Continuous Non-Malleable Key Derivation and Its Application to Related-Key Security

Baodong Qin[1,2], Shengli Liu[1,⋆], Tsz Hon Yuen[3], Robert H. Deng[4], and Kefei Chen[5]

1. Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
2. Southwest University of Science and Technology, Mianyang 621010, China
3. Huawei, Singapore
4. School of Information Systems, Singapore Management University, Singapore 178902
5. School of Science, Hangzhou Normal University, Hangzhou 310036, China
qinbaodong@sjtu.edu.cn, slliu@sjtu.edu.cn, Yuen.Tsz.Hon@huawei.com, robertdeng@smu.edu.sg, kfchen@sjtu.edu.cn

Abstract. Related-Key Attacks (RKAs) allow an adversary to observe the outcomes of a cryptographic primitive under not only its original secret key e.g., s, but also a sequence of modified keys $\phi(s)$, where ϕ is specified by the adversary from a class Φ of so-called Related-Key Derivation (RKD) functions. This paper extends the notion of non-malleable Key Derivation Functions (nm-KDFs), introduced by Faust et al. (EUROCRYPT'14), to *continuous* nm-KDFs. Continuous nm-KDFs have the ability to protect against any a-priori *unbounded* number of RKA queries, instead of just a single time tampering attack as in the definition of nm-KDFs. Informally, our continuous non-malleability captures the scenario where the adversary can tamper with the original secret key repeatedly and adaptively. We present a novel construction of continuous nm-KDF for any polynomials of bounded degree over a finite field. Essentially, our result can be extended to richer RKD function classes possessing properties of *high output entropy and input-output collision resistance*. The technical tool employed in the construction is the one-time lossy filter (Qin et al. ASIACRYPT'13) which can be efficiently obtained under standard assumptions, e.g., DDH and DCR. We propose a framework for constructing Φ-RKA-secure IBE, PKE and signature schemes, using a continuous nm-KDF for the same Φ-class of RKD functions. Applying our construction of continuous nm-KDF to this framework, we obtain the first RKA-secure IBE, PKE and signature schemes for a class of polynomial RKD functions of bounded degree under *standard* assumptions. While previous constructions for the same class of RKD functions all rely on non-standard assumptions, e.g., d-extended DBDH assumption.

Keywords: Related-key attacks, non-malleable key derivation, one-time lossy filter

1 Introduction

Traditionally, cryptographic security notions assume that an adversary can only observe the input/output behavior of the system and thus has only "black-box" access to the system. In a real life, however, it may be far from this case. Recent research [8] has shown that an adversary may learn some information about the secret key/internal state through physical side channels (e.g., timing [21] and power consumption [22]) and/or influence the way that the secret key/internal state is used via physical access to a hardware device (e.g., heating it or cutting wires to inject faults [7,8]). These two types of attacks are usually distinguished as "leakage" and "tampering" attacks respectively. In this paper, we consider how to design algorithms enabling devices resilient to tampering attacks when the devices are "leakage-proof" but not "tamper-proof". Specifically, we focus on tampering attacks on the key stored in a cryptographic hardware device. The key might be a signing key of a certificate authority or a decryption key of an encryption cryptosystem. Such tampering attacks are firstly formalized by Bellare and Kohno [5], as Related-Key Attacks (RKAs) in the context of pseudorandom functions/permutations.

⋆ Corresponding author.

Model of RKA Security. Following [4], we view a system as a composition of algorithms (code), public parameters, public keys (if any) and secret keys. Among these components, public parameters are system-wide, meaning that they are generated beforehand and independent of users and hence their public/secret keys. In an implementation, these parameters are part of the algorithm code and stored in a tamper-proof hardware device. Hence, only the public and secret keys are subject to RKAs.

Suppose that $\mathcal{CS}_{\mathrm{PP}}(s, x)$ is a cryptographic system parameterized by a public parameter PP. It admits a secret key s and a message x as input. For example, if $\mathcal{CS}$ has a decryption functionality, then s is a decryption key and x is a ciphertext. The RKA security model for $\mathcal{CS}$ is formalized by a class Φ of admissible key transformations (also named Related-Key Deriving (RKD) functions). An RKA adversary has the ability to repeatedly and adaptively choose x and a (tampering) function $\phi \in \Phi$, and then observe the outcome of $\mathcal{CS}_{\mathrm{PP}}(\phi(s), x)$ under this modified key $\phi(s)$. If the system is still secure, we say $\mathcal{CS}$ is Φ-RKA secure. Unless stated otherwise, in this paper, the RKA-security model allows an adversary to ask for a-priori unbounded number of RKD queries.

1.1 Motivation

It is not an easy task to design a provably secure scheme under RKAs for an especially large non-trivial class of RKD functions. To date, there are few constructions of RKA-secure primitives. The state-of-the-art RKA-security protects against a-priori unbounded number of queries for polynomials of bounded degree. However, all of them rely on non-standard assumptions, e.g., the d-extended DBDH (decisional bilinear Diffie-Hellman) assumption in the RKA-secure IBE [6] (and the degree of RKD polynomials is limited to d). There are generic approaches that use non-malleable codes [15,17] and non-malleable key derivation functions [17] to protect against tampering attacks even for function class richer than the polynomial one. However, both of them only consider *single time* tampering attack, *not* capturing the scenario of related-key attacks in which the adversary can continuously tamper with the original secret key. Indeed, as far as we know, no formal result shows how to achieve RKA security using these two primitives. Recently, Faust et al. [16] proposed an extension of the standard non-malleable codes, namely *continuous* non-malleable codes which cover the case that allows multiple tampering queries. However, this model relies on self-destruct mechanism, in which tampering queries must be terminated if an invalid code is detected (i.e., the decoding returns $\perp$). Moreover, their continuous non-malleable codes are realized in the split-state model [14] where an encoding is divided into two parts and the tampering must be applied to the two parts independently.

A natural question is whether we can define a stronger security model (than that of [16]) for continuous non-malleable codes or KDFs that can be used to achieve RKA security? Furthermore, can we achieve such continuous non-malleability for larger class of RKD functions under *standard* assumptions? In this paper, we provide affirmative answers to the two questions in the setting of key derivation functions.

1.2 Continuous Non-Malleable KDFs

Usually, a key derivation function KDF is equipped with another two probabilistic polynomial-time (PPT) algorithms: $\mathsf{KDF.Sys}$ and $\mathsf{KDF.Sample}$. The former takes as input a security parameter 1^κ and outputs a system parameter PP; the later takes as input PP, and outputs a derivation key s and a public key π. The key derivation function KDF is implicitly indexed by the system parameter PP and takes as input (s, π) to derive a key $r = \mathsf{KDF}_\pi(s)$ in polynomial-time. At a high level, we can always view (s, π) together as a derivation key. Since π is publicly accessible, any efficient adversary can tamper with π at its will. For this reason, we only explicitly specify the class Φ of tampering functions over the secret key space in this paper. We omit π if the derivation function does not take the public key as input, for example in [17]. The standard security notion for KDFs requires that the derived key r is indistinguishable from a random key if the adversary only knows the system parameter and the public key. Recently, Faust et al. [17] introduced the notion of non-malleable KDFs which, roughly speaking, guarantees that r is still random even if the adversary obtains another value $\mathsf{KDF}(s')$ as long as $s' \neq s$.

As shown previously, the standard non-malleability cannot protect against tampering attacks in some stateless settings where the adversary can continue to tamper with the original keys. To overcome this

drawback, we introduce a new notion, namely continuous non-malleable KDFs, as a natural extension of the standard non-malleability. The continuous non-malleability for function class Φ is defined by the following two experiments: $\mathsf{Real}_{\mathsf{KDF}}(\Phi, \kappa)$ and $\mathsf{Sim}_{\mathsf{KDF}}(\Phi, \kappa)$, in which the derivation key function involves the public key π as an auxiliary input.

1. The challenger generates PP and samples (s, π). In experiment $\mathsf{Real}_{\mathsf{KDF}}$, the challenger computes $r^* = \mathsf{KDF}_\pi(s)$, while in experiment $\mathsf{Sim}_{\mathsf{KDF}}$, the challenger samples r^* uniformly at random from its range.
2. The adversary $\mathcal{A}$ is given (PP, π) and the challenge key r^*.
3. $\mathcal{A}$ can repeatedly and adaptively query the following oracle with (ϕ, π') for any polynomially many times:

$$\text{If } (\phi(s), \pi') = (s, \pi), \text{ return } \mathsf{same}^\star; \text{ else, return } \mathsf{KDF}_{\pi'}(\phi(s)),$$

   where $\phi \in \Phi$, and π' is chosen by $\mathcal{A}$ at its will.

The continuous non-malleability requires that any PPT adversary has negligible advantage in distinguishing the above two experiments.

Though the adversary may tamper with s and π in a different (not necessarily independent) way, we stress that this is not a tampering attack as defined in the split-state model [14,17]. The reason is that π is a public key, any tampered result of π is provided by the adversary *at its will*, instead of being computed by the challenger. As shown in [18], it is impossible to prevent against continuous tampering attacks without any further assumption. Indeed, the continuous non-malleability achieved in the work of Faust et al. [16] limits to self-destruct and split-state model. Our new model above removes these two restrictions, hence is stronger. We will show in Section 4.1 that key derivation functions are still achievable in our new stronger security model, as long as we give a proper restriction (see Definition 1) on the tampering function classes.

Note that in our security model, we consider not only a continuously tampering adversary, but also an adaptive adversary which is allowed to access the tampering oracle after seeing the challenge derived key r^*. It might be of independent interest to consider a non-adaptive tampering adversary, which is only allowed to access the tampering oracle *before* receiving r^*.

1.3 Our Contributions

We summarize our contributions in the following and then detail the techniques that are used in our construction of continuous non-malleable KDF.

