simulation extractable non-interactive zero-knowledge proofs (tSE-NIZK) proposed by Dodis et al. (ASIACRYPT 2010), which are usually needed in constructing schemes with leakage and tamper-resilience. Especially, our SXDHbased signature and PKE schemes are more efficient than the existing schemes in the leakage and tamper-resilient setting: our signature scheme has only 4 group elements in the signature, which is about  $5 \times \sim 8 \times$ shorter, and our PKE scheme has only 6 group elements in the ciphertext, which is about  $1.3 \times \sim 3.3 \times$  shorter. Finally, we note that our signature scheme is the *first* one achieving strong existential unforgeability in the leakage and tamper-resilient setting, where strong existential unforgeability has important applications in building more complex primitives such as signcryption and authenticated key exchange.

**Keywords:** digital signature, public-key encryption, leakage attacks, tampering attacks

## 1 Introduction

Traditionally, when analyzing and proving security of cryptographic schemes, it is always assumed that the only way for an adversary to get information about the secret keys is through *black-box access* to the cryptographic devices. In reality, however, an adversary may go far beyond black-box access, and obtain secret key information by directly accessing/tampering with the memory or the internal computation of the devices. To deal with these threats, *leakage and tamper-resilient* cryptography emerges with the aim of designing provably secure cryptographic schemes in such scenarios.

Leakage-resilient security. The motivation for leakage-resilient cryptography is the increasing popularity of various side-channel attacks [23, 29, 28], including timing measurements, power analysis, electromagnetic measurements and microwave attacks, through which an adversary can recover partial information about the secret keys. Such a capability is usually formulated by a leakage oracle, which allows the adversary to specify arbitrary leakage functions L and obtain the results L(sk) of applying L to the secret key sk. Leakage-resilient security requires that the cryptographic schemes remain secure even for the adversary who has access to the leakage oracle. In this work, we focus on the bounded leakageresilient security [1, 32], where the total amount |L(sk)| of leakage information is less than the whole secret key |sk| and in particular bounded.

**Tamper-resilient security.** The attacks that leakage-resilient cryptography considers are in fact *passive* attacks, while the adversary may also launch *active* attacks such as fault injection and memory tampering attacks [6, 19], through which the adversary can force the cryptographic devices to operate under a different but related secret key, and observe the input-output behaviour of the device under the modified secret key. The theoretical treatment of such attacks was initiated by Bellare and Kohno [3], where the capability of adversaries is modeled by a class of tampering functions  $\mathcal{T}$  on the secret key space. Tamper-resilient security stipulates that the cryptographic schemes remain secure even for the adversary who has access to the schemes executed under the related keys T(sk), with  $T \in \mathcal{T}$  chosen by the adversary.

As observed by Gennaro et al. [21], it is *impossible* to achieve tamper-resilient security against any polynomial number of arbitrary tampering queries, without making further assumptions, such as *key-updating* or *self-destruct* mechanism<sup>1</sup>.

Leakage and Tamper-resilient public-key cryptography. In light of the fact that physical attacks in the real world include both passive and active attacks, Kalai et al. [26] initiate the study of designing public-key cryptographic schemes that are resilient to *both* leakage and tampering attacks.

Up to now, there are several models for leakage and tamper-resilient security. The first model is proposed by Kalai et al. [26] and considers *continual tampering* and leakage (*i.e.*, the CTL model). This model provides a very strong security guarantee, but at the price of inevitably relying on key-updating or self-destruct mechanism. Kalai et al. [26] construct a signature scheme in the CTL model using a true-simulation extractable non-interactive zero-knowledge (tSE-NIZK)

<sup>&</sup>lt;sup>1</sup> Key-updating mechanism enables the secret key to be periodically updated. Selfdestruct mechanism enables the cryptographic device to blow up and erase all internal states, including sk, once a tampering attempt is detected.

proof system [12] as a building block. As shown in [12], tSE-NIZK can be built generically from a chosen-ciphertext attack (CCA) secure public-key encryption (PKE) and a regular NIZK. However, even using the efficient Groth-Sahai NIZK [22], it would lead to a tSE-NIZK with proof consisting of at least 20 group elements, and so does the signature of the resulting signature scheme. Kalai et al. [26] also present a PKE scheme with chosen-plaintext attack (CPA) secure in the CTL model, meaning that the adversary is not allowed to observe the effect of tampering on the decryption oracle. Fujisaki et al. [18] further investigate how to construct CCA-secure PKE in the CTL model, and present a scheme based on the one-time lossy filter technique [33]. The ciphertext of their PKE scheme consists of about 8 group elements.

The second model is introduced by Damgård et al. [10] and considers both bounded leakage and bounded tampering (i.e., the BLT model). Here bounded tampering means that the adversary is only allowed to make a limited number of tampering queries, and consequently, it does not need key-updating or selfdestruct mechanisms. To achieve BLT security, they propose a novel approach which reduces tampering to leakage. The benefit of this approach is that it could achieve tampering-resilience against arbitrary function class  $\mathcal{T}$ . However, the approach suffers from two disadvantages, one being that the amount of leakage tolerated by the leakage-resilience is largely decreasing, and the other being that for PKE it does not allow "post-challenge" tampering queries<sup>2</sup>. Under this approach, Damgård et al. [10] propose a signature scheme from  $\Sigma$ -protocol via the Fiat-Shamir heuristic [17] in the random oracle model and a CCA-secure PKE scheme from tSE-NIZK [12]. Faonio et al. [15] also follow the approach, and prove that the leakage-resilient signature scheme in [12] and the leakageresilient CCA-secure PKE scheme in [33] are secure in the BLT model. However, the signature scheme also uses tSE-NIZK, and its signature consists of more than 34 group elements. Their PKE scheme avoids the use of NIZK, but the ciphertext is over composite order groups and has a length of more than 5000 bits at the 128-bit security level, which corresponds to about 19 group elements in typical prime order groups (where each group element is about 256 bits).

The third model is formalized by Sun et al. [36] and is called *the leakage and tampering-resilient (LTR) model.* This model also considers bounded leakage, but for tampering, it allows an unbounded number of tampering queries, while the tampering functions are restricted in a predefined function class  $\mathcal{T}$ , the same as the tampering-resilient security introduced by Bellare and Kohno [3]. Similar to the BLT model, the LTR model does not need key-updating or self-destruct mechanisms, and not only that, it also allows "post-challenge" tampering queries for PKE. Subsequently, Sun et al. [35] strengthen the LTR model to *the strong LTR (sLTR) model*, by imposing only minimal restrictions on the adversary's decryption queries. These two works [36, 35] focus on PKE, and construct CCAsecure PKE schemes from tSE-NIZK and new variants of hash proof systems [9] in the LTR model and the sLTR model, respectively. Their schemes achieve

<sup>&</sup>lt;sup>2</sup> Namely, the adversary is not allowed to make any tampering queries after it receives the challenge ciphertext.

tamper-resilience against affine function class. However, due to the inefficiency of tSE-NIZK, the ciphertext of their schemes would consist of at least 20 group elements. Accordingly, Sun et al. [35] leave the construction of CCA-secure PKE in the sLTR model without using tSE-NIZK as an interesting future work.

The fourth model is due to Chakraborty and Rangan [7] and extends the BLT model in the presence of *split-state mechanism*<sup>3</sup>. This model is called the *post-challenge BLT (pcBLT) model*, since it serves as an alternative way to make "post-challenge" tampering (and also leakage) queries for PKE possible. Chakraborty and Rangan [7] also focus on PKE and construct a CCA-secure PKE scheme from tSE-NIZK in the pcBLT model. Similarly, the ciphertext of their scheme would consist of at least 20 group elements.

There are also other models such as the line of research which protects cryptosystems against leakage and tampering attacks by leveraging (leakage-resilient) non-malleable codes [13, 30, 16, 25]. However, these works usually rely on hardware requirements such as key-updating, self-destruct or split-state mechanisms, and the proposed schemes are more like feasibility results and less efficient.

**Our Contributions.** In this work, we study the design of efficient signature and PKE schemes in the sLTR model, without using tSE-NIZK (or other heavy building blocks). Our contributions are three-fold.

- We formalize the strong LTR (sLTR) model for signature schemes, as a counterpart to the sLTR model for PKE introduced in [35]. Here "strong" means the strong existential unforgeability of signatures, which even guarantees that the adversary cannot forge a new signature for an already signed message. Moreover, for the adversary to win, we impose only minimal restrictions on the forgery produced by the adversary, thus our security provides a very strong guarantee (see Remark 1 for more discussions).
- We give direct constructions of signature and CCA-secure PKE schemes in the sLTR model. Both of the schemes are designed in the standard model, over asymmetric pairing groups and without using tSE-NIZK, thus accomplishing the interesting future work left by Sun et al. [35].

Both of our schemes are proven secure based on the standard matrix decisional Diffie-Hellman (MDDH) assumptions [14], which cover the standard symmetric external DH (SXDH)<sup>4</sup> and k-Linear assumptions. Our signature scheme achieves leakage-resilience with leakage rate<sup>5</sup>  $\frac{1}{4} - o(1)$  and our PKE scheme with leakage rate  $\frac{1}{3} - o(1)$ . Both of our schemes achieve tamperresilience against affine function class, the same as the existing schemes [36, 35] in the (s)LTR model.

<sup>&</sup>lt;sup>3</sup> Split-state mechanism ensures that the secret key is split into two (or more) disjoint parts and the adversary can obtain leakages from each of the secret key parts independently and tamper each of the parts independently.

<sup>&</sup>lt;sup>4</sup> SXDH is a standard assumption that simply requires the DDH assumption to hold in both source groups  $\mathbb{G}_1$  and  $\mathbb{G}_2$  of the asymmetric pairing groups.

<sup>&</sup>lt;sup>5</sup> Leakage rate is defined as the ratio of the leakage amount that can be tolerated to the secret key size.

Our SXDH-based schemes are more efficient than the existing schemes in the leakage and tamper-resilient setting (i.e., no matter in the CTL, BLT, LTR, sLTR, or pcBLT model) [26, 18, 10, 15, 36, 35, 7]. More precisely, our signature scheme has only 4 group elements in the signature, which is about  $5 \times \sim 8 \times$  shorter, and our PKE scheme has only 6 group elements in the ciphertext, which is about  $1.3 \times \sim 3.3 \times$  shorter. We refer to Remark 2 and Remark 4 for a detailed efficiency analysis of our schemes.

• To our best knowledge, our signature scheme is the *first* one achieving strong existential unforgeability in the leakage and tamper-resilient setting. We note that strong existential unforgeability has important applications in building more complex primitives such as signcryption [2] and authenticated key exchange (AKE) [11], where it can help signcryption to achieve ciphertext integrity [4] and AKE to achieve strong notion of "matching conversations" security [5]. We also stress that the Generalized Boneh-Shen-Waters (GBSW) transform [34], which converts a (non-strongly) secure signature scheme to a strongly secure one with the help of chameleon hash, does *not* work in the presence of leakage and tampering. The reason is, the resulting signature scheme contains the trapdoor of chameleon hash in its secret key, thus the leakage and tampering of secret key means the leakage and tampering of trapdoor, which is not supported by the security of chameleon hash.

## 2 Preliminaries

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

For two random variables X and Y, the min-entropy of X is defined as  $\mathbf{H}_{\infty}(X) := -\log(\max_{x} \Pr[X = x])$ , and the statistical distance between X and Y is defined as  $\Delta(X, Y) := \frac{1}{2} \cdot \sum_{x} |\Pr[X = x] - \Pr[Y = x]|$ .

**Lemma 1** (Leftover Hash Lemma [24]). Let  $\mathcal{H} = \{H : \mathcal{X} \to \mathcal{Y}\}$  be a family of universal hash functions, i.e., for any  $x_1 \neq x_2 \in \mathcal{X}$ ,  $\Pr[H(x_1) = H(x_2)] \leq 1/|\mathcal{Y}|$ , where  $H \leftarrow_{\mathfrak{s}} \mathcal{H}$ . Then for any random variable X on  $\mathcal{X}$ , it holds that  $\Delta((H, H(X)), (H, U)) \leq \sqrt{|\mathcal{Y}| \cdot 2^{-\mathbf{H}_{\infty}(X)}}$ , where  $H \leftarrow_{\mathfrak{s}} \mathcal{H}$  and  $U \leftarrow_{\mathfrak{s}} \mathcal{Y}$ .

## 2.1 Digital Signatures

**Definition 1 (SIG).** A digital signature (SIG) scheme SIG = (Setup, Gen, Sign, Vrfy) with message space  $\mathcal{M}$  consists of four PPT algorithms:

- pp ←<sub>s</sub> Setup: The setup algorithm outputs a public parameter pp, which serves as an implicit input of other algorithms.
- $(vk, sk) \leftarrow \text{s} \text{Gen}(pp)$ : Taking pp as input, the key generation algorithm outputs a pair of verification key and signing key (vk, sk).
- $-\sigma \leftarrow s \operatorname{Sign}(sk, m)$ : Taking as input a signing key sk and a message  $m \in \mathcal{M}$ , the signing algorithm outputs a signature  $\sigma$ .
- $-0/1 \leftarrow \mathsf{Vrfy}(vk, m, \sigma)$ : Taking as input a verification key vk, a message  $m \in \mathcal{M}$  and a signature  $\sigma$ , the deterministic verification algorithm outputs a bit indicating whether  $\sigma$  is a valid signature for m w.r.t. vk.

Correctness requires that for all pp  $\leftarrow$ s Setup,  $(vk, sk) \leftarrow$ s Gen(pp) and  $m \in \mathcal{M}$ , it holds that  $\Pr \left[ \sigma \leftarrow$ s Sign $(sk, m) : Vrfy(vk, m, \sigma) = 1 \right] \ge 1 - negl(\lambda)$ .

### 2.2 Public-Key Encryption

**Definition 2 (PKE).** A public-key encryption (*PKE*) scheme PKE = (Setup, Gen, Enc, Dec) with message space  $\mathcal{M}$  consists of four *PPT* algorithms:

- $pp \leftarrow_s Setup$ : The setup algorithm outputs a public parameter pp, which serves as an implicit input of other algorithms.
- $-(pk, sk) \leftarrow \text{s} \text{Gen}(pp)$ : Taking pp as input, the key generation algorithm outputs a pair of public key and secret key (pk, sk).
- $ct \leftarrow_{s} \mathsf{Enc}(pk, m)$ : Taking as input a public key pk and a message  $m \in \mathcal{M}$ , the encryption algorithm outputs a ciphertext ct.
- $-m/\perp \leftarrow \mathsf{Dec}(sk, ct)$ : Taking as input a secret key sk and a ciphertext ct, 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 pp  $\leftarrow$ s Setup,  $(pk, sk) \leftarrow$ s Gen(pp) and  $m \in \mathcal{M}$ , it holds that  $\Pr\left[ct \leftarrow$ s En $(pk, m) : \text{Dec}(sk, ct) = m\right] \ge 1 - \text{negl}(\lambda)$ .

## 2.3 Collision-Resistant Hash Functions

**Definition 3 (Collision-resistant hash functions).** A family of hash functions  $\mathcal{H}$  is collision-resistant, if for any PPT adversary  $\mathcal{A}$ , it holds that

$$\mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{A}}(\lambda) := \Pr[H \leftarrow \mathfrak{H}, (x_1, x_2) \leftarrow \mathfrak{K}\mathcal{A}(H) : x_1 \neq x_2 \land H(x_1) = H(x_2)] \le \mathsf{negl}(\lambda)$$

#### 2.4 Pairing Groups and MDDH Assumptions

Let PGGen be a PPT algorithm outputting a description of pairing group 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 prime order  $p > 2^{2\lambda}$ ,  $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$  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 [14]. 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  $\mathbb{G}_s$ . 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 Span $([\mathbf{A}]_s) := \{[\mathbf{Ar}]_s | \mathbf{r} \in \mathbb{Z}_p^m\} \subseteq \mathbb{G}_s^n$ . 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  $[\mathbf{B}]_2$ , we let  $e([\mathbf{A}]_1, [\mathbf{B}]_2) := [\mathbf{AB}]_T$ .

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  $\mathcal{U}_k := \mathcal{U}_{k+1,k}$ .

**Definition 4** ( $\mathcal{D}_{\ell,k}$ -**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  $\operatorname{Adv}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}^{\operatorname{mddh}}(\lambda) := |\operatorname{Pr}[\mathcal{A}([\mathbf{A}]_s, [\mathbf{Aw}]_s) = 1] - \operatorname{Pr}[\mathcal{A}([\mathbf{A}]_s, [\mathbf{u}]_s) = 1] | \leq \operatorname{negl}(\lambda)$ , 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 [14], where  $\mathcal{LIN}_k$ :  $\mathbf{A} = \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_k \\ 1 & \cdots & 1 \end{pmatrix} \in \mathbb{Z}_p^{(k+1) \times k}.$ 

MDDH also covers the standard symmetric external DH (SXDH) assumption, which simply requires the DDH assumption to hold both in  $\mathbb{G}_1$  and  $\mathbb{G}_2$ .

Several relations among MDDH assumptions parameterized by different matrix distributions were established in [14, 20].

**Lemma 2**  $(\mathcal{D}_{\ell,k}\text{-MDDH} \Rightarrow \mathcal{U}_k\text{-MDDH} [14] \Rightarrow \mathcal{U}_{\ell',k}\text{-MDDH} [20])$ . For any PPT adversary  $\mathcal{A}$ , there exists a PPT  $\mathcal{B}$  s.t.  $\operatorname{Adv}_{\mathcal{U}_k,\mathbb{G}_s,\mathcal{A}}^{\operatorname{mddh}}(\lambda) \leq \operatorname{Adv}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{B}}^{\operatorname{mddh}}(\lambda)$ . For any PPT  $\mathcal{A}$ , there exists a PPT  $\mathcal{B}$  s.t.  $\operatorname{Adv}_{\mathcal{U}_{\ell',k},\mathbb{G}_s,\mathcal{A}}^{\operatorname{mddh}}(\lambda) \leq \operatorname{Adv}_{\mathcal{U}_k,\mathbb{G}_s,\mathcal{B}}^{\operatorname{mddh}}(\lambda)$ .

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).

We also define the  $\mathcal{D}_{\ell,k}$ -Kernel Matrix DH ( $\mathcal{D}_{\ell,k}$ -KerMDH) assumption according to [31] which is a natural search variant of the  $\mathcal{D}_{\ell,k}$ -MDDH assumption.

**Definition 5** ( $\mathcal{D}_{\ell,k}$ -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  $\operatorname{Adv}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}^{\mathsf{kmdh}}(\lambda) := \Pr\left[[\mathbf{x}]_{3-s} \in \mathbb{G}_{3-s}^{\ell} \leftarrow \mathfrak{A}([\mathbf{A}]_s) : \mathbf{x}^{\top}\mathbf{A} = \mathbf{0} \land \mathbf{x} \neq \mathbf{0}\right] \leq \operatorname{negl}(\lambda)$ , where the probability is over  $\mathbf{A} \leftarrow \mathfrak{D}_{\ell,k}$ .

