Public-Key Cryptosystems Resilient to Continuous Tampering and Leakage of Arbitrary Functions

Submission to ASIACRYPT 2016

No Institute Given

Abstract. We present the first chosen-ciphertext secure public-key encryption schemes resilient to continuous tampering of arbitrary (efficiently computable) functions. Since it is impossible to realize such schemes without a self-destruction or key-updating mechanism, our proposals allow for either of them. As in the previous works resilient to this type of tampering attacks, our schemes also tolerate bounded or continuous memory leakage attacks at the same time. Unlike the previous works, our schemes have efficient instantiations. We also prove that there is no secure digital signature scheme resilient to arbitrary tampering functions against a stronger variant of the continuous tampering attack, even if it has a self-destruction mechanism.

Keywords: public-key encryption, digital signature, continuous tampering attacks, and bounded or continuous memory leakage.

1 Introduction

We study the tampering attack security, or equivalently the related-key attack security, of public-key cryptosystems. The tampering attacks allow an adversary to modify the secret of a target cryptographic device and observe the effect of the changes at the output. For instance, the tampering attacks are mounted on the IND-CCA game of a public-key encryption (PKE) scheme, where an adversary may tamper with the secret-key and observe the output of the decryption oracle with the tampered secret.

Theoretical treatment of tampering attack starts independently by Gennaro et al. [21] and Bellare and Kohno [6]. The former treats *arbitrary* (efficiently computable) tampering functions, whereas the latter considers a *restricted* class of tampering functions. Allowing tampering of all (efficiently computable) functions is very challenging.

Gennaro et al. [21] make a strong compromise that a trusted-third party may publish its verification key (of a secure digital signature scheme) as a part of public parameters where an adversary is not allowed to modify the parameters, and each user may obtain a digital signature on their *secret* issued by the trusted-third party. We call this model **the on-line model** or **the algorithmic tamper-proof security model** [21]. Bellare and Kohno [6] assume no trusted party. However, subsequent works [4, 5, 7, 35, 28, 33, 20] allow a trusted party to play a minimum role, where it makes a public parameter, but once it did, it does nothing. An adversary is not allowed to modify the public parameter. We call this model **the common reference model**.

Gennaro et al. [21] suggested that it is *impossible* to realize chosen-ciphertext attack (CCA) secure PKE and digital signature schemes resilient to all tampering functions even in the on-line model. Therefore, they allowed a cryptosystem to **self-destruct**, meaning that when detecting tampering, a cryptographic device can erase all internal data and an adversary cannot gain any additional information from it.

Other possible ways to bypass the impossibility result are (1) to allow a device to update its inner secret with fresh randomness (**key-updating** mechanism) [26], and (2) to allow an adversary to submit a *bounded* number of tampering queries (**bounded tampering**) [14].

Tampering is further classified into **persistent** or **non-persistent** (due to [25]). In **the persistent tampering attacks**, each tampering is applied to the current version of the secret that has been overwritten by the previous tampering function, so the previous secret is lost, i.e., when an adversary queries (ϕ_1, x_1) and (ϕ_2, x_2) to device $G(s, \cdot)$ in this order, it receives $G(\phi_1(s), x_1)$ and $G(\phi_2(\phi_1(s)), x_2)$, where ϕ_1, ϕ_2 are tampering functions and x_1, x_2 are inputs to device G. In **the non-persistent tampering attacks**, a tampering is always applied to the original secret, i.e., an adversary receives $G(\phi_1(s), x_1)$ and $G(\phi_2(s), x_2)$ when submitting the same queries above. We insist that for PKE and digital signature schemes **without a key-updating mechanism**, *the non-persistent tampering attacks are stronger than the persistent tampering attacks*, because an adversary that breaks a cryptosystem in the persistent tampering attacks also breaks the same system in the non-persistent tampering attacks. It is not clear in a cryptosystem with a key-updating mechanism that the relation holds.

In this paper we focus on the common reference string (CRS) model (as mentioned above), where we assume a public parameter is generated by a trusted third party and assume that an adversary is not allowed to modify it. This setting is common in many prior works, e.g., [4, 5, 7, 35, 28, 26, 14, 33, 20].

At Crypto 2011, Kalai, Kanukurthi, and Sahai [26] considered **the continual tampering and leakage (CTL) model**, where persistent tampering is assumed, and PKE and digital signature schemes are allowed to have a key-updating algorithm, which updates a secret key with fresh (non-tampered) randomness in every time period. This security model is considered in the CRS model. The proposed PKE scheme is one-bit-message encryption scheme based on [10] and is only chosen-plaintext attack (CPA) secure. Therefore, in their CTL security model for PKE, an adversary is NOT allowed to access the decryption oracle, which means that an adversary cannot observe the effect of tampering at the output of the decryption oracle. Instead, it can observe the effect of tampering at the output of the leakage oracle. We note that this tampering attack is not trivially implied by the leakage attack, because tampered secret $\phi(sk)$ is updated and the adversary can observe a partial information on the updated secret, say $L(\mathsf{Update}(\phi(sk)))$, from the leakage oracle. Their digital signature scheme (with a key-update mechanism) is constructed based on their CTL secure PKE scheme with simulation-sound non-interactive zero-knowledge proofs, which is simply inefficient. They also considered a digital signature scheme without a key-update mechanism and its security in the so-called continuous tampering and bounded leakage (CTBL) model. The digital signature scheme may self-destruct (otherwise, it is impossible to prove the security). They claim the security of the digital signature scheme against the persistent tampering attack. Remember for a digital signature scheme without a key-updating mechanism, the non-persistent tampering attacks are stronger than the persistent tampering attacks. We later prove that it is impossible to construct a digital signature scheme without a key-updating mechanism that is resilient to continuous tampering against the non-persistent tampering attack (even if the scheme has a self-destructive mechanism).

At Asiacrypt 2013, Damgård, Faust, Mukherjee, and Venturi [14] proposed **the bounded leakage and tampering (BLT) model**. This setting allows **a bounded number** of non-persistent tampering, as well as bounded memory leakage, in the CRS model, where PKE has neither of self-destructive nor key-updating mechanism. In the BLT model of PKE, in addition to accessing bounded memory leakage oracle, an adversary is allowed to submit *a bounded number* of "pre-challenge" tampering queries (ϕ, CT) to the decryption oracle and receive $\mathbf{D}(\phi(sk), \mathsf{CT})$. It may also access the decryption oracle with the original secret-key both in the pre-challenge and post-challenge stages, as in the normal IND-CCA game. They presented a generic construction of IND-CCA BLT secure PKE scheme from an IND-CPA BLT secure PKE scheme with tSE NIZK proofs [15]. An instantiation of an IND-CPA BLT secure PKE scheme is BHHO PKE scheme [9]. Using the technique of [2], they also consider a variant of the floppy model [2], called **the ι-Floppy model**, where each user has individual secret y different from secret-key sk and is allowed to execute "invisible key updates", i.e., to update their secret key sk with (non-tampered) secret y and (non-tampered) flesh randomness.

1.1 Our Results

We study continuous tampering attacks of arbitrary functions against PKE and digital signature schemes, in the presence of bounded or continuous memory leakage attacks. Due to the impossibility result, we allow PKE and digital signature schemes to have either self-destructive or key-updating mechanism. There is no IND-CCA PKE scheme resilient to post-challenge tampering of arbitrary functions [14]. Indeed, one can break any PKE scheme, by observing the output of the decryption oracle after tampering with the following effciently computable function:

$$\phi(sk) = \begin{cases} sk & \text{if } \mathbf{D}(sk, \mathsf{CT}^*) = m_0, \text{ where } \mathsf{CT}^* \text{ is a challenge ciphertext.} \\ \perp & \text{otherwise.} \end{cases}$$

This attack is unavoidable even with self-destruction, key-updating, and bounded persistent/non-persistent tampering in the on-line model (i.e., in the strongest compromised model). Therefore, we allow tampering queries only in the pre-challenge stage against a PKE scheme.

We present the first chosen-ciphertext secure PKE schemes secure against both continuous tampering and bounded or continuous memory leakage of arbitrary functions. Interestingly, we show that, by putting some parameters in the common reference string and providing a self-destructive mechanism to the decryption algorithm, Qin and Liu's PKE scheme [31] is CTBL-CCA secure, meaning that it is IND-CCA secure resilient to both continuous tampering and bounded memory leakage attacks. We also propose the first CTL-CCA secure PKE scheme, meaning that it is IND-CCA secure resilient to both continuous tampering and continual memory leakage attacks. Our security definitions basically model the non-persistent tampering attack, but it is straightforward to modify them to the persistent one. It is easy to show that any PKE schemes without a key-updating mechanism that is CTBL-CCA secure against the non-persistent tampering attack is still CTBL-CCA secure against the persistent tampering attack. So are our CTBL-CCA secure PKE schemes. It is not clear in the case of PKE schemes with a key-updating mechanism that the same relation holds.

We show that it is impossible to construct a secure digital signature scheme resilient to the (continuous) **non-persistent** tampering attack even if it has a self-destructive mechanism. If the key-updating mechanism runs only when a tampering is detected, any digital signature scheme with the key-updating mechanism is insecure, either.

Comparison Among Continuous Tampering Models. Table 1 classifies the security models related to our continuous tampering models. Here $\mathsf{b\text{-}tamp}$ indicates bounded tampering and $\mathsf{c\text{-}tamp}$

indicates continuous tampering. Similarly, b-leak indicates bounded memory leakage and c-tamp indicates continuous memory leakage. persist indicates persistent tampering and n-persist indicates non-persistent tampering. per./n-per. indicates that the result in this row is effective for both persistent and non-persistent tampering. c-tamp^{-} indicates the case of KKS signature scheme with a key-updating mechanism [26], which allows an adversary to submit a *bounded* number of tampering queries within each time period, although the number of tampering queries overall is unbounded. Our result is given in the gray area. Our CTL model imposes a more severe condition in that the scheme is allowed to update secret keys only when it detects tampering.

Table 1. Comparison: Continuous Tampering Models and Results

Primitives	Self-Dest.	Key Update	Tampering	Leakage	Security	Notes	Results
PKE	w/o.	w/o.	b-tamp	b-leak	CCA	per./n-per.	DFMV [14]
PKE	w/o.	w.	c-tamp	c-leak	CCA	ιFloppy	DFMV [14]
PKE	w.	w.	b-tamp	-	CCA	post-tamp.	Impossible([14])
PKE	w/o.	w/o.	c-tamp	-	CCA	per./n-per.	Impossible ([21])
PKE	w/o.	w.	c-tamp	c-leak	CPA	persist	KKS [26]
PKE	w.	w/o.	c-tamp	b-leak	CCA	per./n-per.	This work
PKE	w/o.	w.	c-tamp	c-leak	CCA	n-persist	This work
Sig	w/o.	w/o.	c-tamp	-	CMA	per./n-per.	Impossible ([21])
Sig	w.	w/o.	c-tamp	b-leak	?	persist	KKS [26]
Sig	w/o.	w.	c-tamp^{-}	c-leak	CMA	persist	KKS [26]
Sig	w.	w/o.	c-tamp	-	CMA	n-persist	Impossible (This work)
Sig	w/o.	w.	c-tamp	-	CMA	n-persist	Impossible (This work)

1.2 Other Related Works

Considering a restricted class of tampering functions, we briefly mention two lines of works.

One research stream derives from Bellare and Kono's [6], who study tampering (or equivalently related-key) resilient security against specific primitives, such as pseudo-random function families (PRFs), PKE, and identity-based encryption (IBE) schemes. By restricting tampering functions, post-challenge tampering queries can be treated at PKE. Currently, it is known that there is an IBE scheme (and hence, converted to PKE and digital signature schemes) resilient to polynomial functions [7] (in the CRS model). Qin et al. [33] recently claimed a broader class, but it is not correct [20] (Indeed, there is a counter example [3]). Recently, Fujisaki and Xagawa proposed an IBE scheme resilient to some kind of invertible functions [20]. In the above works, tampering is non-persistent and the target cryptosystems have neither self-destruction nor key-updating mechanism.

The other line of works comes from algebraic manipulation detection (AMD) codes [11, 12] and non-malleable codes (NMC) [18], whose codes can detect tampering of a certain class of functions. Dziembowski, Pietrzak, and Wichs [18] presented NMC and its application to tamper-resilient security. In their model, a PKE scheme allows for both self-destruction and key-updating mechanisms. An adversary submit to a target device G tampering queries (ϕ, x) with $\phi \in \Phi$. If the decoding fails, i.e., $\mathsf{Dec}(\phi(\mathsf{Enc}(s)) = \perp$, then G self-destructs. Otherwise, it returns $G(s, x)$ and updates $\mathsf{Enc}(s)$. Faust, Mukherjee, Nielsen, and Ventrui [19] considered *continuous NMC* and apply it to tamper and leakage resilient security (in the split-state model). Recently, Jafargholi and Wichs [25] presented NMCs for a bounded number of any subset of a very broader class of tampering functions. However, since an adversary must choose the subset before seeing the parameters of the codes, this result is not effective against the setting of this paper.

2 Preliminaries

For $n \in \mathbb{N}$ (the set of natural numbers), $[n]$ denotes the set $\{1, \ldots, n\}$. We let $\mathsf{negl}(\kappa)$ to denote an unspecified function $f(\kappa)$ such that $f(\kappa) = \kappa^{-\omega(1)} = 2^{-\omega(1)\log\kappa}$, saying that such a function is negligible in κ. We write PPT and DPT algorithms to denote probabilistic polynomial-time and deterministic poly-time algorithms, respectively. For PPT algorithm A, we write $y \leftarrow A(x)$ to denote the experiment of running A for given x, picking inner coins r uniformly from an appropriate domain, and assigning the result of this experiment to the variable y, i.e., $y = A(x; r)$. Let $X = \{X_\kappa\}_{\kappa\in\mathbb{N}}$ and $Y = \{Y_\kappa\}_{\kappa\in\mathbb{N}}$ be probability ensembles such that each X_κ and Y_κ are random variables ranging over $\{0,1\}^\kappa$. The (statistical) distance between X_κ and Y_κ is $\mathsf{Dist}(X_\kappa : Y_\kappa) \triangleq \frac{1}{2} \cdot |\Pr_{s\in\{0,1\}^\kappa}[X = s] - \Pr_{s\in\{0,1\}^\kappa}[Y = s]|$. We say that two probability ensembles, X and Y, are statistically indistinguishable (in κ), denoted $X \overset{\mathrm{s}}{\approx} Y$, if $\mathsf{Dist}(X_\kappa : Y_\kappa) = \mathsf{negl}(\kappa)$. In particular, we denote by $X \equiv Y$ to say that X and Y are identical. We say that X and Y are computationally indistinguishable (in κ), denoted $X \overset{\mathrm{c}}{\approx} Y$, if for every non-uniform PPT D (ranging over $\{0,1\}$), $\{D(1^\kappa, X_\kappa)\}_{\kappa\in\mathbb{N}} \overset{\mathrm{s}}{\approx} \{D(1^\kappa, Y_\kappa)\}_{\kappa\in\mathbb{N}}$.

2.1 Entropy and Extractor

The min-entropy of random variable X is defined as $\mathsf{H}_\infty(X) = -\log(\max_x \Pr[X = x])$. We say that a function $\mathsf{Ext} : \{0,1\}^{\ell_s} \times \{0,1\}^n \to \{0,1\}^m$ is an (k, ϵ)-strong extractor if for any random variable X such that $X \in \{0,1\}^n$ and $\mathsf{H}_\infty(X) > k$, it holds that $\mathsf{Dist}((S, \mathsf{Ext}(S, X)), (S, U_m)) \leq \epsilon$, where S is uniform over $\{0,1\}^{l_s}$. Let $\mathcal{H} = \{H\}$ be a family of hash functions $H : \{0,1\}^n \to \{0,1\}^m$. $\mathcal{H}$ is called a family of universal hash functions if $\forall\ x_1, x_2 \in \{0,1\}^n$ with $x_1 \neq x_2$, $\Pr_{H\leftarrow\mathcal{H}}[H(x_1) = H(x_2)] = 2^{-m}$. Then, The Leftover Hash Lemma (LHL) states the following.

Lemma 1 (Leftover Hash Lemma). *Assume that the family $\mathcal{H}$ of functions $H : \{0,1\}^n \to \{0,1\}^m$ is a family of universal hash functions. Then for any random variable X such that $X \in \{0,1\}^n$ and $\mathsf{H}_\infty(X) > m$,*

$$\mathsf{Dist}((H, H(X)), (H, U_m)) \leq \frac{1}{2}\sqrt{2^{-(\mathsf{H}_\infty(X)-m)}},$$

where H is a random variable uniformly chosen over $\mathcal{H}$ and U_m is a random variable uniformly chosen over $\{0,1\}^m$.

Therefore, H constructs a $(k, 2^{-(k/2+1)})$-strong extractor where $k = \mathsf{H}_\infty(X) - m$.

