Non-Malleable Functions and Their Applications

Yu Chen *
yuchen.prc@gmail.com

Baodong Qin †
baodong.qin@gmail.com

Jiang Zhang ‡
jiangzhang09@gmail.com

Yi Deng §
ydeng.cas@gmail.com

Sherman S.M. Chow ¶
sherman.chow@gmail.com

Abstract

We formally study "non-malleable functions" (NMFs), a general cryptographic primitive which simplifies and relaxes "non-malleable one-way/hash functions" (NMOWHFs) introduced by Boldyreva et al. (ASIACRYPT 2009) and refined by Baecher et al. (CT-RSA 2010). NMFs consider deterministic functions, rather than probabilistic one-way/hash functions in the literature of NMOWHFs.

We mainly follow Baecher et al. to formalize a game-based definition. Roughly, a function f is non-malleable if given an image $y^* \leftarrow f(x^*)$ for a randomly chosen x^*, it is hard to output a mauled image y with a ϕ from some transformation class s.t. $y = f(\phi(x^*))$. A distinctive strengthening of our non-malleable notion is that $\phi(x^*) = x^*$ is always allowed. We also consider non-malleability in adaptive setting, which stipulates non-malleability maintains even when an inversion oracle is available.

We investigate the relations between non-malleability and one-wayness in depth. In the non-adaptive setting, we show that for any achievable transformation class, non-malleability implies one-wayness for poly-to-one functions but not vise versa. In the adaptive setting, we show that for most algebra-induced transformation class, adaptive non-malleability (ANM) is equivalent to adaptive one-wayness (AOW) for injective functions. These two results establish interesting theoretical connections between non-malleability and one-wayness for functions, which extend to trapdoor functions as well, and thus resolve some open problems left by Kiltz et al. (EUROCRYPT 2010). Notably, the implication AOW $\Rightarrow$ ANM not only yields constructions of NMFs from adaptive trapdoor functions, which partially solves an open problem posed by Boldyreva et al. (ASIACRYPT 2009), but also provides key conceptual insight into addressing copy attacks in the context of related-key attacks (RKA).

Finally, we show that NMFs lead to a simple black-box construction of continuous non-malleable key derivation functions recently proposed by Qin et al. (PKC 2015), which have proven to be very useful in achieving RKA-security for numerous cryptographic primitives.

Keywords: non-malleable functions, one-way functions, algebra-induced transformations, related-key attacks, copy attacks, key derivation

*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences & Department of Information Engineering, the Chinese University of Hong Kong.

†School of Computer Science and Technology, Southwest University of Science and Technology, China.

‡State Key Laboratory of Cryptology, Beijing, China.

§State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences.

¶Department of Information Engineering, the Chinese University of Hong Kong.

1 Introduction

Non-malleability is an important notion for cryptographic primitives which ensures some level of independence of outputs with respect to related inputs. This notion, first treated formally in the seminal work [DDN00], has been studied extensively for many randomized primitives, such as commitments [CIO98, FF00, CKOS01, PR05], encryptions [BS99], zero-knowledge proofs [Sah99, OPV08, LPTV10], obfuscations [CV09], and codes [DPW10, FMVW14, FMNV14]. However, little attention has been paid on deterministic primitives. Particularly, the study dedicated to non-malleability for deterministic functions, which is arguably the most basic primitive, is still open. With the goal to fill this gap, we initiate the study of non-malleability for deterministic functions in this work.

1.1 Related Work

Non-Malleable One-Way and Hash Functions. Boldyreva et al. [BCFW09] initiated the foundational study of non-malleable one-way and hash functions (NMOWHFs).[1] They gave a simulation-based definition of non-malleability, basically saying that for any adversary mauling a function value y^* into a related value y there exists a simulator which does just well even without seeing y^*. They provided a construction of NMOWHFs from perfectly one-way hash functions (POWHF) and simulation-sound non-interactive zero-knowledge proof of knowledge (NIZKPoK). However, they regarded this construction as a feasibility result due to its inefficiency. They also discussed applications of NMOWHFs to partially instantiating random oracles in the Bellare-Rogaway encryption scheme [BR93] and OAEP [BF06], as well as enhancing the security of cryptographic puzzles.

Being aware of several deficiencies in the simulation-based non-malleable notion [BCFW09],[2] Baecher et al. [BFS11] reverted the core idea behind non-malleability and proposed a game-based definition which is more handy to work with. Their definition avoids simulator completely and rather asks: given a function value $y^* \leftarrow f(x^*)$ of an unknown preimage x^*, no probabilistic polynomial time (PPT) adversary is able to output a mauled image y together with a transformation ϕ from a prefixed transformation class Φ such that $y = f(\phi(x^*))$. To demonstrate the usefulness of their game-based definition, they proved that the strengthened Merkle-Damgård transformation satisfies their non-malleable notion w.r.t. bit flips, and their non-malleable notion suffices for improving security of the Bellare-Rogaway encryption scheme.

We identify the following gaps in the NMOWHFs literature [BCFW09, BFS11].

- Both [BCFW09] and [BFS11] considered non-malleability for a very general syntax of functions, comprising both classical one-way functions and collision resistant hash functions. In their cases, the underlying functions could be probabilistic and are assumed to be one-way.[3] Despite such treatment is of utmost generality, it is somewhat bulky and even inapplicable for some natural applications, e.g., when the functions are probabilistic, two independent parties computing with the same input will not necessarily get the same output [BCFW09]. Moreover, to some extent, it blurs the relations between non-malleability and one-wayness.
- The game-based non-malleable notion [BFS11] is not strong enough in the sense that the adversary is restricted to output $\phi \in \Phi$ such that $\phi(x^*) \neq x^*$. Note that Φ is introduced to capture all admissible transformations chosen by the adversary, this restriction translates

[1] Historically, Boldyreva et al. [BCFW09] aggregated both one-way functions and hash functions under the term hash functions for simplicity.

[2] See [BFS11] for a detailed discussion on simulation-based non-malleable notion.

[3] Recall that the basic design principle for hash functions is one-wayness.

to the limit that Φ does not contain ϕ that has fixed points, which is undesirable because many widely used transformations (e.g., affine functions and polynomials) are excluded.

- Boldyreva et al.'s construction of NMOWHF is in the standard model, but the uses of POWHF and NIZKPoK render it probabilistic, and inefficient for practical applications [BCFW09] (e.g., cryptographic puzzles for network protocols). The strengthened Merkle-Damgård transformation does constitute an efficient NMOWHF [BFS11], but its non-malleability inherently relies on modeling the compression function as a random oracle [BFS11]. An efficient, deterministic solution in the standard model was left open [BCFW09].
- Though NMOWHFs are powerful, their cryptographic applications are only known for partially instantiating random oracles for some public-key encryption schemes and enhancing the design of cryptographic puzzles. Further applications of NMOWHFs in other areas were expected [BCFW09].

(Adaptive) One-Way Functions. As a fundamental primitive, one-way functions [DH76] and their variants [PPV08, CD08] have been studied extensively. Roughly, one-way functions are a family of deterministic functions where each particular function is easy to compute, but most are hard to invert on average.

Kiltz et al. [KMO10] introduced a strengthening of trapdoor one-way functions called adaptive one-way trapdoor functions (ATDFs), which maintain one-wayness even the adversary is given access to an inversion oracle. They gave a black-box construction of chosen-ciphertext secure public-key encryption (CCA-secure PKE) from ATDFs, and show how to construct ATDFs from either lossy TDFs [PW08] or correlated-product TDFs [RS10]. Their work suggested a number of open problems; in particular, considering non-malleability for TDFs, exploring its relation to existing notions for TDFs and implications for PKE, and realizing them from standard assumptions.

1.2 Motivation

Based on the above discussion, we find that the state of the art of NMOWHFs is not entirely satisfactory. In particular, the study of non-malleability dedicated to deterministic functions and its relation to one-wayness are still open.

In this work, we continue the study of non-malleable primitive but restrict our attention to *deterministic* functions, rather than *probabilistic one-way/hash* functions considered in prior works. Apart from being a natural question which deserves study in its own right, a direct treatment of deterministic functions (without imposing any other cryptographic property) provides three main benefits: first, it shares the same underlying object of "classical" one-way functions and hence allows us to explore the relations between non-malleability and one-wayness; second, this may further lead to efficient constructions of deterministic NMFs in the standard model, by leveraging a vast body of works on one-way functions; third, deterministic primitives are more versatile, making deterministic NMFs more attractive being used a building block for higher-level cryptographic protocols.

In summary, we are motivated to consider the following intriguing questions:

What is the strong yet handy non-malleable notion for deterministic functions? What are the relations between non-malleability and one-wayness? Can we construct efficient deterministic NMFs in the standard model? Are there some new appealing applications of deterministic NMFs?

1.3 Our Contributions

We give positive answers to the above questions, which we sketch as below.

Non-Malleable Functions. In Section 3, we introduce a new cryptographic primitive called deterministic NMFs,[4] which simplifies and relaxes NMOWHFs in that the underlying functions are deterministic and not required to have any cryptographic property. Informally, NMFs stipulate no PPT adversary is able to modify a function value into a meaningfully related one. We mainly follow the game-based approach [BFS11] to define non-malleability for deterministic functions w.r.t. related-preimage deriving transformation[5] (RPDT) class Φ, that is, given $y^* \leftarrow f(x^*)$ for a randomly chosen x^*, no PPT adversary is able to output a $\phi \in \Phi$ and a function value y such that $y = f(\phi(x^*))$.

In our definition, adversary's power is neatly expressed through Φ and there is no other restriction. In particular, $\phi(x^*) = x^*$ is always allowed even when $y = y^*$, whereas existing definition of NMOWHFs [BFS11, Section 3.1] demands $\phi(x^*) \neq x^*$. As we will see in Section 7 and Section 8, this strengthening surfaces as an important property when applying to the area of RKA security. We also introduce adaptive NMFs, which remain non-malleable even an adversary has access to an inversion oracle. This stronger notion is desirable when NMFs are used in more adversarial environment, as we shown in Section 8.4.

Novel Properties of RPDTs. Our non-malleable notion is stronger if Φ is larger. With the purpose of capturing broad yet achievable RPDT class, in Section 4 we introduce two novel properties for RPDT class that we call *bounded root space* (BRS) and *sampleable root space* (SRS). Let id and ϕ_c represent identity transformation and any constant transformation respectively. The two properties demand that for each $\phi \in \Phi$, the root spaces of composite transformations $\phi - \phi_c$ and $\phi - \mathsf{id}$ are polynomially bounded and allow efficient uniform sampling.

BRS and SRS are general enough in that they are met by most algebra-induced transformations considered in the literature, including linear functions, affine functions, and low degree polynomials (with id and ϕ_c being punctured). We let $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$ denote the general RPDT class satisfying the BRS & SRS properties.

Relations Among Non-Malleability and One-Wayness. In Section 5 and Section 6, we investigate the relations among non-malleability and one-wayness in depth. For a pictorial (rough) summary, see Figure 1.

In the non-adaptive setting, we show that w.r.t. any achievable RPDT class Φ, non-malleability (NM) implies one-wayness (OW) for poly-to-one functions (cf. Definition 3.1), but not vise versa. This rigorously confirms the intuition that in common cases NM is strictly stronger than OW. In the adaptive setting, we show that w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$, adaptive non-malleability (ANM) is equivalent to adaptive one-wayness (AOW) for injective functions. While the implication ANM $\Rightarrow$ AOW is obvious, the converse is much more technically involved. In Section 5.3, we prove the implication AOW $\Rightarrow$ ANM via a novel algebraic technique, leveraging the injectivity of the underlying functions and the BRS & SRS properties of $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$. The rough idea of is: if an adversary breaks non-malleability (outputting a mauled image along with a transformation), the reduction can obtain a solvable equation about the preimage and thus contradicts the assumed one-wayness.

All these results indicate that the preimage size is a fundamental parameter of NMFs. We also note that all the above results apply equally well to trapdoor functions. Most importantly,

[4] In the rest of this paper, we will omit the adjective "deterministic" and simply say NMFs when the context is clear.

[5] We use the term transformation to highlight that ϕ has the same domain and range. RPDT is refereed to as admissible transformation in [BFS11].

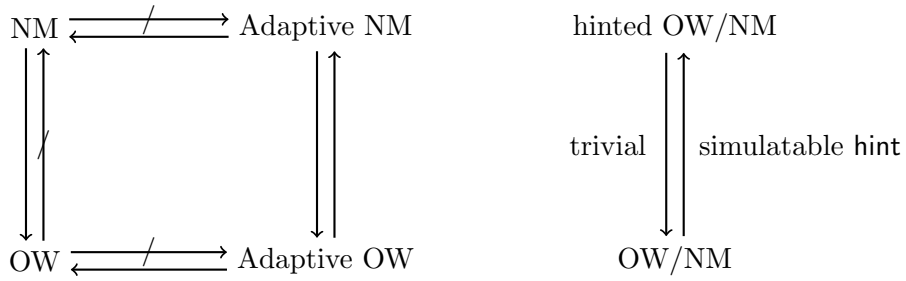

Figure 1: Let unhatched arrows represent implications, and hatched arrows represent separations. The left figure is a rough overview of relations among (adaptive) Φ-non-malleability and (adaptive) one-wayness for deterministic functions. See Section 5 for concrete requirements on Φ and the underlying functions. The right figure depicts the relation between standard one-wayness/non-malleability and hinted one-wayness/non-malleability. See Section 6 for details.

the equivalence AOW $\Leftrightarrow$ ANM answers the aforementioned open problems left by Kiltz et al. [KMO10].

Both OW and NM can be studied with auxiliary information of preimage x^*, which is modeled by a hint function $\mathsf{hint}(x^*)$. We refer to the standard (default) notions without hint as hint-free notions, and refer to the ones with hint as hinted notions. Compared to hint-free notions, hinted ones are generally more useful for cryptographic applications, as we will demonstrate in Section 8. While hinted notions trivially implies hint-free ones, the converse becomes more subtle. In Section 6, we show that w.r.t. statistically/computationally simulatable $\mathsf{hint}(x^*)$, hinted notions are implied by hint-free ones.