- Introduce the notion of *continuous* non-malleable Key Derivation Function (cnm-KDF) for an a-priori class of RKD functions Φ. Informally, we say a key derivation function KDF is continuously non-malleable with respect to Φ, namely Φ-cnm-KDF, if the output of KDF is still pseudo-random even if a PPT adversary tampers with its original key repeatedly and adaptively with function $\phi \in \Phi$.
- Provide a simple construction of continuous non-malleable KDF for the $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$-class of polynomial functions of bounded degree d over finite field $\mathbb{F}$. The construction exploits the functionality of one-time lossy filter (introduced by Qin et al. [24]) and some basic properties of polynomial functions over finite field.
  - We also generalize the polynomial function class $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$ to a larger function class, namely *High Output Entropy and Input-Output Collision Resistance (HOE&IOCR)*, which we denote by $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}}$. Function class $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}}$ possesses similar properties as polynomial functions. We show that our result works well even in such a richer RKD function class $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}}$ ($\Phi_{\mathsf{iocr}}^{\mathsf{hoe}} \supseteq \Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$).
  - The state-of-the-art One-Time Lossy Filters (OT-LFs) [24,25] suggest that OT-LFs can be instantiated from standard assumptions including the Decisional Diffie-Hellman (DDH) assumption and the Decisional Composite Residuosity (DCR) assumption. This leads to instantiations of $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$-cnm-KDF (w.r.t. $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}}$-cnm-KDF) based on standard assumptions.
- Propose a simple framework which transforms a traditional (non-RKA secure) IBE to a Φ-RKA-secure IBE with the help of Φ-cnm-KDF.

- The available standard-assumption-based IBEs and $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$-cnm-KDF suggests the first instantiations of $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$-RKA-secure IBE from standard assumption.
- Applying the transformation from Φ-RKA-secure IBE to PKE and signature schemes [4,6], we immediately obtain $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$-RKA-secure CCA-PKE and signature schemes under standard assumptions.

A Closer Look at Our Techniques. Our construction of continuous non-malleable KDF employs three cryptographic primitives: one-time lossy filter [24], pairwise independent hash function and one-time signature. A one-time lossy filter $\mathsf{LF}_t(\cdot)$ is a family of (one-way) functions parameterized by a tag t. The tag t can be either injective corresponding to an injective function, or lossy corresponding to a lossy function. One-time lossy filter has the following properties: (1) Injective and lossy tags are computationally indistinguishable; (2) There is a trapdoor to efficiently sample a lossy tag. However, without this trapdoor, it is hard to find a *non-injective* [1] tag even given one lossy tag. Recall that a family of pairwise independent hash functions $\mathcal{H}$ is an average-case strong extractor as long as its input has sufficiently large average min-entropy [13]. In our construction, we simply use h to derive the key $r = h(s)$, where $h \leftarrow_R \mathcal{H}$ and s is a random derivation key. Associated with the derivation key s is a public key computed by $\pi = t||\mathsf{LF}_t(s)$, where t is a random LF (injective) [2] tag. At a high level, π provides a knowledge proof of s so that an adversary who can compute a correct proof π' that corresponds to $\phi(s)$ must already know $\phi(s)$. To guarantee such property, in the proof, the tag t is moving from injective to lossy making π reveal only constant amount of information of s. Suppose that s is modified to $\phi(s)$ and π to any value $\pi' = t'||y'$. If $t' \neq t$, t' will be an injective tag with overwhelming probability and hence $\mathsf{LF}_{t'}(\cdot)$ is injective. So, if $\phi(s)$ has high residual min-entropy, the adversary should have negligible probability to correctly guess the value $\mathsf{LF}_{t'}(\phi(s))$. A challenging problem is that the adversary may reuse the lossy tag t, i.e., $t' = t$. To solve this problem, we apply a one-time signature scheme to π, guaranteeing that if t is reused, then $\pi' = \pi$ with overwhelming probability. Recall that a lossy tag is indistinguishable from an injective tag, and hence with overwhelming probability if $\pi' = \pi$, then $\phi(s) = s$. So, such case occurs unless $(\phi(s), \pi') = (s, \pi)$. Now, it only leaves us to discuss the entropy of $\phi(s)$ and the probability of $\phi(s) = s$. A simple property (for detail, see Lemma 3) is that for any non-constant polynomial, $\phi(s)$ has nearly the same entropy as s and if ϕ is not the identity function, then $\phi(s)$ equals s with negligible probability, as long as s has sufficiently large entropy. This concludes that except trivial queries (including the case $(\phi, \pi') = (\mathsf{id}, \pi)$ and the case ϕ is constant), it is hard to generate a valid proof π' for $\phi(s)$.

1.4 Related Work and Remarks

So far, there are not many RKA-secure primitives available and the main constructions are limited to PRFs [3,1], symmetric encryption [2,19,6], IBE [6], signature [6], and public-key encryption [28,6,23]. In particular, Bellare et al. [4] presented an almost complete understanding of the relations among these RKA-primitives. For example, RKA-secure PRFs can make any non-RKA secure primitive constructed from PRFs to be secure against RKAs. However, almost all of the realizations are secure only against simple and *claw-free*[3] RKD functions e.g., linear functions [2,28,23]. It may become more challenging to immunize a cryptographic primitive against non-linear and non-claw-free functions, e.g., affine and polynomial functions. One inherent reason is that a simulator, without the secret key s, is hard to detect dangerous queries such that $\phi(s) = s$ if ϕ is non-claw-free. To overcome this issue, all these methods [19,6,1] rely on non-standard assumptions, e.g., the d-extended DBDH (decisional bilinear Diffie-Hellman) assumption used in [6], from which the simulator is able to compute $\phi(s)$ (in the exponent) for any polynomial ϕ of bounded degree d.

Another approach that may be used to achieve RKA-security in a general way is the tamper-resilient codes, including algebraic manipulation detection codes [11] and (continuous) non-malleable codes [15,17,16]. The secret key stored on the device is now the encoded version of the original key using such a code. These codes considered very practical tampering functions. However, as we mentioned before, their security models

[1] In some case, a tag may be neither injective nor lossy.

[2] With overwhelming probability, a random tag is injective.

[3] A class of RKD functions is called *claw-free*, if for any distinct RKD functions $\phi \neq \phi'$ and all $s \in S$, $\phi(s) \neq \phi'(s)$

have some limitations, e.g., one-time tampering query or split-state model, which are inherent obstacles for capturing the scenario of RKAs security. Recently, Damgård et al.[12] showed that tamper-resilience (even combined with leakage-resilience) can be achieved for arbitrary key relations by restricting the number of adversary's tampering queries.

Concurrent Work. Jafargholi and Wichs [20] considered the same security level of continuous non-malleability and showed that continuous non-malleable codes are achievable if the tampering functions are polynomials or have few fixed points and high entropy (like the properties of HOE&IOCR functions). In contrast to ours, their results are constructed in the information-theoretic setting. However, the parameter in their construction [16, Corollary 5.6] depends on the number of tampering queries and the size of tampering function class. For efficient codes, the degree of polynomials must be set to some polynomial $d = d(\kappa)$. Additionally, they initiated a general study of continuous non-malleable codes and defined four variants of continuous non-malleability depending on (1) whether a tampering is persistent or non-persistent, meaning that any successive tampering function is applied to the former modified codeword or always applied to the original codeword, (2) whether we can self-destruct or not, meaning that we can stop the experiment if a codeword is invalid or the adversary can continue to tamper. Clearly, non-persistent tampering and no self-destruct require stronger model and is just the model considered in this paper.

Organization. We present our RKD function class in Section 3. We present the notion of continuous non-malleable KDF and its construction in Section 4. An application of continuous non-malleable KDF to the RKA-secure IBE is given in Section 5.

2 Preliminary

Notations. Throughout the paper, $\mathbb{N}$ is the set of natural numbers and $\kappa \in \mathbb{N}$ is the security parameter. If S is a finite set, then $s \leftarrow_R S$ denotes the operation of picking an element s from S uniformly at random. We call a function negl negligible in κ, if for every positive polynomial $\mathsf{poly}(\cdot)$ there exists an N such that for all $\kappa > N$, $\mathsf{negl}(\kappa) < 1/\mathsf{poly}(\kappa)$. We say that an event E happens with overwhelming probability, if it happens with probability $1 - \mathsf{negl}(\kappa)$. "PPT" stands for probabilistic polynomial-time. An algorithm A is PPT if it, on input x, computes $A(x)$ using randomness and its running time is bounded by $\mathsf{poly}(\kappa)$.

Average Min-entropy. The statistical distance between two random variables X and Y over a finite set Ω is $\Delta(X, Y) = \frac{1}{2}\sum_{\omega\in\Omega} |\Pr[X = \omega] - \Pr[Y = \omega]|$. We say that two variables are ϵ-close if their statistical distance is at most ϵ. The min-entropy of a random variable X is $H_\infty(X) = -\log(\max_x \Pr[X = x])$. Dodis et al. [13] formalized the notion of average min-entropy that captures the unpredictability of X conditioned on a random variable Y. Formally, it is defined as $\widetilde{H}_\infty(X|Y) = -\log(E_{y\leftarrow Y}[2^{-H_\infty(X|Y=y}])$.

We recall the following useful properties of average min-entropy from [13].

Lemma 1 ([13]). *Let X, Y and Z be random variables. Then*

1. *If Y has at most 2^r possible values and Z is any random variable, then $\widetilde{H}_\infty(X|(Y, Z)) \geq \widetilde{H}_\infty(X|Z) - r$.*
2. *For any $\delta > 0$, the conditional entropy $H_\infty(X|Y = y)$ is at least $\widetilde{H}_\infty(X|Y) - \log(1/\delta)$ with probability at least $1 - \delta$ over the choice of y.*

Average-Case Extractors [13]. A function $\mathsf{Ext} : \{0,1\}^n \times \mathcal{H} \to \{0,1\}^m$ is an efficient average-case (n, ν, m, ϵ)-strong extractor, if for all pairs of random variables (X, Z) such that $X \in \{0,1\}^n$ and $\widetilde{H}_\infty(X|Z) \geq \nu$, we have $\Delta((Z, h, \mathsf{Ext}(X, h)), (Z, h, U_m)) \leq \epsilon$, where h is uniform over $\mathcal{H}$ and U_m is uniform over $\{0,1\}^m$.