We use the notion of the average conditional min-entropy defined by Dodis et al.[17] and its "chain rule". Define the average conditional min-entropy of random variable X given random variable Y as

$$\widetilde{\mathsf{H}}_\infty(X|Y) \triangleq -\log\big(\mathop{\mathbf{E}}_{y\leftarrow Y}[\max_x \Pr[X = x|Y = y]]\big) = -\log\big(\mathop{\mathbf{E}}_{y\leftarrow Y}[2^{-H_\infty(X|Y=y)}]\big).$$

Lemma 2 ("Chain Rule" for Average Min-Entropy [17]). *When random variable Z takes at most 2^r possible values (i.e., $\#\mathsf{Supp}(Z) = 2^r$) and X, Y are random variables, then*

$$\widetilde{\mathsf{H}}_\infty(X|(Y,Z)) \geq \widetilde{\mathsf{H}}_\infty((X,Y)|Z) - r \geq \widetilde{\mathsf{H}}_\infty(X|Z) - r.$$

In particular,

$$\widetilde{\mathsf{H}}_\infty(X|Z) \geq \mathsf{H}_\infty(X,Z) - r \geq \mathsf{H}_\infty(X) - r.$$

Dodis et al.[17] proved that any strong extractor is an average-case strong extractor for an appropriate setting of the parameters. As a special case, they showed any family of universal hash functions is an average-case strong extractor along with the following generalized version of the leftover hash lemma:

Lemma 3 (Generalized Leftover Hash Lemma [17]). *Assume that the family $\mathcal{H}$ of functions $H : \{0,1\}^n \to \{0,1\}^m$ is a family of universal hash functions. Then for any random variables, X and Z,*

$$\mathsf{Dist}((H, H(X), Z), (H, U_m, Z)) \leq \frac{1}{2}\sqrt{2^{-(\widetilde{\mathsf{H}}_\infty(X|Z)-m)}},$$

where H is a random variable uniformly chosen over $\mathcal{H}$ and U_m is a random variable uniformly chosen over $\{0,1\}^m$.

2.2 Random Subspace Lemmas

The following random subspace lemma was given by Brakerski et al. [10]. We use a better bound given by Agrawal et al. [2].

Lemma 4 ([10, 2]). *Let $2 \leq d < t \leq n$ and $\lambda < (d-1)\log(q)$. Let $\mathcal{W} \subset \mathbb{F}_q^n$ be an arbitrary vector subspace in $\mathbb{F}_q^n$ of dimension t. Let $L : \{0,1\}^* \to \{0,1\}^\lambda$ be an arbitrary function. Then, we have*

$$\mathsf{Dist}\Big(\Big(\mathbf{A}, L(\mathbf{A}\boldsymbol{v})\Big), \Big(\mathbf{A}, L(\boldsymbol{u})\Big)\Big) \leq \frac{n}{2}\sqrt{\frac{2^\lambda}{q^{d-1}}},$$

where $\mathbf{A} := (\boldsymbol{a_1}, \ldots, \boldsymbol{a_d}) \leftarrow \mathcal{W}^d$ (seen as a $n \times d$ matrix), $\boldsymbol{v} \leftarrow \mathbb{F}_q^d$, and $\boldsymbol{u} \leftarrow \mathcal{W}$.

Brakerski et al. bound is $\sqrt{\frac{2^\lambda}{q^{d-3}}}$. The above bound is better because $\frac{n}{2} < q$ where $n = poly(\kappa)$.

The following is an affine version of Lemma 4.

Lemma 5. *Let $2 \leq d < t \leq n$ and $\lambda < (d-1)\log(q)$. Let $\boldsymbol{x} \in \mathbb{F}_q^n$ be an arbitrary vector. Let $\mathcal{W} \subset \mathbb{F}_q^n$ be an arbitrary vector subspace in $\mathbb{F}_q^n$ of dimension t. Let $L : \{0,1\}^* \to \{0,1\}^\lambda$ be an arbitrary function. Then, we have*

$$\mathsf{Dist}\Big(\Big(\mathbf{A}, L(\boldsymbol{x} + \mathbf{A}\boldsymbol{v})\Big), \Big(\mathbf{A}, L(\boldsymbol{x} + \boldsymbol{u})\Big)\Big) \leq \frac{n}{2}\sqrt{\frac{2^\lambda}{q^{d-1}}},$$

where $\mathbf{A} := (\boldsymbol{a_1}, \ldots, \boldsymbol{a_d}) \leftarrow \mathcal{W}^d$ (seen as a $n \times d$ matrix), $\boldsymbol{v} \leftarrow \mathbb{F}_q^d$, and $\boldsymbol{u} \leftarrow \mathcal{W}$.

Proof. Let $\mathbf{W} \in \mathbb{F}_q^{n\times t}$ be a matrix whose column vectors span $\mathcal{W}$, i.e., $\mathcal{W} = \mathsf{span}(\mathbf{W})$. Now, we have

$$\begin{aligned}
&\mathsf{Dist}\Big(\big(\mathbf{A}, L(\boldsymbol{x} + \mathbf{A}\boldsymbol{v})\big), \big(\mathbf{A}, L(\boldsymbol{x} + \boldsymbol{u})\big)\Big)\\
&= \mathsf{Dist}\Big(\big(\mathbf{W}\mathbf{R_a}, L(\boldsymbol{x} + \mathbf{W}\mathbf{R_a}\boldsymbol{v})\big), \big(\mathbf{W}\boldsymbol{r_a}, L(\boldsymbol{x} + \mathbf{W}\boldsymbol{r_u})\big)\Big) \quad (\text{ where } \mathbf{A} = \mathbf{W}\mathbf{R_a}\ \boldsymbol{u} = \mathbf{W}\boldsymbol{r_u})\\
&= \mathsf{Dist}\Big(\big(\mathbf{W}\mathbf{R_a}, L'(\mathbf{R_a}\boldsymbol{v})\big), \big(\mathbf{W}\mathbf{R_a}, L'(\boldsymbol{r_u})\big)\Big) \quad (\text{ where } L'(\boldsymbol{y}) := L(\boldsymbol{x} + \mathbf{W}\boldsymbol{y}))\\
&\le \mathsf{Dist}\Big(\big(\mathbf{R_a}, L'(\mathbf{R_a}\boldsymbol{v})\big), \big(\mathbf{R_a}, L'(\boldsymbol{r_u})\big)\Big) \le \frac{n}{2}\sqrt{\frac{2^{\lambda}}{q^{d-1}}},
\end{aligned}$$

where $\mathbf{R_a} \leftarrow \mathbb{F}_q^{t\times d}$, $\boldsymbol{v} \leftarrow \mathbb{F}_q^d$, and $\boldsymbol{r_u} \leftarrow \mathbb{F}_q^t$.

2.3 Hash Proof Systems

We recall the notion of the hash proof systems introduced by Cramer and Shoup [13]. Let $\mathcal{C}, \mathcal{K}, \mathcal{SK}$, and $\mathcal{PK}$ be efficiently samplable sets and let $\mathcal{V}$ be a subset in $\mathcal{C}$. Let $\Lambda_{sk} : \mathcal{C} \to \mathcal{K}$ be a hash function indexed by $sk \in \mathcal{SK}$. A hash function family $\Lambda : \mathcal{SK} \times \mathcal{C} \to \mathcal{K}$ is projective if there is a projection $\mu : \mathcal{SK} \to \mathcal{PK}$ such that $\mu(sk) \in \mathcal{PK}$ defines the action of Λ_{sk} over subset $\mathcal{V}$. That is to say, for every $C \in \mathcal{V}$, $K = \Lambda_{sk}(C)$ is uniquely determined by $\mu(sk)$ and C. Λ is called γ-*entropic* [27] if for all $pk \in \mathcal{PK}$, $C \in \mathcal{C}\backslash\mathcal{V}$, and all $K \in \mathcal{K}$,

$$\Pr[K = \Lambda_{sk}(C) | (pk, C)] \le 2^{-\gamma},$$

where the probability is taken over $sk \xleftarrow{\mathsf{U}} \mathcal{SK}$ with $pk = \mu(sk)$. We note that this Λ is originally called $2^{-\gamma}$-universal$_1$ in [13]. By definition, we note that $\mathsf{H}_\infty(\Lambda_{sk}(C)|(pk, C)) \ge \gamma$ for all $pk \in \mathcal{PK}$ and $C \in \mathcal{C}\backslash\mathcal{V}$.

Λ is called ϵ-smooth [13] if $\mathsf{Dist}((pk, C, \Lambda_{sk}(C)), (pk, C, K)) \le \epsilon$, where $sk \xleftarrow{\mathsf{U}} \mathcal{SK}$, $K \xleftarrow{\mathsf{U}} \mathcal{K}$ and $C \xleftarrow{\mathsf{U}} \mathcal{C}\backslash\mathcal{V}$ are chosen *at random* and $pk = \mu(sk)$.

A hash proof system $\mathsf{HPS} = (\mathsf{HPS.param}, \mathsf{HPS.pub}, \mathsf{HPS.priv})$ consists of three algorithms such that $\mathsf{HPS.param}$ takes 1^κ and outputs an instance of $\mathsf{params} = (\mathsf{group}, \Lambda, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \mu)$, where group contains some additional structural parameters and Λ is a projective hash function family associated with $(\mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \mu)$ as defined above. The deterministic public evaluation algorithm $\mathsf{HPS.pub}$ takes as input $pk = \mu(sk)$, $C \in \mathcal{V}$ and a witness w such that $C \in \mathcal{V}$ and returns $\Lambda_{sk}(C)$. The deterministic private evaluation algorithm takes $sk \in \mathcal{SK}$ and returns $\Lambda_{sk}(C)$, without taking withness w for C (if it exists). A hash proof system HPS as above is said to have a hard subset membership problem if two random elements $C \in \mathcal{C}$ and $C' \in \mathcal{C}\backslash\mathcal{V}$ are computationally indistinguishable, that is, $\{C \,|\, C \xleftarrow{\mathsf{U}} \mathcal{C}\}_{\kappa\in\mathbb{N}} \overset{\mathsf{c}}{\approx} \{C' \,|\, C' \xleftarrow{\mathsf{U}} \mathcal{C}\backslash\mathcal{V}\}_{\kappa\in\mathbb{N}}$.

2.4 All-But-One Injective Functions

We recall all-but-one injective functions (ABO) [32], which is a simple variant of all-but-one injective trap-door functions [30].

A collection of (n, ℓ_{lf})-all-but-one injective functions with branch collection $\mathcal{B} = \{B_\kappa\}_{\kappa\in\mathbb{N}}$ is given by a tuple of PPT algorithms $\mathsf{ABO} = (\mathsf{ABO.gen}, \mathsf{ABO.eval})$ with the following properties:

– $\mathsf{ABO.gen}$ is a PPT algorithm that takes 1^κ and any branch $b^* \in B_\kappa$, and outputs a function index i_{abo} and domain $\mathcal{X}$ with 2^n elements.
– $\mathsf{ABO.eval}$ is a DPT algorithm that takes i_{abo}, b, and $x \in \mathcal{X}$, and computes $y = \mathsf{ABO.eval}(i_{\mathsf{abo}}, b, x)$.

We require that (n, ℓ_{lf})-all-but-one injective functions given by ABO satisfies the following properties:

1. For any $b \neq b^* \in B_\kappa$, $\mathsf{ABO.eval}(i_{\mathsf{abo}}, b, \cdot)$ computes an injective function over the domain $\mathcal{X}$.
2. The number of elements in the image of $\mathsf{ABO.eval}(i_{\mathsf{abo}}, b^*, \cdot)$ over the domain $\mathcal{X}$ is at most $2^{\ell_{\mathsf{lf}}}$.
3. For any $b, b^* \in B_\kappa$, $\{\mathsf{ABO.gen}(1^\kappa, b)\}_{\kappa\in\mathbb{N}} \overset{\mathrm{c}}{\approx} \{\mathsf{ABO.gen}(1^\kappa, b^*)\}_{\kappa\in\mathbb{N}}$.

We note that ABO functions can be efficiently constructed under the DDH assumption and the DCR assumption (See Sec. C).

3 Continuous Tampering and Bounded Leakage Resilient CCA (CTBL-CCA) Secure Public-Key Encryption

A public-key encryption (PKE) scheme consists of the following four algorithms $\mathit{\Pi} = (\mathsf{Setup}, \mathbf{K}, \mathbf{E}, \mathbf{D})$: The setup algorithm Setup is a PPT algorithm that takes 1^κ and outputs public parameter ρ. The key-generation algorithm $\mathbf{K}$ is a PPT algorithm that takes ρ and outputs a pair of public and secret keys, (pk, sk). The encryption algorithm $\mathbf{E}$ is a PPT algorithm that takes public parameter ρ, public key pk and message $m \in \mathcal{M}$, and produces ciphertext $\mathsf{ct} \leftarrow \mathbf{E}_\rho(pk, m)$; Here $\mathcal{M}$ is uniquely determined by pk. The decryption algorithm $\mathbf{D}$ is a DPT algorithm that takes ρ, sk and presumable ciphertext ct, and returns message $m = \mathbf{D}_\rho(sk, \mathsf{ct})$. We require for correctness that for every sufficiently large $\kappa \in \mathbb{N}$, it always holds that $\mathbf{D}_\rho(sk, \mathbf{E}_\rho(pk, m)) = m$, for every $\rho \in \mathsf{Setup}(1^\kappa)$, every (pk, sk) generated by $\mathbf{K}(\rho)$, and every $m \in \mathcal{M}$.

We say that PKE $\mathit{\Pi}$ is **self-destructive** *if the decryption algorithm can erase all inner states including sk and does not work any more, when receiving an invalid ciphertext* ct *(i.e.,* $\mathbf{D}_\rho(sk, \mathsf{ct}) = \bot$*).* We assume that public parameter ρ is **system-wide**, i.e., *fixed beforehand and independent of users, and the only public and secret keys are subject to the tampering attacks.* This model is justified in the environment where the common public parameter could be hardwired into the algorithm codes and stored on tamper-proof hardware or distributed via a public channel where tampering is infeasible or could be easily detected.

CTBL-CCA Security. For PKE $\mathit{\Pi}$ and an adversary $A = (A_1, A_2)$, we define the experiment $\mathsf{Expt}^{\mathsf{ctbl\text{-}cca}}_{\mathit{\Pi},A,(\mathit{\Phi}_1,\mathit{\Phi}_2,\lambda)}(\kappa)$ as in Fig. 1. A may adaptively submit (unbounded) polynomially many queries (ϕ, ct) to oracle RKDec [1], but it should be in $\mathit{\Phi}_i$ appropriately. A may also adaptively submit (unbounded) polynomially many queries L to oracle Leak, before seeing the challenge ciphertext ct^*. The total amount of leakage on sk must be bounded by some λ bit length. We note that if $\mathit{\Pi}$ has the **self-destructive** property, RKDec does not answer any further query, or simply return $\bot$, after it receives an invalid ciphertext such that $\mathbf{D}_\rho(\phi(\mathsf{sk}), \mathsf{ct}) = \bot$. We define the advantage of A against $\mathit{\Pi}$ with respects $(\mathit{\Phi}_1, \mathit{\Phi}_2)$ as

$$\mathsf{Adv}^{\mathsf{ctbl\text{-}cca}}_{\mathit{\Pi},A,(\mathit{\Phi}_1,\mathit{\Phi}_2,\lambda)}(\kappa) \triangleq |\, 2\Pr[\mathsf{Expt}^{\mathsf{ctbl\text{-}cca}}_{\mathit{\Pi},A,(\mathit{\Phi}_1,\mathit{\Phi}_2,\lambda)}(\kappa) = 1] - 1\,|.$$

We say that $\mathit{\Pi}$ is $(\mathit{\Phi}_1, \mathit{\Phi}_2, \lambda)$-CTBL-CCA secure if $\mathsf{Adv}^{\mathsf{ctbl\text{-}cca}}_{\mathit{\Pi},A,(\mathit{\Phi}_1,\mathit{\Phi}_2,\lambda)}(\kappa) = \mathsf{negl}(\kappa)$ for every PPT A.

[1] A tampering function is called a related-key derivation (RKD) function in [6, 4].

$\mathsf{Expt}^{\mathsf{ctbl\text{-}cca}}_{\Pi,A,(\Phi_1,\Phi_2,\lambda)}(\kappa)$: $\rho \leftarrow \mathsf{Setup}(1^\kappa)$; $(\mathsf{pk},\mathsf{sk}) \leftarrow \mathbf{K}(\rho)$; $(m_0, m_1, st) \leftarrow A_1^{\mathsf{RKDec}_{\Phi_1}(\cdot,\cdot),\mathsf{Leak}_\lambda(\cdot)}(\rho,\mathsf{pk})$ such that $\lvert m_0\rvert = \lvert m_1\rvert$; $\beta^* \leftarrow \{0,1\}$; $\mathsf{ct}^* \leftarrow \mathbf{E}_\rho(\mathsf{pk}, m_{\beta^*})$; $\beta \leftarrow A_2^{\mathsf{RKDec}_{\Phi_2}(\cdot,\cdot)}(st,\mathsf{ct}^*)$; If $\beta = \beta^*$, then return 1; otherwise 0.	$\mathsf{RKDec}_\Phi(\phi,\mathsf{ct})$: If $\mathsf{ct} = \mathsf{ct}^*$ queried by A_2, then return $\perp$; If $\mathbf{D}_\rho(\phi(\mathsf{sk}),\mathsf{ct}) = \perp$, then erase sk. Return $\mathbf{D}_\rho(\phi(\mathsf{sk}),\mathsf{ct})$. $\mathsf{Leak}_\lambda(L_i)$: (L_i: i-th query of A.) If $\sum_{j=1}^{i} \lvert L_j(\mathsf{sk})\rvert > \lambda$ then return $\perp$; Else return $L_i(\mathsf{sk})$.