Benefits of AOW $\Rightarrow$ ANM. Given the fact that ATDFs are efficiently realizable from a variety of hardness assumptions, the implication AOW $\Rightarrow$ ANM immediately gives rise to efficient deterministic NMFs w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$ in the standard model. This partially resolves an open question raised in [BCFW09].[6] In Appendix A, by using the technique underlying AOW $\Rightarrow$ ANM, we prove that the Merkle-Damgård transformation is actually $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$-non-malleable. This greatly improves prior result [BFS11], and thus provides an efficient candidate of NMFs w.r.t. large RPDT class, though in the random oracle model.

Apart from yielding efficient constructions of NMFs, we find the implication AOW $\Rightarrow$ ANM also has benefit at other place. In Section 7, we discuss how the high-level idea underlying AOW $\Rightarrow$ ANM provides a key insight in the RKA area, that is, resilience against copy attacks w.r.t. most algebra-induced related-key deriving class is in fact a built-in security.

Applications of NMFs. In [BCFW09], Boldyreva et al. showed how to use NMOWHFs to design cryptographic puzzles. We note that poly-to-one NMFs can be plugged into that design instead of NMOWHFs, making it more applicable for practical network protocols.

In Section 8, we revisit continuous non-malleable key derivation functions (KDFs) recently proposed by Qin et al. [QLY+15], which have proven to be useful in achieving RKA-security for numerous cryptographic primitives. The existing construction of continuous non-malleable KDFs is somewhat complicated, which employs one-time lossy filter, one-time signature and pairwise-independent functions as ingredients. We propose an exquisitely simple and elegant construction of continuous non-malleable KDFs based solely on poly-to-one NMFs. Compared

[6] We say "partially" since Boldyreva et al. [BCFW09] posed the question of constructing efficient, deterministic NMFs in the context of their simulation-based definition.

to Qin et al's construction, our construction not only has potential advantages in efficiency, but also admits a direct and modular proof.

1.4 Additional Related Work

Non-malleable codes. Dziembowski, Pietrzak and Wichs [DPW10] introduced the notion of "non-malleable codes" (NMCs) which relaxes the notion of error-correction and error-detection codes. Roughly, NMCs require that given a code $c^* \leftarrow \mathsf{NMC}(m^*)$ for a source-message m^*, the decoded message m of the tampered codeword $c = \phi(c^*)$ is either equal or completely unrelated to m^*. We note that NMFs are somehow dual to NMCs. The duality comes from the fact that NMFs stipulate given $y^* \leftarrow \mathsf{NMF}(x^*)$, $\mathsf{NMF}(\phi(x^*))$ is still hard to compute. Very informally, we can think of in NMCs the tampering takes place on code (could be interpreted as image of message), whereas in NMFs the "tampering" takes place on preimage.

Correlated-input hash functions. Goyal, O'Neill and Rao [GOR11] studied correlated-input hash functions (CIHs), which maintain security when the adversary sees hash values $h(c_i(r))$ of related inputs $c_i(r)$ sharing the same random coins, where c_i is a sequence of circuits chosen by the adversary. In particular, unpredictable CIHs require that no PPT adversary is able to predicate $h(c_{n+1}(r))$ after seeing $h(c_i(r))$ for $i \in [n]$. NMFs can be roughly viewed as a weakening of unpredictable CIHs by restricting $n = 1$ and $c_1 = \mathsf{id}$. Yet, our motivation, definitional framework, as well as techniques are quite different from their work. Until now, instantiation of unpredictable CIHs is only known w.r.t. specific circuit class (tie to scheme algebra), and based on specific number-theoretical assumption.

2 Preliminaries

Basic Notations. For a distribution or random variable X, we write $x \leftarrow X$ to denote the operation of sampling a random x according to X. For a set X, we use $x \xleftarrow{R} X$ to denote the operation of sampling x uniformly at random from X, and use $|X|$ to denote its size. We denote $\lambda \in \mathbb{N}$ as the security parameter. Unless described otherwise, all quantities are implicit functions of λ (we reserve $n(\lambda)$ and $m(\lambda)$ to denote the input length and output length of a function respectively), and all cryptographic algorithms (including the adversary) take λ as an input.

We use standard asymptotic notation O, o, Ω, and ω to denote the growth of functions. We write $\mathsf{poly}(\lambda)$ to denote an unspecified function $f(\lambda) = O(\lambda^c)$ for some constant c. We write $\mathsf{negl}(\lambda)$ to denote some unspecified function $f(\lambda)$ such that $f(\lambda) = o(\lambda^{-c})$ for every constant c. We say that a probability is overwhelming if it is $1 - \mathsf{negl}(\lambda)$, and a probability is noticeable if it is $\Omega(1/\mathsf{poly}(\lambda))$. A probabilistic polynomial time (PPT) algorithm is a randomized algorithm that runs in time $\mathsf{poly}(\lambda)$. If $\mathcal{A}$ is a randomized algorithm, we write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n; r)$ to indicate that $\mathcal{A}$ outputs z on inputs $(x_1, \ldots, x_n)$ and random coins r. We will omit r and write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n)$.

Implications and Separations. Consider security notions A and B for a cryptographic primitive Π, we say

- $A \Rightarrow B$: if all constructions of Π meeting security notion A also meet security notion B.
- $A \not\Rightarrow B$: if there exists a construction of Π which meets security notion A but does not meet security notion B.

Following [BDPR98], we call a result of the first type an *implication*, and a result of the second type a *separation*. If $A \Rightarrow B$, we say A is stronger than B. If we further have $B \not\Rightarrow A$, we say A is strictly stronger than B. If we further have $B \Rightarrow A$, we say A is equivalent to B.

3 One-Way and Non-Malleable Functions

We first recall the general syntax of a family of efficiently computable deterministic functions.

Definition 3.1 (Efficiently Computable Deterministic Functions)**.** A family of efficiently computable functions $\mathcal{F}$ consists of three polynomial time algorithms $(\mathsf{Gen}, \mathsf{Samp}, \mathsf{Eval})$ such that:

- Sample a function: $\mathsf{Gen}(\lambda)$ outputs a function index $i \in I_\lambda$. Each value of i output by $\mathsf{Gen}(\lambda)$ defines a deterministic function $f_i : D_\lambda \to R_\lambda$.
- Sample a preimage: $\mathsf{Samp}(\lambda)$ samples a random preimage $x \in D_\lambda$ according to some distribution $\mathcal{C}_\lambda$ over D_λ.[7] Typically $\mathcal{C}_\lambda$ is a uniform distribution over D_λ, and we simply write $x \xleftarrow{R} D_\lambda$ in this case.
- Evaluate a function: on input $(i, x) \in I_\lambda \times D_\lambda$, $\mathsf{Eval}(i, x)$ outputs $f_i(x)$.

In the rest of this work, we simply say $\mathcal{F}$ is a family of functions when the context is clear. For an element $y \in R_\lambda$ we denote its preimage set under f_i by $f_i^{-1}(y) = \{x \in D_\lambda : f_i(x) = y\}$. We say $\mathcal{F}$ is injective if each $f_i \in \mathcal{F}$ is injective. Following [BHSV98], we measure the amount of "non-injectivity" by looking at the maximum preimage size. Specifically, we say that $\mathcal{F}$ has polynomially bounded preimage size if $|f_i^{-1}(y)| \leq \mathsf{poly}(\lambda)$ for all $f_i \in \mathcal{F}$, all $y \in R_\lambda$ and all $\lambda \in \mathbb{N}$. For brevity, we simply say $\mathcal{F}$ is poly-to-one.

We say $\mathcal{F}$ is a family of trapdoor functions if $\mathsf{Gen}(\lambda)$ additionally outputs a trapdoor td_i, and there is a PPT algorithm $\mathsf{TdInv}(td_i, y)$ that computes a preimage $x \in f_i^{-1}(y)$. If a value y is not in the image $f_i(D_i)$, i.e., $f_i^{-1}(y)$ is empty, then the behavior of $\mathsf{TdInv}(td_i, y)$ is unspecified.

Remark 3.1. When things are clear from the context, we will slightly abuse the notation for simplicity and write: I for I_λ, D for D_λ, R for R_λ, $\mathcal{C}$ for $\mathcal{C}_\lambda$, td for td_i, $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$ for $(i \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda),\ f := f_i)$. The above definition considers the domains and ranges that depend only on λ. It is easy to generalize the definition so that the domains and ranges also depend on the function index i.

Next, we recall the notion of one-wayness and formally define the notion of non-malleability for deterministic functions. We also define the corresponding adaptive notions, in which the adversary is given access to an inversion oracle $\mathcal{O}_{\mathsf{inv}}(\cdot)$. For trapdoor functions, $\mathcal{O}_{\mathsf{inv}}(y) := \mathsf{TdInv}(td, y)$. For functions without trapdoor, $\mathcal{O}_{\mathsf{inv}}(y)$ returns a preimage $x \in f^{-1}(y)$ if $y \in f(D)$, while its behavior is unspecified otherwise. We emphasize that in the security experiments of adaptive notions the challenger is not necessarily to be efficient and could be unbounded for simulating $\mathcal{O}_{\mathsf{inv}}(\cdot)$.

Definition 3.2 (One-Wayness and Adaptive One-Wayness)**.** $\mathcal{F}$ is one-way if for any PPT adversary $\mathcal{A}$ its advantage $\mathsf{Adv}^{\mathrm{ow}}_{\mathcal{A},\mathcal{F}}(\lambda)$ defined in the security experiment below is negligible in λ:

$$\mathsf{Adv}^{\mathrm{ow}}_{\mathcal{A},\mathcal{F}}(\lambda) = \Pr\left[x \in f^{-1}(y^*) : \begin{array}{l} f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda); \\ x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda); y^* \leftarrow f(x^*); \\ x \leftarrow \mathcal{A}(f, y^*); \end{array}\right].$$

$\mathcal{F}$ is adaptively one-way if one-wayness maintains even when $\mathcal{A}$ is allowed to query $\mathcal{O}_{\mathsf{inv}}(\cdot)$ on any point other than y^*.

Definition 3.3 (Hardcore Functions)**.** Let $\mathcal{H}$ be a family of functions that maps D_λ to $\{0,1\}^{m(\lambda)}$. $\mathcal{H}$ is a hardcore of $\mathcal{F}$ if for any PPT adversary $\mathcal{A}$ its advantage $\mathsf{Adv}^{\mathrm{rand}}_{\mathcal{A},\mathcal{H}}(\lambda)$ defined in the security

[7] Virtually all "interesting" security notions are achievable only for well-spread distributions $\mathcal{C}_\lambda$ (i.e., with super-logarithmic min-entropy). Therefore, we will stick to requirement in this work.

experiment below is negligible in λ:

$$\mathsf{Adv}^{\mathrm{rand}}_{\mathcal{A},\mathcal{H}}(\lambda) = \Pr\left[b = b' : \begin{array}{l} f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda); h \leftarrow \mathcal{H}.\mathsf{Gen}(\lambda, f); \\ x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda); y^* \leftarrow f(x^*); \\ r_0^* \leftarrow h(x^*); r_1^* \xleftarrow{\mathrm{R}} \{0,1\}^m; \\ b \xleftarrow{\mathrm{R}} \{0,1\}; \\ b' \leftarrow \mathcal{A}(f, h, y^*, r_b^*); \end{array}\right] - \frac{1}{2}.$$

The well-known Goldreich-Levin theorem [GL89] says that if $\mathcal{F}$ is one-way, then it has a hardcore $\mathcal{H}$. More precisely, Goldreich and Levin [GL89] showed that the inner product of preimage x with a random string r (the latter could be viewed as part of the description of h) is a hardcore predicate (which is a special hardcore function with one-bit outputs) for any OWFs.

Definition 3.4 (Non-Malleability and Adaptive Non-Malleability)**.** Let Φ be a RPDT class defined over the domain D. $\mathcal{F}$ is Φ-non-malleable if for any PPT adversary $\mathcal{A}$ its advantage $\mathsf{Adv}^{\mathrm{nm}}_{\mathcal{A},\mathcal{F}}$ defined in the security experiment below is negligible in λ:

$$\mathsf{Adv}^{\mathrm{nm}}_{\mathcal{A},\mathcal{F}}(\lambda) = \Pr\left[\phi \in \Phi \wedge y = f(\phi(x^*)) : \begin{array}{l} f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda); \\ x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda); y^* \leftarrow f(x^*); \\ (\phi, y) \leftarrow \mathcal{A}(f, y^*); \end{array}\right].$$

$\mathcal{F}$ is adaptively Φ-non-malleable if Φ-non-malleability maintains even when $\mathcal{A}$ is allowed to query $\mathcal{O}_{\mathsf{inv}}(\cdot)$ on any point other than y^*.

We give several technical remarks about the above notions.

Impossible classes. Obviously, our non-malleable notion is impossible to realize w.r.t. RPDT class that contains "regular" transformations, namely, identity transformation id and constant transformations ϕ_c. If Φ contains id, an adversary can simply win by outputting (id, y^*). If Φ contains ϕ_c, an adversary can win by outputting $(\phi_c, f(c))$. It is easy to see that inclusion of the transformations *near to* the regular ones [8] will also make Φ-non-malleability unachievable. In this regard, we call the regular transformations and the transformations near to the regular ones as "dangerous" transformations. So, a primary task is to distill the characterizations on Φ to exclude "dangerous" transformations as well as maintain its generality to the largest extent.

Parameterized Adaptivity. Let q be the maximum number of inversion queries that an PPT adversary is allowed to make in the experiments of adaptive one-wayness/non-malleability. Typically q is assumed to be polynomially bounded and omitted from the definitions. Nevertheless, explicitly parameterizing adaptive notions with q yields more refined notions, namely q-adaptive one-wayness/non-malleability. Clearly, adaptive notions degenerate to non-adaptive ones when $q = 0$. We will adopt refined adaptive notions in Section 5.3 to give a dedicated relation between adaptive one-wayness and adaptive non-malleability.