Lemma 2 ([13]). *Let $\mathcal{H}$ be a family of pairwise independent hash functions from $\{0,1\}^n$ to $\{0,1\}^m$. If $X \in \{0,1\}^n$, $\widetilde{H}_\infty(X|Z) \geq \nu$ and $m \leq \nu - 2\log 1/\epsilon$, then $\Delta((Z, h, h(X)), (Z, h, U_m)) \leq \epsilon$, where $h \leftarrow_R \mathcal{H}$ and U_m is uniform over $\{0,1\}^m$. In other words, the above family of pairwise independent hash functions can be used as an efficient average-case (n, ν, m, ϵ)-strong extractor.*

One-Time Lossy Filter. We adopt the notion of one-time lossy filter from [24]. An $(\mathcal{X}, \ell_{\mathsf{LF}})$-OT-LF LF consists of three PPT algorithms: (1) $\mathsf{LF.Gen}(1^\kappa)$, on input 1^κ, outputs an evaluation key ek_{LF} and a trapdoor td_{LF}. The evaluation key defines a tag space $\mathcal{T} = \{0,1\}^* \times \mathcal{T}_c$ that contains two disjoint subsets, the subset of lossy tags $\mathcal{T}_{loss} \subseteq \mathcal{T}$ and that of injective tags $\mathcal{T}_{inj} \subseteq \mathcal{T}$. A tag $t \in \mathcal{T}$ consists of an auxiliary tag $t_a \in \{0,1\}^*$ and a core tag $t_c \in \mathcal{T}_c$. The trapdoor td_{LF} allows to efficiently sample a lossy tag. (2) $\mathsf{LF.Eval}(ek_{\mathsf{LF}}, t, X)$, on input a tag t and a preimage $X \in \mathcal{X}$, computes $\mathsf{LF}_{ek_{\mathsf{LF}},t}(X) \in \mathcal{Y}$. (3) $\mathsf{LF.LTag}(td_{\mathsf{LF}}, t_a)$, on input an auxiliary tag t_a, computes a core tag t_c such that $t = (t_a, t_c)$ is lossy.

Besides the above functionalities, LF should satisfy the following properties:

Lossiness. If t is injective, so is the function $\mathsf{LF}_{ek_{\mathsf{LF}},t}(\cdot)$. If t is lossy, then $\mathsf{LF}_{ek_{\mathsf{LF}},t}(X)$ has at most $2^{\ell_{\mathsf{LF}}}$ possible values.

Indistinguishability. For any PPT adversary $\mathcal{A}$, it is hard to distinguish a lossy tag from a random tag, i.e., the following advantage is negligible in κ,

$$\mathsf{Adv}^{\mathrm{ind}}_{\mathsf{LF},\mathcal{A}}(\kappa) := \left|\Pr\left[\mathcal{A}(ek_{\mathsf{LF}}, (t_a, t_c)) = 1\right] - \Pr\left[\mathcal{A}(ek_{\mathsf{LF}}, (t_a, t'_c)) = 1\right]\right|,$$

where $(ek_{\mathsf{LF}}, td_{\mathsf{LF}}) \leftarrow \mathsf{LF.Gen}(1^\kappa)$, $t_a \leftarrow \mathcal{A}(ek_{\mathsf{LF}})$, $t_c \leftarrow \mathsf{LF.LTag}(td_{\mathsf{LF}}, t_a)$ and $t'_c \leftarrow_R \mathcal{T}_c$.

Evasiveness. For any PPT adversary $\mathcal{A}$, it is hard to generate a non-injective tag even given a lossy tag, i.e., the following advantage is negligible in κ,

$$\mathsf{Adv}^{\mathrm{eva}}_{\mathsf{LF},\mathcal{A}}(\kappa) := \Pr\left[\begin{array}{l|l} & (ek_{\mathsf{LF}}, td_{\mathsf{LF}}) \leftarrow \mathsf{LF.Gen}(1^\kappa) \\ (t'_a, t'_c) \neq (t_a, t_c) \wedge & t_a \leftarrow \mathcal{A}(ek_{\mathsf{LF}}) \\ (t'_a, t'_c) \in \mathcal{T} \setminus \mathcal{T}_{inj} & : t_c \leftarrow \mathsf{LF.LTag}(td_{\mathsf{LF}}, t_a) \\ & (t'_a, t'_c) \leftarrow \mathcal{A}(ek_{\mathsf{LF}}, (t_a, t_c)) \end{array}\right].$$

One-Time Signature. A one-time signature scheme OTS consists of four (probabilistic) polynomial-time algorithms: (1) $\mathsf{OTS.Sys}(1^\kappa)$, on input 1^κ, outputs a public parameter pp; (2) $\mathsf{OTS.Gen}(\mathrm{pp})$, on input pp, outputs a verification/signing key pair $(vk, sigk)$; (3) $\mathsf{OTS.Sig}(sigk, m)$, on input a message m, outputs a signature σ; (4) $\mathsf{OTS.Vrf}(vk, m, \sigma)$, on input a message/signature pair (m, σ), outputs 1 if σ is indeed a signature of m or 0 otherwise. We say that OTS is strongly secure against chosen-message attacks, if for any stateful PPT adversary $\mathcal{A}$, the following advantage is negligible in κ,

$$\mathsf{Adv}^{\mathrm{cma}}_{\mathsf{OTS},\mathcal{A}}(\kappa) := \Pr\left[\begin{array}{l|l} & \mathrm{pp} \leftarrow \mathsf{OTS.Sys}(1^\kappa) \\ & (vk, sigk) \leftarrow \mathsf{OTS.Gen}(\mathrm{pp}) \\ (m', \sigma') \neq (m, \sigma) \wedge & m \leftarrow \mathcal{A}(\mathrm{pp}, vk) \\ \mathsf{OTS.Vrf}(vk, m', \sigma') = 1 & : \sigma \leftarrow \mathsf{OTS.Sig}(sigk, m) \\ & (m', \sigma') \leftarrow \mathcal{A}(\sigma) \end{array}\right].$$

3 Properties of RKD Functions over Finite Fields

A class Φ of Related-Key Derivation (RKD) functions over $\mathcal{S}$ is a set of functions, all with the same domain and range $\mathcal{S}$. Suppose that $\mathbb{F}$ is a finite field such that $|\mathbb{F}| \geq 2^n$ for some positive integer n. Let $d \geq 0$ be any fixed integer. Define $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$ to be the set of all polynomial functions over $\mathbb{F}$ with degree bounded by d. Clearly, $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$ includes the identity function $f = \mathsf{id}$ (i.e., $f(x) = x$) and all the constant functions (denoted by $\mathsf{cf} = \{f_c : \mathbb{F} \to c\}_{c \in \mathbb{F}}$). We introduce the following simple lemma.

Lemma 3. *Let $\mathbb{F}$ and $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$ be defined as above. Let X be any random variable over $\mathbb{F}$ such that $H_\infty(X) \geq n$. For any $f \in \Phi_{\mathbb{F}}^{\mathsf{poly}(d)} \backslash \mathsf{cf}$, then $H_\infty(f(X)) \geq n - \log d$ and for any $f \in \Phi_{\mathbb{F}}^{\mathsf{poly}(d)} \backslash \{\mathsf{id}\}$ then $\Pr[f(X) = X] \leq \frac{d}{2^n}$.*

Proof. For any polynomial $f \in \Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$, let B_f denote the set of all solutions x over $\mathbb{F}$ such that $f(x) = 0$. Clearly, if f is not identically zero, then $|B_f|$ is bounded by d. For any fixed value $a \in \mathbb{F}$, if f is not a constant

function, then $f'(x) = f(x) - a$ is not identically zero. This shows that $f'(x) = 0$ has at most d solutions x, i.e., $|B_{f'}| \leq d$. Then,

$$\Pr_{x \leftarrow_R \mathbb{F}}[f(x) = a] = \Pr_{x \leftarrow_R \mathbb{F}}[x \in B_{f'}] \leq \frac{d}{|\mathbb{F}|} \leq \frac{d}{2^n}.$$

Hence, $H_\infty(f(X)) = -\log(\max_{a \in \mathbb{F}} \Pr[f(X) = a]) \geq n - \log d$.

Similar to the above analysis, if f is not identity function, then $f''(x) = f(x) - x$ is not identically zero. Hence $f''(x) = 0$ has at most d solutions over $\mathbb{F}$, i.e., $|B_{f''}| \leq d$. Then

$$\begin{aligned}
\Pr[f(X) = X] &= \sum_{a \in \mathbb{F}} \Pr[f(X) = a \wedge X = a] \\
&= \sum_{a \in B_{f''}} \Pr[f(X) = a \wedge X = a] + \sum_{a \in \mathbb{F} \setminus B_{f''}} \Pr[f(X) = a \wedge X = a] \\
&= \Pr_{x \leftarrow_R \mathbb{F}}[x \in B_{f''}] + 0 \\
&\leq \tfrac{d}{2^n}.
\end{aligned}$$

This completes the proof of Lemma 3. □

Remark 1. In our main result (see Theorem 1), we restrict the RKD function class to polynomials as the proof needs the properties stated in Lemma 3. In fact, we can extend it to any RKD function class that has similar properties as polynomials. We call such function class *High Output Entropy and Input-Output Collision Resistant (HOE&IOCR)* function class, which is formally defined in Definition 1.

Definition 1 (HOE&IOCR RKD function class). *Let $\mathcal{S}$ be a set with super-polynomial size in the security parameter κ. The RKD function class $\Phi^{\mathsf{hoe}}_{\mathsf{iocr}} : \mathcal{S} \to \mathcal{S}$ is called the class of* High Output Entropy and Input-Output Collision Resistance *(HOE&IOCR) as long as it satisfies the following properties.*

- *(High Output Entropy) When S is chosen uniformly at random from $\mathcal{S}$, for each $\phi \in \Phi^{\mathsf{hoe}}_{\mathsf{iocr}} \setminus \mathsf{cf}$, the entropy $H_\infty(\phi(S))$ is sufficiently large, i.e., $2^{-H_\infty(\phi(S))}$ is negligible in κ;*
- *(Input-Output Collision Resistance) For each $\phi \in \Phi^{\mathsf{hoe}}_{\mathsf{iocr}} \setminus \{\mathsf{id}\}$, the probability $\Pr[\phi(S) = S]$ is negligible in κ.*

Clearly, $\Phi^{\mathsf{poly}(d)}_{\mathbb{F}} \subseteq \Phi^{\mathsf{hoe}}_{\mathsf{iocr}}$, and $\Phi^{\mathsf{poly}(d)}_{\mathbb{F}}$ satisfies $\mathcal{S} = \mathbb{F}$, $H_\infty(S) \geq n$, $H_\infty(\phi(S)) \geq n - d$ and $\Pr[\phi(S) = S] \leq \frac{d}{2^n}$.