Fig. 1. The experiment of the CTBL-CCA game.

We say that Π is CTBL-CCA secure if it is $(\Phi_{\mathsf{all}}, \{\mathfrak{id}\}, \lambda)$-CTBL-CCA secure, where Φ_{all} is the class of all efficiently computable functions and $\mathfrak{id}$ denotes the identity function.

Remark 1. This security definition models **non-persistent** tampering. However, it is obvious that the persistent tampering version of CTBL-CCA security can be similarly defined.

We now state the following fact.

Theorem 1. *Suppose a PKE scheme Π without a key-updating mechanism is* CTBL-CCA *secure against the non-persistent tampering attacks. Then, Π is also* CTBL-CCA *secure against the persistent tampering attacks.*

Proof. For a PKE scheme without a key-updating mechanism, persistent tampering queries

$$(\phi_1, \mathsf{ct}_1), (\phi_2, \mathsf{ct}_2), \ldots, (\phi_\ell, \mathsf{ct}_\ell)$$

can be simulated non-persistent tampering queries as

$$(\phi_1, \mathsf{ct}_1), (\phi_2 \circ \phi_1, \mathsf{ct}_2), \ldots, (\phi_\ell \circ \cdots \circ \phi_1, \mathsf{ct}_\ell).$$

Leakage functions in the persistent tampering attack are also simulated as $L' = L \circ \phi_1 \cdots \circ \phi_{ell}$, where $\phi_1, \ldots, \phi_\ell$ denote all persistent tampering functions submitted before leakage function L is submitted. So, if Π is CTBL-CCA secure against the non-persistent tampering attacks, then it is CTBL-CCA secure against the persistent tampering attacks. ■

4 The CTBL-CCA Secure PKE Scheme

Let $\mathsf{HPS} = (\mathsf{HPS.param}, \mathsf{HPS.pub}, \mathsf{HPS.priv})$ be a hash proof system (described in Sec. 2.3). Let $\mathsf{ABO} = (\mathsf{ABO.gen}, \mathsf{ABO.eval})$ be a collection of all-but-one injective (ABO) functions (described in Sec. 2.4). Let TCH be a target collision resistant hash family. Let $\mathcal{H} = \{H \mid H : \{0,1\}^n \to \{0,1\}^{\ell_m}\}$ be a family of universal hash functions with $n = |\mathcal{K}|$. Let $\mathsf{OTSig} = (\mathsf{otKGen}, \mathsf{otSign}, \mathsf{otVrfy})$ a strong one-time signature scheme (Appendix A). We assume $\mathsf{vk} = 0 \notin \mathsf{otKGen}$.

At Asiacrypt 2013, Qin and Liu [31] proposed a new framework of bounded memory leakage resilient IND-CCA secure PKE schemes. Qin-Liu scheme is obtained by applying *a one-time lossy* *filter* to a natural hash-proof-system based PKE scheme, such as the encryption of m is constructed as $CT = (C, H, e)$ where $C \leftarrow \mathcal{V}$ with w, $H \leftarrow \mathcal{H}$, and $e = m \oplus H(\mathsf{HPS.pub}(PK, C, w))$, whereas the decryption is done by computing $m = e \oplus H(\mathsf{HPS.priv}(SK, C))$. Naor and Segev [29] proved that such PKE schemes are resilient to bounded leakage.

We describe a slight modification of Qin-Liu PKE scheme in Fig. 1. The difference is that (1) our construction divides the original key generation algorithm into the Setup algorithm and the key generation algorithm and puts ρ in the common reference string, and (2) replaces a one-time lossy filter with a combination of a strong one-time signature scheme and ABO injective function.

We have the following theorem.

Set-Up Algorithm $\mathsf{Setup}(1^\kappa)$:
$\mathsf{params} \leftarrow \mathsf{HPS.param}(1^\kappa)$
where $\mathsf{params} =$
$(\mathsf{group}, \Lambda, \mathcal{C}, \mathcal{V}, \mathcal{SK}, \mathcal{PK}, \mu)$.
$\mathsf{T} \leftarrow \mathsf{TCH}$ where $\mathsf{T} : \{0,1\}^* \to B_\kappa$.
Set $b^* = 0$ as the lossy branch.
$\iota_{\mathsf{abo}} \leftarrow \mathsf{ABO.gen}(1^\kappa, b^*)$.
$A(\cdot,\cdot) := \mathsf{ABO.eval}(\iota_{\mathsf{abo}}, \cdot, \cdot)$.
Return $\rho = (\mathsf{T}, \mathsf{params}, A(\cdot,\cdot))$.

Key Generation Algorithm $\mathbf{K}(\rho)$:
$sk \leftarrow \mathcal{SK}$.
Set $pk := \mu(sk)$.
Set $PK := pk$ and $SK := sk$.
Return (PK, SK)

Encryption Algorithm $\mathbf{E}_\rho(PK, m)$:
To encrypt a message $m \in \mathbb{G}$,
$C \overset{\mathsf{U}}{\leftarrow} \mathcal{V}$ with witness w.
$K = \mathsf{HPS.pub}(pk, C, w)$.
$(\mathsf{vk}, \mathsf{otsk}) \leftarrow \mathsf{otKGen}(1^\kappa)$
$\pi = A_{b^*}(\mathsf{T}(\mathsf{vk}), K)$. $H \leftarrow \mathcal{H}$.
$e = m \oplus H(K)$.
$\sigma \leftarrow \mathsf{otSign}(\mathsf{otsk}, (C, e, \mathsf{vk}, \pi))$.
Return $\mathsf{CT} = (C, e, H, \mathsf{vk}, \pi, \sigma)$.

Decryption Algorithm $\mathbf{D}_\rho(SK, \mathsf{CT})$:
To decrypt a ciphertext CT,
Parse CT into $(C, e, H, \mathsf{vk}, \pi, \sigma)$.
If $\mathsf{Vrfy}(\mathsf{vk}, (C, e, H, \mathsf{vk}, \pi), \sigma) \neq 1$,
then aborts.
Else $K = \Lambda_{sk}(C)$.
If $\pi \neq A_{b^*}(\mathsf{T}(\mathsf{vk})), K)$,
then aborts.
Else return $m = e \oplus H(K)$.

Fig. 2. The CTBL-CCA secure PKE scheme based on Qin and Liu's PKE

Theorem 2. *Let* HPS *be a* γ*-entropic hash proof system. Let* ABO *be* (n, ℓ_{lf})*-all-but-one injective function where* $n = \log|\mathcal{K}|$*. We assume the PKE scheme in Fig. 2 is self-destructive. Then, it is* $(\Phi_{\mathsf{all}}, \{\mathsf{i}\mathfrak{d}\}, \lambda)$*-CTBL-CCA secure, as long as* $\lambda(\kappa) \leq \gamma - \ell_{\mathsf{lf}} - \ell_m - \eta - \log(1/\epsilon)$ *where* $\eta(\kappa) = \omega(\log \kappa)$ *and* $\epsilon = 2^{-\omega(\log \kappa)}$*, and for any PPT adversary* A *with at most* Q *queries to* RKDec *oracle,* $\mathsf{Adv}^{\mathsf{ctbl\text{-}cca}}_{\Pi, A, (\Phi_{\mathsf{all}}, \{\mathsf{i}\mathfrak{d}\}, \lambda)}(\kappa) \leq$

$$2\epsilon_{\mathsf{tcr}} + 2\epsilon_{\mathsf{otsig}} + 4\epsilon_{\mathsf{lossy}} + 4\epsilon_{\mathsf{SD}} + 2^{-\eta+1} + Q \cdot 2^{-(\gamma - \eta - \lambda - \ell_{\mathsf{lf}} - \ell_m - 1)} + 2\epsilon,$$

where $\epsilon_{\mathsf{otsig}}$*,* $\epsilon_{\mathsf{lossy}}$*, and* ϵ_{SD} *denote some negligible functions such that* $\mathsf{Adv}^{\mathsf{ot}}_{\mathsf{OTSig}, B}(\kappa) \leq \epsilon_{\mathsf{otsig}}$*,* $\mathsf{Adv}^{\mathsf{lossy}}_{\mathsf{ABO}, B'}(\kappa) \leq \epsilon_{\mathsf{lossy}}$*, and* $\mathsf{Adv}^{\mathsf{SD}}_{\mathsf{HPS}, D}(\kappa) \leq \epsilon_{\mathsf{SD}}$ *for any PPT adversaries,* B*,* B' *and* D*, respectively.*

Proof Idea. Qin-Liu PKE scheme is leakage resilient. So, it is tempting to use the leakage oracle to simulate the RKDec oracle. However, the strategy is ineffective against *continual* tampering, because Qin-Liu PKE scheme is just *bounded* leakage resilient. We instead adopt the following proof strategy.

Let $\mathsf{CT}^* = (C^*, e^*, H^*, \mathsf{vk}^*, \pi^*, \sigma^*)$ be the challenge ciphertext and b^* be the challenge bit. Let $K^* = \Lambda_{SK}(C^*)$ and $e^* = m_{b^*} \oplus H^*(K^*)$.

In the early steps of the proof, we replace $C^* \in \mathcal{V}$ with $C^* \notin \mathcal{V}$ and set $\mathsf{T}(\mathsf{vk}^*)$ as a lossy branch, as expected. We then consider the pre-challenge tampering queries. Our interest in the pre-challenge stage is how to remain the entropy of K^* for $C^* \notin \mathcal{V}$, while answering leakage and tampering queries. When an adversary submits tampering query (ϕ, CT) and receives $\perp$, the revealing entropy of $\Lambda_{SK}(C^*)$ is just $\log(1/p)$-bit where $p = \Pr[\mathbf{D}(\phi(SK), \mathsf{CT}) = \perp]$. This comes from the following simple lemma.

Lemma 6. *For any random variables, X and Z, $\mathsf{H}_\infty(X|Z = z) \geq \mathsf{H}_\infty(X) - \log\Big(\frac{1}{\Pr[Z=z]}\Big)$.*

Proof. For any $z \in Z$,

$$\begin{aligned} -\log\Big(\max_x\Big(\Pr[X = x|Z = z]\Big)\Big) &= -\log\left(\max_x\left(\frac{\Pr[X = x \wedge Z = z]}{\Pr[Z = z]}\right)\right) \\ &\geq -\log\Big(\max_x\Big(\Pr[X = x]\Big)\Big) - \log\Big(\frac{1}{\Pr[Z = z]}\Big). \end{aligned} \tag{1}$$

■

By the lemma above, we have $\mathsf{H}_\infty(\Lambda_{SK}(C^*)|\mathbf{D}(\phi(SK), \mathsf{CT}) = \perp) \geq \mathsf{H}_\infty(\Lambda_{SK}(C^*)) - \log(1/p)$.

We then consider the case of $\mathbf{D}(\phi(SK), \mathsf{CT}) \neq \perp$ (still in the pre-challenge stage). This case is not equivalent to the former case, because RKDec oracle returns $\mathbf{D}(\phi(SK), \mathsf{CT})$, which would possibly reveal some additional information on $\Lambda_{SK}(C^*)$ except for $\mathbf{D}(\phi(SK), \mathsf{CT}) \neq \perp$. In a general case, one can use a "loose" bound such that $\widetilde{\mathsf{H}}_\infty(\Lambda_{SK}(C^*)|\mathbf{D}(\phi(SK), \mathsf{CT})) \geq \mathsf{H}_\infty(\Lambda_{SK}(C^*)) - \lambda$ where $\lambda = \log\Big(\mathbf{D}(\phi(SK), \mathsf{CT})\Big)$. As mentioned above, it is too loose in the continual tampering attack. We instead observe that (the slight modified) Qin-Liu PKE scheme satisfies the following equation.

$$\mathsf{H}_{\mathsf{sh}}\Big(\mathbf{D}(\phi(SK), \mathsf{CT}) \quad | \quad (\mathbf{D}(\phi(SK), \mathsf{CT}) \neq \perp), (\phi, \mathsf{CT}), PK\Big) = 0, \tag{2}$$

where $\mathsf{H}_{\mathsf{sh}}(X)$ denotes the Shannon entropy of random variable X (i.e., $\mathsf{H}_{\mathsf{sh}}(X) := \mathbf{E}_{x \leftarrow X}[\log \frac{1}{\Pr[X=x]}]$). This means that, given PK and (ϕ, CT), $\mathbf{D}(\phi(SK), \mathsf{CT})$ is fixed if $\mathbf{D}(\phi(SK), \mathsf{CT}) \neq \perp$ holds. Therefore, conditioned that CT is a valid ciphertext with respects to $\phi(SK)$,

$$\begin{aligned} \widetilde{\mathsf{H}}_\infty(\Lambda_{SK}(C^*)|\mathbf{D}(\phi(SK), \mathsf{CT})) &= \widetilde{\mathsf{H}}_\infty(\Lambda_{SK}(C^*)|\mathbf{D}(\phi(SK), \mathsf{CT}), (\mathbf{D}(\phi(SK), \mathsf{CT}) \neq \perp)) \\ &= \mathsf{H}_\infty(\Lambda_{SK}(C^*)|\mathbf{D}(\phi(SK), \mathsf{CT}) \neq \perp) \\ &\geq \mathsf{H}_\infty(\Lambda_{SK}(C^*)) - \log\Big(\frac{1}{\mathbf{D}(\phi(SK), \mathsf{CT}) \neq \perp}\Big). \end{aligned} \tag{3}$$

Therefore, in the pre-challenge stage of (the slight modified) Qin-Liu PKE scheme, letting $\Pr[\mathbf{D}(\phi(SK), \mathsf{CT}) \neq \perp] = p$,

– If RKDec rejects (ϕ, CT), the entropy loss of $\Lambda_{SK}(C^*)$ from the output of RKDec is $\log\Big(\frac{1}{1-p}\Big)$.

– If RKDec does not reject (ϕ, CT), the entropy loss on $\Lambda_{SK}(C^*)$ from the output of RKDec is $\log\left(\frac{1}{p}\right)$.

As one can see, the adversary can get nothing new about $\Lambda_{SK}(C^*)$ if $p = 1$. If $p = \mathsf{negl}(\kappa)$, it would only get little about $\Lambda_{SK}(C^*)$ (i.e.,$\log(1/1-p) \approx 0$) either, because the oracle rejects it with an overwhelming probability. Remember that the decryption algorithm *self-destructs* when it rejects a ciphertext. Hence, the adversary's best strategy is to submit a sequence of queries with $p = \mathsf{non\text{-}negl}$ and to hope that RKDec oracle accepts as long a prefix of the sequence as possible. The leakage amount of $\Lambda_{SK}(C^*)$ is bounded by $\eta = \omega(\log \kappa)$ bit.

We now consider the post-challenge (tampering) queries, $(\mathfrak{id}, \mathsf{CT})$, i.e., the normal decryption queries, where $\mathsf{CT} = (C, e, H, \mathsf{vk}, \pi, \sigma)$. In the post-challenge stage, our interest is how to prevent $H^*(\Lambda_{SK}(C^*))$ from revealing any partial information. Even one bit leakage would possibly break the system. To achieve the goal, we need to reject any invalid ciphertext during the decryption simulation. The "reject" probability relies on the entropy of $K = \Lambda_{SK}(C)$ for $C \notin \mathcal{V}$. Since the underlying hash proof system is γ-entropic, we can see that the remaining entropy of K is at least $\gamma - \lambda - \eta - \ell_{\mathsf{lf}} - \ell_m$. Here, λ and η are leakage amount via leakage and RKDec oracles in the pre-challenge stage, respectively. $2^{\ell_{\mathsf{lf}}}$ denotes the number of possible elements of π^*, where $A(\mathsf{T}(\mathsf{vk}^*), \cdot)$ is lossy. ℓ_m is the length of $H^*(\Lambda_{SK}(C^*))$. Then, the probability that the simulator *cannot* reject an invalid ciphertext is at most $2^{-(\gamma-\lambda-\eta-\ell_{\mathsf{lf}}-\ell_m)}$.

To summarize all the above, (1) just after the pre-challenge stage, the remaining entropy of $\Lambda_{SK}(C^*)$ for $C^* \notin \mathcal{V}$ is almost $\mathsf{H}_\infty(\Lambda_{SK}(C^*)) - \lambda - \eta$. By applying an appropriate universal hash H^*, we can obtain $H^*(\Lambda_{SK}(C^*))$ that is statistically close to a true uniform string. So, CT^* conceals message m_{b^*} in the statistical sense. (2) In the post-challenge stage, we can prevent $H^*(\Lambda_{SK}(C^*))$ from revealing any partial information with an overwhelming probability $1 - Q \cdot 2^{-(\gamma-\lambda-\eta-\ell_{\mathsf{lf}}-\ell_m)}$, where Q is the total number of decryption queries in the post-challenge stage. The proposal now satisfies the target security.