Hinted Notions. In the non-malleable notions of one-way/hash functions considered in previous works [BCFW09, BFS11], in addition to the challenge y^*, the adversary is also given some hint of x^* to capture the auxiliary information that might has been collected from previous actions that involve x^*. The hint of x^* is modeled by $\mathsf{hint}(x^*)$, where hint is a probabilistic function from D_λ to $\{0,1\}^{m(\lambda)}$. Analogously, in the security experiments of both one-wayness

[8] Roughly, we say f is near to g if they outputs agree on most inputs.

and non-malleability for deterministic functions, we can also make the adversaries more powerful by giving them $\mathsf{hint}(x^*)$.[9] We say the resulting notions are hinted, and say the original notions are hint-free. Hinted notions are very useful in cryptographic applications in which the adversaries may obtain some auxiliary information about x^* other than merely its image y^*, as we demonstrate in Section 8.

Next, we first seek for achievable yet large RPDT class in Section 4, then explore the connections among non-malleability and one-wayness in Section 5, working with hint-free notions for simplicity. We postpone the study of the relations between hint-free notions and hinted ones to Section 6, since we need some result obtained in Section 5 as prerequisite.

4 Related-Preimage Deriving Transformation Class

Following [BFS11], our notion of non-malleability for a family of deterministic functions is defined w.r.t. a RPDT class Φ, in which $\phi : D \to D$ maps a preimage to a related preimage. We require transformations in Φ should be efficiently recognizable and computable. Hereafter, we use id to denote the identity transformation $f(x) = x$ and use cf to denote the set of all constant transformations $\{\phi_c(x) = c\}_{x \in D}$. When D under addition forms a group, we use 0 to denote the identity. For $\phi_1, \phi_2 \in \Phi$, we define $\phi := \phi_1 - \phi_2$ as $\phi(x) = \phi_1(x) - \phi_2(x)$.

As remarked before, we cannot hope to achieve non-malleability for any RPDT class Φ. We are thus motivated to distill some characterizations on Φ that make non-malleability achievable while keeping Φ still general enough. Towards this goal, we introduce two novel properties for RPDT classes as below.

Definition 4.1 (Bounded Root Space)**.** Let $r(\lambda)$ be a quantity of λ. A transformation ϕ has $r(\lambda)$-bounded root space if $|\phi^{-1}(0)| \leq r(\lambda)$. A RPDT class Φ has $r(\lambda)$-bounded root space if for each $\phi \in \Phi$ and each $\phi_c \in \mathsf{cf}$, the composite transformations $\phi' = \phi - \mathsf{id}$ and $\phi' = \phi - \phi_c$ both have $r(\lambda)$-bounded root space.

Definition 4.2 (Sampleable Root Space)**.** A transformation ϕ has sampleable root space if there exists a PPT algorithm SampRS that takes ϕ as input and outputs an element from $\phi^{-1}(0)$ uniformly at random.[10] A RPDT class Φ has sampleable root space if for each $\phi \in \Phi$ and each $\phi_c \in \mathsf{cf}$, the composite transformations $\phi' = \phi - \mathsf{id}$ and $\phi'' = \phi - \phi_c$ both have sampleable root spaces.

In this work, we restrict our attention to root spaces whose size is polynomially bounded,[11] i.e., $r(\lambda) \leq \mathsf{poly}(\lambda)$. Hereafter, we let $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$ denote the RPDT class satisfying the bounded root space (BRS) & sampleable root space (SRS) properties. The BRS property immediately rules the regular transformations out of Φ and stipulates that each $\phi \in \Phi$ is far away from regular ones, i.e., having at most polynomial intersection points with them. As we will see shortly, with the confining of the BRS property, an adversary's correct solution (ϕ, y) such that $f(\phi(x^*)) = y$ provides enough information about x^* and thus reduces the min-entropy of x^* to $O(\log(\lambda))$. The SRS property further guarantees that a polynomial-time reduction can extract the right x^* with noticeable probability.

[9] Clearly, to make the hinted notions achievable, hint must meets some necessary condition. For instance of hinted non-malleability, hint should be at least uninvertible (finding the exact preimage is infeasible). We prefer to keep the definition as general as possible, so we do not explicitly impose concrete restriction to hint in definition.

[10] If $\phi^{-1}(0)$ is empty, then this algorithm simply outputs a distinguished symbol $\perp$.

[11] We will continue to use BRS to denote poly-bounded root space for simplicity.

Remark 4.1. Recent works [JW15, QLY+15] introduced two general properties called high output entropy (HOE) and input-output collision resistance (IOCR) for transformation class Φ. The former states that for each $\phi \in \Phi$, the min-entropy of $\phi(x)$ is sufficiently high when $x \xleftarrow{R} D$, i.e., $\mathsf{H}_\infty(\phi(x)) = \omega(\log \lambda)$. The latter states that for each $\phi \in \Phi$, $\Pr[\phi(x) = x] = \mathsf{negl}(\lambda)$ when $x \xleftarrow{R} D$. We observe here that BRS implies HOE & IOCR. To see this, notice that: (1) for each $c \in D$ the equation $\phi(x) - c = 0$ having at most polynomial number of roots implies that $\max_{c \in D} \Pr[\phi(x) = c] \leq \mathsf{poly}(\lambda)/|D| = \mathsf{negl}(\lambda)$ when $x \xleftarrow{R} D$; (2) the equation $\phi(x) - x = 0$ having at most polynomial number of roots implies that $\Pr[\phi(x) = x] \leq \mathsf{poly}(\lambda)/|D| = \mathsf{negl}(\lambda)$ when $x \xleftarrow{R} D$. We can alternatively think of the BRS property captures the characterization that all $\phi \in \Phi$ are far from regular transformations in an algebraic view.

The notion of root sampleable RPDTs (RPDT class that meets the SRS property) is reminiscent of the notion of preimage sampleable functions introduced in [GPV08]. The former one is weaker than the latter one in that it only insists two special forms of transformations are preimage sampleable at zero point obeying uniform distribution. We note that it suffices to relax uniform distribution to some appropriate distribution.

We conclude this section by showing that the BRS & SRS properties are met by most algebra-induced transformation classes (excluding id and cf) considered in the literature, which we recall as below.

Group-induced transformations. When D under $\odot$ forms a group $\mathbb{G}$, let $\Phi^{\mathrm{lin}} = \{\phi_a\}_{a \in \mathbb{G}}$ with $\phi_a(x) = a \odot x$ be the class of linear transformations, which generalize several important classes, for example, "bit flips" (exclusive or, XOR) $\phi_a(x) = a \oplus x$ and modular additions $\phi_a(x) = a + x \bmod 2^n$ when $D = \{0,1\}^n$.

Ring-induced transformations. When D under addition $+$ and multiplication $\cdot$ forms a ring $\mathbb{R}$, let $\Phi^{\mathrm{aff}} = \{\phi_{a,b}\}_{a,b \in \mathbb{R}}$ with $\phi_{a,b}(x) = ax + b$ be the class of affine transformations.

Field-induced transformations. When D under addition $+$ and multiplication $\cdot$ forms a field $\mathbb{F}$, let p be the characteristic of $\mathbb{F}$ and $d \geq 0$ be any fixed integer. Let $\Phi^{\mathrm{poly(d)}} = \{\phi_q\}_{q \in \mathbb{F}_d(x)}$ with $\phi_q(x) = q(x)$ be the class of polynomial functions, where $\mathbb{F}_d(x)$ denotes single variable polynomials over $\mathbb{F}$ with degree bounded by d. When d and p are small (i.e., $d = \mathsf{poly}(\lambda)$ and $p = \mathsf{poly}(\lambda)$), one can find all roots for any $q \in \mathbb{F}_d(x)$ in polynomial time $O(d^3 p)$ using Berlekamp's algorithm [Ber70]. When d is small but p is large, one can find all roots for any $q \in \mathbb{F}_d(x)$ in expected polynomial time $O(d^{2+\varepsilon} + d^{1+\varepsilon} \log p)$ using Gathen and Shoup's algorithm [vzGS92].

It is easy to verify that $\Phi^{\mathrm{lin}} \backslash \mathsf{id}$, $\Phi^{\mathrm{aff}} \backslash (\mathsf{id} \cup \mathsf{cf})$, and $\Phi^{\mathrm{poly}(d)} \backslash (\mathsf{id} \cup \mathsf{cf})$ for $d = \mathsf{poly}(\lambda)$ all satisfy the BRS and SRS properties.

5 Relations Among Non-Malleability and One-Wayness

In this section, we explore the relations among (adaptive) non-malleability and (adaptive) one-wayness for deterministic functions. For simplicity, we work with hint-free notions. All the results obtained extend naturally among hinted notions.

5.1 Non-Malleability ⇒ One-Wayness

Lemma 5.1. *For any achievable RPDT class Φ, Φ-Non-Malleability ⇒ One-Wayness when $\mathcal{F}$ is poly-to-one.*

Proof. Suppose there is an adversary $\mathcal{A}$ that breaks the one-wayness of $\mathcal{F}$ with non-negligible probability, then we can build an algorithm $\mathcal{B}$ that breaks non-malleability of $\mathcal{F}$ also with non-negligible probability. $\mathcal{B}$ works by simulating $\mathcal{A}$'s challenger in the one-wayness experiment as follows:

Setup: Given $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$ and a challenge $y^* \leftarrow f(x^*)$ for $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$, $\mathcal{B}$ forwards (f, y^*) to $\mathcal{A}$.

Attack: When $\mathcal{A}$ outputs its solution x against one-wayness, $\mathcal{B}$ simply picks a random $\phi \in \Phi$, then outputs $(\phi, f(\phi(x))$ as its solution.

Since $\mathcal{F}$ is poly-to-one, then conditioned on $\mathcal{A}$ succeeds ($x \in f^{-1}(y^*)$), we have $\Pr[x = x^*|y^*] \geq 1/\mathsf{poly}(\lambda)$, where the probability is over the choice of $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$. This is because there are at most $\mathsf{poly}(\lambda)$ values x such that $f(x) = y^*$, and they are all equally like in $\mathcal{A}$'s view. Therefore, if $\mathcal{A}$ breaks the one-wayness of $\mathcal{F}$ with non-negligible probability, then $\mathcal{B}$ breaks the non-malleability of $\mathcal{F}$ also with non-negligible probability. This proves the lemma. □

The above reduction loses a factor of $1/\mathsf{poly}(\lambda)$. When $\mathcal{F}$ is injective, the reduction becomes tight.

5.2 One-Wayness $\not\Rightarrow$ Non-Malleability

Lemma 5.2. *One-Wayness $\not\Rightarrow$ $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$-Non-Malleability.*

Proof. Let $\mathcal{F}$ be a family of one-way functions. To prove this lemma, we show how to modify $\mathcal{F}$ into $\mathcal{F}'$ so that $\mathcal{F}'$ is still one-way but malleable w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$. Suppose $\mathcal{F}.\mathsf{Gen}(\lambda)$ outputs a function $f : \{0,1\}^n \to \{0,1\}^m$, we construct $\mathcal{F}'.\mathsf{Gen}(\lambda)$ as follows: run $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, output a function $f' : \{0,1\}^{n+1} \to \{0,1\}^{m+1}$ where $f'(x||\beta) := f(x)||\beta$ and β denotes the last bit of its input.

Claim 1. *$\mathcal{F}'$ is one-way.*

Proof. It is easy to see that $\mathcal{F}'$ inherits the one-wayness from $\mathcal{F}$. We omit the proof here due to its straightforwardness. □

Claim 2. *$\mathcal{F}'$ is $(\Phi^{\mathrm{xor}} \backslash \mathsf{id})$-malleable.*

Proof. Given f' and a challenge $y'^* = f'(x'^*)$ where $x'^* = x^*||\beta^*$ is randomly chosen from $\{0,1\}^{n+1}$, we build an adversary $\mathcal{A}'$ against the non-malleability of $\mathcal{F}'$ as follows: parse y'^* as $y^*||\beta^*$, set $a = 0^n||1$, then output ϕ_a together with $y' = y^*||(\beta^* \oplus 1)$. It is easy to see that $\phi_a \in \Phi^{\mathrm{xor}} \backslash \mathsf{id}$ and $y' = f'(x^*||(\beta^* \oplus 1)) = f'(\phi_a(x'^*))$. This proves Claim 2. □

As shown in Section 4, Φ^{xor} is a special case of group-induced class, and thus $\Phi^{\mathrm{xor}} \backslash \mathsf{id} \subseteq \Phi^{\mathrm{srs}}_{\mathrm{brs}}$. The lemma immediately follows from the above two claims. □

While this is just a contrived counterexample for one particular attempt, there exist more natural counterexamples. For instance, a Φ-homomorphic one-way function[12] f is also Φ-malleable since $f(x^*) = y^*$ implies $f(\phi(x^*)) = \phi(y^*)$. All these counterexamples indicate that functions with nice algebraic structure are unlikely to be non-malleable.

[12] Φ-homomorphism means that for any $\phi \in \Phi$ and any $x \in D$, $f(\phi(x)) = \phi(f(x))$.

5.3 Adaptive Non-Malleability $\Leftrightarrow$ Adaptive One-Wayness

Lemma 5.3. *For any achievable RPDT class Φ, q-Adaptive Φ-Non-Malleability $\Rightarrow$ q-Adaptive One-Wayness when $\mathcal{F}$ is poly-to-one.*

Proof. The proof can be easily adapted from that of Lemma 5.1. We omit it here for its straightforwardness. □

Lemma 5.4. *$(q+1)$-Adaptive One-Wayness $\Rightarrow$ q-Adaptive $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$-Non-Malleability when $\mathcal{F}$ is injective.*

We first outline the high-level idea of the proof. Since the task of finding the preimage x^* appears to be harder than that of mauling its image, the chief technical difficulty is how to utilize the power of an adversary $\mathcal{A}$ against adaptive non-malleability to break adaptive one-wayness. First, it is instructive to see that a challenge instance of one-wayness already provides an equation about x^*, that is, $f(x^*) = y^*$. When $\mathcal{A}$ outputs its solution (ϕ, y) against non-malleability, the reduction immediately obtains another equation about x^*, that is, $f(\phi(x^*)) = y$. However, these two equations are hard to solve on their own due to the involvement of f (which could be complex). Luckily, by utilizing either the injectivity of f or the inversion oracle, the reduction is able to obtain a new solvable equation about x^* without the presence of f: (1) for the case of $y = y^*$, the reduction gets $\phi(x^*) = x^*$ due to the injectivity of f; (2) for the case of $y \neq y^*$, the reduction first queries the inversion oracle at point y, then gets $\phi(x^*) = \mathcal{O}_{\mathsf{inv}}(y)$. In both cases, the reduction successfully confines x^* in a poly-bounded root space (due to the BRS property), then correctly extracts it with noticeable probability (due to the SRS property). This justifies the usefulness of BRS & SRS properties. See the formal proof as follows.