4 Continuous Non-Malleable Key Derivation

A key derivation function consists of three (PPT) algorithms: (1) The public parameter generation algorithm $\mathsf{KDF.Sys}(1^\kappa)$, on input 1^κ, outputs a system parameter PP, which defines the derivation key space $\mathcal{S}$ and the derived key space $\{0,1\}^m$. (2) $\mathsf{KDF.Sample}(\mathrm{PP})$, on input PP, samples a random derivation key $s \in \mathcal{S}$ and computes a public key, denoted by π. (3) The deterministic algorithm $\mathsf{KDF}_\pi(s)$, on input (s, π), outputs a derived key r or the special symbol $\perp$, indicating that π is an invalid proof of s. The standard security notion of KDF guarantees that r is (computationally or information theoretically) indistinguishable from a uniform over $\{0,1\}^m$ even given the public parameter PP and the proof π.

The notion of non-malleable key derivation [17] was firstly introduced by Faust et al. at Eurocrypt 2014. Intuitively, a function KDF is a non-malleable key derivation function if $\mathsf{KDF}(s)$ [4] is statistically close to uniform even given the output of KDF applied to a related input s' as long as $s' \neq s$. The non-malleability for a key derivation function aims to capture the scenario of one-time tampering attack for tampering function family with all circuits of bounded size. In this section, we extend it to the notion of continuous non-malleability (see Fig. 1) for an a-priori class Φ of RKD functions, making it possible to protect against multiple-time tampering attacks on a fixed secret key (i.e., RKAs).

[4] In [17], the key derivation is defined in the information theoretic setting, not taking π as an auxiliary input, i.e., π is empty.

Definition 2 (Continuous non-malleable KDFs). *Let Φ be a class of RKD functions over the same domain and range $\mathcal{S}$. We say that* $(\mathsf{KDF.Sys}, \mathsf{KDF.Sample}, \mathsf{KDF})$ *is a (Φ, ϵ)-continuous non-malleable key derivation function if for any* ***stateful*** *PPT adversary $\mathcal{A}$,*

$$|\Pr[\mathcal{A}(\mathsf{Real}_{\mathsf{KDF}}(\Phi, \kappa)) = 1] - \Pr[\mathcal{A}(\mathsf{Sim}_{\mathsf{KDF}}(\Phi, \kappa)) = 1]| \leq \epsilon.$$

The experiments $\mathsf{Real}_{\mathsf{KDF}}(\Phi, \kappa)$ and $\mathsf{Sim}_{\mathsf{KDF}}(\Phi, \kappa)$ are defined in Fig. 1 (Suppose that $\mathcal{A}$ makes at most $Q(\kappa)$ queries).

Experiment $\mathsf{Real}_{\mathsf{KDF}}(\Phi, \kappa)$:	Experiment $\mathsf{Sim}_{\mathsf{KDF}}(\Phi, \kappa)$:
$\text{PP} \leftarrow \mathsf{KDF.Sys}(1^\kappa)$	$\text{PP} \leftarrow \mathsf{KDF.Sys}(1^\kappa)$
$s\|\|\pi \leftarrow \mathsf{KDF.Sample}(\text{PP})$ // $s \in \mathcal{S}$	$s\|\|\pi \leftarrow \mathsf{KDF.Sample}(\text{PP})$ //$s \in \mathcal{S}$
$r = \mathsf{KDF}_\pi(s)$	$r \leftarrow_R \{0,1\}^m$
For $i = 1$ to $Q(\kappa)$	For $i = 1$ to $Q(\kappa)$
$(\phi, \pi') \leftarrow \mathcal{A}(\text{PP}, r, \pi)$ // $\phi \in \Phi$	$(\phi, \pi') \leftarrow \mathcal{A}(\text{PP}, r, \pi)$ // $\phi \in \Phi$
If $\phi(s)\|\|\pi' = s\|\|\pi$	If $\phi(s)\|\|\pi' = s\|\|\pi$
return $\mathsf{same}^\star$.	return $\mathsf{same}^\star$.
Else	Else
return $\mathsf{KDF}_{\pi'}(\phi(s))$.	return $\mathsf{KDF}_{\pi'}(\phi(s))$.

Fig. 1. Experiments for continuous non-malleable KDFs

4.1 The Construction

In this subsection, we construct a continuous non-malleable key derivation function with respect to $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$ from one-time lossy filter.

Let $(\mathsf{LF.Gen}, \mathsf{LF.Eval}, \mathsf{LF.LTag})$ be a collection of one-time lossy filters with domain $\mathcal{S}$ (such that $\mathcal{S} \subseteq \mathbb{F}$), range $\mathcal{Y}$, residual leakage ℓ_{LF} and tag space $\mathcal{T} = \{0,1\}^* \times \mathcal{T}_c$. Let $\mathcal{H}$ be a family of pairwise independent hash functions from domain $\mathcal{S}$ to range $\{0,1\}^m$. Let $(\mathsf{OTS.Sys}, \mathsf{OTS.Gen}, \mathsf{OTS.Sig}, \mathsf{OTS.Vrf})$ be a strongly secure one-time signature with verification key space $\mathcal{K}_{\mathsf{OTS}}$ and signature space Σ. Define $\Pi := \mathcal{T} \times \mathcal{Y} \times \Sigma$. The construction is given in Fig. 2.

- $\mathsf{KDF.Sys}(1^\kappa)$: It runs $(ek_{\mathsf{LF}}, td_{\mathsf{LF}}) \leftarrow \mathsf{LF.Gen}(1^\kappa)$ and $\text{PP}_{\mathsf{OTS}} \leftarrow \mathsf{OTS.Sys}(1^\kappa)$, chooses $h \leftarrow_R \mathcal{H}$, and returns $\text{PP}_{\mathsf{KDF}} := (ek_{\mathsf{LF}}, \text{PP}_{\mathsf{OTS}}, h)$.
- $\mathsf{KDF.Sample}(\text{PP}_{\mathsf{KDF}})$: It runs $(vk, sigk) \leftarrow \mathsf{OTS.Gen}(\text{PP}_{\mathsf{OTS}})$, chooses $s \leftarrow_R \mathcal{S}$ and $t_c \leftarrow_R \mathcal{T}_c$, and computes

$$y = \mathsf{LF}_{ek_{\mathsf{LF}}, (vk, t_c)}(s) \text{ and } \sigma = \mathsf{OTS.Sig}(sigk, t_c||y).$$

  Let $\pi := t||y||\sigma$ and $t := (vk, t_c)$. Finally, it returns $s||\pi$.
- $\mathsf{KDF}_\pi(s)$: It parses π as $t||y||\sigma$ and t as (vk, t_c). If the following two equations

$$\mathsf{LF}_{ek_{\mathsf{LF}}, (vk, t_c)}(s) = y \tag{1}$$

$$\mathsf{OTS.Vrf}(vk, t_c||y, \sigma) = 1 \tag{2}$$

  hold simultaneously, it returns $r = h(s)$; else it returns $\perp$.

Fig. 2. Continuous non-malleable KDF w.r.t. RKD functions $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$

Theorem 1. *The KDF given in Fig. 2 is* $(\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}, \epsilon)$*-continuously non-malleable. Concretely, for any* $\delta > 0$ *and any PPT adversary* $\mathcal{A}$ *that makes at most* $Q(\kappa)$ *queries and breaks the continuous non-malleability with advantage* ϵ*, there exist adversaries* $\mathcal{B}$*,* $\mathcal{B}'$ *and* $\mathcal{B}''$ *of roughly the same time complexity as* $\mathcal{A}$*, such that*

$$\epsilon \leq 2\Big(\mathsf{Adv}^{\mathrm{cma}}_{\mathsf{OTS},\mathcal{B}}(\kappa) + \mathsf{Adv}^{\mathrm{ind}}_{\mathsf{LF},\mathcal{B}'}(\kappa) + Q(\kappa)\cdot \mathsf{Adv}^{\mathrm{eva}}_{\mathsf{LF},\mathcal{B}''}(\kappa) + Q(\kappa)\cdot\Big(\delta + \tfrac{d\cdot 2^{m+\ell_{\mathsf{LF}}+\log 1/\delta}}{|\mathcal{S}|-Q(\kappa)+1}\Big) + \epsilon_{\mathcal{H}}\Big),$$

where $\mathcal{S}$ *and* ℓ_{LF} *respectively are the domain and residual leakage of the one-time lossy filter,* m *is the output length of the pairwise independent hash,* d *is the maximum degree of RKD functions and* $\log|\mathcal{S}| \geq \max\{\ell_{\mathsf{LF}} + m + 2\log 1/\epsilon_{\mathcal{H}}, \ell_{\mathsf{LF}} + m + \log 1/\delta\}$*. Taking into account that* ϵ *should be negligible in the security parameter* κ*, we may choose negligible* δ *and* $\epsilon_{\mathcal{H}}$*, and choose a OT-LF with sufficiently large domain* $\mathcal{S}$ *such that* $\log|\mathcal{S}| = \ell_{\mathsf{LF}} + m + \omega(\log\kappa)$*. Moreover, the degree of RKD functions can be made to* 2^κ *as long as* $\log|\mathcal{S}| = \ell_{\mathsf{LF}} + m + \omega(\log\kappa) + \kappa$*.*

Games:	Key derivation rules :	r:	t_c:
Game_0	R0: If $\phi(s)\|\|\pi' = s\|\|\pi$, return $\mathsf{same}^\star$, else if Eq. (1) and Eq. (2) hold, return $\mathsf{KDF}_{\pi'}(\phi(s))$, else return $\perp$.	$r = h(s)$	$t_c \leftarrow_R \mathcal{T}_c$
Game_1	**R1**: If $(\phi,\pi') = (\mathsf{id},\pi)$, return $\mathsf{same}^\star$. If $\phi = \mathsf{id}$, $(vk', t_c') = (vk, t_c)$, but $y' \neq y$, return $\perp$. If $\phi = \phi_c$ is a constant function, return $\mathsf{KDF}_{\pi'}(c)$. R0: As in Game_0.	$r = h(s)$	$t_c \leftarrow_R \mathcal{T}_c$
Game_2	R1: As in Game_1. **R2**: If $vk' = vk$, but $(t_c'\|\|y', \sigma') \neq (t_c\|\|y, \sigma)$, return $\perp$. R0: As in Game_1.	$r = h(s)$	$t_c \leftarrow_R \mathcal{T}_c$
Game_3	R1: As in Game_2. R2: As in Game_2. **R3**: If $\pi' = \pi$, but $\phi(s) \neq s$, return $\perp$. R0: As in Game_2.	$r = h(s)$	$t_c \leftarrow_R \mathcal{T}_c$
Game_4	The same as in Game_3.	$r = h(s)$	$t_c \leftarrow \mathsf{LF.LTag}(td_{\mathsf{LF}}, vk)$
Game_5	R1: As in Game_4. R2: As in Game_4. *R3*: Replaced by **R0'**. *R0*: Replaced by **R0'**. **R0'**: Return $\perp$.	$r = h(s)$	$t_c \leftarrow \mathsf{LF.LTag}(td_{\mathsf{LF}}, vk)$
Game_6	As in Game_5.	$r \leftarrow_R \{0,1\}^m$	$t_c \leftarrow \mathsf{LF.LTag}(td_{\mathsf{LF}}, vk)$
Game_7	As in Game_0.	$r \leftarrow_R \{0,1\}^m$	$t_c \leftarrow_R \mathcal{T}_c$

Fig. 3. Changes in each game

Proof. We prove it through a sequence of games played between a simulator Sim and a fixed PPT adversary $\mathcal{A}$. The initial game (i.e., Game_0) is the experiment $\mathsf{Real}_{\mathsf{KDF}}(\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}, \kappa)$ and the final game is the experiment $\mathsf{Sim}_{\mathsf{KDF}}(\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}, \kappa)$ as defined in Fig. 1. Denote by S_i the output of $\mathcal{A}$ in Game_i.