Proof of Theorem 2. Here we provide the formal proof of Theorem 2 by using the standard game-hopping strategy. We denote by S_i the event that adversary A wins in **Game** i.

– **Game 0**: This game is the original CTBL-CCA game, where $\mathsf{CT}^* = (C^*, e^*, H^*, \mathsf{vk}^*, \pi^*, \sigma^*)$ denotes the challenge ciphertext. By definition, $\Pr[S_0] = \Pr[\beta = \beta^*]$ and $\mathsf{Adv}^{\mathsf{tbl\text{-}cca}}_{\Pi,A,(\Phi_{\mathsf{all}},\{\mathfrak{id}\},\lambda)}(\kappa) = |2\Pr[S_0] - 1|$.
– **Game 1**: This game is identical to **Game 0**, except that when we produce the challenge ciphertext CT^*, *we instead computes* $K^* = \mathsf{HPS.priv}(sk, C^*)$ and do the same experiment at the other steps. The change is just conceptual and hence, it holds that $\Pr[S_0] = \Pr[S_1]$.
– **Game 2**: This game is identical to **Game 1**, except that we change the rule of the winning condition of A as follows: When adversary A submits (ϕ, CT) to oracle RKDec where $\mathsf{CT} = (C, e, H, \mathsf{vk}, \pi, \sigma)$, we aborts the game and regard A as a loser, if $\mathsf{T}(\mathsf{vk}) = \mathsf{T}(\mathsf{vk}^*)$ *but* CT *is still valid.* This happens only when $\mathsf{T}(\mathsf{vk}) = \mathsf{T}(\mathsf{vk}^*)$ with $\mathsf{vk} \neq \mathsf{vk}^*$ or the case that A breaks the strong one-time signature scheme. Hence, we have $\Pr[S_1] - \Pr[S_2] \leq \epsilon_{\mathsf{tcr}} + \epsilon_{\mathsf{otsig}}$ (by Lemma 9).
– **Game 3**: This game is identical to **Game 2**, except that we produce ρ and CT^* as follows: Before the step 3 in the set-up Setup, we run $(\mathsf{vk}^*, \mathsf{otsk}^*) \leftarrow \mathsf{otKGen}(1^\kappa)$ and set $b^* = \mathsf{T}(\mathsf{vk}^*)$. Then we do the same things in the subsequent steps. We produce the challenge ciphertext CT^* similarly in **Game 2** except that we instead use $(\mathsf{vk}^*, \mathsf{otsk}^*)$ generated in the set-up phase. The

difference between the probabilities of events, S_2 and S_3, are close because of indistinguishability between injective and lossy branches. Indeed, we have $\Pr[S_2] - \Pr[S_3] \le 2\epsilon_{\mathsf{lossy}}$ (by Lemma 11).

- **Game 4**: This game is identical to **Game 3**, except that when producing CT^*, we instead picks up $C^* \stackrel{\mathsf{U}}{\leftarrow} \mathcal{C}\backslash\mathcal{K}$. We then have $\Pr[S_3] - \Pr[S_4] \le 2\epsilon_{\mathsf{SD}}$ (by Lemma 11).
- **Game 5**: In this game, we change the rule of the game as described below. Let (ϕ, CT) be a tampering query of A and m be the reply of RKDec, where $\mathsf{CT} = (C, e, H, \mathsf{vk}, \pi, \sigma)$, and let view be A's view just before sending query (ϕ, CT). Here, we stress that when $\mathbf{D}(\phi(SK), \mathsf{CT}) \ne \perp$,

$$\mathsf{H}_{\mathsf{sh}}\Big(\Lambda_{\phi(SK)}(C) \,|\, (\phi, \mathsf{CT}), (\mathbf{D}(\phi(SK), \mathsf{CT}) \ne \perp), \mathsf{view}\Big) = 0$$

where $\mathsf{H}_{\mathsf{sh}}(X)$ denotes the Shannon entropy of random variable X, because $A(\mathsf{T}(\mathsf{vk}), \cdot)$ is a injective function and hence, given π (in CT), $\Lambda_{\phi(SK)}(C)$ is already fixed. This means that returning $m = \mathbf{D}(\phi(SK), C)$ does not reveal any additional information except $\mathbf{D}(\phi(SK), C) \ne \perp$ in the information theoretical sense. Here, we say that a sequence of tampering queries made by A is *η-challenging*, if there is a prefix of the sequence made by A such that oracle RKDec does not accept the prefix except with probability of $2^{-\eta}$. Let RDview be the random variable of the transcript between adversary A and oracle RKDec in the pre-challenge stage and let

$$\mathrm{rdv} = \{(\phi_1, \mathsf{CT}_1, m_1), \ldots, (\phi_{q'}, \mathsf{CT}_{q'}, m_{q'})\} \text{ where } q' \le Q.$$

be a transcript. If rdv is η-challenging, there is the minimum $q'' \le q'$ such that

$$\Pr[\mathrm{RDview} = \mathrm{rdv}] \le \Pr\Big[\wedge_{i=1}^{q"}\Big(\mathbf{D}(\phi_i(SK), \mathsf{CT}_i) \ne \perp\Big)\Big] \le 2^{-\eta}.$$

We now describe the rule of this game. This game is identical to the previous game except that RKDec "self-destructs" at the q''-th tampering query of η-challenging rdv, even if RKDec accepts the q''-th tampering query. (If it can reject an earlier tampering query, it self-destructs at the point.) This experiment is just conceptual and is not required to be executed in a polynomial time. By Lemma 9, $\Pr[S_4] - \Pr[S_5] \le 2^{-\eta}$, because the prefix is accepted at most $2^{-\eta}$.
- **Game 6**: In this game, for all post-challenge (decryption) query $(\mathfrak{id}, \mathsf{CT})$ of A, *we return $\perp$ if* $C \in \mathcal{C}\backslash\mathcal{V}$. This experiment is just conceptual and is not required to be executed in a polynomial time. We evaluate the min-entropy of $K = \Lambda_{SK}(C)$ derived from the post-challenge tampering query. Let Lview be the random variable of the transcript between adversary A and oracle Leak in the pre-challenge stage. When the first post-challenge decryption query is made, by the "chain rule" of the average-min entropy,

$$\widetilde{\mathsf{H}}_{\infty}(K|(\mathrm{RDview}, \mathrm{Lview}, \pi^*, H^*(K^*))) \ge \widetilde{\mathsf{H}}_{\infty}(K|\mathrm{RDview}) - \lambda - \ell_{\mathsf{lf}} - \ell_m,$$

where $2^{\ell_{\mathsf{lf}}}$ denotes the number of elements in the image of "lossy" function $\pi^* = A(\mathsf{T}(\mathsf{vk}^*), \cdot)$, and ℓ_m is the length of $H^*(K^*)$.
By lemma 6, we have

$$\mathsf{H}_{\infty}(K|\mathrm{RDview} = \mathrm{rdv}) \ge \mathsf{H}_{\infty}(K) - \log\Big(\frac{1}{\Pr[\mathrm{RDview} = \mathrm{rdv}]}\Big) \ge \mathsf{H}_{\infty}(K) - \eta.$$

The second inequality comes from $\Pr[\mathrm{RDview} = \mathrm{rdv}] \ge 2^{-\eta}$, because if rdv is η-challenging, the adversary cannot make a post-challenge decryption query. Therefore, for $C \in \mathcal{C}\backslash\mathcal{V}$,

$$\widetilde{\mathsf{H}}_{\infty}(K|\mathrm{RDview}) = -\log\Big(\mathop{\mathbf{E}}_{\mathrm{rdv}\leftarrow\mathrm{RDview}}[2^{-\mathsf{H}_{\infty}(K|\mathrm{RDview}=\mathrm{rdv})}]\Big) \ge \gamma - \eta,$$

because Λ is γ-entropic. Therefore,

$$\widetilde{\mathsf{H}}_\infty(K|(\mathrm{RDview}, \mathrm{Lview}, \pi^*, H(K^*))) \geq \gamma - \eta - \lambda - \ell_{\mathsf{lf}} - \ell_m.$$

Since $\mathsf{T}(\mathsf{vk}^*) \neq \mathsf{T}(\mathsf{vk})$,

$$\widetilde{\mathsf{H}}_\infty(\pi|(\mathrm{RDview}, \mathrm{Lview}, \pi^*, H(K^*))) = \widetilde{\mathsf{H}}_\infty(K|(\mathrm{RDview}, \mathrm{Lview}, \pi^*, H(K^*))),$$

where $\pi = \Lambda_{\mathsf{T}(\mathsf{vk}^*)}(\mathsf{T}(\mathsf{vk}), K)$ (injective). This means that RKDec accepts CT with $C \in \mathcal{C}\backslash\mathcal{V}$ only with probability $2^{-(\gamma-\eta-\lambda-\ell_{\mathsf{lf}}-\ell_m)}$. Assuming that A submits Q queries to RKDec in total, the probability that RKDec accepts at least one CT with $C \in \mathcal{C}\backslash\mathcal{V}$ is bounded by $Q \cdot 2^{-(\gamma-\eta-\lambda-\ell_{\mathsf{lf}}-\ell_m)}$. Hence, we have (by Lemma 9),

$$\Pr[S_5] - \Pr[S_6] \leq Q \cdot 2^{-(\gamma-\eta-\lambda-\ell_{\mathsf{lf}}-\ell_m)}.$$

– **Game 7:** This is the last game we make. This game is identical to **Game 6** except that we replace $H^*(K^*)$ with a uniformly random string from $\{0,1\}^{\ell_m}$. Then it is clear that $\Pr[S_7] = \frac{1}{2}$ because the view of A is independent of β^*. We now show that the advantages between **Game 6** and **Game 7** are statistically close. Since all post-challenge queries of "invalid" ciphertexts are rejected, the average min-entropy of K^* even after all post-challenge queries are made is equivalent to

$$\widetilde{\mathsf{H}}_\infty(K^*|(\mathrm{RDview}, \mathrm{Lview}, \pi^*)) \geq \widetilde{\mathsf{H}}_\infty(K^*|\mathrm{RDview}) - \lambda - \ell_{\mathsf{lf}} \geq \gamma - \eta - \lambda - \ell_{\mathsf{lf}}.$$

Remember that $\lambda \leq \gamma - \eta - \ell_{\mathsf{lf}} - \ell_m - \log(1/\epsilon)$ and H^* is independent of the view of the post-challenge decryption. By the generalized left-over hash lemma, $H^*(K^*)$ is ϵ-close to the uniform distribution on $\{0,1\}^{\ell_m}$. So, Lemma 10 gives us $\Pr[S_6] - \Pr[S_7] \leq \epsilon$.

By summing up the above inequalities, we have

$$\Pr[S_0] \leq \frac{1}{2} + \epsilon_{\mathsf{tcr}} + \epsilon_{\mathsf{otsig}} + 2\epsilon_{\mathsf{lossy}} + 2\epsilon_{\mathsf{SD}} + 2^{-\eta} + Q \cdot 2^{-(\gamma-\eta-\lambda-\ell_{\mathsf{lf}}-\ell_m)} + \epsilon,$$

and conclude the proof of the theorem, with $\mathsf{Adv}^{\mathsf{ctbl\text{-}cca}}_{\Pi,A,(\Phi_{\mathsf{all}},\{\mathsf{i}\mathfrak{d}\},\lambda)}(\kappa) = 2\Pr[S_0] - 1$. ■

An Instantiation of CTBL-CCA Secure PKE with $1 - o(1)$ Leakage Rate. We remark that even if we start with a hash proof system resilient to $1 - o(1)$ leakage rate, we cannot obtain a CTBL-CCA secure PKE scheme with $1 - o(1)$ leakage rate in general. To obtain an optimal leakage rate, we require $\frac{\gamma}{|SK|} = 1 - o(1)$ for a γ-entropic hash proof system. The cryptosystems of Boneh et al. [9] and Naor-Segev [29] do not satisfy the condition, although they are IND-CPA secure resilient to $1 - o(1)$ leakage rate.

Let $n = pq$ be a composite number of distinct odd primes, p and q, and $1 \leq d < p, q$ be a positive integer. It is known that $\mathbb{Z}^\times_{n^{d+1}} \cong \mathbb{Z}_{n^d} \times (\mathbb{Z}/n\mathbb{Z})^\times$ and any element in $\mathbb{Z}^\times_{n^{d+1}}$ is uniquely represented as $(1+n)^\delta \gamma^{n^d} \pmod{n^{d+1}}$ for some $\delta \in \mathbb{Z}_{n^d}$ and $\gamma \in (\mathbb{Z}/n\mathbb{Z})^\times$. For $\delta \in \mathbb{Z}_{n^d}$, we write $\mathbf{E}^{\mathsf{dj}}(\delta)$ to denote a subset in $\mathbb{Z}^\times_{n^{d+1}}$ such that $\mathbf{E}^{\mathsf{dj}}(\delta) = \{(1+n)^\delta \gamma^{n^d} \,|\, \gamma \in (\mathbb{Z}/n\mathbb{Z})^\times\}$. It is well known that for any two distinct $\delta, \delta' \in \mathbb{Z}_{n^d}$, it is computationally hard to distinguish a random element in $\mathbf{E}^{\mathsf{dj}}(\delta)$ from a random element in $\mathbf{E}^{\mathsf{dj}}(\delta')$ as long as the decision computational residue (DCR) assumption holds true. Let $\mathcal{C} = \mathbb{Z}^\times_{n^{d+1}}$ and $\mathcal{V} = \mathbf{E}^{\mathsf{dj}}(0)$. Let $\mathcal{SK} = \{0, 1, \ldots, n^{d+1}\} \subset \mathbb{Z}$.

Let $g \in \mathcal{V}$ and $\mathcal{PK} = \{\mu(sk) \,|\, \mu(sk) = g^{sk} \pmod{n^{d+1}}$ where $sk \in \mathcal{SK}\}$ $(= \mathbf{E}^{\mathsf{dj}}(0))$. For $C \in \mathcal{C}$, define $\Lambda_{sk}(C) = C^{sk} \pmod{n^{d+1}}$. Then, $\Lambda : \mathcal{SK} \times \mathcal{C} \to \mathcal{V}$ is projective and $d\log(n)$-entropic and a hash proof system HPS is constructed on Λ. In addition, $\frac{\text{leakge bound}}{\text{the length of secret-key}} = \frac{d\log(n) - \omega(\log(\kappa))}{(d+1)\log(n)} = 1 - o(1)$.

Corollary 1. *By applying the DCR-based hash proof system above and the DCR based instantiation of ABO injective function in Appendix C to the PKE scheme in Fig. 2, it becomes a* CTBL-CCA *secure PKE scheme with* $1 - o(1)$ *bounded memory leakage rate under the DCR assumption.*

5 Continuous Tampering and Leakage Resilient CCA (CTL-CCA) Secure Public-Key Encryption

We say that PKE has a **key-updating** mechanism *if there is a PPT algorithm* Update *that takes* ρ *and* sk *and returns an "updated" secret key* $sk' = \mathsf{Update}_\rho(sk)$. We assume that the key-updating mechanism Update can be activated only when the decryption algorithm rejects a ciphertext. Therefore, one cannot update his secret key unless the decryption algorithm has detected tampering. We require for $\Pi = (\mathsf{Setup}, \mathsf{Update}, \mathbf{K}, \mathbf{E}, \mathbf{D})$ that for every sufficiently large $\kappa \in \mathbb{N}$ and ever $I \in \mathbb{N}$, it always holds that $\mathbf{D}_\rho(sk_i, \mathbf{E}_\rho(pk, m)) = m$, for every $\rho \in \mathsf{Setup}(1^\kappa)$, every $(pk, sk_0) \in \mathbf{K}(\rho)$, and every $sk_i \in \mathsf{Update}_\rho(sk_{i-1})$ for $i \in [I]$, and every $m \in \mathcal{M}$.