The following lemma shows that the  $\mathcal{D}_{\ell,k}$ -KerMDH assumption is 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 Span( $[\mathbf{A}]_s$ ). **Lemma 3**  $(\mathcal{D}_{\ell,k}\text{-}\mathbf{MDDH} \Rightarrow \mathcal{D}_{\ell,k}\text{-}\mathbf{Ker}\mathbf{MDH}$  [31]). For any PPT adversary  $\mathcal{A}$ , there exists a PPT  $\mathcal{B}$  s.t.  $\operatorname{Adv}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{A}}^{\mathsf{kmdh}}(\lambda) \leq \operatorname{Adv}_{\mathcal{D}_{\ell,k},\mathbb{G}_s,\mathcal{B}}^{\mathsf{mddh}}(\lambda) + 1/(p-1).$ 

## 3 More Efficient SIG with Leakage and Tamper-Resilience

In this section, we present a direct and efficient construction of signature scheme with leakage and tamper-resilience, over asymmetric pairing groups based on the MDDH assumptions.

Concretely, in Subsect. 3.1, we formalize the leakage and tamper-resilient security for signature schemes, i.e., the *strong* LTR-CMA (sLTR-CMA) security, and then in Subsect. 3.2 and Subsect. 3.3, we present our signature scheme and its security proof, respectively.

## 3.1 Definition of sLTR-CMA Security

The standard security notion for signatures is existential unforgeability under chosen-message attacks (EUF-CMA). Here we extend it to  $(\kappa, \mathcal{T})$ -sLTR-CMA, parameterized by an integer  $\kappa$  and a function set  $\mathcal{T}$ : it additionally considers *leakages attacks*, where the total amount of leakage is bounded by  $\kappa$  bits, and *tampering attacks*, where the tampering functions are chosen from  $\mathcal{T}$ . Moreover, it provides *strong* existential unforgeability which further guarantees that the adversary cannot even forge a new signature for a message that it has ever queried. Below we present the formal definition of  $(\kappa, \mathcal{T})$ -sLTR-CMA security.

**Definition 6** (sLTR-CMA Security for SIG). Let  $\kappa = \kappa(\lambda) \in \mathbb{N}$ , and  $\mathcal{T}$  be a set of functions from  $\mathcal{SK}$  to  $\mathcal{SK}$  where  $\mathcal{SK}$  is the secret key space. A signature scheme SIG = (Setup, Gen, Sign, Vrfy) is  $(\kappa, \mathcal{T})$ -sLTR-CMA secure, if for any PPT adversary  $\mathcal{A}$ , it holds that  $\operatorname{Adv}_{\operatorname{SIG},\mathcal{A},\kappa,\mathcal{T}}^{\operatorname{SIG}}(\lambda) := \Pr[\operatorname{Exp}_{\operatorname{SIG},\mathcal{A},\kappa,\mathcal{T}}^{\operatorname{SIT-cma}} \Rightarrow 1] \leq \operatorname{negl}(\lambda)$ , where the experiment  $\operatorname{Exp}_{\operatorname{SIG},\mathcal{A},\kappa,\mathcal{T}}^{\operatorname{SIT-cma}}$  is defined in Fig. 1.

$Exp^{sltr-cma}_{SIG,\mathcal{A},\kappa,\mathcal{T}}$ :		$\mathcal{O}_{\mathrm{SIGN}}(T,m)$ :
$pp \gets_{\$} Setup,$	$(vk,sk) \leftarrow \texttt{s} Gen(pp)$	If $T \notin \mathcal{T}$ : Return $\perp$
$\mathcal{Q}_{id} := \emptyset$	$/\!\!/ {\rm Record}$ the signing queries	$\sigma \leftarrow_{\$} Sign(T(sk),m)$
	$/\!\!/ {\rm under}$ the identity function	If $T = id$ : $\mathcal{Q}_{id} := \mathcal{Q}_{id} \cup \{(m, \sigma)\}$
$\ell := 0$	$/\!\!/ {\rm Record}$ the leakage length	Return $\sigma$
$(m^*,\sigma^*) \leftarrow *$ .	$\mathcal{A}^{\mathcal{O}_{ ext{Sign}}(\cdot,\cdot),\mathcal{O}_{ ext{Leak}}(\cdot)}(pp,vk)$	$\mathcal{O}_{\text{LEAK}}(L)$ : #at most $\kappa$ leakage bits
If $((m^*, \sigma^*) \notin$	$\mathcal{Q}_{id}) \wedge (Vrfy(vk, m^*, \sigma^*) = 1):$	If $\ell +  L(sk)  > \kappa$ : Return $\perp$
Return	1;	$\ell := \ell +  L(sk) $
Else: Return	0	Return $L(sk)$

**Fig. 1.** The  $(\kappa, \mathcal{T})$ -sLTR-CMA security experiment  $\mathsf{Exp}_{\mathsf{SIG},\mathcal{A},\kappa,\mathcal{T}}^{\mathsf{sltr-cma}}$  for SIG, where id denotes the identity function and |L(sk)| denotes the bit-length of L(sk).

Remark 1 (On the formalization of sLTR-CMA security). In the experiment  $\operatorname{Exp}_{\mathsf{SIG},\mathcal{A},\kappa,\mathcal{T}}^{\mathsf{shtr-cma}}$  defined in Fig. 1, oracle  $\mathcal{O}_{\mathsf{SIGN}}$  captures the ability of the adversary to implement tampering attacks and obtain signatures under tampered signing keys T(sk) with  $T \in \mathcal{T}$ , and oracle  $\mathcal{O}_{\mathsf{LEAK}}$  captures the ability of the adversary to implement leakage attacks and obtain bounded leakage information L(sk) about the signing key.

For the adversary to win, the condition  $(m^*, \sigma^*) \notin \mathcal{Q}_{id}$  is the minimal restriction on the adversary's forgery, since otherwise the adversary can query  $\mathcal{O}_{SIGN}(id, m^*)$  for an arbitrary message  $m^*$  to obtain a signature  $\sigma^*$  and simply output  $(m^*, \sigma^*)$  as the forgery, and as a result, the adversary would trivially win and it is impossible to achieve the above security.

If we replace the condition  $(m^*, \sigma^*) \notin \mathcal{Q}_{id}$  with a stronger one, namely requiring  $m^*$  to be different from all messages that the adversary has queried  $\mathcal{O}_{SIGN}$ , we call it (non-strong) LTR-CMA security with standard existential unforgeability. Furthermore, if  $\mathcal{T}$  contains only the identity function id, we obtain the leakageresilience security, while if  $\kappa = 0$ , we obtain the tamper-resilience security. If both  $\mathcal{T} = \{id\}$  and  $\kappa = 0$ , we recover the standard EUF-CMA security.

#### 3.2 Construction of SIG from MDDH

Now we present our direct construction of sLTR-CMA secure SIG scheme over asymmetric pairing groups based on the MDDH assumptions. Let  $\mathcal{D}_k$  be a matrix distribution with  $k \in \mathbb{N}$ , and let  $\mathcal{H}$  be a family of collision resistant hash functions from  $\{0,1\}^*$  to  $\mathbb{Z}_p$ . Our SIG scheme SIG = (Setup, Gen, Sign, Vrfy) is shown in Fig. 2, where the message space is  $\mathcal{M} = \{0,1\}^*$  and the secret key space is  $\mathcal{SK} = \mathbb{Z}_p^{(k+1)\times(k+1)}$ . Correctness of SIG follows by inspection: for any honestly generated signature  $\sigma = ([\mathbf{c}]_1, [\mathbf{d}]_1)$ , we have  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}$  with  $\mathbf{w} \leftarrow_{\mathbb{S}} \mathbb{Z}_p^k$  and  $[\mathbf{d}]_1 = \mathbf{K}[\mathbf{c}]_1 + [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1 \mathbf{w} = [(\mathbf{K} + \mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1$ , which directly implies

$$e([\mathbf{c}^{\top}]_{1}, [(\mathbf{K}^{\top} + \mathbf{K}_{0}^{\top} + \tau\mathbf{K}_{1}^{\top})\mathbf{A}]_{2}) = e([\mathbf{c}^{\top}(\mathbf{K}^{\top} + \mathbf{K}_{0}^{\top} + \tau\mathbf{K}_{1}^{\top})]_{1}, [\mathbf{A}]_{2}) = e([\mathbf{d}^{\top}]_{1}, [\mathbf{A}]_{2}).$$

Moreover, since **U** output by  $\mathcal{D}_k$  is of full rank,  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w} \neq [\mathbf{0}]_1$  holds as long as  $\mathbf{w} \neq \mathbf{0}$  holds, which happens with overwhelming probability  $1 - 1/p^k$ .

Next, we show its  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CMA security under  $\kappa \leq \log p - \Omega(\lambda)$  bits of leakage information and under the set of affine functions

$$\mathcal{T}_{\mathsf{aff}} = \{ T_{(a,\mathbf{B})} : \mathbf{K} \in \mathcal{SK} \mapsto a\mathbf{K} + \mathbf{B} \in \mathcal{SK} \mid a \in \mathbb{Z}_p, \mathbf{B} \in \mathcal{SK} \}.$$
(1)

**Theorem 1** (( $\kappa$ ,  $\mathcal{T}_{aff}$ )-sLTR-CMA Security of SIG). Let  $\kappa \leq \log p - \Omega(\lambda)$  and let  $\mathcal{T}_{aff}$  be the set of affine functions defined in (1). Assume that the  $\mathcal{D}_k$ -MDDH assumption holds over both  $\mathbb{G}_1$  and  $\mathbb{G}_2$ , and  $\mathcal{H}$  is collision-resistant. Then the SIG scheme in Fig. 2 is ( $\kappa$ ,  $\mathcal{T}_{aff}$ )-sLTR-CMA secure.

Concretely, for any PPT adversary  $\mathcal{A}$  who makes at most Q times of  $\mathcal{O}_{SIGN}$  queries, there exist PPT adversaries  $\mathcal{B}_1, \dots, \mathcal{B}_5$ , such that

$$\begin{split} \mathsf{Adv}^{\mathsf{slr-cma}}_{\mathsf{SIG},\mathcal{A},\kappa,\mathcal{T}_{\mathsf{aff}}}(\lambda) &\leq \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_3}(\lambda) \\ &+ Q \cdot \left(\mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}_4}(\lambda) + \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}_5}(\lambda)\right) + 2^{-\Omega(\lambda)}. \end{split}$$



**Fig. 2.** Construction of SIG = (Setup, Gen, Sign, Vrfy) based on MDDH, where the framed boxes and the gray boxes are used to help explain the intuitions behind the construction in Remark 3.

The proof of Theorem 1 is postponed to Subsect. 3.3. Before presenting the formal proof, we give a detailed efficiency analysis and explain the main intuitions of our SIG construction in the following two remarks, respectively.

**Remark 2 (Efficiency of our SIG).** Let  $x \cdot \mathbb{G}$  denote x elements in a group  $\mathbb{G}$ . Our SIG scheme in Fig. 2 is parameterized by the MDDH parameter  $k \in \mathbb{N}$ , and has public parameter  $pp : (3k^2 + 3k) \cdot \mathbb{G}_1 + (3k^2 + 3k) \cdot \mathbb{G}_2$ , verification key  $vk : (k^2 + k) \cdot \mathbb{G}_2$ , signing key  $sk : (k^2 + 2k + 1) \cdot \mathbb{Z}_p$  and signature  $\sigma : (2k+2) \cdot \mathbb{G}_1$ . The verification involves  $(2k^2 + 2k)$  pairing operations.

For k = 1, we get an efficient SIG scheme with  $pp : 6 \cdot \mathbb{G}_1 + 6 \cdot \mathbb{G}_2$ , verification key  $vk : 2 \cdot \mathbb{G}_2$ , signing key  $sk : 4 \cdot \mathbb{Z}_p$  and signature  $\sigma : 4 \cdot \mathbb{G}_1$ , and the verification involves only 4 pairing operations. The resulting SIG scheme is  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CMA secure based on the standard SXDH assumption, and supports  $\kappa = \log p - \Omega(\lambda)$  bits key leakage. The leakage rate (i.e.,  $\kappa$ /bit-length of sk) is  $\frac{\log p - \Omega(\lambda)}{4 \log p} = \frac{1}{4} - o(1)$  asymptotically as p grows.

**Remark 3 (Intuitions of our SIG).** On a high level, our SIG in Fig. 2 can be parsed as two components: the terms in framed boxes (which are related to  $\mathbf{K}_0$  and the terms in gray boxes (which are related to  $\mathbf{K}_0$  and  $\mathbf{K}_1$ ).

Our first idea is to let  $sk = \lfloor \mathbf{K} \rfloor$  involve only term of the first component. With such a design, to achieve sLTR-CMA security, we only need to analyze the first component in the leakage and tampering-resilient setting, while for the second component we can analyze it without being disturbed by the leakage and tampering attacks on it. Our second idea is to integrate the two components carefully during the generation (and verification) of signatures, such that the terms  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$ in the second component can trigger randomness of certain forms to hide (partial information of) the terms  $\mathbf{K}[\mathbf{c}]_1$  in the first component, so that the signatures generated under *tampered* signing keys do not leak much information about  $sk = \mathbf{K}$  beyond vk to the adversary in the sLTR-CMA security experiment. Consequently, the signing oracle is of no use to the adversary, and the sLTR-CMA security of our SIG essentially reduces to the security against no-message attacks (i.e., where the adversary obtains no signatures) in the key leakage setting, which is much easier to achieve and is mainly guaranteed by the first component.

Below we explain the intuitions behind these two components in more detail.

INTUITIONS BEHIND THE FIRST COMPONENT. Intuitively, the terms in framed boxes can be viewed as a *publicly verifiable* function on  $\mathbb{G}_1^{k+1}$ :

- the function is defined by  $sk = \mathbf{K}$ , and it maps  $[\mathbf{c}]_1 \in \mathbb{G}_1^{k+1}$  to  $\mathbf{K}[\mathbf{c}]_1$ ; - given  $vk = [\mathbf{K}^\top \mathbf{A}]_2$ , one can verify the correctness of function value

$$\mathbf{d}]_1 = \mathbf{K}[\mathbf{c}]_1 \tag{2}$$

publicly via pairing equations:

$$e([\mathbf{c}^{\top}]_1, [\mathbf{K}^{\top} \mathbf{A}]_2) = e([\mathbf{d}^{\top}]_1, [\mathbf{A}]_2).$$
(3)

Observe that (2) and (3) are equivalent under the  $\mathcal{D}_k$ -KerMDH assumption on  $[\mathbf{A}]_2$  (which is further implied by the  $\mathcal{D}_k$ -MDDH assumption according to Lemma 3), since otherwise  $[\mathbf{c}^\top \mathbf{K}^\top - \mathbf{d}^\top]_1$  constitutes a non-zero vector in the kernel of  $[\mathbf{A}]_2$ , which is hard to find under the  $\mathcal{D}_k$ -KerMDH assumption.

This publicly verifiable function enjoys a useful property:

- in the presence of only  $vk = [\mathbf{K}^{\top}\mathbf{A}]_2$ , the function value  $\mathbf{K}[\mathbf{c}]_1$  of any  $[\mathbf{c}]_1 \neq [\mathbf{0}]_1$  retains enough entropy from  $\mathbf{K}$ , so that it is information-theoretically hard to produce ( $[\mathbf{c}]_1, [\mathbf{d}]_1$ ) satisfying (2) (and thus computationally hard to satisfy (3) under MDDH assumption).

To see why this property holds more concretely, we can let  $\mathbf{a}^{\perp} \in \mathbb{Z}_p^{k+1}$  be a nonzero vector in the kernel of  $\mathbf{A} \in \mathbb{Z}_p^{(k+1) \times k}$  such that  $(\mathbf{a}^{\perp})^{\top} \mathbf{A} = \mathbf{0}$ , and sample  $sk = \mathbf{K} \leftarrow \mathbb{Z}_p^{(k+1) \times (k+1)}$  equivalently via

$$\mathbf{K} := \widetilde{\mathbf{K}} + \begin{bmatrix} \mathbf{a}^{\perp} \mathbf{k}^{\top} \end{bmatrix}$$

where  $\widetilde{\mathbf{K}} \leftarrow_{\$} \mathbb{Z}_{p}^{(k+1)\times(k+1)}$  and  $\mathbf{k} \leftarrow_{\$} \mathbb{Z}_{p}^{k+1}$ . On the one hand, note that  $\mathbf{k}$  is completely hidden in vk since  $vk = [\mathbf{K}^{\top}\mathbf{A}]_{2} = [(\widetilde{\mathbf{K}}^{\top} + [\mathbf{k}(\mathbf{a}^{\perp})^{\top}])\mathbf{A}]_{2} = [\widetilde{\mathbf{K}}^{\top}\mathbf{A}]_{2}$ . On

the other hand, for any  $[\mathbf{c}]_1 \neq [\mathbf{0}]_1$ , its function value is

$$\mathbf{K}[\mathbf{c}]_1 = (\widetilde{\mathbf{K}} + \mathbf{a}^{\perp} \mathbf{k}^{\top})[\mathbf{c}]_1 = \widetilde{\mathbf{K}}[\mathbf{c}]_1 + \begin{bmatrix} \mathbf{a}^{\perp} (\mathbf{k}^{\top} \mathbf{c}) \end{bmatrix}_1,$$

where the term  $\begin{bmatrix} \mathbf{a}^{\perp}(\mathbf{k}^{\top}\mathbf{c}) \end{bmatrix}$  is uniformly distributed over  $\text{Span}(\mathbf{a}^{\perp}) = \{\gamma \mathbf{a}^{\perp} | \gamma \in \mathbb{Z}_p\}$  due to the randomness of  $\mathbf{k}$  and non-zero of  $\mathbf{c}$ . So the function value  $\begin{bmatrix} \mathbf{K}[\mathbf{c}]_1 \\ \mathbf{k}^{\perp}(\mathbf{k}^{\top}\mathbf{c}) \end{bmatrix}$  has log p bits of entropy conditioned on vk, as shown by the term  $\begin{bmatrix} \mathbf{a}^{\perp}(\mathbf{k}^{\top}\mathbf{c}) \end{bmatrix}$ , and consequently it is hard to produce  $([\mathbf{c}]_1, [\mathbf{d}]_1)$  satisfying (2) and (3).

INSUFFICIENCY OF THE FIRST COMPONENT AND ARISING OF THE SECOND. The first component and the aforementioned useful property serve as the basis for the security of our SIG. In particular, if the adversary  $\mathcal{A}$  does not obtain any signatures in the security experiment, then it is hard for  $\mathcal{A}$  to forge a signature satisfying (3), since the function value  $\mathbf{K}[\mathbf{c}]_1$  has enough entropy (i.e.,  $\log p$  bits entropy) conditioned on vk. Moreover, the argument holds even if the adversary obtains bounded leakage information about  $sk = \mathbf{K}$ , as long as the amount of leakage  $\kappa$  satisfies  $\log p - \kappa \geq \Omega(\lambda)$  so that there are still  $\log p - \kappa \geq \Omega(\lambda)$  bits entropy left in  $\mathbf{K}[\mathbf{c}]_1$ . This shows the security against no-message attacks in the *leakage setting* of our SIG.