Proof. Suppose there is an adversary $\mathcal{A}$ against the adaptive non-malleability of $\mathcal{F}$, then we can build an adversary $\mathcal{B}$ against the adaptive one-wayness of $\mathcal{F}$. $\mathcal{B}$ simulates $\mathcal{A}$'s challenger in the adaptive non-malleability experiment as follows:

Setup: Given $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$ and a challenge $y^* \leftarrow f(x^*)$ for $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$, $\mathcal{B}$ forwards (f, y^*) to $\mathcal{A}$.

Attack: When $\mathcal{A}$ issues an query to the inversion oracle, $\mathcal{B}$ forwards it to its own challenger and sends back the reply. When $\mathcal{A}$ outputs its solution (ϕ, y) against adaptive non-malleability, $\mathcal{B}$ proceeds as follows:

1. Case $y = y^*$: $\mathcal{B}$ runs $\mathsf{SampRS}(\phi')$ to output a random solution of $\phi'(\alpha) = 0$ where $\phi'(\alpha) = \phi(\alpha) - \alpha$.
2. Case $y \neq y^*$: $\mathcal{B}$ queries the inversion oracle $\mathcal{O}_{\mathsf{inv}}(\cdot)$ at point y and gets the response x, then runs $\mathsf{SampRS}(\phi'')$ to output a random solution of $\phi''(\alpha) = 0$ where $\phi''(\alpha) = \phi(\alpha) - x$.

We justify the correctness of $\mathcal{B}$'s strategy as follows. For case 1, conditioned on $\mathcal{A}$ succeeds $(f(\phi(x^*)) = y^*)$, due to the injectivity of $\mathcal{F}$, we have $\phi(x^*) = x^*$, i.e., x^* is a solution of $\phi'(\alpha) = 0$. For case 2, conditioned on $\mathcal{A}$ succeeds $(f(\phi(x^*)) = y)$, due to the injectivity of $\mathcal{F}$, we have $\phi(x^*) = x$, i.e., x^* is a solution of $\phi''(\alpha) = 0$. Taking the two cases together, conditioned on $\mathcal{A}$ succeeds and makes at most q inversion queries, then according to the BRS & SRS properties of $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$, $\mathcal{B}$ will output the right x^* with probability $1/\mathsf{poly}(\lambda)$ by making at most $(q+1)$ inversion queries. We stress that here the probability is taken over the randomness of SampRS but not $\mathcal{F}.\mathsf{Samp}$. Thereby, if $\mathcal{A}$ breaks the q-adaptive non-malleability with non-negligible probability, $\mathcal{B}$ breaks the $(q+1)$-adaptive one-wayness also with non-negligible probability. This proves this lemma. □

Combining Lemma 5.3 and Lemma 5.4 together, we conclude that for injective functions, adaptive $\Phi^{\text{srs}}_{\text{brs}}$-non-malleability is equivalent to adaptive one-wayness.

Remark 5.1. Analogous to the RKA security notion, our non-malleable notion is of "unique" flavor, in which the adversary is only considered to be successful if its output is a related image of the exactly preimage x^* chosen by the challenger. Precisely for this reason, the injectivity of $\mathcal{F}$ is crucial for the reduction from adaptive non-malleability to adaptive one-wayness. If $\mathcal{F}$ is non-injective, the reduction is not guaranteed to get the right equation about x^*. For example, in case $y = y^*$, if the adversary $\mathcal{A}$ always outputs a $\phi \in \Phi$ such that $\phi(x) \neq x$ for any $x \in D$, then the reduction will never get a right solvable equation about x^*.

5.4 Non-Malleability $\not\Rightarrow$ Adaptive Non-Malleability

At first glance, one might think non-malleability does imply adaptive non-malleability based on the intuition that the inversion oracle does not help. Suppose $\mathcal{A}$ is an adversary against adaptive non-malleability. Given $y^* \leftarrow f(x^*)$ for randomly chosen x^* and an inversion oracle, $\mathcal{A}$ is asked to output (ϕ, y) such that $f(\phi(x^*)) = y$. Since $\mathcal{A}$ is not allowed to query the inversion oracle on y^*, it seems the only strategy is to firstly maul y^* to some related y, then query the inversion oracle on y, and use the answer x to help figuring out a transformation ϕ s.t. $\phi(x^*) = x$. As we showed in Lemma 5.1, if $\mathcal{F}$ is non-malleable and poly-to-one, it is also one-way and thus x^* is computationally hidden from $\mathcal{A}$. Thus, it seems impossible for $\mathcal{A}$ to determine ϕ without the knowledge of x^*.

However, the above intuition is deceptive in thinking that the inversion algorithm always behave benignly, namely, returning the preimages of its inputs. Actually, contrived inversion algorithm may reveals critical information (e.g. trapdoor) when its inputs fall outside the image of f, and thus make f couldn't be adaptive non-malleable. This is similar in spirit to the separation NM-CPA $\not\Rightarrow$ IND-CCA1 [BDPR98, Section 3.2] in the public-key encryption setting.

Lemma 5.5. *For any achievable RPDT class Φ, Φ-Non-Malleability $\not\Rightarrow$ Adaptive Φ-Non-Malleability when $\mathcal{F}$ is poly-to-one.*

Proof. Let $\mathcal{F} = (\mathsf{Gen}, \mathsf{Samp}, \mathsf{Eval}, \mathsf{TdInv})$ be a family of Φ-non-malleable functions with trapdoor. We show how to modify $\mathcal{F}$ to $\mathcal{F}' = (\mathsf{Gen}', \mathsf{Samp}', \mathsf{Eval}', \mathsf{TdInv}')$ so that $\mathcal{F}'$ is still Φ-non-malleable but not adaptively Φ-non-malleable. The idea is to make $\mathcal{F}'.\mathsf{TdInv}'$ dangerous, i.e., having its outputs on "ill-formed images" carry the information of trapdoor. Correspondingly, we have to make these "ill-formed images" lie outside f''s well-formed image to ensure the correctness of $\mathcal{F}'.\mathsf{TdInv}'$. For simplicity, we use the bit-prefix trick to split the range into two disjoint sets: the strings with bit-prefix '0' are well-formed images, while the strings with bit-prefix '1' are ill-formed images. We sketch the modification as follows: $\mathcal{F}'.\mathsf{Gen}'(\lambda)$ runs $\mathcal{F}.\mathsf{Gen}(\lambda)$ to generate $f : \{0,1\}^n \to \{0,1\}^m$ and its associated trapdoor td, then outputs $f' : \{0,1\}^n \to \{0,1\}^{m+1}$ along with td where $f'(x) := 0||f(x)$; $\mathcal{F}'$ inherits the same domain sampling algorithm from $\mathcal{F}$; $\mathcal{F}'.\mathsf{Eval}'(x)$ outputs $0||\mathcal{F}.\mathsf{Eval}(x)$; $\mathcal{F}'.\mathsf{TdInv}'(td, y')$ parses y' as $b||y$, if $b = 0$ outputs $\mathcal{F}.\mathsf{TdInv}(td, y)$, else outputs td. Clearly, $\mathcal{F}'$ satisfies the correctness. We then make two claims about the security of $\mathcal{F}'$.

Claim 3. *$\mathcal{F}'$ is Φ-non-malleable.*

Proof. Suppose $\mathcal{A}'$ is an adversary against the Φ-non-malleability against $\mathcal{F}'$, then we show how to construct an adversary $\mathcal{A}$ against the Φ-non-malleability of $\mathcal{F}$. Given f and $y^* \leftarrow f(x^*)$ for $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$, $\mathcal{A}$ is asked to output (ϕ, y) s.t. $f(\phi(x^*)) = y$. $\mathcal{A}$ sends f' (defined as above) and $0||y^*$ to invoke $\mathcal{A}'$. As soon as $\mathcal{A}'$ outputs its solution $(\phi, y' = 0||y)$, $\mathcal{A}$ outputs (ϕ, y) as its answer. By the construction of f', if $\mathcal{A}'$ succeeds ($f'(\phi(x^*)) = y'$), $\mathcal{A}$ also succeeds ($f(\phi(x^*)) = y$). Claim 3 follows. □

Claim 4. $\mathcal{F}'$ is not adaptively Φ-non-malleable.

Proof. Given f and $0||y^* \leftarrow f'(x^*)$ for $x^* \leftarrow \mathcal{F}'.\mathsf{Samp}'(\lambda)$, we construct an adversary $\mathcal{A}'$ against the adaptive Φ-non-malleability of $\mathcal{F}'$ as follows: it queries $\mathcal{O}_{\mathsf{inv}}(\cdot)$ of $\mathcal{F}'$ at point $1||0^m$. According to the definition of $\mathcal{F}'.\mathsf{TdInv}'$, $\mathcal{A}'$ will get td in plain. At this point, knowing the trapdoor td, $\mathcal{A}'$ runs $\mathcal{F}'.\mathsf{TdInv}'(td, 0||y^*)$ itself to compute a preimage $x \in f'^{-1}(0||y^*)$, then picks an arbitrary $\phi \in \Phi$ and outputs $(\phi, 0||f(\phi(x)))$ as its solution. Since $\mathcal{F}$ is poly-to-one, $\mathcal{F}'$ is also poly-to-one. Via an argument similar to that in the proof of Lemma 5.1, we have $\Pr[x = x^*|(0||y^*)] \geq 1/\mathsf{poly}(\lambda)$, where the probability is over the choice of $x^* \leftarrow \mathcal{F}'.\mathsf{Samp}'(\lambda)$. Thus, $\mathcal{A}'$ breaks the adaptive Φ-non-malleability with advantage at least $1/\mathsf{poly}(\lambda)$, which is non-negligible in λ. This proves Claim 4. □

Lemma 5.5 immediately follows from the above two claims. □

In the above, we work with hint-free (standard) non-malleability notion and one-wayness notion for simplicity. It is easy to see that all these relations apply equally well to the hinted non-malleability notion and the hinted one-wayness notion, with respect to the same hint function.

Construction of NMFs. Baecher et al. [BFS11, Construction 4.1] showed that the strengthened Merkle-Damgård (MD) transformation is non-malleable w.r.t. $\Phi^{\mathrm{xor}}\backslash\mathsf{id}$), assuming the compression function is a random oracle. In Appendix A, we improve over their result, showing that the strengthened MD transformation is essentially non-malleable w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$. This result gives us an efficient candidate of NMFs w.r.t. large RPDT class, though in the random oracle model.

As to the construction of NMFs in the standard model, Lemma 5.4 shows that any injective ATDFs are indeed $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$-non-malleable, while [KMO10] demonstrates that injective ATDFs can be constructed from either a number of cryptographic primitives such as correlated-product TDFs [RS10], lossy TDFs [PW08] and CCA-secure deterministic encryption [BBO07] (which in turn can be efficiently constructed from a variety of standard assumptions) or from some specific assumption, e.g. "instance-independent" RSA assumption. This indicates that deterministic NMFs are widely realizable in the standard model, and thus partially resolves an open question raised in [BCFW09].

Finally, we observe that for the purpose of constructing NMFs, 1-ATDFs (which only allows the adversary to query the inversion oracle once) are sufficient. Nevertheless, if 1-ATDFs are strictly weaker than q-ATDFs for $q > 1$ and if it allows more efficient instantiations are still unknown to us. Besides, we are only able to construct NMFs w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$ in this work. Though $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$ is very general (comprising most algebra-induced transformations), it is still of great interest to know if it is possible to go beyond the algebraic barrier.

6 Relation Between Hint-free and Hinted Notions

In this section, we investigate the relations between hint-free notions and hinted notions. Unlike hinted notions obviously imply hint-free ones, if the reverse implication holds crucially depends on the hint functions. It is intriguing to know for what kind of hint functions, hint-free notions do imply hinted notions.

Let $\mathcal{F}$ be a family of deterministic functions, $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$ and $y^* \leftarrow f(x^*)$. Roughly, we say $\mathsf{hint}(x^*)$ is $p(\lambda)$-*statistically simulatable* if there exists a PPT algorithm $\mathcal{R}$ such that $(y^*, \mathcal{R}(y^*)) \approx_s (y^*, \mathsf{hint}(x^*))$ with probability $p(\lambda)$; we say $\mathsf{hint}(x^*)$ is $p(\lambda)$-*computationally simulatable* if there exists a PPT algorithm $\mathcal{R}$ such that based on the hint-free hardness assumption $(y^*, \mathcal{R}(y^*)) \approx_c (y^*, \mathsf{hint}(x^*))$ with probability $p(\lambda)$. The probability is over

the choice of $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$ and the random coins of $\mathcal{R}$. It is easy to see that when $\mathsf{hint}(x^*)$ is either statistically simulatable or computationally simulatable for some noticeable probability $p(\lambda)$, a reduction algorithm is able to create a game with probability $p(\lambda)$ such that it is indistinguishable to the real hinted game, and thus reduces hinted notions to hint-free ones. We exemplify these two cases in Lemma 6.2 and Lemma 6.3, respectively.

Next, we formally study the relation between one-wayness and hinted one-wayness, then show the analogous result also holds between non-malleability and hinted non-malleability for poly-to-one functions.

Lemma 6.1. *For a family of functions $\mathcal{F}$, hinted one-wayness w.r.t. any achievable hint function implies one-wayness.*

Proof. This direction is straightforward and hence the proof is omitted. □

We then turn to the inverse direction. We first show that regardless of the construction of $\mathsf{hint}(\cdot)$, as long as its output length is short, i.e., bounded by $\log(\mathsf{poly}(\lambda))$, then $\mathsf{hint}(x^*)$ is $1/\mathsf{poly}(\lambda)$-perfectly simulatable (a special case of statistically simulatable) and thus one-wayness implies hinted one-wayness.