CTL-CCA Security. For PKE with a key-updating mechanism $\Pi' = (\mathsf{Setup}, \mathsf{Update}, \mathbf{K}, \mathbf{E}, \mathbf{D})$ and an adversary $A = (A_1, A_2)$, we define the experiment $\mathsf{Expt}^{\mathsf{ctl\text{-}cca}}_{\Pi,A,(\Phi_1,\Phi_2,\lambda)}(\kappa)$ as in Fig. 3. A may adaptively submit (unbounded) polynomially many queries (ϕ, ct) to oracle RKDec, but it should be $\phi \in \Phi_i$ appropriately. We remark that secret key sk is updated using (non-tampered) flesh randomness only when the decryption algorithm rejects a ciphertext. A may also adaptively submit (unbounded) polynomially many queries L to oracle Leak, before seeing the challenge ciphertext ct^*. The total amount of leakage on sk must be bounded by some λ bit length within each one period between the key-updating mechanism are activated. We define the advantage of A against Π' with respects to (Φ_1, Φ_2) as

$$\mathsf{Adv}^{\mathsf{ctl\text{-}cca}}_{\Pi,A,(\Phi_1,\Phi_2,\lambda)}(\kappa) \triangleq |\, 2\Pr[\mathsf{Expt}^{\mathsf{ctl\text{-}cca}}_{\Pi,A,(\Phi_1,\Phi_2,\lambda)}(\kappa) = 1] - 1\,|.$$

We say that Π is $(\Phi_1, \Phi_2, \lambda)$-CTL-CCA secure if $\mathsf{Adv}^{\mathsf{ctl\text{-}cca}}_{\Pi,A,(\Phi_1,\Phi_2,\lambda)}(\kappa) = \mathsf{negl}(\kappa)$ for every PPT A.

We say that Π is simply CTL-CCA secure if it is $(\Phi_{\mathsf{all}}, \{\mathfrak{id}\}, \lambda)$-CTL-CCA secure, where Φ_{all} denotes the class of all efficiently computable functions and $\mathfrak{id}$ denotes the identity function.

Remark 2. This security definition models **non-persistent** tampering. However, it is obvious that the persistent tampering version of CTL-CCA security can be similarly defined.

6 The CTL-CCA Secure PKE Scheme

In this section, we present a CTL-CCA-secure PKE scheme. We first provide the intuition behind our construction.

Our starting point is a hash proof system based PKE scheme proposed by Agrawal et al. [2], that is IND-CPA secure resilient to continuous memory leakage in the so-called *Floppy model*, where a decryptor additionally owns secret $\boldsymbol{\alpha}$ to refresh its secret key sk using fresh randomness. The Floppy

$\mathsf{Expt}^{\text{ctl-cca}}_{\Pi,A,(\Phi_1,\Phi_2,\lambda)}(\kappa)$:
  $\rho \leftarrow \mathsf{Setup}(1^\kappa)$;
  $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathbf{K}(\rho)$;
  LEAKSUM $:= 0$;
  $(m_0, m_1, st) \leftarrow A_1^{\mathsf{RKDec}_{\Phi_1}(\cdot,\cdot),\mathsf{Leak}_\lambda(\cdot)}(\rho, \mathsf{pk})$
    such that $|m_0| = |m_1|$;
  $\beta^* \leftarrow \{0,1\}$;
  $\mathsf{ct}^* \leftarrow \mathbf{E}_\rho(\mathsf{pk}, m_{\beta^*})$;
  $\beta \leftarrow A_2^{\mathsf{RKDec}_{\Phi_2}(\cdot,\cdot)}(st, \mathsf{ct}^*)$;
  If $\beta = \beta^*$,
    then return 1; otherwise 0.

$\mathsf{RKDec}_\Phi(\phi, \mathsf{ct})$:
  If $\mathsf{ct} = \mathsf{ct}^*$ queried by A_2,
    then return $\perp$;
  Return $\mathbf{D}_\rho(\phi(\mathsf{sk}), \mathsf{ct})$.
  If $\mathbf{D}_\rho(\phi(\mathsf{sk}), \mathsf{ct}) = \perp$,
    Set $\mathsf{sk} \leftarrow \mathsf{Update}_\rho(\mathsf{sk})$,
    Set LEAKSUM $:= 0$.
  Else do nothing.

$\mathsf{Leak}_\lambda(L)$:
  If LEAKSUM
    $:=$ LEAKSUM $+ |L(\mathsf{sk})| > \lambda$.
    then return $\perp$;
  Else return $L(\mathsf{sk})$.

Fig. 3. The experiment of the CTL-CCA game.

model assumes secret $\boldsymbol{\alpha}$ is not leaked. The Agrawal et al. scheme is as follows: $pk = (g, g^{\boldsymbol{\alpha}}, f)$ is a public key and $sk = \boldsymbol{s}$ is the corresponding secret-key such that $f = g^{<\boldsymbol{\alpha},\boldsymbol{s}>}$, where g is a generator of cyclic group G of prime order q, $\boldsymbol{\alpha}, \boldsymbol{s} \in (\mathbb{Z}/q\mathbb{Z})^n$. In addition, the decryptor owns $\boldsymbol{\alpha}$ as the key-update key. The encryption of message $m \in G$ under pk is $\mathsf{ct} = (g^{\boldsymbol{c}}, e) = (g^{r\boldsymbol{\alpha}}, m \cdot f^r)$, while the decryption is computed as $e \cdot (g^{<\boldsymbol{c},sk>})^{-1}$. The secret key sk is refreshed between each two time periods as $sk := sk + \boldsymbol{\beta}$ where $\boldsymbol{\beta} \leftarrow \ker(\boldsymbol{\alpha})$ is chosen using secret α. Here, $f = g^{<\boldsymbol{\alpha},\boldsymbol{s}>} = g^{<\boldsymbol{\alpha},\boldsymbol{s}+\boldsymbol{\beta}>}$, because $<\boldsymbol{\alpha}, \boldsymbol{\beta}>= 0$.

We first convert this scheme to an IND-CPA secure PKE scheme that is resilient to continuous memory leakage in the model of [10], where the key-update is done without additional secret $\boldsymbol{\alpha}$. To do so, we pick up ℓ independent vectors, $\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell} \in \ker(\boldsymbol{\alpha})$, where $\ell < n-1 = \dim(\ker(\boldsymbol{\alpha}))$, and publish $\tilde{g}^{\mathbf{V}}$ where $\mathbf{V} = (\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}) \in (\mathbb{Z}/q\mathbb{Z})^{n\times\ell}$ is $n \times \ell$ matrix with $\boldsymbol{v_i}$ as i-th column. Here we assume asymmetric pairing groups $(e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ where $g, \tilde{g}$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. We then set $pk = (g, \tilde{g}, g^{\boldsymbol{\alpha}}, \tilde{g}^{\mathbf{V}}, Y)$ and $sk = g^{\boldsymbol{s}}$ such that $Y = e(g, \tilde{g})^{<\boldsymbol{\alpha},\boldsymbol{s}>}$. Here, the encryption of message $m \in \mathbb{G}_T$ under pk is $\mathsf{ct} = (g^{\boldsymbol{c}}, e) = (g^{r\boldsymbol{\alpha}}, m \cdot Y^r)$, while the decryption is computed as $e \cdot K^{-1}$, where $K = e(g^{\boldsymbol{c}}, sk) = e(g, \tilde{g})^{<\boldsymbol{c},\boldsymbol{s}>}$. The secret key sk is refreshed between each two time periods as $sk := sk \cdot \tilde{g}^{\boldsymbol{\beta}}$ where $\boldsymbol{\beta} \leftarrow \mathsf{span}(\mathbf{V}) \subset \ker(\boldsymbol{\alpha})$. We note that random $\tilde{g}^{\boldsymbol{\beta}} = \tilde{g}^{\mathbf{V}\boldsymbol{r'}}$ can be computed using public $\tilde{g}^{\mathbf{V}}$ with random vector $\boldsymbol{r'} \in \mathbb{F}_q^\ell$. This construction is an IND-CPA secure PKE scheme resilient to continuous memory leakage in the sense of [10] under the extended matrix d-linear assumption (on $\mathbb{G}_1$), which is implied by the SXDH assumption. We provide the proof in Appendix E.

This continuous memory leakage CPA secure PKE scheme is hash proof system based where $K = \mathsf{HPS.pub}(Y, g^{r\boldsymbol{\alpha}}, r) = \mathsf{HPS.priv}(g^{r\boldsymbol{\alpha}}, sk) = e(g, \tilde{g})^{<\boldsymbol{\alpha},\boldsymbol{s}>}$. Then, we then filter the hash key K using the one-time lossy filter [31] and finally obtain our construction.

We now describe our full-fledged scheme in Fig. 4.

Asymmetric Pairing. Let GroupG be a PPT algorithm that on input a security parameter 1^κ outputs a bilinear paring $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g, \tilde{g})$ such that; $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic groups of prime order q, $g, \tilde{g}$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively, and a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ satisfies the following properties:

– (Bilinear:) for any $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$, and any $a, b \in \mathbb{Z}_q$, $e(g^a, h^b) = e(g, h)^{ab}$,

– (Non-degenerate:) $e(g, \tilde{g})$ has order q in $\mathbb{G}_T$, and
– (Efficiently computable:) $e(\cdot, \cdot)$ is efficiently computable.

Symmetric External Diffie-Hellman (SXDH) Assumption. The symmetric external DH assumption (SXDH) (on GroupG) is that the DDH problem is hard in both groups, $\mathbb{G}_1$ and $\mathbb{G}_2$. The assumption implies that there is no efficiently computable mapping between $\mathbb{G}_1$ and $\mathbb{G}_2$.

We now present our CTL-CCA secure PKE scheme in Fig. 4.

Set-Up Algorithm $\mathsf{Setup}(1^\kappa)$:
$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g, \tilde{g}) \leftarrow \mathsf{GroupG}$. $\quad \boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \leftarrow (\mathbb{Z}/q\mathbb{Z})^n$.
$\mathbf{V} = (\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}) \leftarrow \Big(\mathsf{Ker}(\boldsymbol{\alpha})\Big)^\ell$, where $\mathbf{V} \in \Big(\mathbb{Z}/q\mathbb{Z}\Big)^{n\times\ell}$ and $\ell \le n-2$.
$g^{\boldsymbol{\alpha}} := (g_1, \ldots, g_n) = (g^{\alpha_1}, \ldots, g^{\alpha_n})$. $\quad \tilde{g}^{\mathbf{V}} := (\tilde{g}^{\boldsymbol{v_1}}, \ldots, \tilde{g}^{\boldsymbol{v_\ell}})$ where $v_i \in (\mathbb{Z}/q\mathbb{Z})^n$ is a column vector.
$\mathsf{T} \leftarrow \mathsf{TCH}$ where $\mathsf{T} : \{0,1\}^* \to B_\kappa$. $\quad$ Set $b^* = 0$ as the lossy branch.
$\iota_{\mathsf{abo}} \leftarrow \mathsf{ABO.gen}(1^\kappa, b^*)$. $\quad A(\cdot,\cdot) := \mathsf{ABO.eval}(\iota_{\mathsf{abo}}, \cdot, \cdot)$.
Return $\rho = (g, \tilde{g}, g^{\boldsymbol{\alpha}}, \tilde{g}^{\mathbf{V}}, \mathsf{T}, A(\cdot,\cdot))$.

Key Generation Algorithm $\mathbf{K}(\rho)$:
$\boldsymbol{s} = (s_1, \ldots, s_n) \leftarrow \Big(\mathbb{Z}/q\mathbb{Z}\Big)^n$.
$\tilde{g}^{\boldsymbol{s}} = (\tilde{g}^{s_1}, \ldots, \tilde{g}^{s_n})$.
$Y = e(g^{\boldsymbol{\alpha}}, \tilde{g}^{\boldsymbol{s}}) = e(g, \tilde{g})^{\langle \boldsymbol{\alpha}, \boldsymbol{s}\rangle}$.
Set $pk := Y$ and $sk := \tilde{g}^{\boldsymbol{s}}$.
Return (pk, sk).

Key Updating Algorithm $\mathsf{Update}(\rho, sk)$:
$\boldsymbol{r'} \leftarrow \Big(\mathbb{Z}/q\mathbb{Z}\Big)^\ell$,
Let $sk = \tilde{g}^{\boldsymbol{s}}$. (See $\tilde{g}^{\boldsymbol{s}}$ as a column vector.)
Set $sk := sk \cdot \tilde{g}^{\mathbf{V}\boldsymbol{r'}} = \tilde{g}^{\boldsymbol{s}+\mathbf{V}\boldsymbol{r'}}$.
$\Big($where $\boldsymbol{\beta} := \mathbf{V}\boldsymbol{r'} \in \mathsf{span}(\mathbf{V})$.$\Big)$
Return sk.

Encryption Algorithm $\mathbf{E}_\rho(pk, m)$:
To encrypt a message $m \in \mathbb{G}_T$,
$r \leftarrow \mathbb{Z}/q\mathbb{Z}$. $\quad K = Y^r$.
$(\mathsf{vk}, \mathsf{otsk}) \leftarrow \mathsf{otKGen}(1^\kappa)$.
$\pi = A(\mathsf{T}(\mathsf{vk}), K)$.
$\boldsymbol{C} = (g^{\boldsymbol{\alpha}})^r$. $\quad e = m \cdot K$.
$\sigma \leftarrow \mathsf{otSign}(\mathsf{otsk}, \boldsymbol{C}, e, \mathsf{vk}, \pi))$.
Return $\mathsf{CT} = (\boldsymbol{C}, e, \mathsf{vk}, \pi, \sigma)$.

Decryption Algorithm $\mathbf{D}_\rho(sk, \mathsf{CT})$:
To decrypt a ciphertext ct,
Parse ct into $(g^{\mathbf{c}}, e, \mathsf{vk}, \pi, \sigma)$.
If $\mathsf{Vrfy}(\mathsf{vk}, (g^{\mathbf{c}}, e, \mathsf{vk}, \pi), \sigma) \neq 1$,
then aborts.
Else $K = e(g^{\mathbf{c}}, sk) = e(g, \tilde{g})^{r\langle \boldsymbol{\alpha}, \boldsymbol{s}\rangle}$.
If $\pi \neq A(\mathsf{T}(\mathsf{vk})), K)$,
then aborts.
Else return $m = e \cdot K^{-1}$.

Fig. 4. Our CTL-CCA secure PKE Scheme

Theorem 3. *The PKE scheme in Fig. 4 is* $(\Phi_{\mathsf{all}}, \{\mathfrak{id}\}, \lambda)$-CTL-CCA *secure, as long as* $\lambda(\kappa) < \log(q) - \ell_{\mathsf{lf}} - \ell_m - \eta - \omega(\log\kappa)$ *with* $\eta(\kappa) = \omega(\log\kappa)$*, and for any PPT adversary* A *with at most* Q *queries to* RKDec *oracle,* $\mathsf{Adv}^{\mathsf{ctl\text{-}cca}}_{\Pi, A, (\Phi_{\mathsf{all}}, \{\mathfrak{id}\}, \lambda)}(\kappa) \le$

$$2\epsilon_{\mathsf{tcr}} + 2\epsilon_{\mathsf{otsig}} + 4\epsilon_{\mathsf{lossy}} + 4\epsilon_{ex} + 2^{-\eta+1} + Q \cdot 2^{-(\log(q)-\eta-\lambda-\ell_{\mathsf{lf}}-\ell_m-1)} + nQ \cdot \sqrt{\frac{2^\lambda}{q^{\ell-1}}} + \sqrt{\frac{2^{\lambda+\ell_{\mathsf{lf}}}}{q^{n-2}}},$$

$\epsilon_{\mathsf{otsig}}$, $\epsilon_{\mathsf{lossy}}$, *and* ϵ_{ex} *denote some negligible functions such that* $\mathsf{Adv}^{\mathsf{ot}}_{\mathsf{OTSig}, B}(\kappa) \le \epsilon_{\mathsf{otsig}}$, $\mathsf{Adv}^{\mathsf{lossy}}_{\mathsf{ABO}, B'}(\kappa) \le \epsilon_{\mathsf{lossy}}$, *and* $\mathsf{Adv}^{ex}_D(\kappa) \le \epsilon_{ex}$ *for any PPT adversaries,* B, B' *and* D*, respectively.*

Proof of Theorem 3. Here we prove the theorem by using the standard game-hopping strategy. We denote by S_i the event that adversary A wins in **Game** i.