Game_0 **(The real experiment):** This is the real experiment $\mathsf{Real}_{\mathsf{KDF}}(\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}, \kappa)$ as defined in Fig. 1. For simplicity, we denote by $\mathrm{PP}_{\mathsf{KDF}} = (ek_{\mathsf{LF}}, \mathrm{PP}_{\mathsf{OTS}}, h)$ the challenge public parameters and denote by $s\|\|\pi$ the challenge sample, where $\pi = t\|\|y\|\|\sigma$, $t = (vk, t_c)$ and vk is the corresponding OTS verification key (with respect to the signing key $sigk$). We write (ϕ,π') as $\mathcal{A}$'s queries, where $\pi' = t'\|\|y'\|\|\sigma'$ and $t' = (vk', t_c')$. Then,

$$\Pr[\mathcal{A}(\mathsf{Real}_{\mathsf{KDF}}(\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}, \kappa)) = 1] = \Pr[S_0 = 1].$$

Game_1 **(Handling trivial queries without the KDF key):** This game is the same as Game_0, except that the simulator uses the new rule R1 to answer some trivial queries as given in Fig. 3. Specifically, for these trivial queries, the simulator never uses the real derivation key s to compute the value of $\mathsf{KDF}_{\pi'}(\phi(s))$. These modifications are just conceptual and hence

$$\Pr[S_1 = 1] = \Pr[S_0 = 1].$$

Game_2 **(Eliminating OTS key reuse):** This game is the same as Game_1, except for a modification to the verification oracle as stated in Fig. 3. Let E_{OTS} denote the event that $\mathcal{A}$ submits a query $(\phi, \pi' = (vk', t'_c)||y'||\sigma')$ such that $vk' = vk$, $(t'_c||y', \sigma') \neq (t_c||y, \sigma)$ but $\mathsf{OTS.Vrf}(vk, t'_c||y', \sigma') = 1$. Clearly, Game_2 is identical to Game_1 unless the event E_{OTS} occurs. We briefly show that if the adversary makes the event E_{OTS} occur, then an efficient algorithm $\mathcal{B}$ can be constructed to break the strong security of OTS using $\mathcal{A}$ as a subroutine.

Given an OTS challenge instance $(\mathsf{PP}_{\mathsf{OTS}}, vk)$, $\mathcal{B}$ runs $(ek_{\mathsf{LF}}, td_{\mathsf{LF}}) \leftarrow \mathsf{LF.Gen}(1^\kappa)$, chooses $h \leftarrow_R \mathcal{H}$, and sets $\mathsf{PP}_{\mathsf{KDF}} := (ek_{\mathsf{LF}}, \mathsf{PP}_{\mathsf{OTS}}, h)$. Then $\mathcal{B}$ samples s, t_c and computes $y = \mathsf{LF}_{ek_{\mathsf{LF}},(vk,t_c)}(s)$ by itself. Also, $\mathcal{B}$ generates σ by querying OTS signing oracle once with $t_c||y$. Since $\mathcal{B}$ knows s, it can answer all the decryption queries (ϕ, π') from $\mathcal{A}$ (recall that decryption does not need the knowledge of the challenge OTS signing key $sigk$). So, $\mathcal{B}$ perfectly simulates the real experiment defined in Game_1 for $\mathcal{A}$. If $\mathcal{A}$ submits a query (ϕ, π') making the event E_{OTS} occur, $\mathcal{B}$ returns $(t'_c||y', \sigma')$ (Note that, $\mathcal{B}$ can check whether the event E_{OTS} occurs or not). From the above observation, we have

$$|\Pr[S_2 = 1] - \Pr[S_1 = 1]| \leq \mathsf{Adv}^{\mathrm{cma}}_{\mathsf{OTS},\mathcal{B}}(\kappa).$$

Game_3 **(Answering a trivial query with the KDF key):** If the adversary submits a query (ϕ, π') such that $\pi' = \pi$ (i.e., $vk' = vk$ and $(t'_c||y', \sigma') = (t_c||y, \sigma)$), the simulator first checks whether $\phi(s) = s$. If not, it returns $\perp$ and halts immediately. Otherwise, the simulator handles it as in Game_2. Recall that, with overwhelming probability, a randomly chosen LF tag (vk, t_c) is injective. So, if $\phi(s) \neq s$, then $\mathsf{LF}_{ek_{\mathsf{LF}},(vk,t_c)}(\phi(s)) \neq y$. This implies that such queries will also be rejected under the rules of Game_2. Hence, with overwhelming probability

$$\Pr[S_3 = 1] = \Pr[S_2 = 1].$$

Game_4 **(From injective to lossy LF tag):** Instead of picking $t_c \in \mathcal{T}_c$ uniformly at random, the simulator computes $t_c := \mathsf{LF.LTag}(td_{\mathsf{LF}}, vk)$.

We show that the difference between Game_3 and Game_4 can be reduced to the indistinguishability of the underlying OT-LF. Given a challenge LF evaluation key ek_{LF}, a PPT algorithm $\mathcal{B}'$ chooses h and $\mathsf{PP}_{\mathsf{OTS}}$, samples s and $(vk, sigk)$ by itself. Then, it queries its injective-lossy tag oracle with query $t_a = vk$. $\mathcal{B}'$ will receive a challenge core tag part t_c. It computes $y = \mathsf{LF}_{ek_{\mathsf{LF}},(vk,t_c)}(s)$ and $\sigma = \mathsf{OTS.Sig}(sigk, t_c||y)$, and sets $\pi = (vk, t_c)||y||\sigma$. It sends $\mathsf{PP}_{\mathsf{KDF}} = (ek_{\mathsf{LF}}, \mathsf{PP}_{\mathsf{OTS}}, h)$ together with π to $\mathcal{A}$. Since $\mathcal{B}'$ knows the KDF key s, it can answer all the queries issued by $\mathcal{A}$. Finally, $\mathcal{B}'$ outputs whatever $\mathcal{A}$ outputs. Clearly, if t_c is sampled from $\mathcal{T}_c$ uniformly at random, then $\mathcal{B}'$ simulates Game_3 perfectly. If t_c is computed by $\mathsf{LF.LTag}(ek_{\mathsf{LF}}, t_a)$, then $\mathcal{B}'$ perfectly simulates Game_4. Hence,

$$|\Pr[S_4 = 1] - \Pr[S_3 = 1]| \leq \mathsf{Adv}^{\mathrm{ind}}_{\mathsf{LF},\mathcal{B}'}(\kappa)$$

for some adversary $\mathcal{B}'$ attacking on the indistinguishability of OT-LF.

Game_5 **(Answering all queries without the KDF key):** In this game, the simulator replaces the rules in step R3 and R0 (relying on the KDF key) with R0' (without relying on the KDF key) as stated in Fig. 3. Note that, the new rule directly rejects all queries except those trivial queries which have already be answered by rule R1. Denote by F the event that $\mathcal{A}$ submits a query (ϕ, π') such that the simulator returns the special symbol $\perp$ in Game_5, but not in Game_4. Also, let E_{ninj} denote the event that among all the queries (ϕ, π'), there exists some non-injective LF tag such that $(vk', t'_c) \neq (vk, t_c)$. Recall that, for the same query (ϕ, π'), if the simulator responds to $\mathcal{A}$ a result not being the special symbol $\perp$ in Game_5,

then the simulator must return the same result as in Game_4. So, unless event F occurs, the two games are identical from the adversary's point of view. By the difference lemma [26, Lemma 1], it follows that $|\Pr[S_5 = 1] - \Pr[S_4 = 1]| \leq \Pr[F]$.
We show the upper bound of the probability $\Pr[F]$ by the following observation

$$\Pr[F] = \Pr[F \wedge E_{\mathsf{ninj}}] + \Pr[F \wedge \overline{E_{\mathsf{ninj}}}] \leq \Pr[E_{\mathsf{ninj}}] + \Pr[F|\overline{E_{\mathsf{ninj}}}]$$

where all probabilities are taken over the randomness used in the experiment in Game_4. The following two lemmas show that both the probabilities $\Pr[E_{\mathsf{ninj}}]$ and $\Pr[F|\overline{E_{\mathsf{ninj}}}]$ are negligible in κ. We postpone to prove them after the main proof.

Lemma 4. *Suppose that $\mathcal{A}$ makes at most $Q(\kappa)$ queries. Then*

$$\Pr[E_{\mathsf{ninj}}] \leq Q(\kappa) \cdot \mathsf{Adv}^{\mathrm{eva}}_{\mathsf{LF},\mathcal{B}''}(\kappa)$$

for some suitable adversary $\mathcal{B}''$ attacking on the evasiveness of OT-LF.