However, in the sLTR-CMA security experiment,  $\mathcal{A}$  can obtain signatures as many as it wants, under *tampered* signing keys  $T_{(a,\mathbf{B})}(sk) = a\mathbf{K} + \mathbf{B}$ . So  $\mathcal{A}$  will obtain multiple  $(a\mathbf{K} + \mathbf{B})[\mathbf{c}]_1$  contained in the signatures  $\sigma = ([\mathbf{c}]_1, [\mathbf{d}]_1)$ , which would leak additional information about  $sk = \mathbf{K}$  beyond vk.

To rescue the above arguments, we resort to the terms in gray boxes. Roughly speaking, we use  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  to hide (partial information of)  $(a\mathbf{K} + \mathbf{B})[\mathbf{c}]_1$  in the generation of  $[\mathbf{d}]_1$ :

$$[\mathbf{d}]_1 = \boxed{(a\mathbf{K} + \mathbf{B})[\mathbf{c}]_1} + [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}, \qquad (4)$$

so that  $\mathcal{A}$  will not learn much information about  $sk = \mathbf{K}$  beyond vk from the obtained signatures, and then we can use the above arguments to show the security of our SIG.

MORE EXPLANATIONS ABOUT THE SECOND COMPONENT. It remains to give the intuitions of the terms in gray boxes in more detail, and in particular, explain how  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  hide  $(a\mathbf{K} + \mathbf{B})[\mathbf{c}]_1$  in the generation of  $[\mathbf{d}]_1$  in (4).

From a high-level perspective, the terms in gray boxes can be viewed as the one-time simulation-sound (OTSS) NIZK scheme proposed by Kiltz and Wee [27, Section 3.3], and they essentially prove that  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}$  belongs to the linear subspace Span( $[\mathbf{U}]_1$ ): in the signing algorithm Sign of our SIG, the term  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  corresponds to the generation of OTSS-NIZK proof; in the

verification algorithm Vrfy, the term  $e([\mathbf{c}^{\top}]_1, [(\mathbf{K}_0^{\top} + \tau \mathbf{K}_1^{\top})\mathbf{A}]_2)$  corresponds to the verification of OTSS-NIZK proof.

- On the one hand, the generation and verification of OTSS-NIZK proofs do not involve any secret key, so they do not introduce additional elements to sk. This is very helpful in the key leakage and tampering-resilient setting, since the leakage and tampering of sk do not affect the terms in gray boxes, and we can use properties of this component without the need of considering any leakage and tampering.
- However, the OTSS property is insufficient for our purpose, since in the security experiment of SIG, the adversary can obtain multiple NIZK proofs  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  contained in the multiple signatures  $\sigma = ([\mathbf{c}]_1, [\mathbf{d}]_1)$ , rather than a single NIZK proof allowed in the OTSS property.

Instead, we resort to another property about the second component  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$ , namely it can trigger randomness of certain forms in a computationally indistinguishable way, as observed in [27]. To be more concrete, we can prove that the multiple pairs of

$$([\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}, [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1 \mathbf{w})$$

contained in the signatures that  $\mathcal{A}$  obtains are computationally indistinguishable from

$$([\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}, \quad [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1 \mathbf{w} + [[\gamma \mathbf{a}^{\perp}]_1]), \quad (5)$$

where  $\gamma \leftarrow \mathbb{Z}_p$  are randomnesses independently chosen for each pair, and  $\mathbf{a}^{\perp} \in \mathbb{Z}_p^{k+1}$  is a non-zero vector in the kernel of  $\mathbf{A} \in \mathbb{Z}_p^{(k+1) \times k}$  such that  $(\mathbf{a}^{\perp})^{\top} \mathbf{A} = \mathbf{0}$ , even conditioned on a single pair

$$([\mathbf{c}^*]_1, [(\mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1)$$

contained in  $\mathcal{A}$ 's forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$  in the case of  $\tau^* \neq \tau$ . Jumping ahead, this corresponds to the game sequence  $\{\mathsf{G}_{4.\eta.0}-\mathsf{G}_{4.\eta.4}\}_{0\leq\eta\leq Q-1}$  and  $\mathsf{G}_{4.Q.0}$  in our security proof in Subsect. 3.3. This property is different from OTSS and is enjoyed by this specific NIZK scheme (in other words, other OTSS-NIZK schemes may not enjoy this property).

However, this property holds only in the case of  $\tau^* \neq \tau$ , i.e., when the following bad event never occurs.

- TagColl: the tag  $\tau^* = H(vk, m^*, [\mathbf{c}^*]_1)$  involved in  $\mathcal{A}$ 's forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$  is *identical* to the tag  $\tau = H(vk' = [(a\mathbf{K} + \mathbf{B})^\top \mathbf{A}]_2, m, [\mathbf{c}]_1)$  involved in some signatures that  $\mathcal{A}$  obtains under *tampered* signing keys.

So to apply this property, we need to first show that the event TagColl can hardly occur. This might be the most technical part of our security proof in Subsect. 3.3 and corresponds to Claim 2 therein. Roughly speaking, we divide TagColl into three sub-cases and analyze them individually to show that they all rarely occur,

by utilizing the concrete algebraic structures of our construction, based on the collision resistance of H and on the MDDH assumption.

Consequently, we can apply the above property, and show that the terms  $[[\gamma \mathbf{a}^{\perp}]_1]$  triggered by  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  in (5) can be used to hide the partial information of  $(a\mathbf{K} + \mathbf{B})[\mathbf{c}]_1$  in the generation of  $[\mathbf{d}]_1$  in (4). To see this more concretely, again, we sample  $sk = \mathbf{K}$  equivalently via

$$\mathbf{K} := \widetilde{\mathbf{K}} + \lfloor \mathbf{a}^{\perp} \mathbf{k}^{ op}$$

where  $\widetilde{\mathbf{K}} \leftarrow \mathbb{Z}_p^{(k+1) \times (k+1)}$  and  $\mathbf{k} \leftarrow \mathbb{Z}_p^{k+1}$ , and then we have

$$\begin{aligned} [\mathbf{d}]_{1} &= \left[ (a\mathbf{K} + \mathbf{B})[\mathbf{c}]_{1} \right] + [\mathbf{K}_{0} + \tau \mathbf{K}_{1})\mathbf{U}]_{1}\mathbf{w} \\ &\approx \left[ (a\mathbf{K} + \mathbf{B})[\mathbf{c}]_{1} \right] + [\mathbf{K}_{0} + \tau \mathbf{K}_{1})\mathbf{U}]_{1}\mathbf{w} + \left[ [\gamma \mathbf{a}^{\perp}]_{1} \right] \\ &= \left[ (a(\widetilde{\mathbf{K}} + \mathbf{a}^{\perp}\mathbf{k}^{\top}) + \mathbf{B})[\mathbf{c}]_{1} \right] + [\mathbf{K}_{0} + \tau \mathbf{K}_{1})\mathbf{U}]_{1}\mathbf{w} + \left[ [\gamma \mathbf{a}^{\perp}]_{1} \right] \\ &= \left[ (a\widetilde{\mathbf{K}} + \mathbf{B})[\mathbf{c}]_{1} \right] + [\mathbf{K}_{0} + \tau \mathbf{K}_{1})\mathbf{U}]_{1}\mathbf{w} + \left[ a\mathbf{a}^{\perp}(\mathbf{k}^{\top}\mathbf{c}) \right] + \left[ \gamma \mathbf{a}^{\perp}_{\perp} \right]_{1}. \end{aligned}$$

Note that the term  $\begin{bmatrix} \gamma \mathbf{a}^{\perp} \\ \gamma \mathbf{a}^{\perp} \end{bmatrix}$  perfectly hides  $\begin{bmatrix} a \mathbf{a}^{\perp} (\mathbf{k}^{\top} \mathbf{c}) \end{bmatrix} = (a \mathbf{k}^{\top} \mathbf{c}) \mathbf{a}^{\perp}$  by the randomness of  $\gamma \leftarrow \mathbb{Z}_p$ , thus the information of  $\mathbf{k}$  is perfectly hidden in the multiple signatures generated under tampered signing keys.

<u>PUTTING Two COMPONENTS TOGETHER.</u> Overall, the two components enjoy specific properties and we carefully integrate the two components in our SIG construction to achieve sLTR-CMA security: the terms  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  in the second component can trigger randomness in the form of  $[\gamma \mathbf{a}^{\perp}]_1$ , which can then be used to hide the terms  $(a\mathbf{K} + \mathbf{B})[\mathbf{c}]_1$  in the first component, so that the signatures generated under *tampered* signing keys do not leak much information about  $sk = \mathbf{K}$  beyond vk to the adversary, and finally the sLTR-CMA security of our SIG follows from the useful property of the first component in the *key leakage* setting.

## 3.3 Proof of Theorem 1

Now we present the formal proof of Theorem 1. Let  $\mathcal{A}$  be any PPT adversary against the  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CMA security of SIG, where  $\mathcal{A}$  makes Q times of  $\mathcal{O}_{SIGN}$  queries. We prove the theorem via a sequence of games  $G_0 - G_3$ ,  $\{G_{4,\eta,0} - G_{4,\eta,4}\}_{0 \le \eta \le Q-1}$  and  $G_{4,Q,0}$ , where  $G_0$  is the  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CMA experiment (cf. Fig. 1), and in  $G_{4,Q,0}$ ,  $\mathcal{A}$  has a negligible advantage. A brief description of differences between adjacent games is summarized in Table 1.

**Game G<sub>0</sub>:** This is the  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CMA experiment (cf. Fig. 1).

**Table 1.** Brief Description of Games  $\mathbf{G}_0 - \mathbf{G}_3$ ,  $\{\mathbf{G}_{4,\eta,0} - \mathbf{G}_{4,\eta,4}\}_{0 \le \eta \le Q-1}$  and  $\mathbf{G}_{4,Q,0}$  for the  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CMA security proof of SIG, where the differences between adjacent games are highlighted in gray boxes. Here column " $\mathcal{O}_{\text{SIGN}}(T_{(a,\mathbf{B})} \in \mathcal{T}_{aff}, m)$ " suggests how a signature  $\sigma = ([\mathbf{c}]_1, [\mathbf{d}]_1)$  is generated: sub-column " $[\mathbf{c}]_1 \leftarrow s$ " refers to the space from which  $[\mathbf{c}]_1$  is chosen; sub-column " $[\mathbf{d}]_1 =$ " shows the computation of  $[\mathbf{d}]_1$ , where  $sk' = \mathbf{K}' = T_{(a,\mathbf{B})}(sk) = a\mathbf{K} + \mathbf{B}$  denotes the tampered signing key. Column " $\mathcal{O}_{\text{LEAK}}$ " shows the output returned by  $\mathcal{O}_{\text{LEAK}}$ . Column "Win's additional check for forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$ " describes the additional check that  $\mathcal{A}$ 's forgery wins, besides the routine check  $(m^*, \sigma^*) \notin \mathcal{Q}_{id} \wedge e([\mathbf{c}^{*\top}]_1, [(\mathbf{K}^\top + \mathbf{K}_0^\top + \tau^* \mathbf{K}_1^\top) \mathbf{A}]_2) = e([\mathbf{d}^{*\top}]_1, [\mathbf{A}]_2) \wedge [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1$ , where  $\mathcal{Q}_{\text{tag}}$  denotes the set of  $\tau$  generated in  $\mathcal{O}_{\text{SIGN}}$  queries.

		$\mathcal{O}_{\mathrm{Sign}}(T_{(a,\mathbf{B})} \in \mathcal{T}_{\mathrm{aff}}, m)$		$\mathcal{O}_{-}(I)$	Win's additional check for forgery	Tradificantian / Annual time
		$[\mathbf{c}]_1 \leftarrow_{\mathrm{s}}$	$[\mathbf{d}]_1 =$	$O_{\text{LEAK}}(L)$	$(m^*,\sigma^*=([\mathbf{c}^*]_1,[\mathbf{d}^*]_1))$	Justification/Assumption
	$G_0$	$Span([\mathbf{U}]_1)$	$\mathbf{K}'[\mathbf{c}]_1 + [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1 \mathbf{w}$	L(sk)		$(\kappa, \mathcal{T}_{aff})$ -sLTR-CMA experiment
	$G_1$	$Span([\mathbf{U}]_1)$	$(\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1$	L(sk)		$G_0=G_1$
	$G_2$	$Span([\mathbf{U}]_1)$	$(\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1$		$[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 +  au^* \mathbf{K}_1) \mathbf{c}^*]_1$	$\mathcal{D}_k$ -KerMDH on $[\mathbf{A}]_2$
	$G_3$	$Span([\mathbf{U}]_1)$	$(\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1$		$[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1, \ \tau^* \notin \mathcal{Q}_{tag}$	Collision-resistance of $\mathcal{H}$ & $\mathcal{D}_k$ -KerMDH on $[\mathbf{A}]_2$
	$G_{4.\eta.0}$	$Span([\mathbf{U}]_1)$	$\begin{cases} (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 & \text{with } \gamma \leftarrow_{\mathfrak{s}} \mathbb{Z}_p, \\ & \text{for the first } \eta \text{ queries} \\ (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1, \text{ for other queries} \end{cases}$	L(sk)	$[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1, \tau^* \notin \mathcal{Q}_{tag}$	$G_3=G_{4.0.0}$
15	$G_{4.\eta.1}$	$\begin{cases} \mathbb{G}_1^{k+1} , & \text{for the } (\eta+1)\text{-th query} \\ \\ Span([\mathbf{U}]_1),  \text{for other queries} \end{cases}$	$\begin{cases} (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \text{ with } \gamma \leftarrow \mathbb{Z}_p, \\ \text{ for the first } \eta \text{ queries} \\ (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1, \text{ for other queries} \end{cases}$	L(sk)	$[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1, \tau^* \notin \mathcal{Q}_{tag}$	$\mathcal{D}_k$ -MDDH on $[\mathbf{U}]_1$
	G <sub>4.η.2</sub>	$\begin{cases} \mathbb{G}_1^{k+1}, & \text{for the } (\eta+1)\text{-th query} \\ Span([\mathbf{U}]_1), \text{for other queries} \end{cases}$	$\begin{cases} (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \text{ with } \gamma \leftarrow \mathbb{Z}_p, \\ \text{for the first } \eta \text{ queries} \\ (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1, \text{ for other queries} \end{cases}$	L(sk)	$\begin{split} [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1, \tau^* \notin \mathcal{Q}_{tag}, \\ [\mathbf{c}^*]_1 \in Span([\mathbf{U}]_1) \end{split}$	Statistical arguments using the leftover entropy in $\mathbf{K}_0, \mathbf{K}_1$
	G <sub>4.η.3</sub>	$\begin{cases} \mathbb{G}_1^{k+1}, & \text{for the } (\eta+1)\text{-th query} \\ Span([\mathbf{U}]_1), \text{for other queries} \end{cases}$	$\begin{cases} (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \text{ with } \gamma \leftarrow \mathbb{Z}_p, \\ \text{for the first } \eta + 1 \\ (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1, \text{ for other queries} \end{cases}$	L(sk)	$\begin{split} [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1, \tau^* \notin \mathcal{Q}_{tag}, \\ [\mathbf{c}^*]_1 \in Span([\mathbf{U}]_1) \end{split}$	Statistical arguments using the leftover entropy in $\mathbf{K}_0, \mathbf{K}_1$
	G <sub>4.η.4</sub>	$\begin{cases} \mathbb{G}_1^{k+1}, & \text{for the } (\eta+1)\text{-th query} \\ Span([\mathbf{U}]_1), \text{for other queries} \end{cases}$	$\begin{cases} (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \text{ with } \gamma \leftarrow_{\mathbb{S}} \mathbb{Z}_p, \\ \text{ for the first } \eta + 1 \text{ queries} \\ (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1, \text{ for other queries} \end{cases}$	L(sk)	$[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1, \tau^* \notin \mathcal{Q}_{tag},$ $\underbrace{[\mathbf{c}^*]_1 \in Span([\mathbf{U}]_1)}$	Statistical arguments using the leftover entropy in $\mathbf{K}_0, \mathbf{K}_1$
	$G_{4.\eta+1.0}$	$Span([\mathbf{U}]_1)$	$\begin{cases} (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \text{ with } \gamma \leftarrow_{\mathbb{S}} \mathbb{Z}_p, \\ \text{ for the first } \eta + 1 \text{ queries} \\ (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1, \text{ for other queries} \end{cases}$	L(sk)	$[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1, \tau^* \notin \mathcal{Q}_{tag}$	$\mathcal{D}_k$ -MDDH on $[\mathbf{U}]_1$
	$G_{4.Q.0}$	$Span([\mathbf{U}]_1)$	$(\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \text{ with } \gamma \leftarrow \mathbb{Z}_p$	L(sk)	$ \begin{bmatrix} [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1, \tau^* \notin \mathcal{Q}_{tag} \end{bmatrix} $	$\Pr[Win] = negl:$ statistical arguments using the leftover entropy in K

Let  $pp = (gpar, [\mathbf{U}]_1, [\mathbf{K}_0\mathbf{U}]_1, [\mathbf{K}_1\mathbf{U}]_1, [\mathbf{A}]_2, [\mathbf{K}_0^{\top}\mathbf{A}]_2, [\mathbf{K}_1^{\top}\mathbf{A}]_2, H)$  and  $(vk = [\mathbf{K}^{\top}\mathbf{A}]_2, sk = \mathbf{K})$ . In this game, when answering an  $\mathcal{O}_{\text{SIGN}}$  query  $(T_{(a,\mathbf{B})} \in \mathcal{T}_{aff}, m)$ , the challenger computes the tampered key  $sk' = \mathbf{K}' := T_{(a,\mathbf{B})}(sk) = a\mathbf{K} + \mathbf{B}$  and  $vk' := [\mathbf{K}'^{\top}\mathbf{A}]_2$ , samples  $\mathbf{w} \leftarrow \mathbb{Z}_p^k$ , and computes  $[\mathbf{c}]_1 := [\mathbf{U}]_1\mathbf{w}$ ,  $\tau := H(vk', m, [\mathbf{c}]_1)$  and  $[\mathbf{d}]_1 := \mathbf{K}'[\mathbf{c}]_1 + [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  using the tampered key  $sk' = \mathbf{K}'$ . Then, the challenger returns  $\sigma := ([\mathbf{c}]_1, [\mathbf{d}]_1)$  to  $\mathcal{A}$ , and further puts  $(m, \sigma)$  to set  $\mathcal{Q}_{id}$  if  $T_{(a,\mathbf{B})}$  is the identity function id. For an  $\mathcal{O}_{\text{LEAK}}$  query L, the challenger returns L(sk) to  $\mathcal{A}$  if the total leakage length is bounded by  $\kappa$ .