- **Game 0**: This game is the original CTL-CCA game, where $\mathsf{CT}^* = (\boldsymbol{C}^*, E^*, \mathsf{vk}^*, \pi^*, \sigma^*)$ denotes the challenge ciphertext. By definition, $\Pr[S_0] = \Pr[b = b^*]$ and $\mathsf{Adv}^{\mathsf{ctl\text{-}cca}}_{\Pi,A,(\Phi_{\mathsf{all}},\{\mathfrak{id}\},\lambda)}(\kappa) = |2\Pr[S_0] - 1|$.
- **Game 1**: In this game, we compute CT^* using sk as follows: Compute $K^* = e(g^{\boldsymbol{c}^*}, sk) = e(g, \tilde{g})^{r\langle \boldsymbol{\alpha}, \boldsymbol{s}\rangle}$ and set $e^* = m_{b^*} \cdot K^*$. This change is just conceptual. Then, $\Pr[S_0] = \Pr[S_1]$.
- **Game 2**: This game is identical to **Game 1**, except that we change the rule of the winning condition of A as follows: When adversary A submits (ϕ, CT) to oracle RKDec where $\mathsf{CT} = (C, e, \mathsf{vk}, \pi, \sigma)$, we aborts the game and regard A as being lost, if $\mathsf{T}(\mathsf{vk}) = \mathsf{T}(\mathsf{vk}^*)$ *but* CT *is still valid.* This happens only when $\mathsf{T}(\mathsf{vk}) = \mathsf{T}(\mathsf{vk}^*)$ with $\mathsf{vk} \neq \mathsf{vk}^*$ or the case that A breaks the strong one-time signature scheme. Hence, we have $\Pr[S_1] - \Pr[S_2] \leq \epsilon_{\mathsf{tcr}} + \epsilon_{\mathsf{otsig}}$ (by Lemma 9).
- **Game 3**: This game is identical to **Game 2**, except that we produce ρ and CT^* as follows: In the set-up Setup, we run $(\mathsf{vk}^*, \mathsf{otsk}^*) \leftarrow \mathsf{otKGen}(1^\kappa)$ and set the lossy branch as $b^* = \mathsf{T}(\mathsf{vk}^*)$. We produce the challenge ciphertext CT^* similarly in **Game 2** except that we instead use $(\mathsf{vk}^*, \mathsf{otsk}^*)$ generated in the set-up phase. The difference between the probabilities of events, S_2 and S_3, are close because of indistinguishability between injective and lossy branches. Then, we have $\Pr[S_2] - \Pr[S_3] \leq 2\epsilon_{\mathsf{lossy}}$.
- **Game 4**: This game is identical to the previous game, except that we choose ℓ independent vectors $\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell} \leftarrow \ker(\boldsymbol{\alpha}, \boldsymbol{c^*})$ and set $\mathbf{V} = (\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell})$. Since $\boldsymbol{c^*} = r^*\boldsymbol{\alpha}$, $\ker(\boldsymbol{\alpha}, \boldsymbol{c^*}) = \ker(\boldsymbol{\alpha})$. Hence, $\Pr[S_3] = \Pr[S_4]$.
- **Game 5**: This game identical to the previous game, except that when producing CT^*, we instead pick up random vector $\boldsymbol{c^*} \leftarrow \mathbb{F}_q^n$. We note that since $\dim(\ker(\boldsymbol{\alpha}, \boldsymbol{c^*})) = n - 2 \geq \ell$, we can still choose ℓ independent vectors $\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}$. The difference between these two games is bounded by the extended matrix d-linear assumption. Then, we have $\Pr[S_4] - \Pr[S_5] \leq 2\epsilon_{\mathsf{ex}}$. We omit the detail, due to the similarity of the proof of Lemma 12 in Appendix E.
- **Game 6**: This game is identical to the previous game, except that in the key-update mechanism, we instead choose $\boldsymbol{\beta} \leftarrow \ker(\boldsymbol{\alpha})$ and update $sk := sk \cdot \tilde{g}^{\boldsymbol{\beta}}$. By the same analysis at **Game 4** in the proof of Appendix E, we have $\Pr[S_5] - \Pr[S_6] \leq \frac{nQ}{2}\sqrt{\frac{2^\lambda}{q^{\ell-1}}}$. We note that the total number of key-updates is, by definition, less than the total number of tampering queries, i.e., $Q' \leq Q$.
- **Game 7**: In this game, we change the rule of the key-updating mechanism. Suppose A submits η-challenging tampering queries. Remember that we say that a sequence of tampering queries is *η-challenging*, if there is a prefix of the sequence of tampering queries made by A such that RKDec does not accepts the prefix except a probability less than $2^{-\eta}$. In this game, we update the secret key when A submits the last query of the minimum prefix of the sequence such that the acceptance probability is less than $2^{-\eta}$. By the same analysis at **Game 5** in the proof of Theorem 2, we have $\Pr[S_6] - \Pr[S_7] \leq 2^{-\eta}$.
- **Game 8**: In this game, for all post-challenge (decryption) query $(\mathfrak{id}, \mathsf{CT})$ of A, *we return* $\perp$ *if* $C \in \mathcal{C}\backslash\mathcal{V}$. By the same analysis at **Game 6** in the proof of Theorem 2, we have $\Pr[S_7] - \Pr[S_8] \leq Q \cdot 2^{-(\log(q) - \eta - \lambda - \ell_{\mathsf{lf}} - \ell_m)}$, where $\gamma = \log(q)$ because the underlying hash proof system is $\log(q)$-entropic.
- **Game 9**: This game is identical to the previous game, except that we pick up random $k^* \leftarrow \mathbb{Z}/q\mathbb{Z}$ and compute $K^* = e(g, \tilde{g})^{k^*}$. This k^* is statistically close to $< \boldsymbol{c^*}, \boldsymbol{s} + \boldsymbol{\beta} >$. The analysis is similar to that at **Game 5** in the proof of Appendix E, except that CT^* additionally reveals

$\pi^* = A(\mathsf{T}(\mathsf{vk}^*), K^*)$. Therefore, we have $\Pr[S_8] - \Pr[S_9] \leq \frac{1}{2}\sqrt{\frac{2^{\lambda+\ell_{\mathsf{lf}}}}{q^{n-2}}}$. By construction, $\Pr[S_9] = \frac{1}{2}$.

To summarize all the above, we have the theorem statement. ■

An Instantiation of CTL-CCA Secure PKE with $\frac{1}{4} - o(1)$ Leakage Rate. We remark that the underlying hash proof system is $\log(q)$-entropic and we have $|sk| = n\log(q)$. By construction, we require $2 \leq \ell < n-1$. Hence, the best parameter for leakage rate is $n = 4$ and $\ell = 2$, where the resulting CTL-CCA secure PKE scheme has $\frac{1}{4} - o(1)$ leakage rate.

7 Impossibility of Non-Persistent Tampering Resilient Signatures

We show that there is no secure digital signature scheme resilient to the non-persistent tampering attacks, if it does not have a key-updating mechanism (See for definition Appendix F). This fact does not contradict [26] (in which they claim a tampering resilient digital signature scheme), because the persistent tampering attack is weaker than the non-persistent attack. To prove our claim, we consider the following adversary. The adversary runs the key-generation algorithm, Gen, and obtains two legitimate pairs of verification and signing keys, (vk_0, sk_0) and (vk_1, sk_1). Then, it sets a set of functions $\{\phi^i_{(sk_0,sk_1)}\}$, such that

$$\phi^i_{(sk_0,sk_1)}(sk) = \begin{cases} sk_0 & \text{if the } i\text{-th bit of } sk \text{ is } 0, \\ sk_1 & \text{otherwise.} \end{cases}$$

For $i = 1, \ldots, |sk|$, the adversary submit $(\phi^i_{(sk_0,sk_1)}, m)$ to the signing oracle and receives σ_i's. Then the adversary finds bit b_i such that $\mathsf{Vrfy}(vk_{b_i}, m, \sigma_i) = 1$ for all i and retrieves the entire secret key sk. This attack is unavoidable because both sk_0 and sk_1 are real secret keys and the signing algorithm cannot detect the tampering attack and cannot self-destruct.

If the key-updating algorithm is allowed to run only when a tampering is detected (which is the case of our definition), then there is no secure digital signature scheme resilient to the non-persistent tampering attacks, even if it has both self-destructive and key-updating mechanisms (See for definition Appendix F).

References

1. *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010* (2010), IEEE Computer Society.
2. Agrawal, S., Dodis, Y., Vaikuntanathan, V., and Wichs, D. On continual leakage of discrete log representations. In Sako and Sarkar [34], pp. 401–420.
3. Anonymous. A note on the RKA security of continuously non-malleable key-derivation function from PKC 2015. Submitted to PKC 2016.
4. Bellare, M., and Cash, D. Pseudorandom functions and permutations provably secure against related-key attacks. In *CRYPTO 2010* (2010), T. Rabin, Ed., vol. 6223 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 666–684.
5. Bellare, M., Cash, D., and Miller, R. Cryptography secure against related-key attacks and tampering. In *ASIACRYPT 2011* (2011), D. H. Lee and X. Wang, Eds., vol. 7073 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 486–503. The full version is available at http://eprint.iacr.org/2011/252.

6. Bellare, M., and Kohno, T. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In *EUROCRYPT 2003* (2003), E. Biham, Ed., vol. 2656 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 491–506.
7. Bellare, M., Paterson, K. G., and Thomason, S. RKA security beyond the linear barrier: IBE, encryption and signatures. In *ASIACRYPT 2012* (2012), X. Wang and K. Sako, Eds., vol. 7658 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 331–348. See also http://eprint.iacr.org/2012/514.
8. Boneh, D., Boyen, X., and Shacham, H. Short group signatures. In *CRYPTO 2004* (2004), M. K. Franklin, Ed., vol. 3152 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 41–55.
9. Boneh, D., Halevi, S., Hamburg, M., and Ostrovsky, R. Circular-secure encryption from decision diffie-hellman. In *CRYPTO 2008* (2008), D. Wagner, Ed., vol. 5157 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 108–125.
10. Brakerski, Z., Kalai, Y. T., Katz, J., and Vaikuntanathan, V. Overcoming the hole in the bucket: Public-key cryptography resilient to continual memory leakage. In *FOCS 2010* [1], pp. 501–510.
11. Cramer, R., Dodis, Y., Fehr, S., Padró, C., and Wichs, D. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In *EUROCRYPT 2008* (2008), N. P. Smart, Ed., vol. 4965 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 471–488. See also http://eprint.iacr.org/2008/030.
12. Cramer, R., Padró, C., and Xing, C. Optimal algebraic manipulation detection codes in the constant-error model. In Dodis and Nielsen [16], pp. 481–501. See also http://eprint.iacr.org/2014/116.
13. Cramer, R., and Shoup, V. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In *EUROCRYPT 2002* (2002), L. R. Knudsen, Ed., vol. 2332 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 45–64.
14. Damgård, I., Faust, S., Mukherjee, P., and Venturi, D. Bounded tamper resilience: How to go beyond the algebraic barrier. In Sako and Sarkar [34], pp. 140–160. See also http://eprint.iacr.org/2013/677 and http://eprint.iacr.org/2013/124.
15. Dodis, Y., Haralambiev, K., López-Alt, A., and Wichs, D. Cryptography against continuous memory attacks. In *FOCS 2010* [1], pp. 511–520. The full version is available at http://eprint.iacr.org/2010/196.
16. Dodis, Y., and Nielsen, J. B., Eds. *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I* (2015), vol. 9014 of *Lecture Notes in Computer Science*, Springer, Heidelberg.
17. Dodis, Y., Ostrovsky, R., Reyzin, L., and Smith, A. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing 38*, 1 (2008), 97–139. Preliminary version in *EUROCRYPT 2004*, 2004.
18. Dziembowski, S., Pietrzak, K., and Wichs, D. Non-malleable codes. In *ICS 2010* (Beijing, China, 2010), A. C.-C. Yao, Ed., Tsinghua University Press, pp. 434–452. The full version is available at http://eprint.iacr.org/2009/608.
19. Faust, S., Mukherjee, P., Nielsen, J. B., and Venturi, D. Continuous non-malleable codes. In *TCC 2014* (2014), Y. Lindell, Ed., vol. 8349 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 465–488.
20. Fujisaki, E., and Xagawa, K. Efficient RKA-secure KEM and IBE schemes against invertible functions. In *Latincrypt 2015* (2015), pp. ??–?? To appear in Latincrypt 2015.
21. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., and Rabin, T. Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In *TCC 2004* (2004), M. Naor, Ed., vol. 2951 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 258–277.
22. Goldreich, O. *Foundations of Cryptography: Volume II – Basic Applications*. Cambridge University Press, 2004.
23. Goldwasser, S., Micali, S., and Rivest, R. L. A digital signature scheme secure against adaptive chosen-message attacks. *SIMA Journal of Computing 17*, 2 (1988), 281–308. Preliminary vesion in *FOCS '84*, 1984.
24. Hofheinz, D., and Kiltz, E. Secure hybrid encryption from weakened key encapsulation. In *CRYPTO 2007* (2007), A. Menezes, Ed., vol. 4622 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 553–571.
25. Jafargholi, Z., and Wichs, D. Tamper detection and continuous non-malleable codes. In Dodis and Nielsen [16], pp. 451–480. See also http://eprint.iacr.org/2014/956.
26. Kalai, Y. T., Kanukurthi, B., and Sahai, A. Cryptography with tamperable and leaky memory. In *CRYPTO 2011* (2011), P. Rogaway, Ed., vol. 6841 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 373–390.
27. Kiltz, E., Pietrzak, K., Stam, M., and Yung, M. A new randomness extraction paradigm for hybrid encryption. In *EUROCRYPT 2009* (2009), A. Joux, Ed., vol. 5479 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 590–609.

28. Liu, F.-H., and Lysyanskaya, A. Tamper and leakage resilience in the split-state model. In *CRYPTO 2012* (2012), R. Safavi-Naini and R. Canetti, Eds., vol. 7417 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 517–532.
29. Naor, M., and Segev, G. Public-key cryptosystems resilient to key leakage. In *CRYPTO 2009* (2009), S. Halevi, Ed., vol. 5677 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 18–35.
30. Peikert, C., and Waters, B. Lossy trapdoor functions and their applications. In *STOC 2008* (2008), R. E. Ladner and C. Dwork, Eds., ACM, pp. 187–196.
31. Qin, B., and Liu, S. Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. In Sako and Sarkar [34], pp. 381–400.
32. Qin, B., and Liu, S. Leakage-flexible cca-secure public-key encryption: Simple construction and free of pairing. In *PKC 2014* (2014), H. Krawczyk, Ed., vol. 8383 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 19–36.
33. Qin, B., Liu, S., Yuen, T. H., Deng, R. H., and Chen, K. Continuous non-malleable key derivation and its application to related-key security. In *PKC 2015* (2015), J. Katz, Ed., vol. 9020 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 557–578.
34. Sako, K., and Sarkar, P., Eds. *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II* (2013), vol. 8270 of *Lecture Notes in Computer Science*, Springer, Heidelberg.
35. Wee, H. Public key encryption against related key attacks. In *Public Key Cryptography* (2012), M. Fischlin, J. Buchmann, and M. Manulis, Eds., vol. 7293 of *Lecture Notes in Computer Science*, Springer, Heidelberg, pp. 262–279.

A Definitions

A.1 Collision-Resistant Hash Function Families

Let $\mathcal{H} = \{H_\iota\}_{\iota \in \mathcal{I}}$ be a keyed hash family of functions from $\{0,1\}^*$ to $\{0,1\}^{p(\kappa)}$, indexed by $\iota \in \mathcal{I}_\kappa$ $(= \mathcal{I} \cap \{0,1\}^\kappa)$, i.e., $H_\iota : \{0,1\}^* \to \{0,1\}^{p(\kappa)}$ for every $\iota \in \mathcal{I}_\kappa$.

A keyed hash-function family $\mathcal{H}$ is called collision-resistant (CR) if, for every non-uniform PPT adversary A, $\Pr[\iota \leftarrow \mathcal{I}_\kappa;\ (x,y) \leftarrow A(\iota) :\ x \neq y\ \wedge\ H_\iota(x) = H_\iota(y)] = \mathsf{negl}(\kappa)$.

A.2 Target Collision-Resistant Hash Function Family

Let $\mathcal{H} = \{H_\iota\}_{\iota \in \mathcal{I}}$ be a keyed hash family of functions from $\{0,1\}^*$ to $\{0,1\}^{p(\kappa)}$, indexed by $\iota \in \mathcal{I}_\kappa$ $(= \mathcal{I} \cap \{0,1\}^\kappa)$, i.e., $H_\iota : \{0,1\}^* \to \{0,1\}^{p(\kappa)}$ for every $\iota \in \mathcal{I}_\kappa$. A keyed hash function family $\mathsf{TCH} = \{T_i\}_{i \in \mathcal{I}}$ is target collision-resistant (TCR) if, for every PPT $A = (A_1, A_2)$, $\mathsf{Adv}^{\mathsf{tcr}}_{\mathsf{TCH},A}(\kappa) = \mathsf{negl}(\kappa)$ where

$$\mathsf{Adv}^{\mathsf{tcr}}_{\mathsf{TCH},A}(\kappa) := \Pr\left[\begin{array}{ll} (x,s) \leftarrow A_1(1^\kappa); & \\ \iota \leftarrow \mathcal{I}_\kappa; & : x \neq y \text{ and } T_\iota(x) = T_\iota(y) \\ y \leftarrow A_2(\iota, s) & \end{array}\right].$$

A.3 One-Time Digital Signature Schemes

A digital signature scheme [23, 22] is given by a triple, $\Sigma = (\mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$, of PPT algorithms, where for every (sufficiently large) $\kappa \in \mathbb{N}$, KGen, the key-generation algorithm, takes as input security parameter 1^k and outputs a pair comprising the verification and signing keys, (vk, sk). Sign, the signing algorithm, takes as input (vk, sk) and message m and produces signature σ. Vrfy, the verification algorithm, takes as input verification key vk, message m and signature σ, and outputs a bit. For completeness, it is required that for all $(vk, sk) \in \mathsf{KGen}(1^\kappa)$ and for all $m \in \{0,1\}^*$, it holds $\mathsf{Vrfy}(vk, m, \mathsf{Sign}(sk, m)) = 1$.