Lemma 5. *Suppose that $\mathcal{A}$ makes at most $Q(\kappa)$ queries. For any $\delta > 0$, we have*

$$\Pr[F|\overline{E_{\mathsf{ninj}}}] \leq Q(\kappa) \cdot \left(\delta + \frac{d \cdot 2^{m+\ell_{\mathsf{LF}}+\log 1/\delta}}{|\mathcal{S}| - Q(\kappa) + 1}\right).$$

Game_6 **(Replacing $h(s)$ by a random string):** This game is the same as Game_5, except that the simulator samples a random string $r \leftarrow_R \{0,1\}^m$ instead of computing $r = h(s)$. Recall that in both Game_5 and Game_6, except r, the simulator never uses the KDF derivation key s to answer $\mathcal{A}$'s queries. So, the adversary does not learn any more information on s through the key derivation oracle $\mathsf{KDF}_{\pi'}(\phi(s))$. Observe that from the adversary's point of view, only the value y may reveal information on s and all other values are independent of s (e.g., $\mathrm{PP}_{\mathsf{KDF}}$ and (vk, t_c)) or are just functions of y (e.g., σ). It holds by the lossiness property of the OT-LF and by Lemma 1 that

$$\widetilde{H}_\infty(s|(\mathrm{PP}_{\mathsf{KDF}}, \pi)) \geq \widetilde{H}_\infty(s|\mathrm{PP}_{\mathsf{KDF}}) - \ell_{\mathsf{LF}} = \log|\mathcal{S}| - \ell_{\mathsf{LF}}.$$

Since $\log|\mathcal{S}| - \ell_{\mathsf{LF}} - 2\log(1/\epsilon_{\mathcal{H}}) \geq m$, by Lemma 2, we have that $h(s)$ is $\epsilon_{\mathcal{H}}$-close to uniform over $\{0,1\}^m$ from $\mathcal{A}$'s point of view. Hence,

$$|\Pr[S_6 = 1] - \Pr[S_5 = 1]| \leq \epsilon_{\mathcal{H}}.$$

Game_7 **(Reversing to answer all queries with the KDF key):** This game is the same as in Game 6, except that the simulator samples $\mathrm{PP}_{\mathsf{KDF}}$ and $s||\pi$, and answers queries (ϕ, π') as in Game_0. Note that, in this game, r is still sampled as in Game_6. Through defining a sequence of reverse games from Game_6 to Game_0, we can prove that

$$|\Pr[S_7 = 1] - \Pr[S_6 = 1]| \leq |\Pr[S_6 = 1] - \Pr[S_0 = 1]|.$$

Observe that, Game_7 is just the simulated experiment $\mathsf{Sim}_{\mathsf{KDF}}(\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}, \kappa)$ and hence

$$\Pr[\mathcal{A}(\mathsf{Sim}_{\mathsf{KDF}}(\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}, \kappa)) = 1] = \Pr[S_7 = 1].$$

Taking all together, Theorem 1 follows. □

Now, we prove Lemma 4 and Lemma 5.

Proof (Proof of Lemma 4). Given a challenge LF evaluation key ek_{LF}, $\mathcal{B}''$ simulates $\mathcal{A}$'s environment in Game_4 as follows. It first picks $\mathrm{PP}_{\mathsf{OTS}} \leftarrow \mathsf{OTS.Sys}(1^\kappa)$, $h \leftarrow_R \mathcal{H}$ and $s \leftarrow_R \mathcal{S}$. It then samples a OTS key pair $(vk, sigk) \leftarrow \mathsf{OTS.Gen}(\mathrm{PP}_{\mathsf{OTS}})$. After that, $\mathcal{B}''$ queries $\mathsf{LF.LTag}(ek_{\mathsf{LF}}, \cdot)$ with vk to obtain the challenge core tag part i.e., $t_c = \mathsf{LF.LTag}(td_{\mathsf{LF}}, vk)$. Next, $\mathcal{B}''$ computes $y = \mathsf{LF}_{ek_{\mathsf{LF}},(vk,t_c)}(s)$ and $\sigma = \mathsf{OTS.Sig}(sigk, t_c||y)$. $\mathcal{B}''$ sends $\mathrm{PP}_{\mathsf{KDF}} = (ek_{\mathsf{LF}}, \mathrm{PP}_{\mathsf{OTS}}, h)$ and $\pi = (vk, t_c)||y||\sigma$ to the adversary $\mathcal{A}$. Since $\mathcal{B}''$ knows the KDF key s, he can answer all the queries as in Game_4. Let $T = \{(vk', t_c')\}$ be the set of tags extracted from $\mathcal{A}$'s queries (ϕ, π') such that $(vk', t_c') \neq (vk, t_c)$. Finally, $\mathcal{B}''$ chooses a tag (vk', t_c') from T uniformly at random as his output. If E_{ninj} occurs, with probability at least $1/Q(\kappa)$, $\mathcal{B}''$ outputs a fresh non-injective tag. Hence, $\Pr[E_{\mathsf{ninj}}] \leq Q(\kappa) \cdot \mathsf{Adv}^{\mathrm{eva}}_{\mathsf{LF},\mathcal{B}''}(\kappa)$. □

Proof (Proof of Lemma 5). Let (ϕ, π') be the first query that does not satisfy the key derivation rules of R1 and R2 in Game_4 and event E_{ninj} does not happen. We call such query invalid query. Recall that an invalid query is always rejected (output $\perp$) in Game_5. We show that it is not rejected in Game_4 with a negligible probability. Clearly, if $(t'_c||y', \sigma')$ is an invalid signature, then (ϕ, π') will be rejected in both Game_4 and Game_5. We consider three cases:

- Case 0: $\pi' = \pi$ and $\phi(s) \neq s$.
- Case 1: $\pi' = \pi$, $\phi \neq \mathsf{id}$, but $\phi(s) = s$.
- Case 2: $vk' \neq vk$ and $\phi \notin \mathsf{cf}$.

Note that, for any query (ϕ, π'), it always satisfies the key derivation rules defined in either R1 or R2, except for the above three cases. Recall that, in the first case, both Game_4 and Game_5 outputs $\perp$. Hence, only the Case 1 and Case 2 may cause the difference between Game_4 and Game_5. Next, we show that the last two cases will be rejected in Game_4 with overwhelming probability.

Observe that in Game_4, only values r and y may contain information on the KDF derivation key s. The other values are independent of s (e.g., $\mathrm{PP}_{\mathsf{KDF}}$ and vk) or just functions of y (e.g., σ). Denote by V the adversary's view in Game_4. From Lemma 1 and the fact that r and y have at most 2^m and $2^{\ell_{\mathsf{LF}}}$ possible values respectively, we have

$$\widetilde{H}_\infty(s|V) = \widetilde{H}_\infty(s|(\mathrm{PP}_{\mathsf{KDF}}, r||\pi)) \geq \widetilde{H}_\infty(s|\mathrm{PP}_{\mathsf{KDF}}) - m - \ell_{\mathsf{LF}}.$$

Recall that s is independent of $\mathrm{PP}_{\mathsf{KDF}}$. So, the average min-entropy of s conditioned on the adversary's point of view is at least $\log|\mathcal{S}| - m - \ell_{\mathsf{LF}}$. According to Lemma 1, for any $\delta > 0$, with probability at least $1 - \delta$,

$$H_\infty(s|V = v) \geq \widetilde{H}_\infty(s|V) - \log 1/\delta \geq \log|\mathcal{S}| - m - \ell_{\mathsf{LF}} - \log 1/\delta$$

over the choice of $V = v$.

According to Lemma 3, for any $\phi \neq \mathsf{id}$, we have

$$\Pr[\phi(s) = s] \leq \frac{d}{2^{H_\infty(s|V=v)}}.$$

So, in Case 1, with probability at least $1 - \delta$,

$$\Pr[\phi(s) = s] \leq \frac{d \cdot 2^{m+\ell_{\mathsf{LF}}+\log 1/\delta}}{|\mathcal{S}|}.$$

Again, according to Lemma 3, for any $\phi \notin \mathsf{cf}$

$$H_\infty(\phi(s)|V = v) \geq H_\infty(s|V = v) - \log d \geq \log|\mathcal{S}| - m - \ell_{\mathsf{LF}} - \log 1/\delta - \log d$$

with probability at least $1 - \delta$.

Recall that event E_{ninj} does not happen, so (vk', t'_c) is an injective tag, which means that $\mathsf{LF}_{ek_{\mathsf{LF}},(vk',t'_c)}(\cdot)$ is injective. As a result, the adversary can correctly guess the value $\mathsf{LF}_{ek_{\mathsf{LF}},(vk',t'_c)}(\phi(s))$ with probability at most $\delta + d \cdot 2^{m+\ell_{\mathsf{LF}}+\log 1/\delta}/|\mathcal{S}|$. Therefore, the first invalid query passes the key derivation rules in Game_4 with probability at most $\delta + d \cdot 2^{m+\ell_{\mathsf{LF}}+\log 1/\delta}/|\mathcal{S}|$.

An almost identical argument holds for all subsequent invalid queries. The only difference is that the adversary can rule out one more value s from each rejection of invalid query. So, R3 or R0 accepts the i-th invalid query with probability at most $\delta + d \cdot 2^{m+\ell_{\mathsf{LF}}+\log 1/\delta}/(|\mathcal{S}| - i + 1)$. Since $\mathcal{A}$ makes at most $Q(\kappa)$ queries, the event $F|\overline{E_{\mathsf{ninj}}}$ occurs with probability at most

$$Q(\kappa) \cdot \left(\delta + \frac{d \cdot 2^{m+\ell_{\mathsf{LF}}+\log 1/\delta}}{|\mathcal{S}| - Q(\kappa) + 1}\right).$$

This finishes the proof of Lemma 5. □

4.2 Instantiations

According to [24,25], OT-LFs can be constructed from standard assumptions including the DDH assumption and the DCR assumption. This results in instantiations of $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$-cnm-KDF (w.r.t. $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}}$-cnm-KDF) based on these standard assumptions.

5 Application to RKA-secure IBE

An identity-based encryption scheme IBE consists of five (PPT) algorithms: (1) $\mathsf{IBE.Sys}(1^\kappa)$, on input 1^κ, outputs a system parameter PP, which defines an identity space $\mathcal{ID}$. (2) $\mathsf{IBE.Gen}(\mathrm{PP})$, on input PP, outputs a master public key mpk and a master secret key msk. (3) $\mathsf{IBE.Ext}(msk, id)$, on input msk and an identity $id \in \mathcal{ID}$, outputs a decryption key dk_{id}. (4) $\mathsf{IBE.Enc}(mpk, id, M)$, on input a message M, outputs a ciphertext C encrypted under mpk and identity id. (5) The deterministic algorithm $\mathsf{IBE.Dec}(dk_{id}, C)$, on input decryption key dk_{id} and ciphertext C, outputs a message M. Correctness requires that for all public parameter $\mathrm{PP} \leftarrow \mathsf{IBE.Sys}(1^\kappa)$, all master public/secret key pair $(mpk, msk) \leftarrow \mathsf{IBE.Gen}(\mathrm{PP})$, all identity id and message M, it always has $\mathsf{IBE.Dec}(dk_{id}, \mathsf{IBE.Enc}(mpk, id, M)) = M$.