At the end of the game,  $\mathcal{A}$  outputs a forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$ . Let Win denote the event that

$$(m^*,\sigma^*) \notin \mathcal{Q}_{\mathsf{id}} \land e([\mathbf{c}^{*\top}]_1,[(\mathbf{K}^\top + \mathbf{K}_0^\top + \tau^*\mathbf{K}_1^\top)\mathbf{A}]_2) = e([\mathbf{d}^{*\top}]_1,[\mathbf{A}]_2) \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1,$$

where  $\tau^* := H(vk, m^*, [\mathbf{c}^*]_1)$ . By definition,  $\mathsf{Adv}^{\mathsf{sltr-cma}}_{\mathsf{SIG},\mathcal{A},\kappa,\mathcal{T}_{\mathsf{aff}}}(\lambda) = \Pr_0[\mathsf{Win}]$ .

<u>**Game G1:**</u> It is the same as G<sub>0</sub>, except that, when answering  $\mathcal{O}_{SIGN}$  queries, the challenger computes  $[\mathbf{d}]_1 := (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1$  directly from  $[\mathbf{c}]_1, \tau$  and  $(\mathbf{K}', \mathbf{K}_0, \mathbf{K}_1)$ , without using the vector  $\mathbf{w}$  for  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}$ .

Since  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}$ , this change is conceptual and  $\Pr_0[\mathsf{Win}] = \Pr_1[\mathsf{Win}]$ .

Game  $G_2$ : It is the same as  $G_1$ , except that, the event Win is now defined as

$$(m^*, \sigma^*) \notin \mathcal{Q}_{\mathsf{id}} \land [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1 \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1$$

Claim 1.  $|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]| \leq \mathsf{Adv}_{\mathcal{D}_k, \mathbb{G}_2, \mathcal{B}_1}^{\mathsf{mddh}}(\lambda) + 1/(p-1)$  for a PPT adversary  $\mathcal{B}_1$  against the  $\mathcal{D}_k$ -MDDH assumption on  $[\mathbf{A}]_2$ .

*Proof.* By VrfyBad denote the event that  $\mathcal{A}$ 's forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$ satisfying  $e([\mathbf{c}^{*\top}]_1, [(\mathbf{K}^\top + \mathbf{K}_0^\top + \tau^* \mathbf{K}_1^\top) \mathbf{A}]_2) = e([\mathbf{d}^{*\top}]_1, [\mathbf{A}]_2)$  but  $[\mathbf{d}^*]_1 \neq [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1$ . Clearly,  $\mathsf{G}_2$  is identical to  $\mathsf{G}_1$  unless VrfyBad occurs, thus  $|\Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}]| \leq \Pr_2[\mathsf{VrfyBad}]$ . To bound  $\Pr_2[\mathsf{VrfyBad}]$ , observe that VrfyBad implies that

$$e(\underbrace{[\mathbf{d}^{*\top}]_1 - [\mathbf{c}^{*\top}(\mathbf{K}^\top + \mathbf{K}_0^\top + \tau^*\mathbf{K}_1^\top)]_1}_{\neq [\mathbf{0}]_1}, [\mathbf{A}]_2) = [\mathbf{0}]_T,$$

i.e.,  $[\mathbf{d}^{*\top}]_1 - [\mathbf{c}^{*\top}(\mathbf{K}^{\top} + \mathbf{K}_0^{\top} + \tau^* \mathbf{K}_1^{\top})]_1$  is a non-zero vector in the kernel of  $[\mathbf{A}]_2$ . Thus VrfyBad rarely occurs under the  $\mathcal{D}_k$ -KerMDH assumption on  $[\mathbf{A}]_2$ , which is further implied by the  $\mathcal{D}_k$ -MDDH assumption on  $[\mathbf{A}]_2$  according to Lemma 3. Consequently,  $\Pr_2[\mathsf{VrfyBad}] \leq \mathsf{Adv}_{\mathcal{D}_k, \mathbb{G}_2, \mathcal{B}_1}^{\mathsf{mddh}}(\lambda) + 1/(p-1)$  and Claim 1 follows.

<u>**Game G\_3:**</u> It is the same as  $G_2$ , except that, when answering  $\mathcal{O}_{SIGN}$  queries, the challenger also puts  $\tau$  to a set  $\mathcal{Q}_{tag}$ , and for the forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$  output by  $\mathcal{A}$ , the event Win is now defined as

$$(m^*,\sigma^*) \notin \mathcal{Q}_{\mathsf{id}} \wedge [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1)\mathbf{c}^*]_1 \wedge [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \wedge \tau^* \notin \mathcal{Q}_{\mathsf{tag}} .$$

Claim 2.  $|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]| \leq \mathsf{Adv}_{\mathcal{H},\mathcal{B}_2}^{\mathsf{cr}}(\lambda) + \mathsf{Adv}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_3}^{\mathsf{mddh}}(\lambda) + 1/(p-1) + 2^{-\Omega(\lambda)}$  for PPT adversaries  $\mathcal{B}_2$  against the collision-resistance of  $\mathcal{H}$  and  $\mathcal{B}_3$  against the  $\mathcal{D}_k$ -MDDH assumption on  $[\mathbf{A}]_2$ .

*Proof.* By TagColl denote the event that  $\mathcal{A}$ 's forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$  satisfying

 $(m^*,\sigma^*) \notin \mathcal{Q}_{\mathsf{id}} \ \land \ [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1)\mathbf{c}^*]_1 \ \land \ [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \ \land \ \tau^* \in \mathcal{Q}_{\mathsf{tag}}.$ 

Clearly,  $G_2$  and  $G_3$  are the same until TagColl occurs, thus  $|\Pr_2[Win] - \Pr_3[Win]| \le \Pr_3[TagColl]$ .

To bound  $Pr_3[TagColl]$ , we divide TagColl into the following three cases:

• Case 1: There exists an  $\mathcal{O}_{SIGN}$  query  $(T_{(a,\mathbf{B})} \in \mathcal{T}_{aff}, m)$ , such that

$$\begin{aligned} \tau^* &= H(vk, m^*, [\mathbf{c}^*]_1) = H(vk', m, [\mathbf{c}]_1) = \tau \in \mathcal{Q}_{\mathsf{tag}} \\ \text{but} \quad (vk, m^*, [\mathbf{c}^*]_1) \neq (vk', m, [\mathbf{c}]_1), \end{aligned}$$

where vk' is the tampered verification key involved in this  $\mathcal{O}_{\text{SIGN}}$  query. Clearly, Case 1 suggests a collision of H, thus  $\Pr_3[\text{Case 1}] \leq \text{Adv}_{\mathcal{H},\mathcal{B}_2}^{\text{cr}}(\lambda)$ .

• Case 2: There exists an  $\mathcal{O}_{SIGN}$  query  $(T_{(a,\mathbf{B})} \in \mathcal{T}_{aff}, m)$ , such that

$$\begin{split} \tau^* &= H(vk, m^*, [\mathbf{c}^*]_1) = H(vk', m, [\mathbf{c}]_1) = \tau \in \mathcal{Q}_{\mathsf{tag}} \\ \text{but} \quad (vk, m^*, [\mathbf{c}^*]_1) = (vk', m, [\mathbf{c}]_1) \ \land \ T_{(a, \mathbf{B})} = \mathsf{id}, \end{split}$$

where id denotes the identity function.

Since  $T_{(a,\mathbf{B})} = \mathrm{id}$ , the tampered signing key  $sk' = \mathbf{K}'$  is in fact the original key  $sk = \mathbf{K}$ , and the tuple  $(m, \sigma = ([\mathbf{c}]_1, [\mathbf{d}]_1))$  involved in this  $\mathcal{O}_{\mathrm{SIGN}}$  query is added to  $\mathcal{Q}_{\mathrm{id}}$ .

Now we show that this case can never occur. On the one hand, TagColl requires  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1)) \notin \mathcal{Q}_{i\mathbf{d}}$  and this case requires  $(m^*, [\mathbf{c}^*]_1) = (m, [\mathbf{c}]_1)$ , so it follows that  $[\mathbf{d}^*]_1 \neq [\mathbf{d}]_1$ . On the other hand, TagColl requires  $[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1)\mathbf{c}^*]_1$  and this case requires  $\tau^* = \tau$ , so we have  $[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1)\mathbf{c}^*]_1 = [(\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1 = [\mathbf{d}]_1$ , which leads to a contradiction. Therefore, this case can never occur, i.e.,  $\Pr_3[\mathbf{Case 2}] = 0$ .

• Case 3: There exists an  $\mathcal{O}_{SIGN}$  query  $(T_{(a,\mathbf{B})} \in \mathcal{T}_{aff}, m)$ , such that

$$\begin{aligned} \tau^* &= H(vk, m^*, [\mathbf{c}^*]_1) = H(vk', m, [\mathbf{c}]_1) = \tau \in \mathcal{Q}_{\mathsf{tag}} \\ \text{but} \quad (vk, m^*, [\mathbf{c}^*]_1) = (vk', m, [\mathbf{c}]_1) \ \land \ T_{(a, \mathbf{B})} \neq \mathsf{id}. \end{aligned}$$

Note that vk' = vk means that  $[\mathbf{K}'^{\top}\mathbf{A}]_2 = [\mathbf{K}^{\top}\mathbf{A}]_2$ , where  $sk' = \mathbf{K}' = T_{(a,\mathbf{B})}(sk) = a\mathbf{K} + \mathbf{B}$  is the tampered signing key. By rearranging terms, it follows that  $[((a-1)\mathbf{K}^{\top} + \mathbf{B}^{\top})\mathbf{A}]_2 = [\mathbf{0}]_2$ . This shows that  $(a-1)\mathbf{K}^{\top} + \mathbf{B}^{\top}$  is a matrix in the kernel of  $[\mathbf{A}]_2$ . We claim that  $(a-1)\mathbf{K}^{\top} + \mathbf{B}^{\top}$  is a non-zero matrix with overwhelming probability  $1 - 2^{-\Omega(\lambda)}$ , which will be shown later. Thus by the  $\mathcal{D}_k$ -KerMDH assumption on  $[\mathbf{A}]_2$  (which is further implied

by the  $\mathcal{D}_k$ -MDDH assumption on  $[\mathbf{A}]_2$  according to Lemma 3), this case can

rarely occurs, and we have  $\Pr_3[\mathbf{Case 3}] \leq \operatorname{Adv}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_3}^{\operatorname{mddh}}(\lambda) + 1/(p-1) + 2^{-\Omega(\lambda)}$ . It remains to show the claim that the matrix  $(a-1)\mathbf{K}^\top + \mathbf{B}^\top$  is non-zero with overwhelming probability  $1 - 2^{-\Omega(\lambda)}$ . By the fact that  $T_{(a,\mathbf{B})} \neq \operatorname{id}$ , there are two sub-cases. The first sub-case is a = 1 and  $\mathbf{B} \neq \mathbf{0}$ . In this sub-case, we have  $(a-1)\mathbf{K}^{\top} + \mathbf{B}^{\top} = \mathbf{B}^{\top}$ , which is clearly non-zero. The second subcase is  $a \neq 1$ . In this sub-case, we will show that  $sk = \mathbf{K}$  contains enough entropy from  $\mathcal{A}$ 's view, so that the matrix  $(a-1)\mathbf{K}^{\top} + \mathbf{B}^{\top}$  is non-zero with overwhelming probability. To see this, let  $\mathbf{u}^{\perp} \in \mathbb{Z}_p^{k+1}$  (resp.,  $\mathbf{a}^{\perp} \in \mathbb{Z}_p^{k+1}$ ) be an arbitrary non-zero vector in the kernel of  $\mathbf{U}$  (resp.,  $\mathbf{A} \in \mathbb{Z}_p^{(resp.)}$ )  $(\mathbf{u}^{\perp})^{\top}\mathbf{U} = \mathbf{0}$  (resp.,  $(\mathbf{a}^{\perp})^{\top}\mathbf{A} = \mathbf{0}$ ). For the convenience of our analysis, we sample  $sk = \mathbf{K} \leftarrow_{\mathbb{S}} \mathbb{Z}_p^{(k+1) \times (k+1)}$  equivalently via

$$\mathbf{K} := \widetilde{\mathbf{K}} + \mu \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top}$$

where  $\widetilde{\mathbf{K}} \leftarrow_{\$} \mathbb{Z}_p^{(k+1) \times (k+1)}$  and  $\mu \leftarrow_{\$} \mathbb{Z}_p$ . Below we analyze the information about  $\mu$  that  $\mathcal{A}$  may obtain in  $G_3$ .

- Firstly, the verification key vk is

$$[\mathbf{K}^{\top}\mathbf{A}]_2 = [(\widetilde{\mathbf{K}}^{\top} + \mu \mathbf{u}^{\perp} (\mathbf{a}^{\perp})^{\top})\mathbf{A}]_2 = [\widetilde{\mathbf{K}}^{\top}\mathbf{A}]_2,$$

thus  $\mu$  is completely hidden.

In  $\mathcal{O}_{SIGN}$  queries, the tampered verification key vk' is

$$\begin{split} [\mathbf{K}'^{\top}\mathbf{A}]_2 &= [(a\mathbf{K}^{\top} + \mathbf{B}^{\top})\mathbf{A}]_2 = [\left(a(\widetilde{\mathbf{K}}^{\top} + \mu \mathbf{u}^{\perp}(\mathbf{a}^{\perp})^{\top}) + \mathbf{B}^{\top}\right)\mathbf{A}]_2 \\ &= [(a\widetilde{\mathbf{K}}^{\top} + \mathbf{B}^{\top})\mathbf{A}]_2, \end{split}$$

thus  $\mu$  is completely hidden. Moreover, since  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}$  with  $\mathbf{w} \leftarrow \mathbb{Z}_p^k$ , we have

$$\begin{split} [\mathbf{d}]_1 &= (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 = (a\mathbf{K} + \mathbf{B} + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{U}]_1 \mathbf{w} \\ &= \left(a(\widetilde{\mathbf{K}} + \mu \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top}) + \mathbf{B} + \mathbf{K}_0 + \tau \mathbf{K}_1\right)[\mathbf{U}]_1 \mathbf{w} \\ &= (a\widetilde{\mathbf{K}} + \mathbf{B} + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{U}]_1 \mathbf{w}, \end{split}$$

thus  $\mu$  is also completely hidden.

– From  $\mathcal{O}_{\text{LEAK}}$  queries,  $\mathcal{A}$  obtains at most  $\kappa$  bits information L(sk) =  $L(\mathbf{K}) = L(\widetilde{\mathbf{K}} + \mu \mathbf{a}^{\perp}(\mathbf{u}^{\perp})^{\top})$  about sk, and also about  $\mu$ .

Overall, the information about  $\mu$  that  $\mathcal{A}$  learns in  $G_3$  is at most  $\kappa$  bits. Thus, there are still  $\log p - \kappa \geq \Omega(\lambda)$  bits of entropy left in  $\mu$ , and also in  $sk = \mathbf{K} = \widetilde{\mathbf{K}} + \mu \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top}$ . Consequently, the probability that the matrix  $(a-1)\mathbf{K}^{\top} + \mathbf{B}^{\top}$  is non-zero is  $\Pr[(a-1)\mathbf{K}^{\top} + \mathbf{B}^{\top} \neq \mathbf{0}] = 1 - \Pr[(a-1)\mathbf{K}^{\top} + \mathbf{B}^{\top} = \mathbf{0}] = 1 - \Pr[\mathbf{K} = (1-a)^{-1}\mathbf{B}] \ge 1 - 2^{-\Omega(\lambda)}$  and the claim follows.

Putting the above three cases together, we have  $\Pr_3[\mathsf{TagColl}] \leq \mathsf{Adv}_{\mathcal{H},\mathcal{B}_2}^{\mathsf{cr}}(\lambda) +$  $\operatorname{Adv}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_3}^{\operatorname{mddh}}(\lambda) + 1/(p-1) + 2^{-\Omega(\lambda)}$ , and Claim 2 follows.

Next, we consider a sequence of games  $\{\mathsf{G}_{4,\eta,0}-\mathsf{G}_{4,\eta,4}\}_{0\leq\eta\leq Q-1}$  and  $\mathsf{G}_{4,Q,0}$ .

<u>**Game**</u>  $\mathbf{G}_{4.\eta,0}, \ 0 \leq \eta \leq Q$ : It is the same as  $\mathbf{G}_3$ , except that, at the beginning of the game, the challenger picks a non-zero vector  $\mathbf{a}^{\perp} \in \mathbb{Z}_p^{k+1}$  in the kernel of  $\mathbf{A}$  such that  $(\mathbf{a}^{\perp})^{\top}\mathbf{A} = \mathbf{0}$ . Moreover, when answering the first  $\eta \mathcal{O}_{\text{SIGN}}$  queries, the challenger computes  $[\mathbf{d}]_1 := (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1$  with  $\gamma \leftarrow_{\text{s}} \mathbb{Z}_p$  chosen uniformly and independently for each query. As for the remaining  $Q - \eta \mathcal{O}_{\text{SIGN}}$  queries, the challenger still computes  $[\mathbf{d}]_1 := (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1$ .

It is clearly that  $G_{4.0.0}$  is identical to  $G_3$ , thus  $Pr_3[Win] = Pr_{4.0.0}[Win]$ .

<u>Game  $G_{4.\eta.1}$ ,  $0 \le \eta \le Q - 1$ </u>: It is the same as  $G_{4.\eta.0}$ , except that, when answering the  $(\eta + 1)$ -th  $\mathcal{O}_{\text{SIGN}}$  query, the challenger samples  $[\mathbf{c}]_1 \leftarrow \mathbb{G}_1^{k+1}$  uniformly at random, instead of computing  $[\mathbf{c}]_1 := [\mathbf{U}]_1 \mathbf{w}$  with  $\mathbf{w} \leftarrow \mathbb{Z}_p^k$ .

It is clearly that  $\mathsf{G}_{4,\eta,0}$  and  $\mathsf{G}_{4,\eta,1}$  are computationally indistinguishable to  $\mathcal{A}$ , under the  $\mathcal{D}_k$ -MDDH assumption on  $[\mathbf{U}]_1$ . Therefore, we have  $|\operatorname{Pr}_{4,\eta,0}[\mathsf{Win}] - \operatorname{Pr}_{4,\eta,1}[\mathsf{Win}]| \leq \mathsf{Adv}_{\mathcal{D}_k,\mathcal{G}_1,\mathcal{B}_4}^{\mathsf{mddh}}(\lambda)$  for a PPT adversary  $\mathcal{B}_4$ .