We say that a digital signature scheme is *one-time* if it can be allowed to sign a message only once per a verification-key. For such a signature scheme, it is natural that an adversary is allowed to access the signing algorithm only once. We say that Σ is strongly one-time secure signature, if for all PPT adversary A, $\mathsf{Adv}^{\mathsf{ot}}_{\Sigma,A}(\kappa) = \mathsf{negl}(\kappa)$, where $\mathsf{Adv}^{\mathsf{ot}}_{\Sigma,A}(\kappa) :=$

$$\Pr\left[\begin{array}{c} (v, sk) \leftarrow \mathsf{otKGen}(1^\kappa); \\ (m, s) \leftarrow A(vk); \\ \sigma \leftarrow \mathsf{otSign}(sk, m); \\ (m^*, \sigma^*) \leftarrow A(s, \sigma^*) \end{array} : \begin{array}{c} \mathsf{Vrfy}(vk, m^*, \sigma^*) = 1 \\ \text{and } (m, \sigma) \neq (m^*, \sigma^*). \end{array}\right].$$

B Computational Hardness Assumptions

Let $\mathcal{G}$ be a PPT algorithm that takes security parameter 1^κ and outputs a triplet $\mathbb{G} = (G, q, g)$ where G is a group of prime order q that is generated by $g \in G$.

***d*-Linear Assumption.** The d-linear assumption [24, 29] (where $d \geq 1$), a generalization of the linear assumption [8], states that there is a PPT algorithm $\mathcal{G}$ such that the following two ensembles are computationally indistinguishable,

$$\left\{\left(\mathbb{G}, g_1, \ldots, g_d, g_{d+1}, g_1^{r_1}, \ldots, g_d^{r_d}, g_{d+1}^{\sum_{i=1}^d r_i}\right)\right\}_{\kappa\in\mathbb{N}}$$

$$\stackrel{c}{\approx} \left\{\left(\mathbb{G}, g_1, \ldots, g_d, g_{d+1}, g_1^{r_1}, \ldots, g_d^{r_d}, g_{d+1}^{r_{d+1}}\right)\right\}_{\kappa\in\mathbb{N}}$$

where $\mathbb{G} \leftarrow \mathcal{G}(1^\kappa)$, and the elements $g_1, \ldots, g_{d+1} \in G$ and $r_1, \ldots, r_{d+1} \in \mathbb{Z}/q\mathbb{Z}$ are chosen independently and uniformly at random. The DDH assumption (on $\mathcal{G}$) is equivalent to 1-linear assumption (on $\mathcal{G}$) and these assumptions are progressively weaker: For every $d \geq 1$, the $(d+1)$-linear assumption is weaker than the d-linear assumption.

Matrix *d*-Linear Assumption. We denote by $\mathrm{Rk}_i(\mathbb{F}_q^{m\times n})$ the set of all $m \times n$ matrices over $\mathbb{F}_q$ with rank i. The matrix d-linear assumption [29] states that there is a PPT algorithm $\mathcal{G}$ such that, for any integers, m and n, and for any $d \leq i \leq j \leq \min(m, n)$, the following two ensembles are computationally indistinguishable,

$$\left\{(\mathbb{G}, g, g^{\mathbf{x}}) \mid \quad \mathbb{G} \leftarrow \mathcal{G}(1^\kappa);\ \mathbf{x} \leftarrow \mathrm{Rk}_i(\mathbb{F}_q^{m\times n})\right\}_{\kappa\in\mathbb{N}}$$

$$\stackrel{c}{\approx} \quad \left\{(\mathbb{G}, g, g^{\mathbf{x}}) \mid \quad \mathbb{G} \leftarrow \mathcal{G}(1^\kappa);\ \mathbf{x} \leftarrow \mathrm{Rk}_j(\mathbb{F}_q^{m\times n})\right\}_{\kappa\in\mathbb{N}}.$$

It is known that breaking the matrix d-Linear assumption implies breaking the d-Linear assumption (on the same $\mathcal{G}$). The following statement holds.

Lemma 7 ([29]). *Breaking the matrix d-Linear assumption is at least as hard as breaking the d-Linear assumption (on the same $\mathcal{G}$).*

Extended Matrix d-Linear Assumption. We state a stronger version of the matrix d-linear assumption, called the extended matrix d-linear assumption [2]. For matrix $\mathbf{x} \in \mathbb{F}_q^{n\times m}$, we write $\ker(\mathbf{x})$ to denote the left kernel of $\mathbf{x}$, i.e.,

$$\ker(\mathbf{x}) = \{\boldsymbol{v} \in \mathbb{F}_q^n \,|\, \boldsymbol{v}^T\mathbf{x} = 0 \in \mathbb{F}_q^{1\times m}\}.$$

Here $\ker(\mathbf{x})$ is a subspace in $\mathbb{F}_q^n$ of dimension $(n - \mathsf{rank}(\mathbf{x}))$. The matrix d-linear assumption means that it is infeasible to distinguish $g^{\mathbf{x_i}}$ from $g^{\mathbf{x_j}}$, where rank-i matrix $\mathbf{x_i}$ and rank-j matrix $\mathbf{x_i}$ are chosen independently and uniformly for any $d \le i < j \le \min(n,m)$. Since $\dim(\ker(\mathbf{x_i})) = n - i$ and $\dim(\ker(\mathbf{x_j})) = n - j$ (with $n - j < n - i$), the matrix d-linear assumption does not hold if an adversary additionally receive $n - i$ independent vectors orthogonal to $\mathbf{x}$. However, one cannot yet distinguish them even if $n - j$ independent vectors orthogonal to $\mathbf{x}$ are given, as long as the matrix d-linear assumption holds true. The extended matrix d-linear assumption [2] states that there is a PPT algorithm $\mathcal{G}$ such that, for any integers, m and n, for any $d \le i \le j \le \min(m,n)$, and for any $\ell \le n - j$, the following two ensembles are computationally indistinguishable,

$$\Big\{(\mathbb{G}, g, g^{\mathbf{x}}, \boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}) \,|\, \mathbb{G} \leftarrow \mathcal{G}(1^\kappa);\, \mathbf{x} \leftarrow \mathrm{Rk}_i(\mathbb{F}_q^{m\times n}); v_1, \ldots, v_\ell \leftarrow \ker(\mathbf{x})\Big\}_{\kappa\in\mathbb{N}}$$

$$\stackrel{c}{\approx} \Big\{(\mathbb{G}, g, g^{\mathbf{x}}, \boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}) \,|\, \mathbb{G} \leftarrow \mathcal{G}(1^\kappa);\, \mathbf{x} \leftarrow \mathrm{Rk}_j(\mathbb{F}_q^{m\times n}); v_1, \ldots, v_\ell \leftarrow \ker(\mathbf{x})\Big\}_{\kappa\in\mathbb{N}}.$$

The following statement holds.

Lemma 8 ([10, 2]). *Breaking the extended matrix d-Linear assumption is at least as hard as breaking the d-Linear assumption (on the same $\mathcal{G}$).*

The proof is implicitly in [10].

Decision Computational Residue (DCR) Assumption. Let $n = pq$ be a composite number of distinct odd primes, p and q, and $1 \le d < p, q$ be a positive integer. We say that the DCR assumption holds if for every PPT A, there exists a parameter generation algorithm Gen such that $\mathsf{Adv}_A^{\mathsf{dcr}}(\kappa) =$

$$\Pr[\mathsf{Expt}_A^{\mathsf{dcr}-0}(\kappa) = 1] - \Pr[\mathsf{Expt}_A^{\mathsf{dcr}-1}(\kappa) = 1]$$

is negligible in κ, where

$\mathsf{Expt}_A^{\mathsf{dcr}-0}(\kappa)$:	$\mathsf{Expt}_{d,A}^{\mathsf{dcr}-1}(\kappa)$:
$n \leftarrow \mathsf{Gen}(1^\kappa)$; $R \stackrel{\mathsf{U}}{\leftarrow} \mathbb{Z}_{n^2}^\times$	$n \leftarrow \mathbb{G}(1^\kappa)$; $R \stackrel{\mathsf{U}}{\leftarrow} \mathbb{Z}_{n^2}^\times$
$c = R^n \bmod n^2$	$c = (1+n)R^n \bmod n^2$
return $A(n, c)$.	*return* $A(n, c)$.

C Instantiation of ABO Injective Functions

C.1 A Matrix Instantiation Based On DDH

Let $\mathcal{G}$ be a PPT algorithm that takes security parameter 1^κ and outputs a triplet $\mathbb{G} = (G, q, g)$ where G is a group of prime order q that is generated by $g \in G$. Let $\mathcal{B} = \{\mathbb{Z}/q\mathbb{Z}\}$ be a branch collection associated with $\mathbb{G} = (G, q, g)$ generated by $\mathcal{G}$.

– $\mathsf{ABO.gen}(1^\kappa, b^*)$ where $b^* \in \mathbb{Z}/q\mathbb{Z}$: Pick up a random column vector $\boldsymbol{u} = (u_i) \in G^\mu$ and a random column vector $\boldsymbol{v} = (v_j) \in G^\mu$. Compute matrix $\mathbf{A} = (A_{i,j}) \in G^{\mu\times\mu}$ as

$$\mathbf{A} = (\boldsymbol{u} \cdot \boldsymbol{v}^T) \boxplus g^{-(b^*)\mathbf{I}_\mu} = \left(u_i v_j g^{-(b^*)\delta_{i,j}}\right) \in G^{\mu\times\mu}$$

where $\boxplus$ denotes the componet-wise product of matrices over G, $\mathbf{I}_\mu \in (\mathbb{Z}/q\mathbb{Z})^{\mu\times\mu}$ is the identity matrix and $\delta_{i,j}$ is Kronecker's delta, i.e., $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise. We note that $\mathsf{rank}(\boldsymbol{u} \cdot \boldsymbol{v}^T) = 1$ and, at least with probability $1 - \frac{2\mu}{q}$, $\mathsf{rank}(A) = \mu$. We let $A(b)$ to denote

$$A(b) := A \boxplus g^{bI_\mu} = \left(u_i v_j g^{(b-b^*)\delta_{i,j}}\right) \in G^{\mu\times\mu}.$$

Finally, output $\iota_{\mathsf{abo}} = A(\cdot)$.

– $\mathsf{ABO.eval}(\iota_{\mathsf{abo}}, b, x)$: On input matrix $X \in (\mathbb{Z}/q\mathbb{Z})^{\mu\times d}$, output

$$\mathsf{ABO.eval}(\iota_{\mathsf{abo}}, b, x) = A(b) \cdot X \in G^{\mu\times d}.$$

This implementation realizes a collection of $(\mu \cdot d\log(q), (\mu - 1)d\log(q))$-all-but-one injective functions (under the DDH assumption).

C.2 DCR Based Instantiation

Let $n = pq$ be a composite number of distinct odd primes, p and q, and $1 \le d < p, q$ be a positive integer. It is known that $\mathbb{Z}^\times_{n^{d+1}} \cong \mathbb{Z}_{n^d} \times (\mathbb{Z}/n\mathbb{Z})^\times$ and any element in $\mathbb{Z}^\times_{n^{d+1}}$ is uniquely represented as $(1+n)^\delta \gamma^{n^d} \pmod{n^{d+1}}$ for some $\delta \in \mathbb{Z}_{n^d}$ and $\gamma \in (\mathbb{Z}/n\mathbb{Z})^\times$. For $\delta \in \mathbb{Z}_{n^d}$, we write $\mathbf{E}^{\mathsf{dj}}(\delta)$ to denote a subset in $\mathbb{Z}^\times_{n^{d+1}}$ such that $\mathbf{E}^{\mathsf{dj}}(\delta) = \{(1+n)^\delta \gamma^{n^d} \mid \gamma \in (\mathbb{Z}/n\mathbb{Z})^\times\}$. It is known that for any two distinct $\delta, \delta' \in \mathbb{Z}_{n^d}$, it is computationally hard to distinguish a random element in $\mathbf{E}^{\mathsf{dj}}(\delta)$ from a random element in $\mathbf{E}^{\mathsf{dj}}(\delta')$ as long as the decision computational residue (DCR) assumption holds true.

– $\mathsf{ABO.gen}(1^\kappa, b^*)$ where $b^* \in \{0,1\}^{d\kappa}$: Pick up $\kappa/2$-bit distinct odd primes p, q and compute $n = pq$. Then choose $\iota_{\mathsf{abo}} \leftarrow \mathbf{E}^{\mathsf{dj}}(-b^*)$. Output ι_{abo}.

– $\mathsf{ABO.eval}(\iota_{\mathsf{abo}}, b, x)$: On input matrix $x \in \mathbb{Z}_{n^d}$, output

$$\mathsf{ABO.eval}(\iota_{\mathsf{abo}}, b, x) = \left(\iota_{\mathsf{abo}} \cdot (1+n)^b\right)^x (\in \mathbf{E}^{\mathsf{dj}}(b - b^*)^x).$$

This implementation realizes a collection of $(d\log(n), \log((p-1)(q-1)))$-all-but-one injective functions (under the DCR assumption).

D Useful Lemmas

Lemma 9. *Let A_1, A_2, and F be events on some probability space. Suppose that $A_1 \cap \neg F$ occurs if and only if $A_2 \cap \neg F$ occurs. Then,* $\Pr[A_1] - \Pr[A_2] \le \Pr[F]$.

Lemma 10. *Let X_1 and X_2 be random variables on $\mathcal{S}$. Then, for every randomized function F,* $\mathsf{Dist}(F(X_1), F(X_2)) \le \mathsf{Dist}(X_1, X_2)$. *The equality holds if each realization f of F is one-to-one.*

The following reduction statement is often used. Let $X = \{X_\kappa\}_{\kappa\in\mathbb{N}}$ and $Y = \{Y_\kappa\}_{\kappa\in\mathbb{N}}$ be ensemble of random variables on $\{\mathcal{S}_\kappa\}_{\kappa\in\mathbb{N}}$. We define the advantage of PPT distinguisher D as

$$\mathsf{Adv}^{\mathsf{ind}}_{(X,Y),D}(\kappa) := \Pr[D(X_\kappa) = 1] - \Pr[D(Y_\kappa) = 1].$$

Consider an arbitrary interactive game between adversary A and challenger C over random variable Z:

- C takes 1^κ and random variable $Z \in \mathcal{S}_\kappa$, and runs A on some input.
- C interact with A following the game.
- At some point, C picks up bit $\beta^* \in \{0,1\}$ at random and changes his behavior depending on chosen bit β^*.
- Finally, A outputs bit β.

We define the advantage of A against C over Z as $\mathsf{Adv}^{\mathsf{ind}}_{(C,Z),A}(\kappa) = 2\Pr[\beta = \beta^*] - 1$ where the probability is taken over the choice of Z and random coins of C and A.

Then, we have the following lemma.

Lemma 11. *For any PPT A, there is some PPT distinguisher D such that*

$$\mathsf{Adv}^{\mathsf{ind}}_{(C,X),A}(\kappa) - \mathsf{Adv}^{\mathsf{ind}}_{(C,Y),A}(\kappa) \le 4\mathsf{Adv}^{\mathsf{ind}}_{(X,Y),D}(\kappa).$$

Namely,

$$\Pr_{C,A}[\beta = \beta^*|X_\kappa] - \Pr_{C,A}[\beta = \beta^*|Y_\kappa] \le 2\mathsf{Adv}^{\mathsf{ind}}_{(X,Y),D}(\kappa).$$

Proof. Construct distinguisher D that takes X_κ or Y_κ and simulates C. Let D outputs 1 if $\beta = \beta^*$ and a random bit otherwise. Then, we have $\mathsf{Adv}^{\mathsf{ind}}_{(X,Y),D}(\kappa) =$

$$\begin{aligned}&\Big(\Pr[\beta = \beta^*|X_\kappa] + \Pr[\beta \ne \beta^*|X_\kappa]\cdot\frac{1}{2}\Big) - \Big(\Pr[\beta = \beta^*|Y_\kappa] + \Pr[\beta \ne \beta^*|Y_\kappa]\cdot\frac{1}{2}\Big)\\ =&\frac{1}{2}\Big(\Pr[\beta = \beta^*|X_\kappa] - \Pr[\beta = \beta^*|Y_\kappa]\Big).\end{aligned}$$

D satisfies the above lemma.

E The Continuous Leakage Resileint CPA PKE Scheme

We propose a IND-CPA secure PKE scheme resilient to continuous memory leakage, based on Agrawal et al. scheme [2].

