Non-Malleable Functions and Their Applications

Yu Chen * Baodong Qin † Jiang Zhang ‡ Yi Deng § Sherman S.M. Chow ¶

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 focus on basic functions, rather than one-way/hash functions considered in the literature of NMOWHFs.

We mainly follow Baecher et al. to formalize a game-based definition for NMFs. 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 transformation ϕ from some prefixed transformation class s.t. $y = f(\phi(x^*))$. A distinctive strengthening of our non-malleable notion is that ϕ such that $\phi(x^*) = x^*$ is allowed. We also consider adaptive non-malleability, which stipulates that non-malleability holds even when an inversion oracle is available.

We investigate the relations between non-malleability and one-wayness in depth. In 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 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 results establish theoretical connections between non-malleability and one-wayness for functions, which extend to trapdoor functions as well, and thus resolve the open problems left by Kiltz et al. (Eurocrypt 2010). We also study the relations between standard OW/NM and hinted OW/NM, where the latter notions are typically more useful in practice. Towards efficient realizations of NMFs, we give a deterministic construction from adaptive trapdoor functions and a randomized construction from all-but-one lossy functions and one-time signature. This partially solves an open problem posed by Boldyreva et al. (Asiacrypt 2009).

Finally, we explore applications of NMFs in security against related-key attacks (RKA). We first show that the implication AOW $\Rightarrow$ ANM provides key conceptual insight into addressing non-trivial copy attacks in RKA security. We then show that NMFs give rise to a generic construction of continuous non-malleable key derivation functions, which have proven to be very useful in achieving RKA security for numerous cryptographic primitives. Particularly, our construction simplifies and clarifies the construction by Qin et al. (PKC 2015).

*State Key Laboratory of Information Security (SKLOIS), Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. & Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. Email: yuchen.prc@gmail.com

†School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang, China. Email: baodong.qin@gmail.com

‡State Key Laboratory of Cryptology, Beijing, China. Email: jiangzhang09@gmail.com

§State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China; & State Key Laboratory of Cryptology, Beijing, China. Email: ydeng.cas@gmail.com

¶Department of Information Engineering, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. Email: sherman@ie.cuhk.edu.hk

Contents

1 Introduction **2**
- 1.1 Related Work . . . 2
- 1.2 Motivation . . . 3
- 1.3 Our Contributions . . . 3
- 1.4 Additional Related Work . . . 5

2 Preliminaries **6**
- 2.1 Randomness Extraction . . . 7
- 2.2 Implications and Separations . . . 7

3 One-Way and Non-Malleable Functions **8**

4 Transformation Class **10**
- 4.1 Concrete Transformation Classes . . . 10
- 4.2 General Transformation Class . . . 11

5 Relations Among Non-Malleability and One-Wayness **12**
- 5.1 Non-Malleability $\Rightarrow$ One-Wayness . . . 12
- 5.2 One-Wayness $\not\Rightarrow$ Non-Malleability . . . 13
- 5.3 Adaptive Non-Malleability $\Leftrightarrow$ Adaptive One-Wayness . . . 14
- 5.4 Non-Malleability $\not\Rightarrow$ Adaptive Non-Malleability . . . 15

6 Relation Between Hint-free and Hinted Notions **16**
- 6.1 OW vs. Hinted-OW . . . 17
- 6.2 NM vs. Hinted-NM . . . 18

7 Constructions of NMFs **19**
- 7.1 Deterministic NMFs from Injective ATDFs . . . 19
- 7.2 Randomized NMFs from ABO Lossy Functions and One-Time Signature . . . 19

8 Built-in Resilience against Non-trivial Copy Attacks **22**
- 8.1 RKA-security Model and Copy Attacks . . . 22
- 8.2 Known Techniques in Tackling Non-trivial Copy Attacks . . . 22
- 8.3 Our Insight in Addressing Non-trivial Copy Attacks . . . 23

9 Application to RKA-secure Authenticated KDFs **24**
- 9.1 Continuous Non-Malleable KDFs, Revisited . . . 24
- 9.2 RKA-secure Authenticated KDFs . . . 25
- 9.3 RKA-secure AKDFs from Non-Malleable Functions . . . 25
- 9.4 Optimizations . . . 27

A An Improved Proof for the Non-Malleability of the Strengthened Merkle-Damgård Transformation **31**

B Missing Cryptographic Primitives **32**
- B.1 Universal Hash Functions . . . 32
- B.2 Strongly Unforgeable One-Time Signatures . . . 33
- B.3 All-But-One Lossy Functions . . . 33
- B.4 One-Time Lossy Filters . . . 33

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 of Dolev, Dwork and Naor [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, the study dedicated to non-malleability for 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 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 definition of non-malleability,[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 for the following: 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-malleability notion w.r.t. bit flips, and their non-malleability 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 one-way and collision resistant hash functions. In their cases, the underlying functions are already one-way.[3] This treatment somewhat blurs the relations between non-malleability and one-wayness in that it implicitly assumes the former notion is not implied by the later one.
- 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 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 inefficient for practical applications [BCFW09] (e.g.,

[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] The basic design principle for cryptographic hash functions is one-wayness.

cryptographic puzzles for network protocols). Baecher et al. [BFS11] proved that the strengthened Merkle-Damgård transformation is non-malleable w.r.t. exclusive or transformation class. However, its non-malleability inherently relies on modeling the compression function as a random oracle [BFS11]. Efficient constructions of NMOWHFs in the standard model was left open since [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. Exploring other applications of NMOWHFs were expected in [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 (OWFs) are a family of 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 (TDFs) called adaptive TDFs (ATDFs), which remain one-way even when 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 showed how to construct ATDFs from either lossy TDFs [PW08] or correlated-product TDFs [RS09]. 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 state of the art of NMOWHFs is not entirely satisfactory. In particular, the study of non-malleability dedicated to basic 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 basic functions, rather than one-way/hash functions considered in prior works [BCFW09, BFS11]. Apart from being a natural question which deserves study in its own right, a direct treatment of 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 NMFs in the standard model, by leveraging a vast body of works on OWFs. Third, the conceptual simplicity makes 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 functions? What are the relations between non-malleability and one-wayness? Can we construct efficient NMFs in the standard model? Are there new appealing applications of NMFs?