**Game**  $G_{4,\eta,2}$ ,  $0 \le \eta \le Q - 1$ : It is the same as  $G_{4,\eta,1}$ , except that, the event Win is now defined as

 $(m^*, \sigma^*) \notin \mathcal{Q}_{\mathsf{id}} \land [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1 \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \land \tau^* \notin \mathcal{Q}_{\mathsf{tag}} \land [\mathbf{c}^*]_1 \in \mathsf{Span}([\mathbf{U}]_1) \land [\mathbf{c}^*]_1 \in \mathsf{Span}([\mathbf{U}]_1) \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \land \tau^* \notin \mathcal{Q}_{\mathsf{tag}} \land [\mathbf{c}^*]_1 \in \mathsf{Span}([\mathbf{U}]_1) \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \land \tau^* \notin \mathcal{Q}_{\mathsf{tag}} \land [\mathbf{c}^*]_1 \in \mathsf{Span}([\mathbf{U}]_1) \land [\mathbf{c}^*]_1 \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \land \tau^* \notin \mathcal{Q}_{\mathsf{tag}} \land [\mathbf{c}^*]_1 \in \mathsf{Span}([\mathbf{U}]_1) \land [\mathbf{c}^*]_1 \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \land \tau^* \notin \mathcal{Q}_{\mathsf{tag}} \land [\mathbf{c}^*]_1 \in \mathsf{Span}([\mathbf{U}]_1) \land [\mathbf{c}^*]_1 \land [\mathbf{c}^*]_1 \neq [\mathbf{c}^*]_1 \land [\mathbf{c}^*]_1 \land$ 

Claim 3.  $|\operatorname{Pr}_{4.\eta.1}[\operatorname{Win}] - \operatorname{Pr}_{4.\eta.2}[\operatorname{Win}]| \le 1/p.$ 

*Proof.* By CBad denote the event that  $\mathcal{A}$ 's forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$  satisfying

 $(m^*, \sigma^*) \notin \mathcal{Q}_{\mathsf{id}} \land [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1 \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \land \tau^* \notin \mathcal{Q}_{\mathsf{tag}} \land [\mathbf{c}^*]_1 \notin \mathsf{Span}([\mathbf{U}]_1).$ 

Clearly,  $G_{4,\eta,1}$  and  $G_{4,\eta,2}$  are the same until CBad occurs, thus  $|\Pr_{4,\eta,1}[Win] - \Pr_{4,\eta,2}[Win]| \le \Pr_{4,\eta,2}[CBad].$ 

Next, we analyze  $\Pr_{4.\eta,2}[\mathsf{CBad}]$ . Let  $\mathbf{u}^{\perp} \in \mathbb{Z}_p^{k+1}$  be a non-zero vector in the kernel of  $\mathbf{U}$  such that  $(\mathbf{u}^{\perp})^{\top}\mathbf{U} = \mathbf{0}$  but  $(\mathbf{u}^{\perp})^{\top}\mathbf{c}^* \neq 0$ . For the convenience of our analysis, we sample  $\mathbf{K}_0, \mathbf{K}_1 \leftarrow \mathbb{Z}_p^{(k+1) \times (k+1)}$  equivalently via

$$\mathbf{K}_0 := \widetilde{\mathbf{K}_0} + \mu_0 \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top}, \quad \mathbf{K}_1 := \widetilde{\mathbf{K}_1} + \mu_1 \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top},$$

where  $\widetilde{\mathbf{K}_0}, \widetilde{\mathbf{K}_1} \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$  and  $\mu_0, \mu_1 \leftarrow \mathbb{Z}_p$ . Below we analyze the information about  $\mu_0$  and  $\mu_1$  that  $\mathcal{A}$  may obtain in  $\mathsf{G}_{4,\eta,2}$ .

- Firstly, the public parameter pp contains  $[\mathbf{K}_0\mathbf{U}]_1, [\mathbf{K}_1\mathbf{U}]_1, [\mathbf{K}_0^{\top}\mathbf{A}]_2, [\mathbf{K}_1^{\top}\mathbf{A}]_2$ . Due to the facts that  $(\mathbf{u}^{\perp})^{\top}\mathbf{U} = \mathbf{0}$  and  $(\mathbf{a}^{\perp})^{\top}\mathbf{A} = \mathbf{0}$ , it is easy to see that

$$[\mathbf{K}_0\mathbf{U}]_1 = [\widetilde{\mathbf{K}_0}\mathbf{U}]_1, [\mathbf{K}_1\mathbf{U}]_1 = [\widetilde{\mathbf{K}_1}\mathbf{U}]_1, [\mathbf{K}_0^{\top}\mathbf{A}]_2 = [\widetilde{\mathbf{K}}_0^{\top}\mathbf{A}]_2, [\mathbf{K}_1^{\top}\mathbf{A}]_2 = [\widetilde{\mathbf{K}}_1^{\top}\mathbf{A}]_2$$

Thus  $\mu_0$  and  $\mu_1$  are completely hidden. Moreover, the verification key vk does not involve  $\mathbf{K}_0$  and  $\mathbf{K}_1$ , so  $\mu_0$  and  $\mu_1$  are also completely hidden.

- In  $\mathcal{O}_{\text{SIGN}}$  queries, the tampered verification key vk' does not involve  $\mathbf{K}_0$  and  $\mathbf{K}_1$ , thus also does not involve  $\mu_0$  and  $\mu_1$ . Next we analyze the information about  $\mu_0$  and  $\mu_1$  contained in  $[\mathbf{d}]_1$ .

\* For the first  $\eta \mathcal{O}_{SIGN}$  queries, we have

$$\begin{split} [\mathbf{d}]_1 &= (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1) [\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \\ &= (\mathbf{K}' + \widetilde{\mathbf{K}_0} + \tau \widetilde{\mathbf{K}_1} + (\mu_0 + \tau \mu_1) \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top}) [\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1. \end{split}$$

Due to fact that  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}$  with  $\mathbf{w} \leftarrow \mathbb{Z}_p^k$ , we have

$$[\mathbf{d}]_1 = (\mathbf{K}' + \widetilde{\mathbf{K}_0} + \tau \widetilde{\mathbf{K}_1})[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1.$$

Therefore,  $\mu_0$  and  $\mu_1$  are completely hidden.

\* For the  $(\eta + 1)$ -th  $\mathcal{O}_{SIGN}$  query, we have

$$\begin{split} [\mathbf{d}]_1 &= (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 \\ &= (\mathbf{K}' + \widetilde{\mathbf{K}_0} + \tau \widetilde{\mathbf{K}_1} + (\mu_0 + \tau \mu_1) \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top})[\mathbf{c}]_1, \end{split}$$

- so the information of  $\mu_0$  and  $\mu_1$  contained in  $[\mathbf{d}]_1$  is limited in  $(\mu_0 + \tau \mu_1)$ . \* For the remaining  $(Q - \eta - 1) \mathcal{O}_{\text{SIGN}}$  queries, we also have  $[\mathbf{d}]_1 = (\mathbf{K}' + \mathbf{M})$
- For the remaining ( $\mathbf{g} = \eta \tau$ )  $\mathcal{C}_{\text{SIGN}}$  queries, we also have  $[\mathbf{d}]_1 = (\mathbf{K} + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 = (\mathbf{K}' + \mathbf{\widetilde{K}_0} + \tau \mathbf{\widetilde{K}_1} + (\mu_0 + \tau \mu_1)\mathbf{a}^{\perp}(\mathbf{u}^{\perp})^{\top})[\mathbf{c}]_1$ . Due to fact that  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}$  with  $\mathbf{w} \leftarrow_{\$} \mathbb{Z}_p^k$ , we have  $[\mathbf{d}]_1 = (\mathbf{K}' + \mathbf{\widetilde{K}_0} + \tau \mathbf{\widetilde{K}_1})[\mathbf{c}]_1$ . Therefore,  $\mu_0$  and  $\mu_1$  are completely hidden.
- From  $\mathcal{O}_{\text{LEAK}}$  queries,  $\mathcal{A}$  obtains leakage information about  $sk = \mathbf{K}$ . It does not involve  $\mathbf{K}_0$  and  $\mathbf{K}_1$ , thus also does not involve  $\mu_0$  and  $\mu_1$ .

Overall, the information that  $\mathcal{A}$  might learn about  $\mu_0$  and  $\mu_1$  is limited in  $(\mu_0 + \tau \mu_1)$ .

For CBad to occur,  $\mathcal{A}$ 's forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$  should satisfy  $(m^*, \sigma^*) \notin \mathcal{Q}_{id}, [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1, \tau^* \notin \mathcal{Q}_{tag}, [\mathbf{c}^*]_1 \notin \mathsf{Span}([\mathbf{U}]_1)$ , and

$$[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1 = [(\mathbf{K} + \widetilde{\mathbf{K}_0} + \tau^* \widetilde{\mathbf{K}_1}) \mathbf{c}^* + (\mu_0 + \tau^* \mu_1) \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top} \mathbf{c}^*]_1$$

Below we argue that  $\mathcal{A}$  can hardly compute such  $[\mathbf{d}^*]_1$ . Since  $\tau^* \notin \mathcal{Q}_{\mathsf{tag}}$ , the term  $(\mu_0 + \tau^* \mu_1)$  is pairwise independent from the information  $(\mu_0 + \tau \mu_1)$  that  $\mathcal{A}$  might learn, thus  $(\mu_0 + \tau^* \mu_1)$  is uniformly distributed over  $\mathbb{Z}_p$  from  $\mathcal{A}$ 's view. Moreover,  $\mathbf{u}^{\perp} \in \mathbb{Z}_p^{k+1}$  is chosen to satisfy  $(\mathbf{u}^{\perp})^{\top} \mathbf{c}^* \neq 0$  since  $[\mathbf{c}^*]_1 \notin \mathsf{Span}([\mathbf{U}]_1)$ . Therefore,  $(\mu_0 + \tau^* \mu_1) \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top} \mathbf{c}^*$  is uniformly distributed over  $\mathsf{Span}(\mathbf{a}^{\perp}) = \{\gamma^* \mathbf{a}^{\perp} | \gamma^* \in \mathbb{Z}_p\}$  from  $\mathcal{A}$ 's view, and consequently,  $\mathcal{A}$  can compute such  $[\mathbf{d}^*]_1$  with probability at most 1/p. This shows that  $\Pr_{4.\eta, 2}[\mathsf{CBad}] \leq 1/p$  and Claim 3 follows.

<u>Game G<sub>4. $\eta$ .3</sub>,  $0 \le \eta \le Q - 1$ </u>: It is the same as G<sub>4. $\eta$ .2</sub>, except that, when answering the  $(\eta + 1)$ -th  $\mathcal{O}_{SIGN}$  query, the challenger computes  $[\mathbf{d}]_1 := (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1$  with  $\gamma \leftarrow \mathbb{Z}_p$ , instead of  $[\mathbf{d}]_1 := (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1$ .

Claim 4.  $\left|\operatorname{Pr}_{4.\eta,2}[\operatorname{Win}] - \operatorname{Pr}_{4.\eta,3}[\operatorname{Win}]\right| \leq 1/p.$ 

*Proof.* We will show that  $G_{4,\eta,2}$  and  $G_{4,\eta,3}$  are identically distributed, except with probability 1/p. To see this, let  $\mathbf{u}^{\perp} \in \mathbb{Z}_p^{k+1}$  be a non-zero vector in the kernel of  $\mathbf{U}$  such that  $(\mathbf{u}^{\perp})^{\top}\mathbf{U} = \mathbf{0}$ . Similar to the proof of Claim 3, we sample  $\mathbf{K}_0, \mathbf{K}_1 \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$  equivalently via  $\mathbf{K}_0 := \widetilde{\mathbf{K}}_0 + \mu_0 \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top}$  and  $\mathbf{K}_1 := \widetilde{\mathbf{K}}_1 + \mu_1 \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top}$ , where  $\widetilde{\mathbf{K}}_0, \widetilde{\mathbf{K}}_1 \leftarrow \mathbb{Z}_p^{(k+1)\times(k+1)}$  and  $\mu_0, \mu_1 \leftarrow \mathbb{Z}_p$ . Recall that in the proof of Claim 3, we observe that  $\mu_0$  and  $\mu_1$  are completely hidden in the public parameter **pp**, the verification key vk, all  $\mathcal{O}_{\text{SIGN}}$  queries except the  $(\eta+1)$ th  $\mathcal{O}_{\text{SIGN}}$  query, and  $\mathcal{O}_{\text{LEAK}}$  queries. Moreover, due to the game change in  $G_{4,\eta,2}$ , the event Win checks  $[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1)\mathbf{c}^*]_1$  only if  $[\mathbf{c}^*]_1 \in \text{Span}([\mathbf{U}]_1)$ , and when  $[\mathbf{c}^*]_1 \in \text{Span}([\mathbf{U}]_1)$ , the check becomes

$$\begin{split} [\mathbf{d}^*]_1 &= [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{c}^* + (\mu_0 + \tau^* \mu_1) \mathbf{a}^{\perp} (\mathbf{u}^{\perp})^{\top} \mathbf{c}^*]_1 \\ &= [(\mathbf{K} + \widetilde{\mathbf{K}_0} + \tau^* \widetilde{\mathbf{K}_1}) \mathbf{c}^*]_1, \end{split}$$

where  $\mu_0$  and  $\mu_1$  are also completely hidden. Therefore, the only place that involves  $\mu_0$  and  $\mu_1$  lies in the computation of  $[\mathbf{d}]_1$  in the  $(\eta + 1)$ -th  $\mathcal{O}_{\text{SIGN}}$  query, where in  $\mathsf{G}_{4,\eta,2}$ , we have

$$[\mathbf{d}]_1 = (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 = (\mathbf{K}' + \widetilde{\mathbf{K}_0} + \tau \widetilde{\mathbf{K}_1})[\mathbf{c}]_1 + [(\mu_0 + \tau \mu_1)\mathbf{a}^{\perp}(\mathbf{u}^{\perp})^{\top}\mathbf{c}]_1,$$
(6)

while in  $G_{4.\eta.3}$ , we have

$$[\mathbf{d}]_{1} = (\mathbf{K}' + \mathbf{K}_{0} + \tau \mathbf{K}_{1})[\mathbf{c}]_{1} + [\gamma \mathbf{a}^{\perp}]_{1}$$
  
=  $(\mathbf{K}' + \widetilde{\mathbf{K}_{0}} + \tau \widetilde{\mathbf{K}_{1}})[\mathbf{c}]_{1} + [(\mu_{0} + \tau \mu_{1})\mathbf{a}^{\perp}(\mathbf{u}^{\perp})^{\top}\mathbf{c} + \gamma \mathbf{a}^{\perp}]_{1},$  (7)

with  $\gamma \leftarrow_{s} \mathbb{Z}_{p}$ . Note that due to the game change in  $\mathsf{G}_{4,\eta,1}$ , in the  $(\eta + 1)$ -th  $\mathcal{O}_{\mathrm{SIGN}}$  query,  $[\mathbf{c}]_{1}$  is chosen uniformly from  $\mathbb{G}_{1}^{k+1}$ , thus except with probability 1/p, we have  $(\mathbf{u}^{\perp})^{\top}\mathbf{c} \neq 0$ , and in this case, both the term  $(\mu_{0} + \tau\mu_{1})\mathbf{a}^{\perp}(\mathbf{u}^{\perp})^{\top}\mathbf{c}$  in (6) and the term  $(\mu_{0} + \tau\mu_{1})\mathbf{a}^{\perp}(\mathbf{u}^{\perp})^{\top}\mathbf{c} + \gamma\mathbf{a}^{\perp}$  in (7) are uniformly distributed over  $\mathrm{Span}(\mathbf{a}^{\perp}) = \{\gamma^{*}\mathbf{a}^{\perp} | \gamma^{*} \in \mathbb{Z}_{p}\}$ , due to the randomness of  $\mu_{0}$  and  $\mu_{1}$ . Consequently,  $\mathsf{G}_{4,\eta,2}$  (which computes  $[\mathbf{d}]_{1}$  in the  $(\eta + 1)$ -th  $\mathcal{O}_{\mathrm{SIGN}}$  query according to (6)) and  $\mathsf{G}_{4,\eta,3}$  (which computes  $[\mathbf{d}]_{1}$  in the  $(\eta + 1)$ -th  $\mathcal{O}_{\mathrm{SIGN}}$  query according to (7)) are identically distributed, except with probability 1/p. This shows that  $|\operatorname{Pr}_{4,\eta,2}[\operatorname{Win}] - \operatorname{Pr}_{4,\eta,3}[\operatorname{Win}]| \leq 1/p$ , and Claim 4 follows.

**Game**  $G_{4,\eta,4}$ ,  $0 \le \eta \le Q - 1$ : It is the same as  $G_{4,\eta,3}$ , except that, the event Win is changed back to

$$(m^*, \sigma^*) \notin \mathcal{Q}_{\mathsf{id}} \land [\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1)\mathbf{c}^*]_1 \land [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1 \land \tau^* \notin \mathcal{Q}_{\mathsf{tag}} \land \underline{[\mathbf{c}^*]_1 \in \mathsf{Span}([\mathbf{U}]_1)}$$

The transition from  $G_{4.\eta.3}$  to  $G_{4.\eta.4}$  is reverse to the transition from  $G_{4.\eta.1}$  to  $G_{4.\eta.2}$ . Similar to Claim 3, we have the following claim.

Claim 5. 
$$|\Pr_{4.\eta.3}[Win] - \Pr_{4.\eta.4}[Win]| \le 1/p.$$

Now we analyze the difference between  $G_{4,\eta,4}$  and  $G_{4,\eta+1,0}$ . The only difference is the distribution of  $[\mathbf{c}]_1$  in the  $(\eta + 1)$ -th  $\mathcal{O}_{SIGN}$  query, where in  $G_{4,\eta,4}$ ,

$$\begin{split} [\mathbf{c}]_1 &\leftarrow_{\mathtt{s}} \mathbb{G}_1^{k+1} \text{ is chosen uniformly at random, while in } \mathsf{G}_{4.\eta+1.0}, \ [\mathbf{c}]_1 := [\mathbf{U}]_1 \mathbf{w} \\ \text{with } \mathbf{w} &\leftarrow_{\mathtt{s}} \mathbb{Z}_p^k. \text{ It is clearly that } \mathsf{G}_{4.\eta.4} \text{ and } \mathsf{G}_{4.\eta+1.0} \text{ are computationally indistinguishable to } \mathcal{A}, \text{ under the } \mathcal{D}_k\text{-MDDH} \text{ assumption on } [\mathbf{U}]_1. \text{ Therefore, we have} \\ \big| \operatorname{Pr}_{4.\eta.4}[\mathsf{Win}] - \operatorname{Pr}_{4.\eta+1.0}[\mathsf{Win}] \big| \leq \mathsf{Adv}_{\mathcal{D}_k,\mathbb{G}_1,\mathcal{B}_5}^{\mathsf{mddh}}(\lambda) \text{ for a PPT adversary } \mathcal{B}_5. \end{split}$$

Finally, we arrive at  $G_{4,Q,0}$ , which is restated as follows.