Lemma 6.2 (statistically simulatable case)**.** *For a family of functions $\mathcal{F}$, one-wayness implies hinted one-wayness w.r.t. any hint function with output length bounded by* $\log(\mathsf{poly}(\lambda))$.

Proof. Let $\mathcal{A}$ be an adversary against hinted one-wayness of $\mathcal{F}$ with advantage $\mathsf{Adv}^{\mathrm{how}}_{\mathcal{A},\mathcal{F}}(\lambda)$. We build an adversary $\mathcal{B}$ against one-wayness by using $\mathcal{A}$'s power. Given (f, y^*) where $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, $y^* \leftarrow f(x^*)$ for $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$, $\mathcal{B}$ simply makes a random guess of $\mathsf{hint}(x^*)$, then sends $(f, y^*, \mathsf{hint}(x^*))$ to $\mathcal{A}$. Finally, $\mathcal{B}$ forwards $\mathcal{A}$'s solution as its solution. Since the output length is bounded by $\log(\mathsf{poly}(\lambda))$, then $\mathcal{B}$ guesses the right hint value and thus simulate perfectly with probability $1/\mathsf{poly}(\lambda)$. Thereby, we conclude that $\mathsf{Adv}^{\mathrm{ow}}_{\mathcal{B},\mathcal{F}}(\lambda) \geq \mathsf{Adv}^{\mathrm{how}}_{\mathcal{A},\mathcal{F}}(\lambda)/\mathsf{poly}(\lambda)$. The lemma immediately follows. □

We then show that for some specific hint functions with output length could possibly beyond $\log(\mathsf{poly}(\lambda))$, $\mathsf{hint}(x^*)$ is computationally simulatable assuming the one-wayness of $\mathcal{F}$ and thus hint-free one-wayness also implies hinted one-wayness in this case.

Lemma 6.3 (computationally simulatable case)**.** *For a family of functions $\mathcal{F}$, one-wayness implies hinted one-wayness w.r.t. the following specific hint function:*

$$\mathsf{hint}(x; b) = \begin{cases} h(x) & \text{if } b = 0 \\ r \xleftarrow{\mathrm{R}} \{0,1\}^{m(\lambda)} & \text{if } b = 1 \end{cases} \tag{1}$$

Here, $h : D \to \{0,1\}^{m(\lambda)}$ denotes the hardcore function for $f \in \mathcal{F}$. It is well-defined when $\mathcal{F}$ is one-way.

Proof. The high-level idea of the proof is to show that assuming the one-wayness of $\mathcal{F}$, $\mathsf{hint}(x^*; b)$ for $x^* \xleftarrow{\mathrm{R}} X$ and $b \xleftarrow{\mathrm{R}} \{0,1\}$ is 1-computationally simulatable. We prove this theorem via a sequence of games. Let $\mathcal{A}$ be an adversary against the hinted one-wayness of $\mathcal{F}$ w.r.t. the hint function defined as above. Let S_i be the event that $\mathcal{A}$ wins in Game i.

Game 0 (The real experiment): $\mathcal{CH}$ interacts with $\mathcal{A}$ in the real hinted one-wayness experiment w.r.t. the hinted function defined as above. According to the definition, we have:

$$\mathsf{Adv}^{\mathrm{how}}_{\mathcal{A},\mathcal{F}}(\lambda) = \Pr[S_0]. \tag{2}$$

Game 1 (Modify the hint function): The same as Game 0 except that the hint function $\mathsf{hint}(x^*; b)$ is modified to $\widetilde{\mathsf{hint}}(x^*; b)$, which ignores its input (x^*, b) and always returns a random value $r \xleftarrow{R} \{0,1\}^{m(\lambda)}$. Observe that in this case the hint value carries no information of x^*.

We now state and prove two claims that establish the main theorem.

Claim 5. Game 0 and Game 1 are computationally indistinguishable, assuming the hint-free one-wayness of $\mathcal{F}$.

Proof. Since one-wayness of $\mathcal{F}$ implies pseudorandomness w.r.t. its hardcore $\mathcal{H}$, it suffices to show Game 0 and Game 1 are computationally indistinguishable based on the pseudorandomness of $\mathcal{H}$ defined in Definition 3.3. We show how to turn a distinguisher $\mathcal{A}$ into an algorithm $\mathcal{B}$ against the pseudorandomness of $\mathcal{H}$.

Given (f, h, y^*, r^*_β) where $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, h is a hardcore function for f, $y^* \leftarrow f(x^*)$ for $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$, and r^*_β is $h(x^*)$ if $\beta = 0$ or a random string from $\{0,1\}^{m(\lambda)}$ if $\beta = 1$, $\mathcal{B}$ is asked to determine the value of β. $\mathcal{B}$ picks a random bit b and computes the hint value as follows:

$$\mathsf{hint}'(x^*; b) = \begin{cases} r^*_\beta & \text{if } b = 0 \\ r \xleftarrow{R} \{0,1\}^{m(\lambda)} & \text{if } b = 1 \end{cases}$$

$\mathcal{B}$ then sends $(f, y^*, \mathsf{hint}'(x^*))$ to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs a bit b' ($b' = 0$ indicates Game 0 and $b' = 1$ indicates Game 1), and $\mathcal{B}$ forwards b' to its own challenger. It is easy to verify that if $\beta = 0$ then $\mathsf{hint}(x^*; b) = \mathsf{hint}'(x^*; b)$ and thus $\mathcal{B}$ perfectly simulates Game 0; if $\beta = 1$ then $\widetilde{\mathsf{hint}}(x^*; b) = \mathsf{hint}'(x^*; b)$ and thus $\mathcal{B}$ perfectly simulates Game 1. Therefore, $\mathcal{B}$ breaks the pseudorandomness of $\mathcal{H}$ with at least the same advantage as $\mathcal{A}$ distinguishes Game 0 and Game 1. By assuming the one-wayness of $\mathcal{F}$, Game 0 and Game 1 are computationally indistinguishable. This proves the Claim 5. □

Claim 6. No PPT adversary has non-negligible advantage in Game 2 assuming the one-wayness of $\mathcal{F}$.

Proof. Suppose $\mathcal{A}$ is a PPT adversary that has non-negligible advantage in Game 2. We show how to use $\mathcal{A}$'s power to break the one-wayness of $\mathcal{F}$. Given the one-wayness challenge (f, y^*) where $y^* \leftarrow f(x^*)$ for randomly chosen x^*, $\mathcal{B}$ simply assigns $\widetilde{\mathsf{hint}}(x^*; b)$ to be a random string from $\{0,1\}^{m(\lambda)}$, then sends $(f, y^*, \widetilde{\mathsf{hint}}(x^*; b))$ to $\mathcal{A}$ as the challenge. Finally, $\mathcal{A}$ outputs its solution, and $\mathcal{B}$ forwards it to its own challenger. Clearly, $\mathcal{B}$ perfectly simulates Game 1. Therefore, $\mathcal{B}$ breaks the one-wayness of $\mathcal{F}$ with at least the same advantage as $\mathcal{A}$ succeeds in Game 1. By assuming the one-wayness of $\mathcal{F}$, $\mathcal{A}$'s advantage must be negligible in λ. This proves the Claim 6. □

From Claim 5 and Claim 6, we have $\Pr[S_1] - \Pr[S_0] = \mathsf{negl}(\lambda)$ and $\Pr[S_1] = \mathsf{negl}(\lambda)$. Putting all the above together, we have $\mathsf{Adv}^{\mathrm{how}}_{\mathcal{A},\mathcal{F}}(\lambda) = \mathsf{negl}(\lambda)$ assuming the one-wayness of $\mathcal{F}$. In other words, one-wayness implies hinted one-wayness w.r.t. such specific hint function defined as above. The lemma immediately follows. □

The above results apply naturally to the adaptive setting.

Remark 6.1. It is easy to see that the above results also hold between hinted non-malleability and hint-free non-malleability for poly-to-one $\mathcal{F}$. Particularly, to see hinted NM w.r.t. the hint function defined in Equation (1) is implied by hint-free NM, just note that such hint function is 1-computationally simulatable assuming the one-wayness of $\mathcal{F}$ (as we shown in Lemma 6.3), which in turn implied by the non-malleability of $\mathcal{F}$ when $\mathcal{F}$ is poly-to-one (Lemma 5.1).

7 Built-in Resilience against Copy Attacks

Here, we extend the idea underlying the implication AOW $\Rightarrow$ ANM further still to address copy attacks in the RKA area. We begin by briefly introducing the background of RKA security and defining what it means for "copy attacks".

Related key security. Traditional security models assume that the internal states (e.g., secret keys and random coins) of cryptographic hardware device are completely protected from the adversary. However, practical fault injection techniques [BS97, BDL97] demonstrate that the adversaries are able to launch related-key attacks (RKAs), namely inducing modifications to the keys stored in cryptographic hardware device and subsequently observe the outcome under the modified keys. Bellare and Kohno [BK03] initiated a theoretical study of RKA security. Their results mainly focused on PRF/PRP, and their constructions were further improved by [BC10, ABPP14]. So far, the study of RKA security expands to other primitives, such as private-key encryption [AHI11], public-key encryption [Wee12], signature [BPT12], and identity-based encryption [BPT12].

RKA-security Model and Copy Attacks. In the RKA-security model, modifications to the secret keys are modeled by related-key deriving transformation (RKDT) class Φ, and cryptographic hardware device is modeled by algorithm $\mathsf{Func}(sk, x)$, where $\mathsf{Func}(sk, \cdot)$ denotes some keyed-operations (e.g., signing, decryption) and x denotes its input (e.g., message, ciphertext). A cryptographic primitive is said to be RKA-secure if it remains secure when the adversary is given access to a RKA oracle $\mathcal{O}_{\mathsf{rka}}(\phi, x) := \mathsf{Func}(\phi(sk), x)$.

Let x^* denote the challenge in the security experiment. The RKA queries $\langle \phi, x^* \rangle$ where $\phi(sk) = sk$ essentially capture a category of attacks known as "copy attacks". Among copy attacks, we refer to the ones with $\phi = \mathsf{id}$ as trivial copy attacks and the rest as non-trivial copy attacks. While trivial copy attacks must be excluded to ensure the meaningfulness of the RKA-security notion, non-trivial copy attacks should be allowed since they are possible in practice (e.g., fault injection attacks [BS97, BDL97]). However, attaining resilience against non-trivial copy attacks turns out to be difficult.

Almost all the known constructions of RKA-secure primitives achieve RKA security by exploiting so called Φ-key-malleability as a vital property. Loosely speaking, this property provides a PPT algorithm T such that $\mathsf{Func}(\phi(sk), x) = \mathsf{Func}(sk, \mathsf{T}(\phi, x))$. Let $\mathcal{O}(x) := \mathsf{Func}(sk, x)$ be the original oracle of the starting primitive. With such property, the reduction is able to reduce the RKA security to the original security of the starting primitive by simulating the RKA oracle via the original oracle, that is, answering $\mathcal{O}_{\mathsf{rka}}(\phi, x)$ with $\mathcal{O}(\mathsf{T}(\phi, x))$. However, a subtlety in the above strategy is that the original oracle $\mathcal{O}(\cdot)$ will deny query $\langle x^* \rangle$. As a consequence, the reduction is unable to handle non-trivial copy attacks (i.e., answering RKA queries $\langle \phi, x^* \rangle$ where $\phi \neq \mathsf{id}$ but $\phi(sk) = sk$) as before.

Prior works paid a lot of effort on this problem. To date, there are three methods dealing with non-trivial copy attacks in the literature. The first method is assuming Φ is claw-free and contains id. Recall that claw-freeness requires that for all distinct $\phi, \phi' \in \Phi$ and all $x \in D$, $\phi(x) \neq \phi'(x)$. With this assumption, such a ϕ is not in Φ and non-trivial copy attacks are automatically ruled out. This is exactly the technical reason of why numerous constructions of Φ-RKA-secure-primitives [BK03, Luc04, BC10, GL10] are restricted to claw-free Φ. However, as pin-pointed by [BCM11, ABPP14], this assumption is undesirable because many natural and practical RKDT classes are not claw-free. The second method is directly modifying the RKA security experiment to disallow RKA queries $\langle \phi, x^* \rangle$ where $\phi \neq \mathsf{id}$ but $\phi(sk) = sk$. Such method evades non-trivial copy attacks only in the conceptual sense by adopting a potentially weaker RKA notion. It also brings a new technical challenge, that is, checking if $\phi(sk) = sk$ without knowing sk. To overcome this hurdle, existing works either require the starting primitives to

meet extra properties like Φ-fingerprinting [Wee12, JLLM13, LLJ14] in the context of public-key encryption or resort to ad-hoc transform like identity-renaming [BPT12] in the context of identity-based encryption.[13] The third method is relying on Φ-key-collision-security [ABPP14] in the context of PRFs, which requires that for a random key k it is impossible to find two distinct $\phi_1, \phi_2 \in \Phi$ such that $\phi_1(k) = \phi_2(k)$. However, such property is only known to hold w.r.t. specific Φ under concrete number-theoretical assumptions.

Insight in Addressing Copy Attacks. As discussed above, non-trivial copy attacks have not been well addressed at a general level. Being aware of the similarity between our non-malleability notion and the RKA security notion, we are curious to know if our strengthening of allowing $\phi(x^*) = x^*$ can shed light on this problem. Recall that in the proof of Lemma 5.4 for the case of $y = y^*$, we essentially proved that by assuming the one-wayness of f, no PPT adversary is able to find a $\phi \in \Phi^{\mathsf{srs}}_{\mathsf{brs}}$ such that $\phi(x^*) = x^*$ with non-negligible probability. The high-level idea is that as long as the adversary is able to find such a $\phi \in \Phi^{\mathsf{srs}}_{\mathsf{brs}}$, then a reduction can obtain an efficiently solvable equation about x^*. Somewhat surprisingly, this idea immediately indicates that w.r.t. RKDT class $\Phi = \Phi^{\mathsf{srs}}_{\mathsf{brs}} \cup \mathsf{id} \cup \mathsf{cf}$, resilience against non-trivial copy attacks is in fact a built-in immunity guaranteed by the security of starting primitives.