1.3 Our Contributions

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

Non-malleable functions. In Section 3, we introduce a new cryptographic primitive called non-malleable functions (NMFs), which simplifies and relaxes NMOWHFs in that the underlying functions are not required to have any cryptographic property. Loosely speaking, NMFs stipulate that 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 functions w.r.t. a transformation class Φ, that is, given $y^* \leftarrow f(x^*)$ for a randomly chosen x^*, no PPT adversary is able to output a transformation $\phi \in \Phi$ and a function value y such that $y = f(\phi(x^*))$.

In our definition, adversary's power is neatly expressed through Φ. Particularly, ϕ such that $\phi(x^*) = x^*$ 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 8 and Section 9, 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 when the adversary has access to an inversion oracle. This stronger notion is desirable when NMFs are used in more adversarial environment, as we will show in Section 9.4.

Novel properties of transformation class. Our non-malleability notion is stronger if Φ is larger. To capture broad yet achievable transformation class, in Section 4 we introduce two novel properties 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 transformation 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. Figure 1 shows a (rough) pictorial summary.

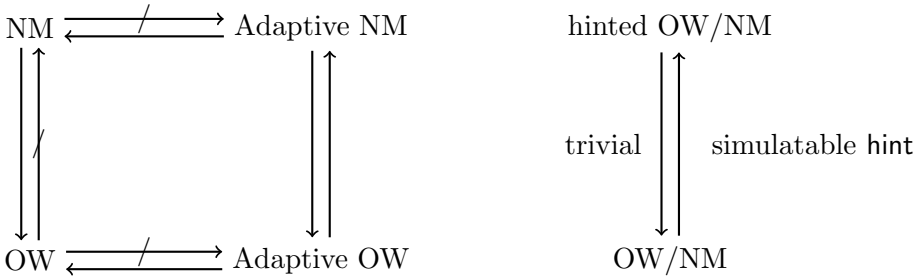

Figure 1: Let unhatched arrows represent implications, and hatched arrows represent separations. The left figure is a rough overview of relations among (adaptive) Φ-non-malleability and (adaptive) one-wayness for deterministic functions. See Section 5 for concrete requirements on Φ and the underlying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

In the non-adaptive setting, we show that w.r.t. any achievable transformation 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 is that:

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

Both OW and NM can be considered with some auxiliary information of preimage x^*, which is modeled by a hint value $\mathsf{hint}(x^*)$ under some hint function $\mathsf{hint}(\cdot)$. 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 9. While hinted notions trivially implies hint-free ones, the converse becomes more subtle. In Section 6, we investigate under what conditions hinted notions are implied by hint-free ones.

Constructions of NMFs. In the random oracle model, by leveraging the proof idea behind AOW $\Rightarrow$ ANM, we prove that the strengthened Merkle-Damgård transformation is actually $\Phi^{\mathsf{srs}}_{\mathsf{brs}} \cup \mathsf{id}$-non-malleable (cf. Appendix A for details). This greatly improves prior result [BFS11] with larger transformation class, and thus provides an efficient candidate of NMFs.

In the standard model, we give two generic constructions of NMFs in Section 7. The first construction is deterministic and non-malleable w.r.t. $\Phi^{\mathsf{srs}}_{\mathsf{brs}} \cup \mathsf{id}$, based on injective ATDFs. The second construction is randomized and non-malleable w.r.t. polynomial transformations, based on all-but-one lossy functions, one-time signature and universal hash functions. Given the fact that all the underlying ingredients are efficiently realizable from a variety of hardness assumptions, we obtain efficient constructions of NMFs in the standard model. This partially[4] resolves an open problem raised in [BCFW09].

Applications of NMFs. Since we have proved that when functions are poly-to-one, non-malleability implies one-wayness. Thereby, poly-to-one NMFs can replace NMOWHFs in the design cryptographic puzzles [BCFW09] for securing practical network protocols.

Apart from yielding efficient constructions of NMFs, we find that the implication AOW $\Rightarrow$ ANM is also useful elsewhere. In Section 8, we discuss how the high-level idea underlying AOW $\Rightarrow$ ANM provides a key insight in the RKA area, that is, resilience against non-trivial copy attacks w.r.t. most algebra-induced related-key transformation classes is in fact a built-in security.

Qin et al. [QLY+15] proposed a new notion called continuous non-malleable key derivation, which had proven to be useful in achieving RKA-security for numerous cryptographic primitives. They also gave a complicated construction of cnm-KDFs. In Section 9, we refine cnm-KDFs with accurate naming and stronger security notion, yielding RKA-secure authenticated KDFs. We propose an exquisitely simple construction of RKA-secure AKDFs from NMFs w.r.t. hardcore values as hint. Our construction admits a direct and modular proof. Particularly, by instantiating our generic construction with randomized NMFs presented in Section 7.2, we simplify and clarify the construction of cnm-KDFs by Qin et al. [QLY+15].

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

[4]We say "partially" since the posed question in [BCFW09] is to construct efficient NMFs in the context of their simulation-based definition.

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 (which 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] undertook the study of correlated-input security for deterministic hash functions, which stipulate security remains even 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. They considered three notions, namely one-wayness, unpredictability and pseudorandomness under correlated-inputs for hash functions. Among them, unpredictable correlated-input hash functions (CIHs) require that no PPT adversary is able to predicate $h(c_{n+1}(r))$ after seeing $h(c_i(r))$ for $i \in [n]$.

Our NMFs consider non-malleability for (possibly randomized) functions. The connection between NMFs and CIHs is that deterministic NMFs can be 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-theoretic 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. When the context is clear, we will omit r and write $z \leftarrow \mathcal{A}(x_1, \ldots, x_n)$.

Let X and Y be two random variables over some countable set S. The *statistical distance* between X and Y is defined as

$$\Delta(X, Y) = \frac{1}{2} \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]|$$

Let $X = \{X_\lambda\}_{\lambda \in \mathbb{N}}$ and $Y = \{Y_\lambda\}_{\lambda \in \mathbb{N}}$ denote two ensembles of random variables indexed by λ. We say that X and Y are statistically indistinguishable, written $X \approx_s Y$, if $\Delta(X_\lambda, Y_\lambda) = \mathsf{negl}(\lambda)$. We say that X and Y are computationally indistinguishable, written $X \approx_c Y$, if the advantage of any PPT algorithm in distinguishing X_λ and Y_λ is $\mathsf{negl}(\lambda)$.