**Game**  $G_{4.Q.0}$ : It is the same as  $G_3$ , except that, at the beginning of the game, the challenger picks a non-zero vector  $\mathbf{a}^{\perp} \in \mathbb{Z}_p^{k+1}$  in the kernel of  $\mathbf{A}$  such that  $(\mathbf{a}^{\perp})^{\top}\mathbf{A} = \mathbf{0}$ . Moreover, when answering *all*  $\mathcal{O}_{\text{SIGN}}$  queries, the challenger computes  $[\mathbf{d}]_1 := (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1$  with  $\gamma \leftarrow \mathbb{Z}_p$  chosen uniformly and independently for each query.

We have the following claim regarding  $Pr_{4.Q.0}[Win]$ .

Claim 6.  $\Pr_{4.Q.0}[\mathsf{Win}] \leq 2^{-\Omega(\lambda)}$ .

*Proof.* For the convenience of our analysis, we sample  $sk = \mathbf{K} \leftarrow \mathbb{Z}_p^{(k+1) \times (k+1)}$  equivalently via

$$\mathbf{K} := \widetilde{\mathbf{K}} + \mathbf{a}^{\perp} \mathbf{k}^{\top}$$

where  $\widetilde{\mathbf{K}} \leftarrow_{\$} \mathbb{Z}_p^{(k+1) \times (k+1)}$  and  $\mathbf{k} \leftarrow_{\$} \mathbb{Z}_p^{k+1}$ . Below we analyze the information about  $\mathbf{k}$  that  $\mathcal{A}$  may obtain in  $\mathsf{G}_{4.Q.0}$ .

- Firstly, the verification key vk is

$$[\mathbf{K}^{ op}\mathbf{A}]_2 = [(\widetilde{\mathbf{K}}^{ op} + \mathbf{k}(\mathbf{a}^{\perp})^{ op})\mathbf{A}]_2 = [\widetilde{\mathbf{K}}^{ op}\mathbf{A}]_2,$$

thus  $\mathbf{k}$  is completely hidden.

– In  $\mathcal{O}_{SIGN}$  queries, the tampered verification key vk' is

$$[\mathbf{K}^{\prime \top} \mathbf{A}]_2 = [(a\mathbf{K}^{\top} + \mathbf{B}^{\top})\mathbf{A}]_2 = [(a(\widetilde{\mathbf{K}}^{\top} + \mathbf{k}(\mathbf{a}^{\perp})^{\top}) + \mathbf{B}^{\top})\mathbf{A}]_2$$
$$= [(a\widetilde{\mathbf{K}}^{\top} + \mathbf{B}^{\top})\mathbf{A}]_2,$$

thus  ${\bf k}$  is completely hidden. Moreover, due to the game change in  ${\sf G}_{4.Q.0},$  we have

$$\begin{aligned} [\mathbf{d}]_1 &= (\mathbf{K}' + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \\ &= (a\mathbf{K} + \mathbf{B} + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \\ &= (a(\widetilde{\mathbf{K}} + \mathbf{a}^{\perp} \mathbf{k}^{\top}) + \mathbf{B} + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [\gamma \mathbf{a}^{\perp}]_1 \\ &= (a\widetilde{\mathbf{K}} + \mathbf{B} + \mathbf{K}_0 + \tau \mathbf{K}_1)[\mathbf{c}]_1 + [a\mathbf{a}^{\perp} \mathbf{k}^{\top} \mathbf{c} + \gamma \mathbf{a}^{\perp}]_1. \end{aligned}$$

Since  $\gamma \leftarrow \mathbb{Z}_p$ , the term  $\gamma \mathbf{a}^{\perp}$  perfectly hides  $a\mathbf{a}^{\perp}\mathbf{k}^{\top}\mathbf{c} = (a\mathbf{k}^{\top}\mathbf{c})\mathbf{a}^{\perp}$ . Consequently,  $\mathbf{k}$  is also completely hidden.

- From  $\mathcal{O}_{\text{LEAK}}$  queries,  $\mathcal{A}$  obtains at most  $\kappa$  bits information  $L(sk) = L(\mathbf{K}) = L(\widetilde{\mathbf{K}} + \mathbf{a}^{\perp} \mathbf{k}^{\top})$  about sk, and also about  $\mathbf{k}$ .

Overall, the information about **k** that  $\mathcal{A}$  learns in  $\mathsf{G}_{4.Q.0}$  is at most  $\kappa$  bits.

For Win to occur,  $\mathcal{A}$ 's forgery  $(m^*, \sigma^* = ([\mathbf{c}^*]_1, [\mathbf{d}^*]_1))$  should satisfy  $(m^*, \sigma^*) \notin \mathcal{Q}_{\mathsf{id}}, [\mathbf{c}^*]_1 \neq [\mathbf{0}]_1, \tau^* \notin \mathcal{Q}_{\mathsf{tag}}$ , and

$$[\mathbf{d}^*]_1 = [(\mathbf{K} + \mathbf{K}_0 + \tau^* \mathbf{K}_1)\mathbf{c}^*]_1 = [(\widetilde{\mathbf{K}} + \mathbf{K}_0 + \tau^* \mathbf{K}_1)\mathbf{c}^* + \mathbf{a}^{\perp} \mathbf{k}^{\top} \mathbf{c}^*]_1.$$

Below we argue that such  $[\mathbf{d}^*]_1$  has high entropy so that  $\mathcal{A}$  can hardly compute it. We first analyze the entropy of  $[\mathbf{d}^*]_1$  in the case  $\kappa = 0$ , i.e., there is no leakage at all. In this case,  $\mathbf{k}$  is uniformly distributed over  $\mathbb{Z}_p^{k+1}$  from  $\mathcal{A}$ 's view, and by  $[\mathbf{c}^*]_1 \neq [\mathbf{0}]_1$ , it follows that  $\mathbf{k}^\top \mathbf{c}^*$  is uniformly distributed over  $\mathbb{Z}_p$  from  $\mathcal{A}$ 's view. Therefore,  $\mathbf{a}^\perp \mathbf{k}^\top \mathbf{c}^*$  is uniformly distributed over  $\mathsf{Span}(\mathbf{a}^\perp) = \{\gamma^* \mathbf{a}^\perp | \gamma^* \in \mathbb{Z}_p\}$ from  $\mathcal{A}$ 's view, and consequently, such  $[\mathbf{d}^*]_1$  has  $\log p$  bits of entropy from  $\mathcal{A}$ 's view. Next, we analyze the entropy of  $[\mathbf{d}^*]_1$  for any  $\kappa \leq \log p - \Omega(\lambda)$ . Even in the presence of  $\kappa$  bits leakage information,  $[\mathbf{d}^*]_1$  still has entropy at least  $\log p - \kappa \geq \Omega(\lambda)$  bits from  $\mathcal{A}$ 's view. Consequently,  $\mathcal{A}$  can compute such  $[\mathbf{d}^*]_1$ with probability at most  $2^{-\Omega(\lambda)}$ . This shows that  $\Pr_{4.Q.0}[\mathsf{Win}] \leq 2^{-\Omega(\lambda)}$  and Claim 6 follows.

Taking all things together, Theorem 1 follows.

## 4 More Efficient PKE with Leakage and Tamper-Resilience

In this section, we present a direct and efficient construction of public-key encryption (PKE) scheme with leakage and tamper-resilience, over asymmetric pairing groups based on the MDDH assumptions.

Concretely, in Subsect. 4.1, we formalize the leakage and tamper-resilient security for PKE, i.e., the *strong* LTR-CCA (sLTR-CCA) security, according to [35], and then in Subsect. 4.2 and Subsect. 4.3, we present our PKE scheme and its security proof, respectively.

#### 4.1 Definition of sLTR-CCA Security

Below we recall the sLTR-CCA security for PKE defined in [35].

**Definition 7** (sLTR-CCA Security for PKE). Let  $\kappa = \kappa(\lambda) \in \mathbb{N}$ , and  $\mathcal{T}$  be a set of functions from  $\mathcal{SK}$  to  $\mathcal{SK}$  where  $\mathcal{SK}$  is the secret key space. A PKE scheme PKE = (Setup, Gen, Enc, Dec) is  $(\kappa, \mathcal{T})$ -sLTR-CCA secure, if for any PPT adversary  $\mathcal{A}$ , it holds that  $\operatorname{Adv}_{\mathsf{PKE},\mathcal{A},\kappa,\mathcal{T}}^{\mathsf{sltr-cca}}(\lambda) := |\operatorname{Pr}[\operatorname{Exp}_{\mathsf{PKE},\mathcal{A},\kappa,\mathcal{T}}^{\mathsf{sltr-cca}} \Rightarrow 1] - \frac{1}{2}| \leq \operatorname{negl}(\lambda)$ , where the experiment  $\operatorname{Exp}_{\mathsf{PKE},\mathcal{A},\kappa,\mathcal{T}}^{\mathsf{sltr-cca}}$  is defined in Fig. 3.

In the experiment  $\mathsf{Exp}_{\mathsf{PKE},\mathcal{A},\kappa,\mathcal{T}}^{\mathsf{sltr-cca}}$  defined in Fig. 3, it imposes only minimal restrictions on the  $\mathcal{O}_{\mathsf{DEC}}$  queries that  $\mathcal{A}$  can make, i.e.,  $(T, ct) \neq (\mathsf{id}, ct^*)$ . This is formulated in [35], as a strengthening of the (non-strong) LTR-CCA security defined in [36] where (T, ct) is subject to  $ct \neq ct^*$ .

$Exp_{PKE,\mathcal{A},\kappa,\mathcal{T}}^{sltr-cca}$ :	$\mathcal{O}_{\mathrm{DFC}}(T, ct)$ :		
$\overline{pp} \leftarrow s Setup, \ (pk, sk) \leftarrow s Gen(pp)$	$\frac{I}{If T \notin \mathcal{T}}: \text{Return } \perp$		
$\ell := 0$ //Record the leakage length	If $(T, ct) = (id, ct^*)$ : Return $\perp$		
$(m_0, m_1, st) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathrm{Dec}}(\cdot, \cdot), \mathcal{O}_{\mathrm{Leak}}(\cdot)}(pp, pk)$	Return $Dec(T(sk), ct)$		
If $ m_0  \neq  m_1 $ : Return $\perp$			
$\beta \leftarrow \{0,1\}$ //Challenge bit	$\mathcal{O}_{\text{LEAK}}(L)$ : //at most $\kappa$ leakage		
$ct^* \leftarrow s Enc(pk, m_\beta)$	If $\ell +  L(sk)  > \kappa$ : Return $\perp$		
$eta' \leftarrow \mathcal{A}^{\mathcal{O}_{ ext{Dec}}(\cdot, \cdot)}(st, ct^*)$	$\ell := \ell +  L(sk) $		
If $\beta' = \beta$ : Return 1; Else: Return 0	Return $L(sk)$		

**Fig. 3.** The  $(\kappa, \mathcal{T})$ -sLTR-CCA security experiment  $\text{Exp}_{\mathsf{PKE},\mathcal{A},\kappa,\mathcal{T}}^{\mathsf{sltr-cca}}$  for PKE, where id denotes the identity function and |L(sk)| denotes the bit-length of L(sk).

#### 4.2 Construction of PKE from MDDH

Now we present our direct construction of sLTR-CCA secure PKE scheme over asymmetric pairing groups based on the MDDH assumptions. Let  $\mathcal{D}_k$  be a matrix distribution with  $k \in \mathbb{N}$ , let  $\mathcal{U}_{k+2,k}$  be the uniform distribution, and let  $\mathcal{H}$  be a family of collision resistant hash functions from  $\{0,1\}^*$  to  $\mathbb{Z}_p$ . Our PKE scheme PKE = (Setup, Gen, Enc, Dec) is shown in Fig. 4, where the message space is  $\mathcal{M} = \mathbb{G}_1$  and the secret key space is  $\mathcal{SK} = \mathbb{Z}_p^{k+2}$ . Correctness of PKE follows by inspection: for any honestly generated ciphertext  $ct = ([\mathbf{c}]_1, [d]_1, [\mathbf{e}]_1)$ , we have  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}, [d]_1 = [\mathbf{k}^\top \mathbf{U}]_1 \mathbf{w} + m = \mathbf{k}^\top [\mathbf{c}]_1 + m$  and  $[\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1 \mathbf{w} =$  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1$ , which implies

$$e([\mathbf{c}^{\top}]_1, [(\mathbf{K}_0^{\top} + \tau \mathbf{K}_1^{\top})\mathbf{A}]_2) = e([\mathbf{c}^{\top}(\mathbf{K}_0^{\top} + \tau \mathbf{K}_1^{\top})]_1, [\mathbf{A}]_2) = e([\mathbf{c}^{\top}]_1, [\mathbf{A}]_2).$$

Next, we show its  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CCA security under  $\kappa \leq \log p - \Omega(\lambda)$  bits of leakage information and under the set of affine functions

$$\mathcal{T}_{\mathsf{aff}} = \{ T_{(a,\mathbf{b})} : \mathbf{k} \in \mathcal{SK} \mapsto a\mathbf{k} + \mathbf{b} \in \mathcal{SK} \mid a \in \mathbb{Z}_p, \mathbf{b} \in \mathcal{SK} \}.$$
(8)

**Theorem 2** (( $\kappa$ ,  $\mathcal{T}_{aff}$ )-sLTR-CCA Security of PKE). Let  $\kappa \leq \log p - \Omega(\lambda)$  and let  $\mathcal{T}_{aff}$  be the set of affine functions defined in (8). Assume that the  $\mathcal{D}_k$ -MDDH assumption holds over both  $\mathbb{G}_1$  and  $\mathbb{G}_2$ , and  $\mathcal{H}$  is collision-resistant. Then the PKE scheme in Fig. 4 is ( $\kappa$ ,  $\mathcal{T}_{aff}$ )-sLTR-CCA secure.

Concretely, for any PPT adversary  $\mathcal{A}$ , there exist PPT adversaries  $\mathcal{B}_1, \cdots, \mathcal{B}_4$ , such that

$$\begin{split} \mathsf{Adv}^{\mathsf{sltr-cca}}_{\mathsf{PKE},\mathcal{A},\kappa,\mathcal{T}}(\lambda) &\leq \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{cr}}_{\mathcal{H},\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_{k+2,k},\mathbb{G}_1,\mathcal{B}_3}(\lambda) \\ &+ \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_{k+2,k},\mathbb{G}_1,\mathcal{B}_4}(\lambda) + 2^{-\Omega(\lambda)}. \end{split}$$

The proof of Theorem 2 is postponed to Subsect. 4.3. Before presenting the formal proof, we give a detailed efficiency analysis and explain the main intuitions of our PKE construction in the following two remarks, respectively.



**Fig. 4.** Construction of  $\mathsf{PKE} = (\mathsf{Setup}, \mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$  based on MDDH, where the framed boxes and the gray boxes are used to help explain the intuitions behind the construction in Remark 5.

**Remark 4 (Efficiency of our PKE).** Let  $x \cdot \mathbb{G}$  denote x elements in a group  $\mathbb{G}$ . Our PKE scheme in Fig. 4 is parameterized by the MDDH parameter  $k \in \mathbb{N}$ , and has public parameter  $pp : (3k^2 + 4k) \cdot \mathbb{G}_1 + (3k^2 + 5k) \cdot \mathbb{G}_2$ , public key  $pk : k \cdot \mathbb{G}_1$ , secret key  $sk : (k + 2) \cdot \mathbb{Z}_p$  and ciphertext  $ct : (2k + 4) \cdot \mathbb{G}_1$ . The decryption involves  $(2k^2 + 3k)$  pairing operations.

For k = 1, we get an efficient PKE scheme with  $pp : 7 \cdot \mathbb{G}_1 + 8 \cdot \mathbb{G}_2$ , public key  $pk : 1 \cdot \mathbb{G}_2$ , secret key  $sk : 3 \cdot \mathbb{Z}_p$  and ciphertext  $ct : 6 \cdot \mathbb{G}_1$ , and the decryption involves only 5 pairing operations. The resulting PKE scheme is  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CCA secure based on the standard SXDH assumption, and supports  $\kappa = \log p - \Omega(\lambda)$  bits key leakage. The leakage rate is  $\frac{\log p - \Omega(\lambda)}{3 \log p} = \frac{1}{3} - o(1)$  asymptotically as p grows.

**Remark 5 (Intuitions of our PKE).** Similar to our SIG scheme proposed in Subsect. 3.2, our PKE in Fig. 4 can also be parsed as two components: the terms in framed boxes (which are related to  $\mathbf{k}$ ) and the terms in gray boxes

(which are related to  $\mathbf{K}_0$  and  $\mathbf{K}_1$ ).

Our first idea is to let  $sk = \lfloor \mathbf{k} \rfloor$  involve only term of the first component, similar to our SIG scheme, so that we only need to analyze the first component in the leakage and tampering-resilient setting.

Our second idea is to use the first component to mask the message during the generation of ciphertext, while use the second component to prove the wellformedness of ciphertext. More concretely, the first component can be viewed as the CPA-secure variant of the Cramer-Shoup PKE scheme [8] (which corresponds to the  $[\mathbf{c}]_1, [d]_1$  in our *ct*), and the second component can be viewed as the onetime simulation-sound (OTSS) NIZK scheme proposed by Kiltz and Wee [27] (which corresponds to the  $[\mathbf{e}]_1$  in our *ct* and essentially proves that  $[\mathbf{c}]_1$  belongs to the linear subspace  $\mathsf{Span}([\mathbf{U}]_1)$ ). The efficiency of our PKE scheme benefits from the efficiency of their schemes. For example, the Kiltz-Wee OTSS-NIZK has a very short proof, which is much shorter than the tSE-NIZK [12] usually required when constructing schemes in the leakage and tamper-resilient setting. However, the sLTR-CCA security of our PKE scheme does not simply follow from the CPA-security of the Cramer-Shoup PKE variant and the OTSS of the Kiltz-Wee NIZK. In fact, our sLTR-CCA security proof also relies on the concrete algebraic structures of the schemes and involves many subtleties similar to the sLTR-CMA security proof of our SIG scheme in Subsect. 3.2, as explained later.

Below we explain the intuitions behind these two components in more detail.

INTUITIONS BEHIND THE FIRST COMPONENT. Intuitively, the terms in framed boxes can be viewed as the CPA-secure variant of the Cramer-Shoup PKE scheme [8]:

- the secret key is  $sk = \mathbf{k}$  and the public key is  $pk = \lfloor [\mathbf{k}^\top \mathbf{U}]_1 \rfloor;$
- the ciphertext of message  $m \in \mathbb{G}_1$  is simply

$$([\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}, \ [d]_1 = \boxed{[\mathbf{k}^\top \mathbf{U}]_1 \mathbf{w}} + m), \text{ with } \mathbf{w} \leftarrow \mathbb{Z}_p^k, \tag{9}$$

and the decryption simply computes  $m = [d]_1 - \mathbf{k}^\top [\mathbf{c}]_1$ .

It is worthwhile to briefly recall why this component is CPA secure. Its CPA security proof consists of two main steps.