We sketch the argument more formally as follows. Let $\mathcal{A}$ be a RKA adversary and denote by E the event that non-trivial attack happens, i.e., $\mathcal{A}$ makes at least one RKA query $\langle \phi, x^* \rangle$ such that $\phi \in \Phi^{\mathsf{srs}}_{\mathsf{brs}}$ and $\phi(sk) = sk$. Let $l(\lambda)$ be the maximum number of RKA queries $\mathcal{A}$ makes. Our aim is to prove $\Pr[E] = \mathsf{negl}(\lambda)$ by only assuming the original security of the starting primitives. Conditioned on E happens, a reduction $\mathcal{R}$ can pick out a non-trivial copy attack query say $\langle \phi, x^* \rangle$ and hence obtains a right equation $\phi(sk) = sk$ about sk, with probability at least $1/l(\lambda)$. Conditioned on getting the right equation, $\mathcal{R}$ can further compute the correct sk with probability $1/\mathsf{poly}(\lambda)$ due to the BRS & SRS properties of $\Phi^{\mathsf{srs}}_{\mathsf{brs}}$. Overall, $\mathcal{R}$ is able to recover sk with probability $\Pr[E]/l(\lambda)\mathsf{poly}(\lambda)$. Since $\mathcal{A}$ is a PPT adversary, $l(\lambda)$ is poly-bounded. Therefore, if $\Pr[E]$ is non-negligible, then $\mathcal{R}$ can recover sk with non-negligible probability. This immediately contradicts the security of the starting primitives, and therefore we must have $\Pr[E] = \mathsf{negl}(\lambda)$.

Our result indicates that w.r.t. RKDT class $\Phi \subseteq \Phi^{\mathsf{srs}}_{\mathsf{brs}} \cup \mathsf{id} \cup \mathsf{cf}$, resilience against non-trivial copy attacks is essentially a built-in security guaranteed by the starting primitives. Previous RKA-secure schemes w.r.t. algebra-induced RKDTs could benefit from this, that is, "weak" RKA security (disallowing non-trivial copy attacks) can be enhanced automatically without resorting to claw-free assumption or additional properties/transformations.

8 Application to RKA-secure Authenticated KDFs

8.1 Continuous Non-Malleable KDFs, Revisit

Recently, Qin et al. [QLY+15] extended non-malleable key derivation functions (KDFs) [FMVW14] to continuous non-malleable KDFs, and showed how to use it to compile numerous cryptographic primitives into RKA-secure ones. In what follows, we briefly recall the syntax, security notion, as well as construction of continuously non-malleable KDFs presented in [QLY+15].

Syntax. KDFs consist of three polynomial time algorithms: (1) $\mathsf{Setup}(\lambda)$, on input λ, outputs system-wide public parameters pp, which define the key space S, the public key space Π, and

[13] Taking Φ-fingerprinting for instance: briefly, this property requires that $\phi(sk) \neq sk$ always invalidates the challenge ciphertext c^*. Notice that queries $\langle \phi, c^* \rangle$ such that $\phi(sk) = sk$ are already forbidden by the definition, the reduction can thus safely reject all RKA queries of the form $\langle \phi, c^* \rangle$ without even looking at ϕ, since either case $\phi(sk) = sk$ or case $\phi(sk) \neq sk$ yields the same output $\perp$ with respect to c^*.

the derived key space $\{0,1\}^m$. (2) $\mathsf{Sample}(pp)$, on input pp, samples a random key $s \xleftarrow{R} S$ and computes public key $\pi \in \Pi$. (3) $\mathsf{Derive}(s,\pi)$, on input (s,π), outputs a derived key $r \in \{0,1\}^m$ or $\perp$ indicating that π is not a valid proof of s.

SECURITY NOTIONS. The continuous non-malleability of KDFs is defined w.r.t. a transformation class Φ, which states that no PPT adversary can distinguish a real derived key $r \leftarrow \mathsf{Derive}(s^*,\pi^*)$ from a random one, even it can continuously query a key derivation oracle $\mathcal{O}^{\Phi}_{\mathsf{derive}}(\cdot,\cdot)$, which on input $\phi \in \Phi$ and $\pi \in \Pi$ returns a special symbol same^* if $(\phi(s^*),\pi) = (s^*,\pi^*)$ and returns $\mathsf{Derive}(\phi(s^*),\pi)$ otherwise.

CONSTRUCTION. Let $\mathrm{LF} = (\mathsf{Gen},\mathsf{Eval},\mathsf{LTag})$ be a collection of one-time lossy filters [QL13] with domain S, range Y, and tag space $T = \{0,1\}^* \times T_c$. Let $\mathrm{OTS} = (\mathsf{Gen},\mathsf{Sign},\mathsf{Vefy})$ be a strongly one-time signature. Let $\mathcal{H}$ be a family of pairwise independent functions from S to $\{0,1\}^m$. The construction is as below.

- $\mathrm{KDF}.\mathsf{Setup}(\lambda)$: run $(ek,td) \leftarrow \mathrm{LF}.\mathsf{Gen}(\lambda)$, pick $h \xleftarrow{R} \mathcal{H}$, output $pp = (ek,h)$. Precisely, pp also includes the public parameters of LF and OTS.
- $\mathrm{KDF}.\mathsf{Sample}(pp)$: run $(vk,sk) \leftarrow \mathrm{OTS}.\mathsf{Gen}(\lambda)$, pick $t_c \xleftarrow{R} T_c$, $s \xleftarrow{R} S$; compute $y \leftarrow \mathrm{LF}.\mathsf{Eval}(ek,(vk,t_c),s)$ and $\sigma \leftarrow \mathrm{OTS}.\mathsf{Sign}(sk,t_c||y)$, then set $t = (vk,t_c,y,\sigma)$, and finally output (s,t).
- $\mathrm{KDF}.\mathsf{Derive}(s,t)$: parse $t = (vk,t_c,y,\sigma)$, output $h(s)$ if $\mathrm{LF}.\mathsf{Eval}(ek,(vk,t_c),s) = y$ and $\mathrm{OTS}.\mathsf{Vefy}(vk,t_c||y,\sigma) = 1$ hold simultaneously, and output $\perp$ otherwise.

Discussion. In standard KDFs, there is no the concept of "public key", and the key derivation algorithm never fails. In contrast, in the KDFs introduced by Qin et al. [QLY+15], each key s is accompanied with an auxiliary "public key" π, and the key derivation algorithm reports failure by outputting $\perp$ if π does not match s. Thus, it is preferable to use the name authenticated KDFs to highlight this functional difference. In addition, π is interpreted as a proof of knowledge of s in [QLY+15] . However, in the context of KDFs the key s is not necessarily belong to any $\mathcal{NP}$ language. In this regard, it is more appropriate to simply view π as a tag of s, which we will denote by t.

We then reconsider its security notion. The continuous non-malleable notion considered in [QLY+15] is potentially weak in that key derivation queries of the form $\langle \phi,\pi^* \rangle$ with $\phi(s^*) = s^*$ are implicitly rejected by returning same^*. As a consequence, this notion cannot guarantee the resilience against non-trivial copy attacks for its enabling RKA-secure schemes. Besides, conventionally non-malleability is used to capture the inability to maul the value of a cryptographic primitive in a controlled way, whereas RKA security ensures that a cryptographic primitive remains secure even an adversary may adaptively learn functions of a sequence of related keys. In light of this distinction, we refer to continuous non-malleability as related-key security and use the term "RKA-secure authenticated KDFs" instead of continuous non-malleable KDFs in the rest of this work.

Qin et al.'s construction requires one-time lossy filter, one-time signature, and pairwise-independent functions as ingredients. Though ingenious, their construction is somewhat complicated and expensive. Its public parameters consist of that of three ingredients as well as an evaluation key; to compute a tag for a random key, its sampling procedure has to generate a fresh key pair for a one-time signature scheme, pick a random tag, evaluate a function and also compute a signature; to derive a random key, its key derivation procedure has to verify a signature and a function value before deriving. Compared to standard KDFs, these do add noticeable storage and computation overhead, which could be critical in resource-constrained scenarios, e.g., embedded systems and low-end smart card.

8.2 RKA-secure Authenticated KDFs

Based on the above discussions, we are motivated to enhance the current security notion as well as give a simple yet efficient construction for RKA-secure authenticated KDFs (AKDFs) w.r.t. general RKDT class. For completeness, we first present authenticated KDFs with the refined terminology and enhanced security notions.

Definition 8.1 (Authenticated KDFs)**.** Authenticated KDFs are given by three polynomial time algorithms as follows:

- $\mathsf{Setup}(\lambda)$: on input λ, output system parameters pp, which define the derivation key space S, the tag space T, and the derived key space $\{0,1\}^m$.
- $\mathsf{Sample}(pp)$: on input pp, pick a random key $s \xleftarrow{\mathrm{R}} S$ computes it associated tag $t \in T$, output (s,t).
- $\mathsf{Derive}(s,t)$: on input a key $s \in S$ and a tag $t \in T$, output a derived key $r \in \{0,1\}^m$ or a rejecting symbol $\perp$ indicating that t is not a valid tag of s.

Definition 8.2 (RKA-security)**.** AKDFs are said to be Φ-RKA-secure w.r.t. RKDT class Φ if for any PPT adversary $\mathcal{A}$ its advantage $\mathsf{Adv}^{\mathrm{rka}}_{\mathcal{A},\mathrm{AKDF}}$ defined in the following experiment is negligible in λ.

$$\mathsf{Adv}^{\mathrm{rka}}_{\mathcal{A},\mathrm{AKDF}}(\lambda) = \Pr\left[b' = b : \begin{array}{l} pp \leftarrow \mathsf{Setup}(\lambda); \\ (s^*,t^*) \leftarrow \mathsf{Sample}(pp); \\ r_0^* \leftarrow \mathsf{Derive}(s^*,t^*), r_1^* \xleftarrow{\mathrm{R}} \{0,1\}^m; \\ b \xleftarrow{\mathrm{R}} \{0,1\}; \\ b' \leftarrow \mathcal{A}^{\mathcal{O}^{\Phi}_{\mathsf{derive}}(\cdot,\cdot)}(pp,t^*,r_b^*); \end{array}\right] - \frac{1}{2}.$$

Here $\mathcal{O}^{\Phi}_{\mathsf{derive}}(\phi,\pi)$ on input $\phi \in \Phi$ and $t \in T$, returns a special symbol same^* only if $\underline{\phi = \mathsf{id} \text{ and } t = t^*}$, and returns $\mathsf{Derive}(\phi(s^*),t)$ otherwise.

Our RKA security notion is strong in the sense that only trivial query (underlined as above) is not allowed. By Qin et al.'s result [QLY+15], one can use RKA-secure AKDFs to transform a cryptographic primitive to a RKA-secure one in a modular way, as long as the key generation algorithm of the primitive takes uniform random coins to generate (public/secret) keys. Notably, this transform naturally transfers our strong RKA security of AKDFs to the resulting RKA-secure primitives.

8.3 RKA-secure AKDFs from Non-Malleable Functions

Before presenting our construction, we first sketch the high-level idea, which we think may be useful in other places. The main technical hurdle in constructing RKA-secure AKDFs is to answer related key derivation queries without knowing the secret key s^*. As we recalled in Section 7, a common approach addressing this hurdle is exploiting key-malleable like property to simulate RKA oracle based on the standard oracle of the starting primitive. However, this approach does not fit for our purpose. On one hand, efficient construction of the starting primitive namely AKDFs is yet unknown to us. On the other hand, key-malleable like property (if exists) is usually tied to some specific algebraic structure and thus cannot yield RKA-security w.r.t. general RKDT class. Here we take a complementary approach, that is, acquiring RKA security from non-malleability. Instead of trying to answer RKA queries, we aim to reject all RKA queries. We do so by stipulating that even after seeing a valid tag t^* of s^*, no PPT adversary is able to generate a legal related key derivation query (ϕ,π) (here legal means t is

a valid tag of $\phi(s^*)$). In this way, the reduction can handle all related key derivation queries without knowing s^*, by simply returning $\perp$.

With this strategy, an incredibly simple construction of RKA-secure AKDFs comes out by twisting NMFs: Let $\mathcal{F}$ be a family of poly-to-one NMFs. The Setup algorithm randomly pick f from $\mathcal{F}$ and let h be a hardcore function of f. To generate a tag for a random key, one simply computes $t \leftarrow f(s)$. Intuitively, t serves as a deterministic non-malleable tag of s. To get a derived key from (s, t), one first checks if $f(s) = t$ and then outputs $r \leftarrow h(s)$ if so. On a high (and not entirely precise) level, due to the non-malleability of the underlying NMFs, all related-key derivation queries can be safely rejected, and thus the pseudorandomness of the derived key can be reduced to the one-wayness of f. A subtlety here is that in addition to t^* the adversary can obtain some auxiliary information about s^*, namely the real or random derived key. In this regard, hinted non-malleability is required for $\mathcal{F}$. We present our generic construction and formal security proof in details as below.

Let $\mathcal{F} = (\mathsf{Gen}, \mathsf{Samp}, \mathsf{Eval})$ be a family of Φ-non-malleable poly-to-one functions and $\mathcal{H}$ be its hardcore that maps D to $\{0,1\}^m$. We show how to build Φ'-RKA-secure AKDFs from it, where $\Phi' = \Phi \cup \mathsf{id} \cup \mathsf{cf}$.[14]

- AKDF.$\mathsf{Setup}(\lambda)$: run $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, $h \leftarrow \mathcal{H}.\mathsf{Gen}(\lambda, f)$, output $pp = (f, h)$.
- AKDF.$\mathsf{Sample}(pp)$: sample $s \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$, compute $t \leftarrow f(s)$, output (s, t).
- AKDF.$\mathsf{Derive}(s, t)$: if $t \neq f(s)$, output $\perp$; otherwise output $r \leftarrow h(s)$.