2.1 Randomness Extraction

The min-entropy of a random variable X over a domain S is the negative (base-2) logarithm of the *unpredictability* of X:

$$\mathsf{H}_\infty(X) = -\log\left(\max_{s\in S}\Pr[X=s]\right).$$

In many natural settings, the variable X is correlated with another variable Y whose value is known to an adversary. In such scenarios, it is most convenient to use the notion of *average min-entropy* as defined by Dodis et al. [DORS08], which captures the *average unpredictability* of X conditioned Y:

$$\tilde{\mathsf{H}}_\infty(X|Y) = -\log\left(\mathrm{E}_{y\leftarrow Y}\left[2^{\mathsf{H}_\infty(X|Y=y)}\right]\right) = -\log\left(\mathrm{E}_{y\leftarrow Y}\left[\max_{s\in S}\Pr[X=s|Y=y]\right]\right)$$

The average min-entropy corresponds to the optimal probability of guessing X, given knowledge of Y. The following bound of average min-entropy was proved in [DORS08].

Lemma 2.1 ([DORS08])**.** *Let X, Y, Z be random variables. If Z has 2^r possible values, $\tilde{\mathsf{H}}_\infty(X|(Y,Z)) \geq \tilde{\mathsf{H}}_\infty(X|Y) - r$. In particular, $\tilde{\mathsf{H}}_\infty(X|Y) \geq \mathsf{H}_\infty(X) - r$.*

In cryptographic applications, we usually need to derive nearly uniform bits from a weakly random source X that has some average min-entropy. This is accomplished via an appropriate type of *randomness extractor*.

Definition 2.1 ([DORS08])**.** A function $\mathsf{ext} : \mathcal{X}\times\{0,1\}^d \to \{0,1\}^\ell$ is an average-case (m,ℓ,ϵ)-strong extractor if for all pairs of random variables (X,Y) such that X is defined over $\mathcal{X}$ and $\tilde{\mathsf{H}}_\infty(X|Y) \geq m$, it holds that:

$$\Delta((\mathsf{ext}(X,s),s,Y),(k,s,Y)) \leq \epsilon,$$

where $s \xleftarrow{\mathrm{R}} \{0,1\}^d$ and $k \xleftarrow{\mathrm{R}} \{0,1\}^\ell$.

Dodis et al. [DORS08] proved that any strong extractor is in fact an average-case strong extractor for appropriate setting of the parameters. As a specific example, they proved that any family of universal hash functions is an average-case strong extractor.

Lemma 2.2 ([DORS08])**.** *Let X and Y be random variables such that $\tilde{\mathsf{H}}_\infty(X|Y) \geq m$. Let $\mathcal{H}$ be a family of universal hash functions from X to $\{0,1\}^\ell$, where $\ell \leq m - 2\log(1/\epsilon)$. Then $\mathcal{H}$ is a average-case (m,ℓ,ϵ)-strong extractor.*

2.2 Implications and Separations

Consider security notions A and B for a cryptographic primitive Π, we say that

- $A \Rightarrow B$: 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 that A is strictly stronger than B. If we further have $B \Rightarrow A$, we say that A is equivalent to B.

3 One-Way and Non-Malleable Functions

We first recall the syntax of efficiently computable functions.

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

- Sample a function: on input a security parameter λ, $\mathsf{Gen}(\lambda)$ outputs a function index s. Each s output by $\mathsf{Gen}(\lambda)$ defines a function $f_s : X_s \to Y_s$.
- Evaluate a function: on input s and $x \in X_s$, $\mathsf{Eval}(s, x)$ outputs $f_s(x)$.
- Verify a function: on input s and $(x, y) \in X_s \times Y_s$, $\mathsf{Vefy}(s, x, y)$ returns "1" if and only if $y = f_s(x)$. Here we note that when f_s is randomized, we simply write $y = f_s(x)$ to indicate that y is a function value of x with some randomness under f_s.

Note that we consider a very general syntax, comprising both deterministic and randomized functions. In our case the evaluation algorithm Eval may be probabilistic, as long as the correctness of function values is verifiable given the pre-image only (via Vefy).

Without loss of generality, we assume the existence of an efficient algorithm that samples x according to some well-spread distribution $\mathcal{C}$ over X_s.[5] A canonical choice of $\mathcal{C}$ is the uniform distribution over X_s. We stick to this choice in this work for simplicity.

We say a function f (possibly randomized) is injective if for any distinct inputs x_1, x_2, we have $f(x_1) \neq f(x_2)$. For an element $y \in Y_s$ we denote its preimage set under f_s by $f_s^{-1}(y) = \{x \in X_s : f_s(x) = y\}$. 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 $|f_s^{-1}(y)| \leq \mathsf{poly}(\lambda)$ for all $f_s \in \mathcal{F}$ and all $y \in Y_s$. 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_s, and there is a PPT algorithm $\mathsf{TdInv}(td_s, y)$ that computes a preimage $x \in f_s^{-1}(y)$. If a value y is not in $\mathrm{Img}(f_s)$, i.e., $f_s^{-1}(y)$ is empty, then the behavior of $\mathsf{TdInv}(td_s, y)$ is unspecified.

Next, we recall the notion of one-wayness and formally define the notion of non-malleability for efficiently computable 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 $x \in f_s^{-1}(y)$ if $y \in \mathrm{Img}(f_s)$, 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 below is negligible in λ:

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

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

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

Definition 3.3 (Hardcore Functions)**.** Let $\mathcal{H} = \{h_s\}$, where each h_s maps X_s to K_s. $\mathcal{H}$ is a family of hardcore functions of $\mathcal{F}$ if for any PPT adversary $\mathcal{A}$ its advantage $\mathsf{Adv}^{\mathrm{rand}}_{\mathcal{A},\mathcal{H}}(\lambda)$ defined below is negligible in λ:

$$\mathsf{Adv}^{\mathrm{rand}}_{\mathcal{A},\mathcal{H}}(\lambda) = \Pr\left[\beta = \beta' : \begin{array}{l} s \leftarrow \mathsf{Gen}(\lambda); \\ x^* \xleftarrow{\mathrm{R}} X_s, y^* \leftarrow f_s(x^*); \\ k_0^* \leftarrow h_s(x^*); k_1^* \xleftarrow{\mathrm{R}} K_s; \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}(f_s, h_s, y^*, k_\beta^*); \end{array}\right] - \frac{1}{2}.$$

The celebrated Goldreich-Levin theorem [GL89] demonstrated a "universal" hardcore function with 1-bit output (hardcore predicate) for every deterministic OWFs. In details, the inner product of preimage x with a random string r (r could also be viewed as part of the description of h_s) constitutes a hardcore predicate. It is easy to verify that this classical result applies to randomized setting as well.