• Firstly, we change the generation of the challenge ciphertext as follows

$$([\mathbf{c}^*]_1 \leftarrow \mathbb{G}_1^{k+2}, \ [d^*]_1 = \boxed{[\mathbf{k}^\top \mathbf{c}^*]_1} + m).$$
(10)

Observe that (10) is computationally indistinguishable from (9) under the  $\mathcal{U}_{k+2,k}$ -MDDH assumption on  $[\mathbf{U}]_1$  (which is further implied by the  $\mathcal{D}_k$ -MDDH assumption according to Lemma 2).

Since [c<sup>\*</sup>]<sub>1</sub> ←<sub>s</sub> G<sub>1</sub><sup>k+2</sup> is uniformly chosen, the mapping k → [[k<sup>T</sup>c<sup>\*</sup>]<sub>1</sub>] indexed by [c<sup>\*</sup>]<sub>1</sub> is a universal hash function.

Note that in the presence of only  $pk = [\mathbf{k}^{\top}\mathbf{U}]_1$ , where  $\mathbf{U} \in \mathbb{Z}_p^{(k+2)\times k}$ ,  $sk = \mathbf{k}$  retains  $2 \log p$  bits of entropy, so it can be extracted by the universal hash function to yield a (statistically close to) uniform element  $[\mathbf{k}^{\top}\mathbf{c}^*]_1 \in \mathbb{G}_1$  in (10) (according to the leftover hash lemma, i.e., Lemma 1). Consequently, the term  $[\mathbf{k}^{\top}\mathbf{c}^*]_1$  in (10) hides the message m.

INSUFFICIENCY OF THE FIRST COMPONENT AND ARISING OF THE SECOND. The first component and its CPA security proof serve as the basis for the security of our PKE. Moreover, the first component is in fact resilient to bounded key leakage, as noted by Naor and Segev in [32]. This is because the above argument for CPA security holds even if the adversary obtains bounded leakage information about  $sk = \mathbf{k}$ , as long as the amount of leakage  $\kappa$  satisfies  $\log p - \kappa \geq \Omega(\lambda)$  so that there are still  $2\log p - \kappa \geq \log p + \Omega(\lambda)$  bits entropy left in  $\mathbf{k}$  to extract a uniform group element  $[\mathbf{k}^{\top}\mathbf{c}^*]_1 \in \mathbb{G}_1$ . This shows the CPA security in the *leakage setting* of our PKE.

However, in the sLTR-CCA security experiment,  $\mathcal{A}$  has also access to a decryption oracle, through which  $\mathcal{A}$  can obtain the decryption results of multiple ciphertexts, under *tampered* secret keys  $T_{(a,\mathbf{b})}(sk) = a\mathbf{k} + \mathbf{b}$ . So the decryption oracle would leak additional information about  $sk = \mathbf{k}$  beyond pk.

To rescue the above arguments, we resort to the terms in gray boxes. Roughly speaking, we use  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  as a OTSS-NIZK, as shown in [27], to prove the well-formedness of ciphertexts. This guarantees that the decryption result of a ciphertext  $ct = ([\mathbf{c}]_1, [d]_1, [\mathbf{e}]_1)$  is not  $\bot$ , i.e.,  $[\mathbf{e}]_1$  satisfies  $e([\mathbf{c}^\top]_1, [(\mathbf{K}_0^\top + \tau \mathbf{K}_1^\top)\mathbf{A}]_2) = e([\mathbf{e}^\top]_1, [\mathbf{A}]_2)$ , only when one of the following two cases occur:

- Case 1:  $[\mathbf{c}]_1 \in \mathsf{Span}([\mathbf{U}]_1), or$
- Case 2: the tag  $\tau = H(pk' = [(a\mathbf{k} + \mathbf{b})^{\top}\mathbf{U}]_1, [\mathbf{c}]_1, [d]_1)$  is *identical* to the tag  $\tau^* = H(pk = [\mathbf{k}^{\top}\mathbf{U}]_1, [\mathbf{c}^*]_1, [d^*]_1)$  involved in the challenge ciphertext  $ct^* = ([\mathbf{c}^*]_1, [d^*]_1, [\mathbf{e}^*]_1).$

If **Case 1** occurs, i.e.,  $[\mathbf{c}]_1 = [\mathbf{U}]_1 \mathbf{w}$  for some  $\mathbf{w} \in \mathbb{Z}_p^k$ , then the decryption result under *tampered* secret key  $T_{(a,\mathbf{b})}(sk) = a\mathbf{k} + \mathbf{b}$  would be  $m = [d]_1 - [(a\mathbf{k}^\top + \mathbf{b}^\top)[\mathbf{c}]_1] = [d]_1 - [(a[\mathbf{k}^\top \mathbf{U}]_1 \mathbf{w} + \mathbf{b}^\top [\mathbf{c}]_1)]$ , which leaks no information about  $sk = \mathbf{k}$  beyond  $pk = [\mathbf{k}^\top \mathbf{U}]_1$  to  $\mathcal{A}$ .

However, for all decryption queries made by  $\mathcal{A}$ , the OTSS property can only ensure *either* **Case 1** or **Case 2** occur. So, it is important for us to prove that **Case 2** can hardly occur and it is always **Case 1** that occurs, then we can use the above argument to show that the decryption oracle does not leak any information about sk beyond pk to  $\mathcal{A}$ .

To show that **Case 2** hardly occurs, we can use similar techniques as the analysis of **TagColl** in the security proof of our SIG, i.e., dividing **Case 2** into three sub-cases and analyzing them individually to show that they all rarely occur, by utilizing the concrete algebraic structures of our construction, based on the collision resistance of H and on the MDDH assumption.

<u>PUTTING Two COMPONENTS TOGETHER.</u> Overall, we carefully design our PKE construction so that the two components interplay with each other properly and help us to achieve sLTR-CCA security: the terms  $[(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{U}]_1\mathbf{w}$  in the second component ensure that the decryption oracle under *tampered* secret keys do not leak any information about  $sk = \mathbf{k}$  beyond pk to the adversary, so that the decryption oracle is of no use to the adversary, and then the sLTR-CCA security of our PKE follows from the CPA security of the first component in the *key leakage* setting.

#### 4.3 Proof of Theorem 2

Now we present the formal proof of Theorem 2. Let  $\mathcal{A}$  be any PPT adversary against the  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CCA security of PKE, where  $\mathcal{A}$  makes Q times of  $\mathcal{O}_{DEC}$  queries. We prove the theorem via a sequence of games  $G_0-G_6$ , where  $G_0$  is the  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CCA experiment, and in  $G_6$ ,  $\mathcal{A}$  has no advantage. A brief description of differences between adjacent games is summarized in Table 2.

Table 2. Brief Description of Games  $G_0-G_6$  for the  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CCA security proof of PKE, where the differences between adjacent games are highlighted in gray boxes. Here column "Challenge ciphertext  $ct^*$ " suggests how the challenge ciphertext  $ct^* = ([\mathbf{c}^*]_1, [d^*]_1, [\mathbf{e}^*]_1)$  is generated: sub-column " $[\mathbf{c}^*]_1 \leftarrow s$ " refers to the space from which  $[\mathbf{c}^*]_1$  is chosen; sub-columns " $[d^*]_1 =$ " and " $[\mathbf{e}^*]_1 =$ " show the computation of  $[d^*]_1$  and  $[\mathbf{e}^*]_1$  respectively, where  $\tau^* := H(pk, [\mathbf{c}^*]_1, [d^*]_1)$ . Column " $\mathcal{O}_{\text{DEC}}$ 's additional check" describes the additional check made by  $\mathcal{O}_{\text{DEC}}$  upon a decryption query  $(T_{(a,\mathbf{b})} \in \mathcal{T}_{\text{aff}}, ct = ([\mathbf{c}]_1, [d]_1, [\mathbf{e}]_1))$ , besides the routine check  $(T_{(a,\mathbf{b})}, ct) \neq (\mathbf{id}, ct^*) \land e([\mathbf{c}^\top]_1, [(\mathbf{K}_0^\top + \tau \mathbf{K}_1^\top)\mathbf{A}]_2) = e([\mathbf{e}^\top]_1, [\mathbf{A}]_2); \mathcal{O}_{\text{DEC}}$  outputs  $\perp$  if the check fails. Column " $\mathcal{O}_{\text{LEAK}}$ " shows the output returned by  $\mathcal{O}_{\text{LEAK}}$ . Recall that  $\mathcal{A}$  is not allowed to query  $\mathcal{O}_{\text{LEAK}}$  after receiving the challenge ciphertext.

	Challenge ciphertext $ct^*$		text $ct^*$	$\mathcal{O}_{\mathrm{DEC}}(T_{(a,\mathbf{b})} \in \mathcal{T}_{aff}, ct = ([\mathbf{c}]_1, [d]_1, [\mathbf{e}]_1))$ 's	$\mathcal{O}_{L-1}(L)$	Justification / Assumption
	$[\mathbf{c}^*]_1 \leftarrow s$	$[d^*]_1 =$	$[\mathbf{e}^*]_1 =$	additional check	CLEAK(L)	Justification/Tissumption
$G_0$	$Span([\mathbf{U}]_1)$	$[\mathbf{k}^\top \mathbf{U}]_1 \mathbf{w}^* + m_\beta$	$[(\mathbf{K}_0 + \tau^* \mathbf{K}_1) \mathbf{U}]_1 \mathbf{w}^*$		L(sk)	$(\kappa, \mathcal{T}_{aff})$ -sLTR-CCA experiment
$G_1$	$Span([\mathbf{U}]_1)$	$\mathbf{k}^{\top}[\mathbf{c}^*]_1 + m_{\beta}$	$(\mathbf{K}_0 + \tau^* \mathbf{K}_1) [\mathbf{c}^*]_1$		L(sk)	$G_0=G_1$
$G_2$	$Span([\mathbf{U}]_1)$	$\mathbf{k}^{\top}[\mathbf{c}^*]_1 + m_{\beta}$	$(\mathbf{K}_0 + \tau^* \mathbf{K}_1)[\mathbf{c}^*]_1$	$[\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1$	L(sk)	$\mathcal{D}_k$ -KerMDH on $[\mathbf{A}]_2$
$G_3$	$Span([\mathbf{U}]_1)$	$\mathbf{k}^\top [\mathbf{c}^*]_1 + m_\beta$	$(\mathbf{K}_0+\tau^*\mathbf{K}_1)[\mathbf{c}^*]_1$	$[\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1, \  au  eq  au^*$	L(sk)	Collision-resistance of $\mathcal{H}$ & $\mathcal{U}_{k+2,k}$ -KerMDH on $[\mathbf{U}]_1$
$G_4$	$\mathbb{G}_1^{k+2}$	$\mathbf{k}^{\top}[\mathbf{c}^*]_1 + m_{\beta}$	$(\mathbf{K}_0 + \tau^* \mathbf{K}_1)[\mathbf{c}^*]_1$	$[\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1, \tau \neq \tau^*$	L(sk)	$\mathcal{U}_{k+2,k}$ -MDDH on $[\mathbf{U}]_1$
$G_5$	$\mathbb{G}_1^{k+2}$	$\mathbf{k}^{\top}[\mathbf{c}^*]_1 + m_{\beta}$	$(\mathbf{K}_0 + \tau^* \mathbf{K}_1)[\mathbf{c}^*]_1$	$[\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1, \tau \neq \tau^*, \ [\mathbf{c}]_1 \in Span([\mathbf{U}]_1)$	L(sk)	Statistical arguments using the leftover entropy in $\mathbf{K}_0, \mathbf{K}_1$
$G_6$	$\mathbb{G}_1^{k+2}$	random	$(\mathbf{K}_0 + \tau^* \mathbf{K}_1) [\mathbf{c}^*]_1$	$[\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1, \tau \neq \tau^*, [\mathbf{c}]_1 \in Span([\mathbf{U}]_1)$	L(sk)	Statistical arguments using the leftover entropy in $\mathbf{K}$
						$\Pr[Win] = \frac{1}{2} \text{ in } G_6$

**<u>Game G\_0</u>**: This is the  $(\kappa, \mathcal{T}_{aff})$ -sLTR-CCA experiment (cf. Fig. 3). Let Win denote the event that  $\beta' = \beta$ . By definition,  $\mathsf{Adv}_{\mathsf{PKE},\mathcal{A},\kappa,\mathcal{T}}^{\mathsf{sltr-cca}}(\lambda) = |\operatorname{Pr}_0[\mathsf{Win}] - \frac{1}{2}|$ .

Let  $\mathbf{pp} = (\mathbf{gpar}, [\mathbf{U}]_1, [\mathbf{K}_0\mathbf{U}]_1, [\mathbf{K}_1\mathbf{U}]_1, [\mathbf{A}]_2, [\mathbf{K}_0^{\top}\mathbf{A}]_2, [\mathbf{K}_1^{\top}\mathbf{A}]_2, H)$  and  $(pk = [\mathbf{k}^{\top}\mathbf{U}]_1, sk = \mathbf{k})$ . In this game, the challenge ciphertext  $ct^*$  that encrypts  $m_\beta$  is generated as follows. The challenger samples  $\mathbf{w}^* \leftarrow \mathbb{Z}_p^k$ , computes  $[\mathbf{c}^*]_1 := [\mathbf{U}]_1\mathbf{w}^*, [d^*]_1 := [\mathbf{k}^{\top}\mathbf{U}]_1\mathbf{w}^* + m_\beta, \tau^* := H(pk, [\mathbf{c}^*]_1, [d^*]_1) \in \mathbb{Z}_p, [\mathbf{e}^*]_1 := [(\mathbf{K}_0 + \tau^*\mathbf{K}_1)\mathbf{U}]_1\mathbf{w}^*$ , and returns the challenge ciphertext  $ct^* := ([\mathbf{c}^*]_1, [d^*]_1, [\mathbf{e}^*]_1)$  to  $\mathcal{A}$ . Upon an  $\mathcal{O}_{\text{DEC}}$  query  $(T_{(a,\mathbf{b})} \in \mathcal{T}_{\text{aff}}, ct = ([\mathbf{c}]_1, [d]_1, [\mathbf{e}]_1))$ , the challenger computes the tampered key  $sk' = \mathbf{k}' := T_{(a,\mathbf{b})}(sk) = a\mathbf{k} + \mathbf{b}$  and  $pk' := [\mathbf{k}'^{\top}\mathbf{U}]_1$ , computes  $\tau := H(pk', [\mathbf{c}]_1, [d]_1)$ , and checks whether  $(T_{(a,\mathbf{b})}, ct) \neq (\mathrm{id}, ct^*) \land e([\mathbf{c}^{\top}]_1, [(\mathbf{K}_0^{\top} + \tau\mathbf{K}_1^{\top})\mathbf{A}]_2) = e([\mathbf{e}^{\top}]_1, [\mathbf{A}]_2)$  holds. If the check passes, the challenger computes  $m := [d]_1 - \mathbf{k}'^{\top}[\mathbf{c}]_1$  using the tampered key  $sk' = \mathbf{k}'$  and returns

m to  $\mathcal{A}$ ; otherwise, the challenger returns  $\perp$ . For an  $\mathcal{O}_{\text{LEAK}}$  query L, the challenger returns L(sk) to  $\mathcal{A}$  if the total leakage length is bounded by  $\kappa$ . Recall that  $\mathcal{A}$  can query  $\mathcal{O}_{\text{DEC}}$  throughout the game, but is only allowed to query  $\mathcal{O}_{\text{LEAK}}$  before receiving the challenge ciphertext.

<u>**Game G1**</u>: It is the same as  $G_0$ , except that, when generating the challenge ciphertext  $ct^*$ , the challenger computes  $[d^*]_1 := \mathbf{k}^{\top}[\mathbf{c}^*]_1 + m_{\beta}$  and  $[\mathbf{e}^*]_1 := (\mathbf{K}_0 + \tau^* \mathbf{K}_1)[\mathbf{c}^*]_1$  directly from  $[\mathbf{c}^*]_1, m_{\beta}, \tau^*$  and  $(\mathbf{k}, \mathbf{K}_0, \mathbf{K}_1)$ , without using the vector  $\mathbf{w}^*$  for  $[\mathbf{c}^*]_1 = [\mathbf{U}]_1 \mathbf{w}^*$ .

Since  $[\mathbf{c}^*]_1 = [\mathbf{U}]_1 \mathbf{w}^*$ , the changes are conceptual and  $\Pr_0[\mathsf{Win}] = \Pr_1[\mathsf{Win}]$ .

**Game G<sub>2</sub>:** It is the same as G<sub>1</sub>, except that, when answering  $\mathcal{O}_{\text{DEC}}(T_{(a,\mathbf{b})} \in \mathcal{T}_{\text{aff}}, ct = ([\mathbf{c}]_1, [d]_1, [\mathbf{e}]_1))$ , the challenger returns  $\perp$  to  $\mathcal{A}$  directly if the following check fails:

$$(T_{(a,\mathbf{b})},ct) \neq (\mathsf{id},ct^*) \land [\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1.$$

 $\begin{array}{l} Claim \ 7. \ \left| \ \Pr_1[\mathsf{Win}] - \Pr_2[\mathsf{Win}] \right| \leq \mathsf{Adv}_{\mathcal{D}_k,\mathbb{G}_2,\mathcal{B}_1}^{\mathsf{mddh}}(\lambda) + 1/(p-1) \ for \ a \ PPT \ adversary \\ \mathcal{B}_1 \ against \ the \ \mathcal{D}_k \text{-}MDDH \ assumption \ on \ [\mathbf{A}]_2. \end{array}$ 

*Proof sketch.* The proof is similar to the proof of Claim 1. Clearly,  $G_2$  is identical to  $G_1$  unless that  $\mathcal{A}$  ever makes a  $\mathcal{O}_{DEC}$  query such that

$$e([\mathbf{c}^{\top}]_1, [(\mathbf{K}_0^{\top} + \tau \mathbf{K}_1^{\top})\mathbf{A}]_2) = e([\mathbf{e}^{\top}]_1, [\mathbf{A}]_2) \land [\mathbf{e}]_1 \neq [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1.$$

We denote such an event by DecBad. Similar to the proof of Claim 1, DecBad rarely happens under the  $\mathcal{D}_k$ -KerMDH assumption on  $[\mathbf{A}]_2$  (which is further implied by the  $\mathcal{D}_k$ -MDDH assumption on  $[\mathbf{A}]_2$  according to Lemma 3). Consequently, Claim 7 follows.

<u>**Game G\_3:**</u> It is the same as  $G_2$ , except that, when answering  $\mathcal{O}_{\text{DEC}}(T_{(a,\mathbf{b})} \in \mathcal{T}_{\text{aff}}, ct = ([\mathbf{c}]_1, [d]_1, [\mathbf{e}]_1))$ , the challenger now returns  $\perp$  to  $\mathcal{A}$  directly if the following check fails:

$$(T_{(a,\mathbf{b})},ct) \neq (\mathsf{id},ct^*) \land [\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1 \land \tau \neq \tau^*$$

where  $\tau := H(pk', [\mathbf{c}]_1, [d]_1)$  and  $\tau^* := H(pk, [\mathbf{c}^*]_1, [d^*]_1)$  are the tags involved in this  $\mathcal{O}_{\text{DEC}}$  query and in the challenge ciphertext  $ct^*$ , respectively.