The RKA security of the above construction follows from the theorem below.

Theorem 8.1. *The above construction of AKDFs is Φ'-RKA-secure if $\mathcal{F}$ is Φ-non-malleable and poly-to-one, where $\Phi' = \Phi \cup \mathsf{id} \cup \mathsf{cf}$.*

Proof. We prove this theorem via a sequence of games. Let S_i be the event that $\mathcal{A}$ wins in Game i.

Game 0 (The real experiment): $\mathcal{CH}$ interacts with $\mathcal{A}$ as follows:

1. $\mathcal{CH}$ picks $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, $h \leftarrow \mathcal{H}.\mathsf{Gen}(\lambda, f)$, sets $pp = (f, h)$; then picks $s^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$, computes $t^* \leftarrow f(s^*)$, $r_0^* \leftarrow h(s^*)$, $r_1^* \xleftarrow{\mathrm{R}} \{0,1\}^m$. Finally, $\mathcal{CH}$ picks $b \xleftarrow{\mathrm{R}} \{0,1\}$, sends (pp, t^*, r_b^*) to $\mathcal{A}$ as the challenge.
2. Upon receiving a RKA key derivation query $\langle \phi, t \rangle$ from $\mathcal{A}$, if $\langle \phi, t \rangle = \langle \mathsf{id}, t^* \rangle$, $\mathcal{CH}$ returns same^*; else $\mathcal{CH}$ returns $h(\phi(s^*))$ if $\phi(s^*) = t$ and $\perp$ otherwise.
3. $\mathcal{A}$ outputs a guess b' for b and wins if $b' = b$.

According to the definition of $\mathcal{A}$, we have:

$$\mathsf{Adv}^{\mathrm{rka}}_{\mathcal{A},\mathrm{AKDF}}(\lambda) = |\Pr[S_0] - 1/2|. \tag{3}$$

Game 1 (Handling trivial queries without s^*): The same as Game 0 except that in step 2 $\mathcal{CH}$ handles trivial queries $\langle \phi, t \rangle$ without s^*. Here the term "trivial" means $\phi \in \mathsf{id} \cup \mathsf{cf}$. We break trivial queries into three cases:

- $\phi = \mathsf{id}$ and $t = t^*$: return same^* indicating the query is illegal.
- $\phi = \mathsf{id}$ and $t \neq t^*$: return $\perp$ indicating the query is invalid. This is because f is a deterministic function hence each s has an unique tag.

[14] As we discussed in Section 3, non-malleability is impossible to achieve if Φ contains id or constant transformations. Thus, we assume $\Phi \cap (\mathsf{id} \cup \mathsf{cf}) = \emptyset$.

- $\phi \in \mathsf{cf}$ and t: suppose ϕ is a constant transform that maps all its inputs to a constant c, then return $h(c)$ if $f(c) = t$ and $\perp$ otherwise.

These modifications are purely conceptual and hence

$$\Pr[S_1] = \Pr[S_0]. \tag{4}$$

Game 2 (Handling all queries without s^*): The same as Game 1 except $\mathcal{CH}$ directly returns $\perp$ for all non-trivial queries $\langle \phi, t \rangle$. Here the term "non-trivial" means $\phi \in \Phi$. Let E be the event that $\mathcal{A}$ issues a non-trivial query $\langle \phi, t \rangle$ such that $t = f(\phi(s^*))$. According to the definitions of Game 1 and Game 2, if this event happens, $\mathcal{CH}$ returns $\perp$ in Game 2, but not in Game 1. It is easy to see that unless event E occurs, Game 1 and Game 2 are identical from adversary's view. By the difference lemma, it follows that:

$$|\Pr[S_2] - \Pr[S_1]| \leq \Pr[E]. \tag{5}$$

We now state and prove two claims that establish the main theorem.

Lemma 8.2. $\Pr[E]$ *is negligible in* λ *assuming the* Φ*-non-malleability of* $\mathcal{F}$*.*

What we need to show is that after seeing t^* and the auxiliary information r_b^* about s^*, no PPT adversary is able to output a valid non-trivial RKA query $\langle \phi, t \rangle$ such that $\phi(s^*) = t$. Therefore, hint-free non-malleability is inadequate and hinted non-malleability is needed. Notice that here the auxiliary information r_b^* is exactly $\mathsf{hint}(s^*; b)$ where hint is the special hint function defined in Equation (1). As we have shown Section 6, hinted non-malleability w.r.t. this hint function is implied by hint-free non-malleability.

Proof. Suppose $\mathcal{B}$ is an adversary against hinted Φ-non-malleability of $\mathcal{F}$ w.r.t. the hint function defined in Equation (1). Given $(f, y^*, \mathsf{hint}(x^*; b))$, where $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, $y^* \leftarrow f(x^*)$ for $x^* \leftarrow \mathcal{F}.\mathsf{Samp}(\lambda)$, and $b \xleftarrow{R} \{0,1\}$. $\mathcal{B}$ simulates $\mathcal{A}$'s challenger in Game 2 as below: set $pp = (f, h)$,[15] $t^* = y^*$, $r_b^* \leftarrow \mathsf{hint}(x^*; b)$, then send (pp, t^*, r_b^*) to $\mathcal{A}$. Here s^* is implicitly set as x^*, which is unknown to $\mathcal{B}$. This is not a problem since according to the definition of Game 2, $\mathcal{B}$ is able to handle all RKA queries correctly without s^*. Let L be the list of all non-trivial queries issued by $\mathcal{A}$. Since $\mathcal{A}$ is a PPT adversary, we have $|L| \leq \mathsf{poly}(\lambda)$. At the end of the simulation, $\mathcal{B}$ picks a random tuple (ϕ, t) from the L list as its answer against hinted Φ-non-malleability. Conditioned on E happens, $\mathcal{B}$ succeeds with probability at least $1/\mathsf{poly}(\lambda)$. Therefore, if $\Pr[E]$ is non-negligible, then $\mathcal{B}$'s advantage is at least $\Pr[E]/\mathsf{poly}(\lambda)$, which is also non-negligible. This breaks the hinted Φ-non-malleability of $\mathcal{F}$, which in turn contradicts the assumed hint-free Φ-non-malleability of $\mathcal{F}$ in this case. The lemma immediately follows. □

Lemma 8.3. $|\Pr[S_2] - 1/2| = \mathsf{negl}(\lambda)$ *assuming the* Φ*-non-malleability of* $\mathcal{F}$*.*

Proof. Since $\mathcal{F}$ is poly-to-one, then according to Lemma 5.1 Φ-non-malleability implies one-wayness, and further implies pseudorandomness of its hardcore $\mathcal{H}$. Thereby, it suffices to prove $|\Pr[S_2] - 1/2| = \mathsf{negl}(\lambda)$ assuming the pseudorandomness of $\mathcal{H}$. Suppose $\mathcal{B}$ is an adversary against pseudorandomness of hardcore $\mathcal{H}$ associated with $\mathcal{F}$. Given (f, h, y^*, r_b^*), where $y^* \leftarrow f(x^*)$ for $x^* \xleftarrow{R} D$ and r_b^* is either $h(x^*)$ when $b = 0$ or a random string from $\{0,1\}^m$ when $b = 1$, $\mathcal{B}$ simulates $\mathcal{A}$'s challenger in Game 2 as follows: set $pp = (f, h)$, $t^* = y^*$, send (pp, t^*, r_b^*) to $\mathcal{A}$. According to the definition of Game 2, $\mathcal{B}$ can handle all the queries without the knowledge of $s^* = x^*$. At the end of the game, $\mathcal{B}$ simply forwards $\mathcal{A}$'s output as its guess. It is easy to see that if $\mathcal{A}$ succeeds, so does $\mathcal{B}$. Therefore, we have $\mathsf{Adv}^{\mathrm{rand}}_{\mathcal{B},\mathcal{H}}(\lambda) \geq |\Pr[S_2] - 1/2|$. By the hypothesis that $\mathcal{H}$ is pseudorandom, we have $|\Pr[S_2] - 1/2| = \mathsf{negl}(\lambda)$. This proves the lemma. □

[15] The description of h is implicit in hint.

Putting it all together, the theorem immediately follows. □

By instantiating our generic construction with poly-to-one NMFs w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$ (which in turn can be constructed from ATDFs), we obtain RKA-secure AKDFs w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id} \cup \mathsf{cf}$.

While both our construction and Qin et al.'s construction are generic, it is still instructive to make a rough comparison. In terms of efficiency, benefit from the fact that building solely from deterministic NMFs, in our construction the public parameters consist of merely the descriptions of a NMF f and a hardcore function h, and the tag generation and authentication procedures are both deterministic. By comparison, Qin et al.'s construction is built from three different cryptographic primitives, and thus its public parameters size is large and its tag generation procedure is randomized. In this regard, our construction has potential advantages over Qin et al.'s construction in terms of small footprint of cryptographic code, compact public parameters size, short tag size, as well as quick tag generation and authentication. In terms of security, our construction is RKA-secure in the strong sense w.r.t. generic RKDT class with a direct and modular proof, whereas Qin et al.'s construction is RKA-secure w.r.t. specific RKDT class [**?**] with a bit involved proof.

8.4 Optimizations

Relaxation on NMFs. We observe that in the above construction, NMFs can be relaxed to non-malleable verifiable relations (NMVRs). In NMVRs, instead of requiring f to be efficiently computable, we only require that the distribution $(x, f(x))$ for a random x is efficiently sampleable and the correctness of sampling is publicly verifiable.[16] It is easy to see that NMVRs are implied by adaptive trapdoor relations (ATDRs) [Wee10] with publicly verifiability. As shown in [XLLL14], publicly verifiable ATDRs can be constructed from all-but-one verifiable lossy trapdoor relations, which permit efficient realizations from a variety of standard assumptions. Combining this result with our observation above, we are able to give more efficient constructions of RKA-secure AKDFs.

Stronger RKA security. In the above RKA security notion for AKDFs, the adversary is only given access to a RKA oracle. In practice, it may also collect some tags and learn the corresponding derivation keys. To defend against such powerful adversaries, it is necessary to make the RKA security stronger by giving the adversary access to a reveal oracle $\mathcal{O}_{\mathsf{reveal}}$ that on input a tag t outputs a corresponding key s.[17] AKDFs satisfying such strong RKA notion can be constructed from adaptive NMFs, which in turn can be constructed from ATDFs. This not only justifies the utility of the adaptive non-malleable notion, but also supports the assertion made by Kiltz et al. [KMO10] that "ATDFs may be useful in the general context of black-box constructions of cryptographic primitives secure against adaptive attacks."

Increasing the length of derivation key. We can always instantiate h via the Goldreich-Levin hardcore predicate [GL89]. Nevertheless, such general instantiation yields only one-bit derived key. We may also obtain a hardcore function with linearly-many hardcore bits either by iteration when $\mathcal{F}$ is a family of one-way permutations or relying on stronger decisional assumptions. A recent work [BST14] provides us an appealing hardcore function with poly-many hardcore bits from any one-way functions, assuming the existence of differing-inputs/indistinguishability obfuscation. In applications of RKA-secure AKDFs where the length of the derived key is of great importance, one can further stretch it by applying a normal PRG.

[16] Here the publicly verifiable property means verification can be done without knowing the secret random coins used in sampling.

[17] Query on the challenge tag t^* is not allowed to avoid trivial attack.

Acknowledgment

We particularly thank Zongyang Zhang for bring the work [BFS11] to our attention. We are grateful to Qiong Huang, Marc Fischlin, Jinyong Chang and Fei Tang for helpful discussions and advice. We also thank the anonymous reviewers of PKC 2016 for their useful comments.

Yu Chen is supported by the National Natural Science Foundation of China (Grant No. 61303257), the IIE's Cryptography Research Project (Grant No. Y4Z0061B02), and the Strategic Priority Research Program of CAS (Grant No. XDA06010701). Baodong Qin is supported by the National Natural Science Foundation of China (Grant No. 61502400, 61373153 and 61572318). Jiang Zhang is supported by the National Basic Research Program of China (Grant No. 2013CB338003). Yi Deng is supported by the National Natural Science Foundation of China (Grant No. 61379141), the IIE's Cryptography Research Project (Grant No. Y4Z0061802), and the State Key Laboratory of Cryptology's Open Project (Grant No. MMKFKT201511). Sherman S.M. Chow is supported by the Early Career Scheme and the Early Career Award of the Research Grants Council, Hong Kong SAR (CUHK 439713), and Direct Grant of the Chinese University of Hong Kong (No. 4055018).

References

[ABPP14] Michel Abdalla, Fabrice Benhamouda, Alain Passelègue, and Kenneth G. Paterson. Related-key security for pseudorandom functions beyond the linear barrier. In *Advances in Cryptology - CRYPTO 2014*, volume 8616 of *LNCS*, pages 77–94. Springer, 2014.

[AHI11] Benny Applebaum, Danny Harnik, and Yuval Ishai. Semantic security under related-key attacks and applications. In *Innovations in Computer Science - ICS 2010*, pages 45–60, 2011.

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In *Advances in Cryptology - CRYPTO 2007*, volume 4622 of *LNCS*, pages 535–552. Springer, 2007.

[BC10] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 666–684. Springer, 2010.

[BCFW09] Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 524–541. Springer, 2009.

[BCM11] Mihir Bellare, David Cash, and Rachel Miller. Cryptography secure against related-key attacks and tampering. In *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 486–503. Springer, 2011.

[BDL97] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults (extended abstract). In *Advances in Cryptology - EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 37–51. Springer, 1997.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology - CRYPTO 1998*, volume 1462 of *LNCS*, pages 26–45. Springer, 1998.

[Ber70] E. R. Berlekamp. Factoring polynomials over large finite fields. *Math. Comput.*, 24, 1970.

[BF06] Alexandra Boldyreva and Marc Fischlin. On the security of OAEP. In *Advances in Cryptology - ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 210–225. Springer, 2006.

[BFS11] Paul Baecher, Marc Fischlin, and Dominique Schröder. Expedient non-malleability notions for hash functions. In *CT-RSA 2011*, volume 6558 of *LNCS*, pages 268–283. Springer, 2011.