Definition 3.4 (Non-Malleability and Adaptive Non-Malleability)**.** Let Φ be a transformation class defined over the domain X. $\mathcal{F}$ 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:

$$\mathsf{Adv}^{\mathrm{nm}}_{\mathcal{A},\mathcal{F}}(\lambda) = \Pr\left[\begin{array}{c} \phi \in \Phi \wedge y = f_s(\phi(x^*)) \\ \wedge (\phi, y) \neq (\mathsf{id}, y^*) \end{array} : \begin{array}{l} s \leftarrow \mathsf{Gen}(\lambda); \\ x^* \xleftarrow{\mathrm{R}} X_s, y^* \leftarrow f_s(x^*); \\ (\phi, y) \leftarrow \mathcal{A}(f_s, y^*); \end{array}\right].$$

$\mathcal{F}$ 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^*.

Remark 3.1. Generally speaking, for a family of functions, the domain and range could be dependent on function index. For simplicity, we assume that they are same for all function index, and write X for X_s, Y for Y_s and K for K_s. When things are clear from the context, we will also suppress the dependence on index and write: f for f_s, h for h_s and td for td_s.

We then proceed to give several technical discussions about the above notions.

Impossible classes. We first observe that function is easily malleable w.r.t. "regular" transformations, namely, identity transformation id and constant transformations ϕ_c. An adversary can trivially win by outputting (id, y^*) or $(\phi_c, f(c))$. It is not hard to see that function is also easily malleable w.r.t. transformations *near to* the regular ones[6]. In this regard, we call the regular transformations and the transformations near to the regular ones as "dangerous" transformations. So, a primary task is distilling the characterizations on admissible transformation class Φ to exclude "dangerous" transformations yet maintaining its generality to the largest extent. We also note that to make our non-malleable notion as strong as possible, Φ could include id with the natural restriction that trivial copy solution (id, y^*) is not considered successful.

Parameterized adaptivity. Let q be the maximum number of inversion queries that a 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, i.e., q-adaptive one-wayness/non-malleability. Clearly, adaptive notions degenerate to non-adaptive ones when $q = 0$. We will adopt the refined adaptive notions in Section 5.3 to give a dedicated relation between adaptive one-wayness and adaptive non-malleability.

[6]Roughly speaking, we say f is near to g if they outputs agree on most inputs. The extent of near is determined by the number of agreed inputs.

Hinted notions. In the non-malleability 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 (possibly probabilistic) function from X to $\{0,1\}^*$.

Analogously, in the security experiments of both one-wayness and non-malleability for functions, we can make the adversaries more powerful by giving them a hint about x^*. We say that the resulting notions are hinted, and 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 will demonstrate in Section 9. Observe that the hint function might be dependent on f, thus in this work we model hint via a family of hint functions.[7]

Next, we first seek for an achievable yet large transformation 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 results in Section 5 as prerequisite.

4 Transformation Class

Following [BFS11], our notion of non-malleability is defined w.r.t. a transformation class Φ, in which $\phi : X \to X$ maps a preimage to a related preimage. We require that transformations in Φ are efficiently recognizable and computable. Hereafter, we denote by id the identity transformation $\phi(x) = x$ and denote by cf the set of all constant transformations $\{\phi_c : \phi_c(x) = c\}_{c\in X}$. In particular, we denote by cf_0 the constant transformation that maps all inputs to zero. 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 transformation class Φ. We are thus motivated to distill some characterizations on Φ that make non-malleability achievable while keep Φ general enough. Towards this goal, we first review some algebra-induced transformation classes considered in the literature, then abstract two novel properties to capture a general transformation class.

4.1 Concrete Transformation Classes

Group-induced transformations. When X under '+' forms a group $\mathbb{G}$, let $\Phi^{\mathrm{lin}} = \{\phi_a\}_{a\in\mathbb{G}}$ with $\phi_a(x) = a + x$ be the class of linear transformations, which generalizes 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 $X = \{0,1\}^n$.

Ring-induced transformations. When X 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 X 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.

[7] To make the hinted notions achievable, hint functions must meet 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 restrictions to hint in definition.

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].

For any $\phi \in \Phi^{\mathsf{poly}(d)}$, let $R_\phi = \{x \in \mathbb{F} : \phi(x) = 0\}$, i.e., the set of all roots of ϕ over $\mathbb{F}$. Clearly, $|R_\phi| \le d$ for any $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}_0$. Suppose $|\mathbb{F}| \ge 2^{n(\lambda)}$, where n is a polynomial of λ. Qin et al. [QLY+15] proved the following lemma.

Lemma 4.1 ([QLY+15])**.** *Let X be any random variable over $\mathbb{F}$ such that $\mathsf{H}_\infty(X) \ge n$. For any $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}$, then $\mathsf{H}_\infty(\phi(X)) \ge n - \log d$. For any $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{id}$, then $\Pr[\phi(X) = X] \le d/2^n$.*

By setting $\log d \le n - \omega(\log \lambda)$, the above lemma shows that for any $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}$ the value $\phi(X)$ is unpredictable, and the probability of $\phi(X) = X$ for any $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{id}$ is negligible. In the following lemma, we prove that similar properties hold in the case that X has a lower bound of average-case min-entropy, which are more handy to use for cryptographic applications.