Claim 8.  $|\Pr_2[\mathsf{Win}] - \Pr_3[\mathsf{Win}]| \leq \mathsf{Adv}_{\mathcal{H},\mathcal{B}_2}^{\mathsf{cr}}(\lambda) + \mathsf{Adv}_{\mathcal{U}_{k+2,k},\mathbb{G}_1,\mathcal{B}_3}^{\mathsf{mddh}}(\lambda) + 1/(p-1) + 2^{-\Omega(\lambda)} \text{ for PPT adversaries } \mathcal{B}_2 \text{ against the collision-resistance of } \mathcal{H} \text{ and } \mathcal{B}_3 \text{ against the } \mathcal{U}_{k+2,k}\text{-MDDH assumption on } [\mathbf{U}]_1.$ 

*Proof sketch.* The proof is similar to the proof of Claim 2. Clearly,  $G_3$  is identical to  $G_2$  unless that  $\mathcal{A}$  ever makes a  $\mathcal{O}_{DEC}$  query such that

$$(T_{(a,\mathbf{b})},ct) \neq (\mathsf{id},ct^*) \land [\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1 \land \tau = \tau^*.$$

We denote such an event by TagColl. Similar to the proof of Claim 2, we can divide the event TagColl into three cases, analyze them individually and finally obtain Claim 8.

<u>Game G\_4</u>: It is the same as G<sub>3</sub>, except that, when generating the challenge ciphertext  $ct^*$ , the challenger samples  $[\mathbf{c}^*]_1 \leftarrow \mathbb{G}_1^{k+2}$  uniformly at random, instead of computing  $[\mathbf{c}^*]_1 := [\mathbf{U}]_1 \mathbf{w}^*$  with  $\mathbf{w}^* \leftarrow \mathbb{Z}_p^k$ .

By the  $\mathcal{U}_{k+2,k}$ -MDDH assumption on  $[\dot{\mathbf{U}}]_1$ ,  $\mathsf{G}_3$  and  $\mathsf{G}_4$  are computationally indistinguishable, and we have  $|\Pr_3[\mathsf{Win}] - \Pr_4[\mathsf{Win}]| \leq \mathsf{Adv}^{\mathsf{mddh}}_{\mathcal{U}_{k+2,k},\mathbb{G}_1,\mathcal{B}_4}(\lambda)$  for a PPT adversary  $\mathcal{B}_4$ .

**Game G<sub>5</sub>:** It is the same as G<sub>4</sub>, except that, when answering  $\mathcal{O}_{\text{DEC}}(T_{(a,\mathbf{b})} \in \mathcal{T}_{\text{aff}}, ct = ([\mathbf{c}]_1, [d]_1, [\mathbf{e}]_1))$ , the challenger now returns  $\perp$  to  $\mathcal{A}$  directly if the following check fails:

$$(T_{(a,\mathbf{b})},ct) \neq (\mathsf{id},ct^*) \land [\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau \mathbf{K}_1)\mathbf{c}]_1 \land \tau \neq \tau^* \land [\mathbf{c}]_1 \in \mathsf{Span}([\mathbf{U}]_1)$$

Claim 9.  $|\operatorname{Pr}_4[\operatorname{Win}] - \operatorname{Pr}_5[\operatorname{Win}]| \le Q/p.$ 

*Proof sketch.* The proof is similar to the proof of Claim 3. Clearly,  $G_5$  is identical to  $G_4$  unless that  $\mathcal{A}$  ever makes a  $\mathcal{O}_{DEC}$  query such that

$$(T_{(a,\mathbf{b})},ct)\neq (\mathsf{id},ct^*) \land [\mathbf{e}]_1 = [(\mathbf{K}_0 + \tau\mathbf{K}_1)\mathbf{c}]_1 \land \tau \neq \tau^* \land [\mathbf{c}]_1 \notin \mathsf{Span}([\mathbf{U}]_1).$$

We denote such an event by CBad. Similar to the proof of Claim 3, we can analyze the information about  $\mathbf{K}_0$  and  $\mathbf{K}_1$  that  $\mathcal{A}$  may obtain in  $\mathbf{G}_5$ , and use the leftover entropy in  $\mathbf{K}_0$  and  $\mathbf{K}_1$  to show that CBad occurs in a particular  $\mathcal{O}_{\text{DEC}}$  query with probability at most 1/p. Consequently, by a union bound over Q times of  $\mathcal{O}_{\text{DEC}}$  queries, Claim 9 follows.

<u>Game G<sub>6</sub></u>: It is the same as G<sub>5</sub>, except that, when generating the challenge ciphertext  $ct^*$ , the challenger samples  $[d^*]_1 \leftarrow \mathbb{G}_1$  uniformly at random, instead of computing  $[d^*]_1 := \mathbf{k}^\top [\mathbf{c}^*]_1 + m_\beta$ .

Claim 10.  $|\Pr_5[\mathsf{Win}] - \Pr_6[\mathsf{Win}]| \le 2^{-\Omega(\lambda)}$ .

*Proof.* We will show that the  $[d^*]_1 := \mathbf{k}^\top [\mathbf{c}^*]_1 + m_\beta$  in  $\mathsf{G}_5$  is statistically close to the  $[d^*]_1 \leftarrow_{\mathfrak{s}} \mathbb{G}_1$  in  $\mathsf{G}_6$ , with statistical distance at most  $2^{-\Omega(\lambda)}$ .

For the convenience of our analysis, we sample  $sk = \mathbf{k} \leftarrow \mathbb{Z}_p^{k+2}$  equivalently via

$$\mathbf{k} := \widetilde{\mathbf{k}} + \mathbf{U}^{\perp} \cdot \mathbf{r},$$

where  $\widetilde{\mathbf{k}} \leftarrow_{\$} \mathbb{Z}_p^{k+2}$  and  $\mathbf{r} \leftarrow_{\$} \mathbb{Z}_p^2$  are uniformly sampled, and  $\mathbf{U}^{\perp} \in \mathbb{Z}_p^{(k+2) \times 2}$  is a non-zero vector in the kernel of  $\mathbf{U} \in \mathbb{Z}_p^{(k+2) \times k}$  such that  $(\mathbf{U}^{\perp})^{\top} \cdot \mathbf{U} = \mathbf{0}$ . Below we analyze the information about  $\mathbf{r}$  that  $\mathcal{A}$  may obtain in  $\mathsf{G}_5$  (except for the challenge ciphertext  $ct^*$ ).

– Firstly, the public key pk is

$$[\mathbf{k}^{\top}\mathbf{U}]_1 = [(\widetilde{\mathbf{k}}^{\top} + \mathbf{r}^{\top} \cdot (\mathbf{U}^{\perp})^{\top})\mathbf{U}]_1 = [\widetilde{\mathbf{k}}^{\top}\mathbf{U}]_1$$

thus  $\mathbf{r}$  is completely hidden.

– In  $\mathcal{O}_{\text{Dec}}$  queries, the tampered public key pk' is

$$[\mathbf{k}'^{\top}\mathbf{U}]_1 = [(a\mathbf{k}^{\top} + \mathbf{b}^{\top})\mathbf{U}]_1 = [(a(\widetilde{\mathbf{k}}^{\top} + \mathbf{r}^{\top} \cdot (\mathbf{U}^{\perp})^{\top}) + \mathbf{b}^{\top})\mathbf{U}]_1 = [(a\widetilde{\mathbf{k}}^{\top} + \mathbf{b}^{\top})\mathbf{U}]_1$$

thus **r** is completely hidden. Moreover, due to the game change in  $G_5$ , the challenger will not output  $m := [d]_1 - \mathbf{k}^{\prime \top}[\mathbf{c}]_1$  unless  $[\mathbf{c}]_1 \in \mathsf{Span}([\mathbf{U}]_1)$ , and for  $[\mathbf{c}]_1 \in \mathsf{Span}([\mathbf{U}]_1)$ , we have

$$m = [d]_1 - \mathbf{k}'^{\top} [\mathbf{c}]_1 = [d]_1 - (a\mathbf{k}^{\top} + \mathbf{b}^{\top}) [\mathbf{c}]_1$$
$$= [d]_1 - (a(\widetilde{\mathbf{k}}^{\top} + \mathbf{r}^{\top} \cdot (\mathbf{U}^{\perp})^{\top}) + \mathbf{b}^{\top}) [\mathbf{c}]_1$$
$$= [d]_1 - (a\widetilde{\mathbf{k}}^{\top} + \mathbf{b}^{\top}) [\mathbf{c}]_1,$$

thus  $\mathbf{r}$  is also completely hidden.

- From  $\mathcal{O}_{\text{LEAK}}$  queries,  $\mathcal{A}$  obtains at most  $\kappa$  bits information  $L(sk) = L(\mathbf{k}) = L(\mathbf{k} + \mathbf{U}^{\perp} \cdot \mathbf{r})$  about sk, and also about  $\mathbf{r}$ .

Overall, the information about  $\mathbf{r}$  that  $\mathcal{A}$  learns in  $\mathsf{G}_5$  (except for the challenge ciphertext  $ct^*$ ) is at most  $\kappa$  bits. Thus, there are still  $2\log p - \kappa = \log p + (\log p - \kappa) \geq \log p + \Omega(\lambda)$  bits of entropy left in  $\mathbf{r}$ , and also in  $sk = \mathbf{k} = \widetilde{\mathbf{k}} + \mathbf{U}^{\perp} \cdot \mathbf{r}$ .

On the other hand, for the challenge ciphertext  $ct^*$ , note that  $[\mathbf{c}^*]_1$  is uniformly chosen from  $\mathbb{G}_1^{k+2}$  due to the game change in  $\mathbf{G}_4$ , thus the mapping  $\mathbf{k} \mapsto \mathbf{k}^\top [\mathbf{c}^*]_1$  indexed by  $[\mathbf{c}^*]_1$  is a universal hash function. By the leftover hash lemma (i.e., Lemma 1),  $\mathbf{k}^\top [\mathbf{c}^*]_1$  is statistically close to the uniform distribution over  $\mathbb{G}_1$ , with statistical distance at most  $\sqrt{p \cdot 2^{-(\log p + \Omega(\lambda))}} = \sqrt{2^{-\Omega(\lambda)}}$ , which is also  $2^{-\Omega(\lambda)}$ . Consequently, the  $[d^*]_1 := \mathbf{k}^\top [\mathbf{c}^*]_1 + m_\beta$  in  $\mathbf{G}_5$  is also statistically close to the uniform distribution, with statistical distance at most  $2^{-\Omega(\lambda)}$ . Therefore,  $\mathbf{G}_5$  and  $\mathbf{G}_6$  are statistically indistinguishable to  $\mathcal{A}$ , and Claim 10 follows.

Finally in  $G_6$ ,  $[d^*]_1$  is uniformly chosen regardless of the value of  $\beta$ , thus the challenge bit  $\beta$  is completely hidden to  $\mathcal{A}$ . Then  $\Pr_6[\mathsf{Win}] = \frac{1}{2}$ .

Taking all things together, and noting that by Lemma 2, the  $\mathcal{U}_{k+2,k}$ -MDDH assumption is implied by the  $\mathcal{D}_k$ -MDDH assumption, Theorem 2 follows.

Acknowledgments. We would like to thank the reviewers for their valuable comments. Shuai Han and Shengli Liu were partially supported by National Natural Science Foundation of China (Grant Nos. 62372292, 61925207), the National Key R&D Program of China under Grant 2022YFB2701500, Guangdong Major Project of Basic and Applied Basic Research (2019B030302008), and Young Elite Scientists Sponsorship Program by China Association for Science and Technology (YESS20200185). Dawu Gu was partially supported by the National Key R&D Program of China under Grant 2020YFA0712302.

#### References

- Akavia, A., Goldwasser, S., Vaikuntanathan, V.: Simultaneous hardcore bits and cryptography against memory attacks. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 474–495. Springer, Heidelberg (Mar 2009). https://doi.org/10.1007/978-3-642-00457-5\_28
- [2] An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (Apr / May 2002). https://doi.org/10.1007/3-540-46035-7\_6
- [3] Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: Biham, E. (ed.) EURO-CRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (May 2003). https://doi.org/10.1007/3-540-39200-9\_31
- [4] Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASI-ACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (Dec 2000). https://doi.org/10.1007/3-540-44448-3\_41
- [5] Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48329-2\_21
- Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 513–525. Springer, Heidelberg (Aug 1997). https://doi.org/10.1007/BFb0052259
- [7] Chakraborty, S., Rangan, C.P.: Public key encryption resilient to post-challenge leakage and tampering attacks. In: Matsui, M. (ed.) CT-RSA 2019. LNCS, vol. 11405, pp. 23–43. Springer, Heidelberg (Mar 2019). https://doi.org/10.1007/978-3-030-12612-4\_2
- [8] Cramer, R., Shoup, V.: A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In: Krawczyk, H. (ed.) CRYPTO'98. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (Aug 1998). https://doi.org/10.1007/BFb0055717
- [9] Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EURO-CRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (Apr / May 2002). https://doi.org/10.1007/3-540-46035-7\_4
- [10] Damgård, I., Faust, S., Mukherjee, P., Venturi, D.: Bounded tamper resilience: How to go beyond the algebraic barrier. In: Sako, K., Sarkar, P. (eds.) ASI-ACRYPT 2013, Part II. LNCS, vol. 8270, pp. 140–160. Springer, Heidelberg (Dec 2013). https://doi.org/10.1007/978-3-642-42045-0\_8
- [11] Diemert, D., Gellert, K., Jager, T., Lyu, L.: More efficient digital signatures with tight multi-user security. In: Garay, J. (ed.) PKC 2021, Part II. LNCS, vol. 12711, pp. 1–31. Springer, Heidelberg (May 2021). https://doi.org/10.1007/978-3-030-75248-4\_1
- [12] Dodis, Y., Haralambiev, K., López-Alt, A., Wichs, D.: Efficient publickey cryptography in the presence of key leakage. In: Abe, M. (ed.) ASI-ACRYPT 2010. LNCS, vol. 6477, pp. 613–631. Springer, Heidelberg (Dec 2010). https://doi.org/10.1007/978-3-642-17373-8\_35
- [13] Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. In: Yao, A.C.C. (ed.) ICS 2010. pp. 434–452. Tsinghua University Press (Jan 2010)

- [14] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147. Springer, Heidelberg (Aug 2013). https://doi.org/10.1007/978-3-642-40084-1\_8
- [15] Faonio, A., Venturi, D.: Efficient public-key cryptography with bounded leakage and tamper resilience. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 877–907. Springer, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53887-6\_32
- [16] Faust, S., Mukherjee, P., Nielsen, J.B., Venturi, D.: Continuous non-malleable codes. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 465–488. Springer, Heidelberg (Feb 2014). https://doi.org/10.1007/978-3-642-54242-8\_20
- [17] Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (Aug 1987). https://doi.org/10.1007/3-540-47721-7\_12
- [18] Fujisaki, E., Xagawa, K.: Public-key cryptosystems resilient to continuous tampering and leakage of arbitrary functions. In: Cheon, J.H., Takagi, T. (eds.) ASI-ACRYPT 2016, Part I. LNCS, vol. 10031, pp. 908–938. Springer, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53887-6\_33
- [19] Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: Concrete results. In: Koç, Çetin Kaya., Naccache, D., Paar, C. (eds.) CHES 2001. LNCS, vol. 2162, pp. 251–261. Springer, Heidelberg (May 2001). https://doi.org/10.1007/3-540-44709-1\_21
- [20] Gay, R., Hofheinz, D., Kiltz, E., Wee, H.: Tightly CCA-secure encryption without pairings. In: Fischlin, M., Coron, J.S. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 1–27. Springer, Heidelberg (May 2016). https://doi.org/10.1007/978-3-662-49890-3\_1
- [21] Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 258–277. Springer, Heidelberg (Feb 2004). https://doi.org/10.1007/978-3-540-24638-1\_15
- [22] Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (Apr 2008). https://doi.org/10.1007/978-3-540-78967-3\_24
- [23] Halderman, J.A., Schoen, S.D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J.A., Feldman, A.J., Appelbaum, J., Felten, E.W.: Lest we remember: Cold boot attacks on encryption keys. In: van Oorschot, P.C. (ed.) USENIX Security 2008. pp. 45–60. USENIX Association (Jul / Aug 2008)
- [24] Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28(4), 1364–1396 (1999). https://doi.org/10.1137/S0097539793244708
- [25] Jafargholi, Z., Wichs, D.: Tamper detection and continuous non-malleable codes. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 451–480. Springer, Heidelberg (Mar 2015). https://doi.org/10.1007/978-3-662-46494-6\_19
- [26] Kalai, Y.T., Kanukurthi, B., Sahai, A.: Cryptography with tamperable and leaky memory. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 373–390. Springer, Heidelberg (Aug 2011). https://doi.org/10.1007/978-3-642-22792-9\_21
- [27] Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (Apr 2015). https://doi.org/10.1007/978-3-662-46803-6\_4

- [28] Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO'96. LNCS, vol. 1109, pp. 104– 113. Springer, Heidelberg (Aug 1996). https://doi.org/10.1007/3-540-68697-5\_9
- [29] Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) CRYPTO'99. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (Aug 1999). https://doi.org/10.1007/3-540-48405-1\_25
- [30] Liu, F.H., Lysyanskaya, A.: Tamper and leakage resilience in the split-state model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 517–532. Springer, Heidelberg (Aug 2012). https://doi.org/10.1007/978-3-642-32009-5\_30
- [31] Morillo, P., Ràfols, C., Villar, J.L.: The kernel matrix Diffie-Hellman assumption. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 729–758. Springer, Heidelberg (Dec 2016). https://doi.org/10.1007/978-3-662-53887-6\_27
- [32] Naor, M., Segev, G.: Public-key cryptosystems resilient to key leakage. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 18–35. Springer, Heidelberg (Aug 2009). https://doi.org/10.1007/978-3-642-03356-8\_2
- [33] Qin, B., Liu, S.: Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 381–400. Springer, Heidelberg (Dec 2013). https://doi.org/10.1007/978-3-642-42045-0\_20
- [34] Steinfeld, R., Pieprzyk, J., Wang, H.: How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 357–371. Springer, Heidelberg (Feb 2007). https://doi.org/10.1007/11967668\_23
- [35] Sun, S., Gu, D., Au, M.H., Han, S., Yu, Y., Liu, J.K.: Strong leakage and tamper-resilient PKE from refined hash proof system. In: Deng, R.H., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds.) ACNS 2019. pp. 486–506 (2019). https://doi.org/10.1007/978-3-030-21568-2.24
- [36] Sun, S., Gu, D., Parampalli, U., Yu, Y., Qin, B.: Public key encryption resilient to leakage and tampering attacks. Journal of Computer and System Sciences 89, 142–156 (2017). https://doi.org/10.1016/j.jcss.2017.03.004