[BHSV98] Mihir Bellare, Shai Halevi, Amit Sahai, and Salil P. Vadhan. Many-to-one trapdoor functions and their ralation to public-key cryptosystems. In *CRYPTO 1998*, volume 1462 of *LNCS*, pages 283–298. Springer, 1998.

[BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: Rka-prps, rka-prfs, and applications. In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 491–506. Springer, 2003.

[BPT12] Mihir Bellare, Kenneth G. Paterson, and Susan Thomson. RKA security beyond the linear barrier: Ibe, encryption and signatures. In *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 331–348. Springer, 2012.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73. ACM, 1993.

[BS97] Eli Biham and Adi Shamir. Differential fault analysis of secret key cryptosystems. In *Advances in Cryptology - CRYPTO 1997*, volume 1294 of *LNCS*, pages 513–525. Springer, 1997.

[BS99] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In *Advances in Cryptology - CRYPTO 1999*, volume 1666 of *LNCS*, pages 519–536. Springer, 1999.

[BST14] Mihir Bellare, Igors Stepanovs, and Stefano Tessaro. Poly-many hardcore bits for any one-way function and a framework for differing-inputs obfuscation. In *Advances in Cryptology - ASIACRYPT 2014*, volume 8874 of *LNCS*, pages 102–121. Springer, 2014.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008*, volume 5126 of *LNCS*, pages 449–460. Springer, 2008.

[CIO98] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-interactive and non-malleable commitment. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, STOC 1998*, pages 141–150. ACM, 1998.

[CKOS01] Giovanni Di Crescenzo, Jonathan Katz, Rafail Ostrovsky, and Adam Smith. Efficient and non-interactive non-malleable commitment. In *Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 40–59. Springer, 2001.

[CV09] Ran Canetti and Mayank Varia. Non-malleable obfuscation. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *LNCS*, pages 73–90. Springer, 2009.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM J. Comput.*, 30(2):391–437, 2000.

[DH76] Whitefield Diffie and Martin E. Hellman. New directions in cryptograpgy. *IEEE Transactions on Infomation Theory*, 22(6):644–654, 1976.

[DPW10] Stefan Dziembowski, Krzysztof Pietrzak, and Daniel Wichs. Non-malleable codes. In *Innovations in Computer Science - ICS 2010*, pages 434–452. Tsinghua University Press, 2010.

[FF00] Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. In *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *LNCS*, pages 413–431. Springer, 2000.

[FMNV14] Sebastian Faust, Pratyay Mukherjee, Jesper Buus Nielsen, and Daniele Venturi. Continuous non-malleable codes. In *Theory of Cryptography - 11th Theory of Cryptography Conference, TCC 2014*, volume 8349 of *LNCS*, pages 465–488. Springer, 2014.

[FMVW14] Sebastian Faust, Pratyay Mukherjee, Daniele Venturi, and Daniel Wichs. Efficient non-malleable codes and key-derivation for poly-size tampering circuits. In *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 111–128. Springer, 2014.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, STOC 1989*, pages 25–32. ACM, 1989.

[GL10] David Goldenberg and Moses Liskov. On related-secret pseudorandomness. In *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010*, volume 5978 of *LNCS*, pages 255–272. Springer, 2010.

[GOR11] Vipul Goyal, Adam O'Neill, and Vanishree Rao. Correlated-input secure hash functions. In *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011*, volume 6597 of *LNCS*, pages 182–200. Springer, 2011.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*, pages 197–206. ACM, 2008.

[JLLM13] Dingding Jia, Xianhui Lu, Bao Li, and Qixiang Mei. RKA secure PKE based on the DDH and HR assumptions. In *Provable Security - 7th International Conference, ProvSec 2013*, volume 8209 of *LNCS*, pages 271–287. Springer, 2013.

[JW15] Zahra Jafargholi and Daniel Wichs. Tamper detection and continuous non-malleable codes. In *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015*, volume 9014 of *LNCS*, pages 451–480. Springer, 2015.

[KMO10] Eike Kiltz, Payman Mohassel, and Adam O'Neill. Adaptive trapdoor functions and chosen-ciphertext security. In *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 673–692. Springer, 2010.

[LLJ14] Xianhui Lu, Bao Li, and Dingding Jia. Related-key security for hybrid encryption. In *Information Security - 17th International Conference, ISC 2014*, volume 8783 of *LNCS*, pages 19–32. Springer, 2014.

[LPTV10] Huijia Lin, Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable zero knowledge proofs. In *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 429–446. Springer, 2010.

[Luc04] Stefan Lucks. Ciphers secure against related-key attacks. In *Fast Software Encryption, 11th International Workshop, FSE 2004*, volume 3017 of *LNCS*, pages 359–370. Springer, 2004.

[OPV08] Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Constant-round concurrent non-malleable zero knowledge in the bare public-key model. In *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008*, volume 5126 of *LNCS*, pages 548–559. Springer, 2008.

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive one-way functions and applications. In *Advances in Cryptology - CRYPTO 2008*, volume 5157 of *LNCS*, pages 57–74. Springer, 2008.

[PR05] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005*, pages 563–572. IEEE Computer Society, 2005.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*, pages 187–196. ACM, 2008.

[QL13] Baodong Qin and Shengli Liu. Leakage-resilient chosen-ciphertext secure public-key encryption from hash proof system and one-time lossy filter. In *Advances in Cryptology - ASIACRYPT 2013*, volume 8270 of *LNCS*, pages 381–400. Springer, 2013.

[QLY+15] Baodong Qin, Shengli Liu, Tsz Hon Yuen, Robert H. Deng, and Kefei Chen. Continuous non-malleable key derivation and its application to related-key security. In *Public-Key Cryptography - PKC 2015*, volume 9020 of *LNCS*, pages 557–578. Springer, 2015.

[RS10] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. *SIAM J. Comput.*, 39(7):3058–3088, 2010.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *FOCS 1999*, pages 543–553. ACM, 1999.

[vzGS92] Joachim von zur Gathen and Victor Shoup. Computing frobenius maps and factoring polynomials (extended abstract). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, STOC 1992*, pages 97–105. ACM, 1992.

[Wee10] Hoeteck Wee. Efficient chosen-ciphertext security via extractable hash proofs. In *Advances in Cryptology - CRYPTO 2010*, volume 6223 of *LNCS*, pages 314–332. Springer, 2010.

[Wee12] Hoeteck Wee. Public key encryption against related key attacks. In *Public Key Cryptography - PKC 2012*, volume 7293 of *LNCS*, pages 262–279. Springer, 2012.

[XLLL14] Haiyang Xue, Xianhui Lu, Bao Li, and Yamin Liu. Lossy trapdoor relation and its applications to lossy encryption and adaptive trapdoor relation. In *Provable Security - 8th International Conference, ProvSec 2014*, volume 8782 of *LNCS*, pages 162–177. Springer, 2014.

A An Improved Proof for the Non-Malleability of the Strengthened Merkle-Damgård Transformation

Strengthened Merkle-Damgård Transformation. Let h be a fixed-length hash function with input length $2\ell(\lambda)$ and output length $\ell(\lambda)$, $iv \in \{0,1\}^{\ell(\lambda)}$ be an initialization vector, pad be a padding function which maps a message $x \in \{0,1\}^*$ of length at most $2^{\ell(\lambda)} - 1$ to multiples of the block length $\ell(\lambda)$ such that the final block contains the message length.[18] The strengthened Merkle-Damgård transformation MD is defined as:

$$\mathsf{MD}^h_{iv}(x) := h^*_{iv}(x_1||\dots||x_k) = h(\cdots h(h(iv, x_1), x_2)\cdots)$$

where $x_1||\dots||x_k = \mathsf{pad}(x)$. In the following we denote by y_i the i-th intermediate value when iterating h, i.e. $h^*_{iv}(x_1||\dots||x_i)$.

Baecher et al. [BFS11, Proposition 4.2] proved that MD is $\oplus$-non-malleable (for fixed-length message) if the compression function h is modeled as random oracle. In the following lemma, we show that MD is essentially $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$-non-malleable.

[18] We limit the length of x to be at most $2^{\ell(\lambda)} - 1$ so that its length can fit into a single block of length $\ell(\lambda)$ bits. This is not a limitation because we assume that all messages considered are of length polynomial in λ and not exponential.

Lemma A.1. *For a random oracle* $h : \{0,1\}^{2\ell(\lambda)} \to \{0,1\}^{\ell(\lambda)}$ *where* $\ell(\lambda) = \mathsf{poly}(\lambda)$*, the hash function* $\mathsf{MD}^h_{iv} : \{0,1\}^{n(\lambda)} \to \{0,1\}^{\ell(\lambda)}$ *is* $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$*-non-malleable w.r.t. arbitrary* hint *as long as* $\tilde{\mathsf{H}}_\infty(x^*|(\mathsf{hint}(x^*), y^*)) \geq \omega(\log \lambda)$ *where* $x^* \xleftarrow{\mathrm{R}} \{0,1\}^{n(\lambda)}$ *and* $y^* = \mathsf{MD}^h_{iv}(x^*)$*.*

Proof. We prove this lemma by showing that if there exists a PPT adversary $\mathcal{A}$ that has non-negligible advantage against $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$-non-malleability of MD^h_{iv} where h is a random oracle, then we can build a PPT adversary $\mathcal{B}$ that contradicts to the hypothesis $\tilde{\mathsf{H}}_\infty(x^*|(\mathsf{hint}(x^*), y^*)) \geq \omega(\log \lambda)$. Here, $h(\cdot)$ is implemented via an external random oracle $\mathcal{O}^h_{\mathsf{ro}}(\cdot)$, which maintains a list L to track random oracle queries. For each fresh random oracle query at point $(a, b) \in \{0,1\}^{2\ell(\lambda)}$, a random value $c \xleftarrow{\mathrm{R}} \{0,1\}^{\ell(\lambda)}$ is chosen and the tuple $\langle (a, b), c\rangle$ is added into L. At the very beginning, $\mathcal{B}$ is given $y^* = \mathsf{MD}^h_{iv}(x^*)$ and $\mathsf{hint}(x^*)$ for a randomly chosen $x^* \xleftarrow{\mathrm{R}} \{0,1\}^{n(\lambda)}$ from its challenger. With the aim to recover x^*, $\mathcal{B}$ invokes $\mathcal{A}$ with $\mathsf{hint}(x^*)$ and y^* and simulates its challenger in the non-malleable security experiment. Let L_A be the subset of L which containing all the tuples indexed by $\mathcal{A}$'s random oracle queries. When $\mathcal{A}$ outputs its solution (ϕ, y), $\mathcal{B}$ recovers x^* via the following steps:

1. Let $x_1||\ldots||x_k = \mathsf{pad}(\phi(x^*))$, $y_0 = iv$, $y_k = y$ and $y_i = \mathcal{O}^h_{\mathsf{ro}}(y_{i-1}, x_i)$ for $1 \leq i \leq k$. $\mathcal{B}$ initiates a counter $j = k$, sets $y'_j = y$. $\mathcal{B}$ then randomly picks a tuple in L_A whose image is y'_j, sets the left part of the preimage as y'_{j-1}, sets the right part of the preimage as x'_j. $\mathcal{B}$ then sets the counter $j = j - 1$ and continues the above operation until $j = 0$. Finally, $\mathcal{B}$ obtains $x'_k, \ldots, x'_1$. We claim that if $\mathcal{A}$ succeeds (i.e., $\mathsf{MD}^h_{iv}(\phi(x^*)) = y$) with some negligible probability $\epsilon(\lambda)$, then $\Pr[\wedge^k_{i=1} x_i = x'_i] \geq \epsilon(\lambda)$. Let Q be the event that during the game $\mathcal{A}$ explicitly queries $\mathcal{O}^f_{\mathsf{ro}}(\cdot)$ at all intermediate points $(y_0, x_1), (y_1, x_2), \ldots, (y_{k-1}, x_k)$, and S be the event that $\mathcal{A}$ succeeds. Then we have:

$$\Pr[S] = \Pr[S \wedge \overline{Q}] + \Pr[S \wedge Q] \leq \Pr[S \wedge \overline{Q}] + \Pr[Q].$$

   Note that $\ell(\lambda) = \mathsf{poly}(\lambda)$, the output of $\mathcal{O}^h_{\mathsf{ro}}(\cdot)$ is unpredictable and $\mathcal{O}^h_{\mathsf{ro}}(\cdot)$ acts like a collision-resistant hash function. The first fact indicates that $\Pr[S \wedge \overline{Q}] = \mathsf{negl}(\lambda)$. The second fact indicates that for each $1 \leq i \leq k$, there is one and only one tuple $\langle (y_{i-1}, x_i), y_i\rangle$ (whose image is y_i) in L_A. Therefore, we must have $y'_j = y_j$ and $x'_j = x_j$ for each $j \in [k]$. This proves the above claim.

2. $\mathcal{B}$ recovers $x' \in \{0,1\}^{n(\lambda)}$ from $(x'_1, \ldots, x'_k)$. According to the above claim, if $\mathcal{A}$ succeeds with non-negligible probability ϵ, then $\Pr[\wedge^k_{i=1} x'_i = x_i] \geq \epsilon(\lambda)$ and thus $\Pr[x' = \phi(x^*)] \geq \epsilon$. $\mathcal{B}$ then runs SampRS to output a random solution of equation $\phi(\alpha) - x' = 0$ as its answer. Combine the fact $\Pr[x' = \phi(x^*)] \geq \epsilon(\lambda)$ and the BRS & SRS properties of $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$, we conclude that $\mathcal{B}$ outputs x^* probability $\epsilon/\mathsf{poly}(\lambda)$, which is still non-negligible in λ.

During the recovering procedure, $\mathcal{B}$ only uses the information from $\mathcal{A}$'s random oracle queries. Therefore, the existence of $\mathcal{B}$ contradicts the hypothesis that $\tilde{\mathsf{H}}_\infty(x^*|(\mathsf{hint}(x^*), y^*)) \geq \omega(\log \lambda)$. This proves the above lemma. □