RKA-SECURE IBE. We recall the Φ-RKA security of IBE schemes from [4]. In the context of IBE, an RKA adversary is allowed to access a decryption key generation oracle: $\mathcal{O}_{msk}^{\Phi}(\cdot,\cdot)$, on input $(\phi, id) \in \Phi \times \mathcal{ID}$, it returns $\mathsf{IBE.Ext}(\phi(msk), id)$. Besides this, the oracle initializes an empty set $I := \emptyset$ and $id^* = \perp$. For an RKA query (ϕ, id), if $\phi(msk) = msk$ [5], it adds id to the set $I := I \cup \{id\}$, and if id equals the challenge identity id^*, it returns $\perp$ directly. An IBE scheme is Φ-RKA secure, if for any PPT adversary $\mathcal{A}$, the following advantage

$$\mathsf{Adv}^{\mathrm{rka}}_{\mathsf{IBE},\mathcal{A}}(\kappa) := \left| \Pr\left[b' = b : \begin{array}{l} \mathrm{PP} \leftarrow \mathsf{IBE.Sys}(1^\kappa) \\ (mpk, msk) \leftarrow \mathsf{IBE.Gen}(\mathrm{PP}) \\ (M_0, M_1, id^*, St) \leftarrow \mathcal{A}^{\mathcal{O}^{\Phi}_{msk}(\cdot,\cdot)}(\mathrm{PP}, mpk) \\ b \leftarrow_R \{0,1\}, C \leftarrow \mathsf{IBE.Enc}(mpk, id^*, M_b) \\ b' \leftarrow \mathcal{A}^{\mathcal{O}^{\Phi}_{msk}(\cdot,\cdot)}(St, C) \end{array} \right] - \frac{1}{2} \right|$$

is negligible in κ, where M_0 and M_1 are two equal length messages. Clearly, if Φ only contains the identity function id, then the above definition is just the traditional CPA-security of IBE schemes [9].

Suppose that $\mathsf{IBE.Gen}(\mathrm{PP})$ utilizes an m-bit random string as the internal coin for generating mpk and msk. We write r explicitly in the key generation algorithm, i.e., $\mathsf{IBE.Gen}(\mathrm{PP}; r) = (mpk, msk)$ (a deterministic algorithm w.r.t. input (PP, r)).

The IBE Construction. Starting from a $(\Phi, \epsilon_{\mathsf{KDF}})$-continuous non-malleable KDF ($\mathsf{KDF.Sys}$, $\mathsf{KDF.Sample}$, KDF) and a CPA-secure IBE scheme ($\mathsf{IBE.Sys}$, $\mathsf{IBE.Gen}$, $\mathsf{IBE.Ext}$, $\mathsf{IBE.Enc}$, $\mathsf{IBE.Dec}$), we construct a new IBE scheme ($\overline{\mathsf{IBE}}.\mathsf{Sys}$, $\overline{\mathsf{IBE}}.\mathsf{Gen}$, $\overline{\mathsf{IBE}}.\mathsf{Ext}$, $\overline{\mathsf{IBE}}.\mathsf{Enc}$, $\overline{\mathsf{IBE}}.\mathsf{Dec}$) as follows:

- $\overline{\mathsf{IBE}}.\mathsf{Sys}(1^\kappa)$: It runs $\mathrm{PP}_{\mathsf{KDF}} \leftarrow \mathsf{KDF.Sys}(1^\kappa)$ and $\mathrm{PP}_{\mathsf{IBE}} \leftarrow \mathsf{IBE.Sys}(1^\kappa)$, and returns $\mathrm{PP}_{\overline{\mathsf{IBE}}} = (\mathrm{PP}_{\mathsf{KDF}}, \mathrm{PP}_{\mathsf{IBE}})$.
- $\overline{\mathsf{IBE}}.\mathsf{Gen}(\mathrm{PP}_{\overline{\mathsf{IBE}}})$: It samples $s||\pi \leftarrow \mathsf{KDF.Sample}(\mathrm{PP}_{\mathsf{KDF}})$ and computes $r = \mathsf{KDF}_\pi(s)$. Then, it computes $(mpk, msk) = \mathsf{IBE.Gen}(\mathrm{PP}_{\mathsf{IBE}}; r)$ and returns master public key $\overline{mpk} = (mpk, \pi)$ and secret key $\overline{msk} = (s, \pi)$.
- $\overline{\mathsf{IBE}}.\mathsf{Ext}(\overline{msk}, id)$: For $\overline{msk} = (s, \pi)$, it computes $r = \mathsf{KDF}_\pi(s)$. If r is the special symbol $\perp$, it returns $\perp$ and halts. Otherwise, it computes $(mpk, msk) = \mathsf{IBE.Gen}(\mathrm{PP}_{\mathsf{IBE}}; r)$ and returns $\overline{dk}_{id} = \mathsf{IBE.Ext}(msk, id)$.
- $\overline{\mathsf{IBE}}.\mathsf{Enc}(\overline{mpk}, id, M)$: It first parses $\overline{mpk}$ as (mpk, π) and then returns $\overline{C} = \mathsf{IBE.Enc}(mpk, id, M)$.
- $\overline{\mathsf{IBE}}.\mathsf{Dec}(\overline{dk}_{id}, \overline{C})$: It returns $\mathsf{IBE.Dec}(\overline{dk}_{id}, \overline{C})$.

Theorem 2. *If* KDF *is* $(\Phi, \epsilon_{\mathsf{KDF}})$*-continuously non-malleable and* IBE *is CPA-secure, then the above construction is a* Φ*-RKA secure IBE scheme. Concretely, for any PPT adversary* $\mathcal{A}$*, there exist KDF distinguisher* $\mathcal{D}$ *and adversary* $\mathcal{B}$ *of roughly the same complexity as* $\mathcal{A}$ *such that*

$$\mathsf{Adv}^{\mathrm{rka}}_{\overline{\mathsf{IBE}},\mathcal{A}}(\kappa) \leq \epsilon_{\mathsf{KDF}} + \mathsf{Adv}^{\mathrm{cpa}}_{\mathsf{IBE},\mathcal{B}}(\kappa).$$

[5] If msk contains some public information, for example in our construction $msk = (s, \pi)$ where π is completely given to an adversary, we define $\phi(msk) = (\phi(s), \pi')$ and π' is implicitly defined in the adversary's query (ϕ, id).

Proof. We prove it through two games: Game_0 and Game_1. The former is just the original experiment of RKA-security and the later is slightly different from the former in which the internal coin r is replaced by a uniform random string. We depict the difference between these two games in Fig. 4.

	In Game_0:	In Game_1:
Master public key	$s\|\|\pi \leftarrow \mathsf{KDF.Sample}(\mathrm{PP}_{\mathsf{KDF}})$	$s\|\|\pi \leftarrow \mathsf{KDF.Sample}(\mathrm{PP}_{\mathsf{KDF}})$
	$r = \mathsf{KDF}_\pi(s)$	$\boxed{r \leftarrow_R \{0,1\}^m}$
	$(mpk, msk) = \mathsf{IBE.Gen}(\mathrm{PP}_{\mathsf{IBE}}; r)$	$(mpk, msk) = \mathsf{IBE.Gen}(\mathrm{PP}_{\mathsf{IBE}}; r)$
	Return $\overline{mpk} = (mpk, \pi)$.	Return $\overline{mpk} = (mpk, \pi)$.
Dec. key oracle Input: (ϕ, id)	If $\phi(s)\|\|\pi' = s\|\|\pi$, set $I := I \cup \{id\}$.	If $\phi(s)\|\|\pi' = s\|\|\pi$, set $I := I \cup \{id\}$ and return $\boxed{\mathsf{IBE.Ext}(msk, id)}$. Else compute
	$r' = \mathsf{KDF}_{\pi'}(\phi(s))$	$r' = \mathsf{KDF}_{\pi'}(\phi(s))$
	If $r' = \perp$, return $\perp$. Else, compute $(mpk', msk') = \mathsf{IBE.Gen}(\mathrm{PP}_{\mathsf{IBE}}; r')$	If $r' = \perp$, return $\perp$. Else, compute $(mpk', msk') = \mathsf{IBE.Gen}(\mathrm{PP}_{\mathsf{IBE}}; r')$
	Return $\overline{dk}_{id} \leftarrow \mathsf{IBE.Ext}(msk', id)$.	Return $\overline{dk}_{id} \leftarrow \mathsf{IBE.Ext}(msk', id)$.

Fig. 4. Differences between Game_0 and Game_1

Denote by S_0 and S_1 the event that $\mathcal{A}$ successfully guesses the random coin b in Game_0 and Game_1 respectively. We show shortly that

$$|\Pr[S_0] - \Pr[S_1]| \leq \epsilon_{\mathsf{KDF}} \tag{3}$$

$$|\Pr[S_1] - 1/2| \leq \mathsf{Adv}^{\mathrm{cpa}}_{\mathsf{IBE},\mathcal{B}}(\kappa). \tag{4}$$

Clearly,

$$\mathsf{Adv}^{\mathrm{rka}}_{\overline{\mathsf{IBE}},\mathcal{A}}(\kappa) = |\Pr[S_0] - 1/2|.$$

This completes the proof of Theorem 2. □

Proof (Proof of Eq. (3)). Given $(\mathrm{PP}_{\mathsf{KDF}}, r, \pi)$ where r either equals $\mathsf{KDF}_\pi(s)$ or a uniform random string, the simulator chooses $\mathrm{PP}_{\mathsf{IBE}}$ and computes $(mpk, msk) = \mathsf{IBE.Gen}(\mathrm{PP}_{\mathsf{IBE}}; r)$. It sends $\overline{mpk} = (mpk, \pi)$ to the adversary and keeps the secret key msk. The simulator answers $\mathcal{A}$'s decryption key queries (ϕ, id) as follows: It sends (ϕ, π') to the KDF oracle and obtains the value r'. If $r' = \mathsf{same}^\star$, the simulator returns $\mathsf{IBE.Ext}(msk, id)$ to $\mathcal{A}$ and updates $I := I \cup \{id\}$. If $r' = \perp$, the simulator returns $\perp$. Otherwise, the simulator computes $(mpk', msk') = \mathsf{IBE.Gen}(\mathrm{PP}_{\mathsf{IBE}}; r')$ and returns $\mathsf{IBE.Ext}(msk', id)$ to $\mathcal{A}$. After the phase of decryption key queries, $\mathcal{A}$ submits two equal-length messages (M_0, M_1) and a challenge identity id^*. The simulator picks $b \leftarrow_R \{0,1\}$ and returns $\overline{C} = \mathsf{IBE.Enc}(mpk, id^*, M_b)$ to $\mathcal{A}$. Finally, the simulator outputs what $\mathcal{A}$ outputs. Recall that, the symbol $\mathsf{same}^\star$ implies $\phi(s)||\pi' = s||\pi$. So, if $r = \mathsf{KDF}_\pi(s)$, the simulator perfectly simulates Game_0. While if r is a uniform string, the simulator simulates Game_1. This completes the proof of Eq. (3). □