Lemma 4.2. *Let X be any random variable over $\mathbb{F}$ and Y is any random variable such that $\tilde{\mathsf{H}}_\infty(X|Y) \ge m$. For any $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}$, then $\tilde{\mathsf{H}}_\infty(\phi(X)|Y) \ge m - \log d$. For any $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{id}$, then $\Pr[\phi(X) = X|Y] \le d/2^m$.*

Proof. We first prove the first property. For any $c \in \mathbb{F}$, if $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}$, then $\phi' \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}_0$ where $\phi'(x) = \phi(x) - c$ and $|R_{\phi'}| \le d$.

$$\begin{aligned}
\mathrm{E}_{y \leftarrow Y}\left(\max_{c \in \mathbb{F}} \Pr[\phi(X) = c | Y = y]\right) &= \mathrm{E}_{y \leftarrow Y}\left(\textstyle\sum_{x_i \in R_{\phi'}} \Pr[X = x_i | Y = y]\right) \\
&\le \mathrm{E}_{y \leftarrow Y}\left(d \cdot \max_{c \in \mathbb{F}} \Pr[X = c | Y = y]\right) \\
&= d \cdot \mathrm{E}_{y \leftarrow Y}\left(\max_{c \in \mathbb{F}} \Pr[X = c | Y = y]\right) \le d/2^m
\end{aligned}$$

This proves $\tilde{\mathsf{H}}_\infty(\phi(X)|Y) \ge m - \log d$. We then prove the second property. If $\phi \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{id}$, then $\phi'' \in \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}_0$ where $\phi''(x) = \phi(x) - x$ and $|R_{\phi''}| \le d$.

$$\begin{aligned}
\mathrm{E}_{y \leftarrow Y}\left(\Pr[\phi(X) = X | Y = y]\right) &= \mathrm{E}_{y \leftarrow Y}\left(\textstyle\sum_{x_i \in R_{\phi''}} \Pr[X = x_i | Y = y]\right) \\
&\le \mathrm{E}_{y \leftarrow Y}\left(d \cdot \max_{c \in \mathbb{F}} \Pr[X = c | Y = y]\right) \\
&= d \cdot \mathrm{E}_{y \leftarrow Y}\left(\max_{c \in \mathbb{F}} \Pr[X = c | Y = y]\right) \le d/2^m
\end{aligned}$$

This proves $\Pr[f(X) = X|Y] \le d/2^m$. The lemma immediately follows. □

4.2 General Transformation Class

Let $r(\lambda)$ be a quantity of λ, R_ϕ be the set of roots of ϕ over X. We introduce the following two properties to capture a general transformation class.

Definition 4.1 (Bounded Root Space)**.** A transformation ϕ has $r(\lambda)$-bounded root space if $|R_\phi| \le r(\lambda)$. A transformation class Φ has $r(\lambda)$-bounded root space if for each $\phi \in \Phi$ and each $\phi_c \in \mathsf{cf}$, the composite transformations $\phi' = \phi - \phi_c$ and $\phi'' = \phi - \mathsf{id}$ 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 as input ϕ and outputs an element from R_ϕ uniformly at random.[8] A transformation class Φ has sampleable root space if for each $\phi \in \Phi$ and $\phi_c \in \mathsf{cf}$, the composite transformations $\phi' = \phi - \phi_c$ and $\phi'' = \phi - \mathsf{id}$ both have sampleable root spaces.

Hereafter, we let $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$ denote the transformation class satisfying the bounded root space (BRS) & sampleable root space (SRS) properties. Define the distance between two transformations ϕ_1 and ϕ_2 over X as $\|\phi_1 - \phi_2\| = 1 - |R_{\phi_1-\phi_2}|/|X|$. The BRS property essentially captures the characterization that all $\phi \in \Phi$ are $(1 - r(\lambda)/|X|)$-far away from regular transformations in an algebraic view.

In this work, we are primarily interested in transformation classes with polynomially bounded root spaces[9]. Now, the BRS property rules out dangerous transformations from Φ by stipulating that each $\phi \in \Phi$ is at least $(1 - \mathsf{poly}(\lambda)/|X|)$-far away from regular ones, i.e., having at most polynomially many intersection points with them. The reason of choosing polynomial as the bound to capture the distance to regular transformation will be clear in Section 5.3.

Remark 4.1. Recent works [JW15, QLY+15] introduced two properties called high output entropy (HOE) and input-output collision resistance (IOCR) for transformation class Φ over X. The former states that for each $\phi \in \Phi$, the min-entropy of $\phi(x)$ is sufficiently high when $x \xleftarrow{R} X$, 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 $x \xleftarrow{R} X$. Lemma 4.1 actually states that $\Phi^{\mathsf{poly}(d)} \backslash (\mathsf{id} \cup \mathsf{cf})$ satisfies HOE and IOCR properties if $\log d \leq n - \omega(\log \lambda)$.

We observe that BRS implies HOE & IOCR. To see this, notice that: (1) for each $c \in X$ the equation $\phi(x) - c = 0$ having at most polynomial number of roots implies that $\max_{c \in X} \Pr[\phi(x) = c] \leq \mathsf{poly}(\lambda)/|X| = \mathsf{negl}(\lambda)$ when $x \xleftarrow{R} X$; (2) the equation $\phi(x) - x = 0$ having at most polynomial number of roots implies that $\Pr[\phi(x) = x] \leq \mathsf{poly}(\lambda)/|X| = \mathsf{negl}(\lambda)$ when $x \xleftarrow{R} X$.

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. More precisely, 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 functions, which are possibly randomized. For simplicity, we work with hint-free notions. All the results obtained extend naturally to hinted notions.

5.1 Non-Malleability ⇒ One-Wayness

We first show that in common cases non-malleability implies one-wayness.

Lemma 5.1. *For any achievable transformation class Φ, Φ-Non-Malleability $\Rightarrow$ 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:

[8] If R_f is empty, this algorithm simply outputs a distinguished symbol $\perp$.

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

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

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

Since $\mathcal{F}$ is poly-to-one, 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 $x^* \leftarrow X$. This is because there are at most polynomially many values x such that $f(x) = y^*$, and they are all equally likely 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 lemma follows. □

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

We then show that generally one-wayness does not imply non-malleability, even w.r.t. small transformation class Φ^{xor}.

Lemma 5.2. *One-Wayness $\not\Rightarrow$ Φ^{xor}-Non-Malleability.*

Proof. Let $\mathcal{F}$ be a family of OWFs. 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 $f'(x||b) := f(x)||b$ and b denotes the last bit of its input. We then proceed to prove the following two claims.

Claim 5.3. *$\mathcal{F}'$ is one-way.*

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

Claim 5.4. *$\mathcal{F}'$ is Φ^{xor}-malleable.*

Proof. Given f' and an image $y'^* \leftarrow f'(x'^*)$ where $x'^* = x^*||b^*$ 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 $y^*||b^*$, set $a = 0^n||1$, then output ϕ_a together with $y' = y^*||(b^* \oplus 1)$. It is easy to see that $\phi_a \in \Phi^{\mathrm{xor}} \backslash \mathsf{id}$ and $y' = f'(x^*||(b^* \oplus 1)) = f'(\phi_a(x'^*))$. This proves Claim 5.4. □

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[10] 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.