- The Key Generation Algorithm: Choose $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, g, \tilde{g}) \leftarrow \mathsf{GroupG}$. Pick up a random column vector $\boldsymbol{\alpha} \leftarrow (\mathbb{Z}/q\mathbb{Z})^n$. Pick up ℓ independent column vectors, $\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}$, in $(\mathbb{Z}/q\mathbb{Z})^n$ uniformly from $\mathsf{Ker}(\boldsymbol{\alpha})$ where $2 \le \ell \le n-2$. Set $n \times \ell$ matrix $\mathbf{V} = (\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell})$. Set $g^{\boldsymbol{\alpha}} := (g^{\alpha_1}, \ldots, g^{\alpha_n})^T$. Set $\tilde{g}^{\mathbf{V}} := (\tilde{g}^{\boldsymbol{v_1}}, \ldots, \tilde{g}^{\boldsymbol{v_\ell}})$. Pick up a random column vector $\boldsymbol{s} \leftarrow (\mathbb{Z}/q\mathbb{Z})^n$. Compute $\tilde{g}^{\boldsymbol{s}} = (\tilde{g}^{s_1}, \ldots, \tilde{g}^{s_n})^T$. Compute $Y = e(g^{\boldsymbol{\alpha}}, \tilde{g}^{\boldsymbol{s}}) = e(g, \tilde{g})^{\langle \boldsymbol{\alpha}, \boldsymbol{s}\rangle}$. Set $pk := (g, \tilde{g}, g^{\boldsymbol{\alpha}}, \tilde{g}^{\mathbf{V}}, Y)$ and $sk := \tilde{g}^{\boldsymbol{s}}$. Output (pk, sk).
- The Key Updating Algorithm: Take (pk, sk) as input. Choose a random column vector $\boldsymbol{r'} \leftarrow (\mathbb{Z}/q\mathbb{Z})^\ell$ and compute $\tilde{g}^{\boldsymbol{\beta}} = \tilde{g}^{\mathbf{V}r'}$. Update $sk := sk \cdot \tilde{g}^{\boldsymbol{\beta}} = \tilde{g}^{\boldsymbol{s}+\boldsymbol{\beta}}$. Note that $\boldsymbol{\beta} \in \mathsf{span}(\mathbf{V}) \subset \ker(\boldsymbol{\alpha})$. Output sk.

- The Encryption Algorithm: To encrypt $m \in \mathbb{G}_T$ under pk, pick up random $r \leftarrow \mathbb{Z}/q\mathbb{Z}$. Compute $\boldsymbol{C} = g^{r\boldsymbol{\alpha}}$ and $K = Y^r$. Output $\mathsf{CT} = (\boldsymbol{C}, e)$ where $e = m \cdot K$.
- The Decryption algorithm: To decrypt ciphertext $\mathsf{CT} = (g^{\boldsymbol{c}}, e)$ under sk, compute $K = e(g^{\boldsymbol{c}}, sk)(= e(g, \tilde{g})^{<\boldsymbol{c},\boldsymbol{s}>})$. Output $m = e \cdot K^{-1}$.

We define IND-CPA security of PKE resilient to λ-continuous memory leakage [10] as $(\emptyset, \emptyset, \lambda)$-CTL-CCA security of PKE.

Theorem 4. *The above PKE scheme is* $(\emptyset, \emptyset, \lambda)$-CTL-CCA *secure, as long as* $\lambda(\kappa) < \ell \log(q) - \omega(\log \kappa)$*, and for any PPT adversary* A*,*

$$\mathsf{Adv}^{\mathsf{ctl\text{-}cca}}_{\Pi,A,(\emptyset,\emptyset,\lambda)}(\kappa) \leq 4\epsilon_{\mathsf{ex}} + nQ \cdot \sqrt{\frac{2^\lambda}{q^{\ell-1}}} + \sqrt{\frac{2^\lambda}{q^{n-2}}}$$

where Q *denotes the total number of key-updates in the running time of* A*.*

Proof. Here we prove the theorem by using the standard game-hopping strategy. We denote by S_i the event that adversary A wins in **Game** i.

- **Game 0**: This game is the original game. We write $\mathsf{CT}^* = (g^{\boldsymbol{c}^*}, e^*)$ where $e^* = m_{b^*} \cdot K^*$ to denote the challenge ciphertext. Let us assume that Q is the maximum number of the key-updates. By definition, $\Pr[S_0] = \Pr[b = b^*]$ and $\mathsf{Adv}^{\mathsf{ctl\text{-}cca}}_{\Pi,A,(\emptyset,\emptyset,\lambda)}(\kappa) = |2\Pr[S_0] - 1|$.
- **Game 1**: In this game, we instead produce CT^* as follows: Compute $K^* = e(g^{\boldsymbol{c}^*}, sk) = e(g, \tilde{g})^{r\langle \boldsymbol{\alpha}, \boldsymbol{s}\rangle}$ and set $e^* = m_{b^*} \cdot K^*$. This change is just conceptual. Then, $\Pr[S_0] = \Pr[S_1]$.
- **Game 2**: This game is identical to **Game 1**, except that we choose ℓ independent vectors $\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell} \leftarrow \ker(\boldsymbol{\alpha}, \boldsymbol{c}^*)$ and set $\mathbf{V} = (\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell})$. Since $\boldsymbol{c}^* = r^*\boldsymbol{\alpha}$, $\ker(\boldsymbol{\alpha}, \boldsymbol{c}^*) = \ker(\boldsymbol{\alpha})$. Hence, $\Pr[S_1] = \Pr[S_2]$.
- **Game 3**: This game is identical to **Game 2**, except that when producing CT^*, we instead pick up random vector $\boldsymbol{c}^* \leftarrow \mathbb{F}_q^n$. We note that since $\dim(\ker(\boldsymbol{\alpha}, \boldsymbol{c}^*)) = n - 2 \geq \ell$, we can still choose ℓ independent vectors $\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}$. The difference between these two games is bounded by the extended matrix d-linear assumption.

  Lemma 12. *Under the extended matrix d-linear assumption in Appendix B, we have* $\Pr[S_2] - \Pr[S_3] \leq 2\epsilon_{\mathsf{ex}}$.

  Proof. Let $\mathbf{x} \in (\mathbb{Z}/q\mathbb{Z})^{n\times 2}$ whose columns are $\boldsymbol{\alpha}$ and $\boldsymbol{c}$, i.e., $\mathbf{x} = (\boldsymbol{\alpha}, \boldsymbol{c})$. Let $\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell}$ be ℓ independent random column vectors chosen via $\boldsymbol{v_i} \leftarrow \ker(\mathbf{x}) = \ker(\boldsymbol{\alpha}, \boldsymbol{c})$ and set $\mathbf{V} = (\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell})$. Now given $g^{\mathbf{x}}$ and $\mathbf{V} = (\boldsymbol{v_1}, \ldots, \boldsymbol{v_\ell})$, we can simulate public and secret keys that the adversary sees during the game, as well as the challenge ciphertext. In the case that $\mathsf{rank}(X) = 1$, we perfectly simulate Game 2. In the case that $\mathsf{rank}(X) = 2$, we perfectly simulate Game 3. Therefore, by using Lemma 11, we have $\Pr[S_2] - \Pr[S_3] \leq 2\epsilon_{\mathsf{ex}}$. ■

- **Game 4** is defined as a sequence of $Q + 1$ sub-games denoted by Games, $4.0, \ldots, 4.Q$. For $i = 0, \ldots, Q$, we have
  - **Game** $4.i$: This game is identical to Game 4.0, except that at the last i key-updates, we instead choose $\boldsymbol{\beta} \leftarrow \ker(\boldsymbol{\alpha})$ and update $sk := sk \cdot \tilde{g}^{\boldsymbol{\beta}}$. We insist that the first $Q - i$ key-updates, $\boldsymbol{\beta}$ is chosen from $\mathsf{span}(\mathbf{V})$, whereas in the last i key-updates, it is chosen from $\ker(\boldsymbol{\alpha})$.

Game 4.0 is identical to Game 3. The difference between Games, $4.i$ and $4.i+1$, is bounded by Lemma 5. Indeed, we have

$$\mathsf{Dist}\Big((\mathbf{V}, L(\boldsymbol{s} + \mathbf{V}\boldsymbol{r'})) : (\mathbf{V}, L(\boldsymbol{s} + \boldsymbol{\beta}))\Big) \leq \frac{n}{2}\sqrt{\frac{2^{\lambda}}{q^{\ell-1}}},$$

where $\mathbf{V} \leftarrow \Big(\ker(\boldsymbol{\alpha})\Big)^{\ell}$, $\boldsymbol{r'} \leftarrow (\mathbb{Z}/q\mathbb{Z})^{\ell}$, and $\boldsymbol{\beta} \leftarrow \ker(\boldsymbol{\alpha})$.

Therefore, we have $\Pr[S_{4.i}] - \Pr[S_{4.i+1}] \leq \frac{n}{2}\sqrt{\frac{2^{\lambda}}{q^{\ell-1}}}$, and hence, $\Pr[S_3] - \Pr[S_{4.Q}] \leq \frac{nQ}{2}\sqrt{\frac{2^{\lambda}}{q^{\ell-1}}}$.

– **Game 5**: This game is identical to **Game**$4.Q$, except that we pick up random $k^* \leftarrow \mathbb{Z}/q\mathbb{Z}$ and compute $K^* = e(g, \tilde{g})^{k^*}$. This k^* is statistically close to $< \boldsymbol{c}^*, \boldsymbol{s} + \boldsymbol{\beta} >$. Indeed, by Lemma 3,

$$\mathsf{Dist}((\boldsymbol{c}^*, < \boldsymbol{c}^*, \boldsymbol{s} + \boldsymbol{\beta} >, L(\boldsymbol{s} + \boldsymbol{\beta}), \mathsf{view}) : (\boldsymbol{c}^*, k^*, L(\boldsymbol{s} + \boldsymbol{\beta}), \mathsf{view})) \leq \frac{\sqrt{q}}{2} 2^{-\frac{1}{2}\widetilde{\mathsf{H}}_{\infty}(\boldsymbol{s}+\boldsymbol{\beta}|L(\boldsymbol{s}+\boldsymbol{\beta}), \mathsf{view})},$$

where view is fixed values containing $\boldsymbol{\alpha}$,$\mathbf{V}$, and $< \boldsymbol{\alpha}, \boldsymbol{s} >$ as well as $\tilde{\boldsymbol{\beta}}$'s in $\ker(\boldsymbol{\alpha})$ that are random vectors used in the past key-updates. Since $\boldsymbol{\beta}$ is only random variable in the above $\widetilde{\mathsf{H}}_{\infty}$, we have

$$\widetilde{\mathsf{H}}_{\infty}(\boldsymbol{s} + \boldsymbol{\beta}|L(\boldsymbol{s} + \boldsymbol{\beta}), \mathsf{view}) = \widetilde{\mathsf{H}}_{\infty}(\boldsymbol{\beta}|L(\boldsymbol{s} + \boldsymbol{\beta})) \geq \mathsf{H}_{\infty}(\boldsymbol{\beta}) - \lambda = (n-1)\log(q) - \lambda.$$

Therefore, we have $\Pr[S_{4.Q}] - \Pr[S_5] \leq \frac{1}{2}\sqrt{\frac{2^{\lambda}}{q^{n-2}}}$. By construction, $\Pr[S_5] = \frac{1}{2}$.

To summarize the above, we have $\Pr[S_0] - \frac{1}{2} =$

$$2\epsilon_{\mathsf{ex}} + \frac{1}{2}nQ \cdot \sqrt{\frac{2^{\lambda}}{q^{\ell-1}}} + \frac{1}{2}\sqrt{\frac{2^{\lambda}}{q^{n-2}}}.$$

■

F Continuos Tampering Secure Signature

A digital signature scheme $\Sigma = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Sign}, \mathsf{Vrfy})$ consists four algorithms. Setup, the set-up algoritm, takes as input security parameter 1^k and outputs public parameter ρ. KGen, the key-generation algorithm, takes as input ρ and outputs a pair comprising the verification and signing keys, (vk, sk). Sign, the signing algorithm, takes as input (ρ, sk) and message m and produces signature σ. Vrfy, the verification algorithm, takes as input verification key vk, message m and signature σ, as well as ρ, and outputs a bit. For completeness, it is required that for all $\rho \in \mathsf{Setup}(1^{\kappa})$, all $(vk, sk) \in \mathsf{KGen}(\rho)$ and for all $m \in \{0,1\}^*$, it holds $\mathsf{Vrfy}_{\rho}(vk, m, \mathsf{Sign}_{\rho}(sk, m)) = 1$.

We say that digital signature scheme Σ is **self-destructive**, if the signing algorithm can erase all inner states including sk and does not work any more, when it can detect tampering. We say that digital signature scheme Σ has a **key-updating** mechanism *if there is a PPT algorithm* Update *that takes* ρ *and* sk *and returns an "updated" secret key* $sk' = \mathsf{Update}_{\rho}(sk)$. We assume that the key-updating mechanism Update can be activated only when the signing algorithm detects tampering.

CTBL-CMA Security. For digital signature scheme Σ and an adversary A, we define the experiment $\mathsf{Expt}^{\mathsf{ctbl\text{-}cma}}_{\Pi,A,(\Phi,\lambda)}(\kappa)$ as in Fig. 5. We define the advantage of A against Π with respects Φ as

$$\mathsf{Adv}^{\mathsf{ctbl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa) \triangleq \Pr[\mathsf{Expt}^{\mathsf{ctbl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa) = 1].$$

A may adaptively submit (unbounded) polynomially many queries (ϕ, CT) to oracle RKSign, but it should be $\phi \in \Phi$. A may also adaptively submit (unbounded) polynomially many queries L to oracle Leak. Finally, A outputs (m', σ'). We say that A wins if $\mathsf{Vrfy}(\mathsf{vk}, m', \sigma') = 1$ and m' is not asked to RKSign. We note that if Sig has "self-destructive" property, RKSign does not receive any further query from the adversary or simply returns $\perp$. We say that Σ is (Φ, λ)-CTBL-CMA secure if $\mathsf{Adv}^{\mathsf{tbl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa) = \mathsf{negl}(\kappa)$ for every PPT A.

$\mathsf{Expt}^{\mathsf{ctbl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa)$: $\rho \leftarrow \mathsf{Setup}(1^\kappa)$; $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\rho)$; $(m', \sigma') \leftarrow A^{\mathsf{RKSign}_\Phi(\cdot,\cdot),\mathsf{Leak}_\lambda(\cdot)}(\rho, \mathsf{vk})$ If $m' \in$ List or $\mathsf{Vrfy}_\rho(\mathsf{vk}, m', \sigma') \neq 1$, then return 0; Otherwise 1.	$\mathsf{RKSign}_\Phi(\phi, m)$: $\sigma \leftarrow \mathsf{Sign}_\rho(\phi(\mathsf{sk}), m)$; If $\sigma = \perp$, then erase sk. Else return σ. ——— $\mathsf{Leak}_\lambda(L_i)$: ($L_i$: i-th query of A.) If $\sum_{j=1}^{i} \lvert L_j(\mathsf{sk})\rvert > \lambda$, then return $\perp$; Else return $L_i(\mathsf{sk})$.

Fig. 5. The experiment of the CTBL-CMA game.

CTL-CMA Security. For digital signature scheme $\Sigma = (\mathsf{Setup}, \mathsf{KGen}, \mathsf{Update}, \mathsf{Sign}, \mathsf{Vrfy})$ with a key-updating mechanism and an adversary A, we define the experiment $\mathsf{Expt}^{\mathsf{ctl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa)$ as in Fig. 6. We define the advantage of A against Σ with respects Φ as

$$\mathsf{Adv}^{\mathsf{ctl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa) \triangleq \Pr[\mathsf{Expt}^{\mathsf{ctl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa) = 1].$$

A may adaptively submit (unbounded) polynomially many queries (ϕ, CT) to oracle RKSign, but it should be $\phi \in \Phi$. A may also adaptively submit (unbounded) polynomially many queries L to oracle Leak. Finally, A outputs (m', σ'). We say that A wins if $\mathsf{Vrfy}(\mathsf{vk}, m', \sigma') = 1$ and m' is not asked to RKSign. We say that Σ is (Φ, λ)-CTL-CMA secure if $\mathsf{Adv}^{\mathsf{ctl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa) = \mathsf{negl}(\kappa)$ for every PPT A.

$\mathsf{Expt}^{\mathsf{ctbl\text{-}cma}}_{\Sigma,A,(\Phi,\lambda)}(\kappa)$:	$\mathsf{RKSign}_{\Phi}(\phi, m)$:

$\rho \leftarrow \mathsf{Setup}(1^\kappa)$;
$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\rho)$;
$(m', \sigma') \leftarrow A^{\mathsf{RKSign}_{\Phi}(\cdot,\cdot),\mathsf{Leak}_\lambda(\cdot)}(\rho, \mathsf{vk})$
If $m' \in$ List or $\mathsf{Vrfy}_\rho(\mathsf{vk}, m', \sigma') \neq 1$,
  then return 0;
Otherwise 1.
	$\sigma \leftarrow \mathsf{Sign}_\rho(\phi(\mathsf{sk}), m)$;
	If $\sigma = \perp$,
	  then return $\perp$;
	  Set $\mathsf{sk} \leftarrow \mathsf{Update}_\rho(\mathsf{sk})$;
	  Set LEAKSUM := 0;
	Else return σ.
	———
	$\mathsf{Leak}_\lambda(L)$:
	If LEAKSUM
	  := LEAKSUM + $\lvert L(\mathsf{sk})\rvert > \lambda$,
	  then return $\perp$;
	Else return $L(\mathsf{sk})$.

Fig. 6. The experiment of the CTL-CMA game.