Proof (Proof of Eq. (4)). Given an IBE challenge instance $(\mathrm{PP}_{\mathsf{IBE}}, mpk)$, the simulator samples $\mathrm{PP}_{\overline{\mathsf{KDF}}} \leftarrow \mathsf{KDF.Sys}(1^\kappa)$ and sets $\mathrm{PP}_{\overline{\mathsf{IBE}}} = (\mathrm{PP}_{\mathsf{KDF}}, \mathrm{PP}_{\mathsf{IBE}})$. It also samples $s||\pi \leftarrow \mathsf{KDF.Sample}(\mathrm{PP}_{\mathsf{KDF}})$ and sets $\overline{mpk} = (mpk, \pi)$. Then it sends $(\mathrm{PP}_{\overline{\mathsf{IBE}}}, \overline{mpk})$ to $\mathcal{A}$. To answer $\mathcal{A}$'s decryption key queries (ϕ, id), the simulator first checks whether $\phi(s)||\pi' = s||\pi$. If so, it submits id to its own decryption key generation oracle and forwards the result to $\mathcal{A}$. Since the simulator knows s and it can handle the case $\phi(s)||\pi' \neq s||\pi$ as in Game_1. When $\mathcal{A}$ queries the challenge ciphertext, the simulator forwards (M_0, M_1, id^*) to its own encryption oracle to obtain a challenge ciphertext C. The simulator forwards C to the adversary. Finally, the simulator outputs what $\mathcal{A}$ outputs. Clearly, the simulator perfectly simulates $\mathcal{A}$'s environment in Game_1. If $\mathcal{A}$ succeeds, so does the simulator. This completes the proof of Eq. (4). □

From [27], we have a CPA-secure IBE scheme under the standard DBDH assumption. Subsection 4.2 suggests that $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$-continuously non-malleable KDFs can be constructed from the DDH and DCR assumptions. Consequently, our IBE construction above immediately results in the first IBE that is RKA-secure for class $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$, i.e., the sets of all polynomial functions of bounded degree, under the standard DBDH assumption, and the security follows from Theorem 1 and Theorem 2. We stress that the degree of our RKD polynomial functions is not limited to polynomial size in κ and we can always enlarge the polynomial function class $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$ to class $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}}$ whose functions has high output entropy and input-output collision resistance, as defined in Definition 1. As a result, the $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$-RKA security of IBE can be extended to $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}}$, with $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}} \supseteq \Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$.

Extensions to PKE and Signature. Bellare et al. [6] showed that the CHK [10] IBE-to-CCA-PKE transform and the Naor IBE-to-Sig transform both preserve Φ-RKA security. Thus, we readily obtain $\Phi_{\mathbb{F}}^{\mathsf{poly}(d)}$(also extended to $\Phi_{\mathsf{iocr}}^{\mathsf{hoe}}$)-RKA-secure CCA-PKE and signature schemes under standard assumptions.

On the other hand, the continuous non-malleable KDFs can also be directly used to transform a cryptographic primitive to a RKA secure version in a modular way, as long as the key generation algorithm of the primitive takes uniform random coins r to generate (secret/public) keys. The transformation with the help of cnm-KDF is as follows. First, sample a random derivation key s together with the public key π such that $\mathsf{KDF}_\pi(s) = r$; Then, store s in the cryptographic hardware device. In addition, we append the proof π of s to the public key of the system. When using r, we retrieve it via computing $\mathsf{KDF}_\pi(s)$. By the property of continuous non-malleability, if s is modified to $\phi(s) \neq s$ and π to π', then $r' = \mathsf{KDF}_{\pi'}(\phi(s))$ is either the rejection symbol $\perp$ or a value independent of r. Finally, the Φ-RKA security is reduced to the original security of the primitive.

Acknowledgments. Baodong Qin and Shengli Liu were supported by the National Natural Science Foundation of China (Grant No. 61170229 and 61373153), the Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20110073110016), and the Scientific innovation projects of Shanghai Education Committee (Grant No. 12ZZ021). Kefei Chen was supported by the National Natural Science Foundation of China (Grant No. 61133014). The authors would also like to thank anonymous reviewers for very useful comments and suggestions on a preliminary version of this paper. Special thanks go to the reviewer who pointed out that our result holds for the extended RKD function class defined in Definition 1.

References

1. Abdalla, M., Benhamouda, F., Passelègue, A., Paterson, K.G.: Related-key security for pseudorandom functions beyond the linear barrier. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014, Part I. LNCS, vol. 8616, pp. 77–94. Springer, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-662-44371-2_5 4
2. Applebaum, B., Harnik, D., Ishai, Y.: Semantic security under related-key attacks and applications. In: Chazelle, B. (ed.) Innovations in Computer Science - ICS 2010. pp. 45–60. Tsinghua University Press (2011), http://conference.itcs.tsinghua.edu.cn/ICS2011/content/papers/30.html 4
3. Bellare, M., Cash, D.: Pseudorandom functions and permutations provably secure against related-key attacks. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 666–684. Springer, Heidelberg (2010), http://dx.doi.org/10.1007/978-3-642-14623-7_36 4
4. Bellare, M., Cash, D., Miller, R.: Cryptography secure against related-key attacks and tampering. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 486–503. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-25385-0_26 2, 4, 13
5. Bellare, M., Kohno, T.: A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 491–506. Springer, Heidelberg (2003), http://dx.doi.org/10.1007/3-540-39200-9_31 1
6. Bellare, M., Paterson, K.G., Thomson, S.: RKA security beyond the linear barrier: Ibe, encryption and signatures. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 331–348. Springer, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-34961-4_21 2, 4, 15

7. Biham, E., Shamir, A.: Differential fault analysis of secret key cryptosystems. In: Jr., B.S.K. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 513–525. Springer, Heidelberg (1997), http://dx.doi.org/10.1007/BFb0052259 1
8. Boneh, D., DeMillo, R.A., Lipton, R.J.: On the importance of checking cryptographic protocols for faults (extended abstract). In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 37–51. Springer, Heidelberg (1997), http://dx.doi.org/10.1007/3-540-69053-0_4 1
9. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001), http://dx.doi.org/10.1007/3-540-44647-8_13 13
10. Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004), http://dx.doi.org/10.1007/978-3-540-24676-3_13 15
11. Cramer, R., Dodis, Y., Fehr, S., Padró, C., Wichs, D.: Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 471–488. Springer, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-78967-3_27 4
12. Damgård, I., Faust, S., Mukherjee, P., Venturi, D.: Bounded tamper resilience: How to go beyond the algebraic barrier. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 140–160. Springer, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-42045-0_8 5
13. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. 38(1), 97–139 (2008), http://dx.doi.org/10.1137/060651380 4, 5
14. Dziembowski, S., Pietrzak, K.: Leakage-resilient cryptography. In: FOCS 2008. pp. 293–302. IEEE Computer Society (2008), http://doi.ieeecomputersociety.org/10.1109/FOCS.2008.56 2, 3
15. Dziembowski, S., Pietrzak, K., Wichs, D.: Non-malleable codes. In: Yao, A.C. (ed.) Innovations in Computer Science - ICS 2010. pp. 434–452. Tsinghua University Press (2010), http://conference.itcs.tsinghua.edu.cn/ICS2010/content/papers/34.html 2, 4
16. Faust, S., Mukherjee, P., Nielsen, J.B., Venturi, D.: Continuous non-malleable codes. In: Lindell, Y. (ed.) TCC 2014. LNCS, vol. 8349, pp. 465–488. Springer (2014), http://dx.doi.org/10.1007/978-3-642-54242-8_20 2, 3, 4, 5
17. Faust, S., Mukherjee, P., Venturi, D., Wichs, D.: Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In: Nguyen, P.Q., Oswald, E. (eds.) EUROCRYPT 2014. LNCS, vol. 8441, pp. 111–128. Springer, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-55220-5_7 2, 3, 4, 7
18. Gennaro, R., Lysyanskaya, A., Malkin, T., Micali, S., Rabin, T.: Algorithmic tamper-proof (ATP) security: Theoretical foundations for security against hardware tampering. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 258–277. Springer (2004), http://dx.doi.org/10.1007/978-3-540-24638-1_15 3
19. Goyal, V., O'Neill, A., Rao, V.: Correlated-input secure hash functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-19571-6_12 4
20. Jafargholi, Z., Wichs, D.: Tamper detection and continuous non-malleable codes. Cryptology ePrint Archive, Report 2014/956 (2014), http://eprint.iacr.org/ 5
21. Kocher, P.C.: Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996), http://dx.doi.org/10.1007/3-540-68697-5_9 1
22. Kocher, P.C., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M.J. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999), http://dx.doi.org/10.1007/3-540-48405-1_25 1
23. Lu, X., Li, B., Jia, D.: Related-key security for hybrid encryption. In: Chow, S.S.M., Camenisch, J., Hui, L.C.K., Yiu, S. (eds.) ISC 2014. LNCS, vol. 8783, pp. 19–32. Springer (2014), http://dx.doi.org/10.1007/978-3-319-13257-0_2 4
24. Qin, B., Liu, S.: Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part II. LNCS, vol. 8270, pp. 381–400. Springer, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-42045-0_20 3, 4, 6, 13
25. Qin, B., Liu, S.: Leakage-flexible cca-secure public-key encryption: Simple construction and free of pairing. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 19–36. Springer, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-54631-0_2 3, 13
26. Shoup, V.: Sequences of games: a tool for taming complexity in security proofs. IACR Cryptology ePrint Archive 2004, 332 (2004), http://eprint.iacr.org/2004/332 11
27. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005), http://dx.doi.org/10.1007/11426639_7 15
28. Wee, H.: Public key encryption against related key attacks. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 262–279. Springer, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-30057-8_16 4