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

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

Lemma 5.5. *For any achievable transformation 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 since it is straightforward. □

Lemma 5.6. *$(q+1)$-Adaptive One-Wayness $\Rightarrow$ q-Adaptive $\Phi_{\mathrm{brs}}^{\mathrm{srs}} \cup \mathsf{id}$-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 major technical difficulty is how to utilize the power of an adversary $\mathcal{A}$ against adaptive non-malleability to break adaptive one-wayness.

Suppose $\mathcal{A}$ outputs (ϕ, y) as its solution against non-malleability. If $\phi = \mathsf{id}$, we must have $y \neq y^*$. In this case, the reduction immediately obtains x^* by simply quering $\mathcal{O}_{\mathsf{inv}}$ at y because f is injective. Otherwise, we have $\phi \in \Phi_{\mathrm{brs}}^{\mathrm{srs}}$. Note that the challenge instance of one-wayness has already provided with a relation about x^*, i.e., $\mathsf{Vefy}(f, x^*, y^*) = 1$. Now, the reduction obtains another relation about x^*, that is, $\mathsf{Vefy}(f, \phi(x^*), y) = 1$. However, these two relations 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 an solvable equation about x^* without the presence of f: (1) for the case of $y \neq y^*$, the reduction first queries the inversion oracle at point y, then gets $\phi(x^*) = \mathcal{O}_{\mathsf{inv}}(y)$; (2) for the case of $y = y^*$, the reduction gets $\phi(x^*) = x^*$ due to the injectivity of f. 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 below.

Proof. Suppose there is an adversary $\mathcal{A}$ against the adaptive non-malleability of $\mathcal{F}$, 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 an image $y^* \leftarrow f(x^*)$ for $x^* \leftarrow X$, $\mathcal{B}$ 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 $(\phi, y) \neq (\mathsf{id}, y^*)$ against adaptive non-malleability, $\mathcal{B}$ proceeds as follows:

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

We justify the correctness of $\mathcal{B}$'s strategy as follows. Conditioned on $\mathcal{A}$ succeeds, we have $\mathsf{Vefy}(f, \phi(x^*), y) = 1$. Due to injectivity of $\mathcal{F}$, for case 1 we have $\mathsf{id}(x^*) = x^* = x$; for case 2 we have $\phi(x^*) = x$, i.e., x^* is a solution of $\phi'(\alpha) = 0$; for case 3 we have $\phi(x^*) = x^*$, i.e., x^* is a solution of $\phi''(\alpha) = 0$. Taking the three cases together, conditioned on $\mathcal{A}$ succeeds by making at most q inversion queries, then according to the BRS & SRS properties of $\Phi_{\mathrm{brs}}^{\mathrm{srs}}$, $\mathcal{B}$ will output the right x^* with probability $1/\mathsf{poly}(\lambda)$ by making at most $(q+1)$ inversion

queries. We stress that the probability here is taken over the randomness of SampRS, but not $x \xleftarrow{R} X$. 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 the lemma. □

Combine Lemma 5.5 and Lemma 5.6 together, we conclude that for injective functions, adaptive $(\Phi^{\mathsf{srs}}_{\mathsf{brs}} \cup \mathsf{id})$-non-malleability is equivalent to adaptive one-wayness.

Remark 5.1. Analogous to the RKA security notion, our non-malleability notion is of "unique" flavor, in which the adversary is only considered to be successful if its output is a related image of the preimage x^* exactly 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 $\phi \in \Phi$ such that $\phi(x) \neq x$ for any $x \in X$, 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 $\mathsf{Vefy}(f, \phi(x^*), y) = 1$. 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, i.e., returning the preimages of its inputs. Actually, contrived inversion algorithm may reveal critical information (e.g. trapdoor) when its inputs fall outside the image of f, and thus make f not adaptively 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 5.7. *For any achievable transformation class Φ, Φ-Non-Malleability $\not\Rightarrow$ Adaptive Φ-Non-Malleability when $\mathcal{F}$ is poly-to-one.*

Proof. Let $\mathcal{F} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{TdInv})$ be a family of Φ-non-malleable functions with trapdoor. We show how to modify $\mathcal{F}$ to $\mathcal{F}' = (\mathsf{Gen}', \mathsf{Eval}', \mathsf{TdInv}')$ 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 $\mathcal{F}'.\mathsf{TdInv}'$. To this end, we use the bit-prefix trick to ensure: well-formed images of f' must have bit-prefix '0', while strings with bit-prefix '1' must be 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 $f'(x) := 0||f(x)$; $\mathcal{F}'.\mathsf{Eval}'(f', x)$ outputs $0||\mathcal{F}.\mathsf{Eval}(f, x)$; $\mathcal{F}'.\mathsf{TdInv}'(td, y')$ 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 5.8. *$\mathcal{F}'$ is Φ-non-malleable.*

Proof. Suppose $\mathcal{A}'$ is an adversary against the Φ-non-malleability of $\mathcal{F}'$, 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 $x^* \xleftarrow{R} \{0,1\}^n$, $\mathcal{A}$ is asked to output (ϕ, y) s.t. $\mathcal{F}.\mathsf{Vefy}(f, \phi(x^*), y) = 1$. $\mathcal{A}$ 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 ($\mathcal{F}'.\mathsf{Vefy}(f', \phi(x^*), y') = 1$), $\mathcal{A}$ also succeeds ($\mathcal{F}.\mathsf{Vefy}(f, \phi(x^*), y) = 1$). Claim 5.8 follows. □

Claim 5.9. *$\mathcal{F}'$ is not adaptively Φ-non-malleable.*

Proof. Given f and $0||y^* \leftarrow f'(x^*)$ for $x^* \xleftarrow{R} \{0,1\}^n$, 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 $x^* \leftarrow \{0,1\}^n$. 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 5.9. □

Lemma 5.7 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 and the hinted one-wayness, with respect to the same hint.

6 Relation Between Hint-free and Hinted Notions

In this section, we investigate the relations between hint-free notions and hinted notions. While 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 imply hinted notions.

Let $\mathcal{F}$ be a family of functions, $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, $x^* \xleftarrow{R} X$, $y^* \leftarrow f(x^*)$ and $\mathsf{H} = \{\mathsf{hint}_f\}_{f \in \mathcal{F}}$ be a family of associated hint functions. Henceforth, we drop the subscript and simply write hint to denote the hint function for f. We say H is *p-statistically simulatable* if there exists a PPT reduction algorithm $\mathcal{R}$ such that $(y^*, \mathcal{R}(y^*)) \approx_s (y^*, \mathsf{hint}(x^*))$ holds with probability $p(\lambda)$; we say H is *p-computationally simulatable* if there exists a PPT algorithm $\mathcal{R}$ such that $(y^*, \mathcal{R}(y^*)) \approx_c (y^*, \mathsf{hint}(x^*))$ holds with probability $p(\lambda)$. Both the probabilities are defined over the choice of $x^* \xleftarrow{R} X$ and the random coins of $\mathcal{R}$. Roughly speaking, if $p(\lambda)$ is noticeable, then hint-free notions imply hinted notions. Intuitively, this follows since if the hint value can be simulated by a PPT reduction algorithm, then it is useless since the adversary can generate such hint value on itself. We formalize this intuition as below.

It is easy to see that when H is statistically simulatable for some noticeable probability $p(\lambda)$, then with probability $p(\lambda)$ a reduction algorithm is able to create a game that is statistically close to the real hinted game, and thus reduces hinted notions to hint-free ones.

Things turn out to be subtle when H is computationally simulatable. In this case, a reduction algorithm is able to create a game that is computationally indistinguishable to the real hinted game. However, to argue the adversary's advantages between the real hinted game and the simulated hinted game are negligibly close (for the purpose to reduce the hinted notions to hint-free ones), we have to ensure that the hinted notion is public-coin falsifiable, i.e., a PPT

reduction algorithm can verify if a solution is correct even without knowing the secret coins of the challenge.

We exemplify these two cases in the following lemmas.

6.1 OW vs. Hinted-OW

We first show that regardless of the construction of H, as long as the output length of each hint function is short, i.e., bounded by $O(\log(\lambda))$, then H is $1/O(\lambda^c)$-perfectly simulatable (a special case of statistically simulatable) and thus one-wayness implies hinted one-wayness.

Lemma 6.1 (Statistically Simulatable Case)**.** *Let $\mathcal{F}$ be a family of functions, one-wayness implies hinted one-wayness w.r.t. any family of hint functions with output length bounded by $O(\log(\lambda))$.*

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 $x^* \xleftarrow{\mathrm{R}} X$, $\mathcal{B}$ simply makes a random guess of $\mathsf{hint}(x^*)$, say k^*, then sends (f, y^*, k^*) to $\mathcal{A}$ as the challenge. Finally, $\mathcal{B}$ forwards $\mathcal{A}$'s solution as its solution. Since the output length is bounded by $O(\log(\lambda))$, $\mathcal{B}$ guesses the right hint value and thus simulates perfectly with probability $1/O(\lambda^c)$, where c could be any constant. Thereby, we conclude that $\mathsf{Adv}^{\mathrm{ow}}_{\mathcal{B},\mathcal{F}}(\lambda) \geq \mathsf{Adv}^{\mathrm{how}}_{\mathcal{A},\mathcal{F}}(\lambda)/O(\lambda^c)$. The lemma immediately follows. □

Remark 6.1. By interpreting $\mathsf{hint}(\cdot)$ as a leakage function, this result is in the same spirit to the fact that "essentially all cryptographic schemes are already resilient against small amount of leakage" [DY13].

We then show that for some specific hint functions, their output lengths could possibly go beyond $O(\log(\lambda))$. For instance, if H is a family of hardcore functions for $\mathcal{F}$, then H is 1-computationally simulatable assuming the one-wayness of $\mathcal{F}$, and thus one-wayness also implies hinted one-wayness in this case.

Lemma 6.2 (Computationally Simulatable Case)**.** *Let $\mathcal{F}$ be a family of functions and $\mathcal{H}$ be a family of hardcore functions for $\mathcal{F}$ from X to K. Then one-wayness implies hinted one-wayness w.r.t. $\mathcal{H}$ as the family of hint functions.*

Proof. We prove this theorem by leveraging the fact that $\mathcal{H}$ is 1-computationally simulatable. We proceed via a sequence of games. Let $\mathcal{A}$ be an adversary against the hinted one-wayness of $\mathcal{F}$ w.r.t. $\mathcal{H}$ as the family of hint functions. 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. $\mathcal{A}$ is given $(f, h, y^*, h(x^*))$ as the challenge, where $f \leftarrow \mathcal{F}.\mathsf{Gen}$, $h \leftarrow \mathcal{H}$, $y^* \leftarrow f(x^*)$ for $x^* \xleftarrow{\mathrm{R}} X$. Here, the hint value is set as $k^* \leftarrow h(x^*)$. According to the definition, we have:

$$\mathsf{Adv}^{\mathrm{how}}_{\mathcal{A},\mathcal{F}}(\lambda) = \Pr[S_0]. \tag{1}$$

Game 1 (Modify the hint): The same as Game 0 except that the hint value is set as $k^* \xleftarrow{\mathrm{R}} K$ rather than $h(x^*)$. Observe that in this case the hint carries no information of x^*.

We now state and prove the following two claims that establish the lemma.

Claim 6.3. *If $\mathcal{H}$ is family of hardcore functions for $\mathcal{F}$, then the advantage of any PPT adversary in Game 0 is negligibly close to the advantage in Game 1.*

Proof. We prove this lemma by giving a reduction to the pseudorandomness $\mathcal{H}$. We show how to turn an adversary $\mathcal{A}$ into an algorithm $\mathcal{B}$ against the pseudorandomness of $\mathcal{H}$.

Given (f, h, y^*, k^*_β) where $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, h is a hardcore function for f, $y^* \leftarrow f(x^*)$ for $x^* \xleftarrow{R} X$, and k^*_β is $h(x^*)$ if $\beta = 0$ or a random value from K if $\beta = 1$, $\mathcal{B}$ is asked to determine the value of β. $\mathcal{B}$ forwards (f, y^*, k^*_β) to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs x'. If $\mathcal{A}$ wins, namely $f(x') = y^*$, $\mathcal{B}$ outputs 1. Else, $\mathcal{B}$ outputs 0. It is easy to see that if $\beta = 0$ then the hint value is $h(x^*)$ and thus $\mathcal{B}$ perfectly simulates Game 0; if $\beta = 1$ then $\mathcal{B}$ perfectly simulates Game 1. We analyze $\mathcal{B}$'s advantage as below.

$$\begin{aligned}\mathsf{Adv}^{\mathrm{rand}}_{\mathcal{B}} &= \left|\Pr[\beta = 0]\Pr[\mathcal{B}\text{ outputs }0|\beta = 0] + \Pr[\beta = 1]\Pr[\mathcal{B}\text{ outputs }1|\beta = 1] - \frac{1}{2}\right| \\ &= \left|\frac{1}{2}\left(1 - \Pr[S_0] + \Pr[S_1]\right) - \frac{1}{2}\right| = \frac{1}{2}\left|\Pr[S_1] - \Pr[S_0]\right|\end{aligned}$$

Therefore, $\mathcal{B}$ breaks the pseudorandomness of $\mathcal{H}$ with at least the same advantage of $|\Pr[S_1] - \Pr[S_0]|/2$. This proves the Claim 6.3. □

Claim 6.4. *No PPT adversary has non-negligible advantage in Game 1 assuming the one-wayness of $\mathcal{F}$.*

Proof. Suppose $\mathcal{A}$ is a PPT adversary that has non-negligible advantage in Game 1. 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 $x^* \xleftarrow{R} X$, $\mathcal{B}$ simply picks $k^* \xleftarrow{R} K$, then sends (f, y^*, k^*) 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 6.4. □

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 follows. □

The above results apply naturally to the adaptive setting.

6.2 NM vs. Hinted-NM

Lemma 6.5 (Statistically Simulatable Case)**.** *For a family of functions $\mathcal{F}$, non-malleability implies hinted non-malleability w.r.t. any family of hint functions with output length bounded by $O(\log(\lambda))$.*

The proof is almost identical to that of Lemma 6.1. We omit the details here.

Lemma 6.6 (Computationally Simulatable Case)**.** *Let $\mathcal{F}$ be a family of injective functions $\mathcal{F}$ and $\mathcal{H}$ be a family of hardcore functions for $\mathcal{F}$ from X to K. Then adaptive non-malleability implies hinted adaptive non-malleability w.r.t. transformation class $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$ and $\mathcal{H}$ as the family of hint functions.*

Proof. Recall that Lemma 5.6 establishes the equivalence between adaptive one-wayness and adaptive $\Phi^{\mathrm{srs}}_{\mathrm{brs}}$-non-malleability for a family of injective functions, and this equivalence holds in the hinted setting as well. Combine Lemma 6.2, we conclude that adaptive non-malleability implies hinted adaptive non-malleability w.r.t. transformation class $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$ and hint function family $\mathcal{H}$. □

Remark 6.2. The above lemma states that in the adaptive NM implies adaptive hinted-NM w.r.t. transformation class $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$ and $\mathcal{H}$ as the family of hint functions. However, it is unknown if this result holds in the non-adaptive setting. The reason is that hinted-NM is not public-coin falsifiable, and thus we do not know how to reduce hinted-NM to hint-free NM.

7 Constructions of NMFs

Baecher et al. [BFS11, Construction 4.1] showed that the strengthened Merkle-Damgård (MD) transformation is non-malleable w.r.t. Φ^{xor}, assuming the compression function is a random oracle. In Appendix A, we improve over their result by showing that the strengthened MD transformation is essentially non-malleable w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$. This result gives us an efficient candidate of NMFs w.r.t. large transformation class, though in the random oracle model. In the rest of this section, we provide two efficient constructions of NMFs in the standard model, which partially resolves an open problem raised in [BCFW09].

7.1 Deterministic NMFs from Injective ATDFs

Let $\mathcal{F}$ be a family of injective ATDFs, and $\mathcal{H}$ be a family of Goldreich-Levin hardcore predicates for $\mathcal{F}$. [GL89] proved that $\mathcal{H}$ is a family of hardcore functions for $\mathcal{F}$.

Theorem 7.1. *$\mathcal{F}$ is hinted-Φ-non-malleable w.r.t. $\mathcal{H}$ as the family of hint functions, where $\Phi = \Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$.*

Proof. According to Lemma 5.6, $\mathcal{F}$ is $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$-NM. By Lemma 6.6, $\mathcal{F}$ is also hinted-$\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$-NM w.r.t. $\mathcal{H}$. This finishes the proof. □

[KMO10] demonstrates that injective ATDFs can be instantiated from either a number of cryptographic primitives such as correlated-product TDFs [RS09], 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 assumptions, e.g. "instance-independent" RSA assumption. These instantiations immediately give us efficient constructions of deterministic NMFs.

In the above construction, we employ the Goldreich-Levin hardcore predicate $\mathsf{hc}(\cdot)$ to serve as the hardcore function. Nevertheless, this yields only one-bit hint, which is not applicable when many-bits hint is needed. We may 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 OWFs, assuming the existence of differing-inputs/indistinguishability obfuscation.

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.

7.2 Randomized NMFs from ABO Lossy Functions and One-Time Signature

Now we show how to construct NMFs from ABO lossy functions (cf. Section B.3) and one-time signature (cf. Section B.2). Let ABOLF be a collection of (X, Z, τ)-ABO lossy functions with branch set $B = \{0,1\}^d$, OTS be a strongly unforgeable one-time signature with verification key space $VK \subseteq B$ and signature space Σ. Let $\mathcal{H}$ be a family of universal hash functions

from domain X to $K = \{0,1\}^\ell$, where $\ell \leq n - \tau - 2\log(1/\epsilon)$ for $n = \log|X|$, $\tau = \log|Z|$ and $\epsilon = \mathsf{negl}(\lambda)$. Define $Y := B \times Z \times \Sigma$. We build a family of NMFs $\mathcal{F}$ from X to Y as below.

- $\mathsf{Gen}(\lambda)$: on input a security parameter λ, output $s \leftarrow \text{ABOLF.}\mathsf{Gen}(\lambda, 0^d)$.
- $\mathsf{Eval}(x)$: on input a function index s, first generate a fresh key pair $(vk, sk) \leftarrow \text{OTS.}\mathsf{Gen}(\lambda)$, then compute $z \leftarrow g_{s,vk}(x)$, $\sigma \leftarrow \text{OTS.}\mathsf{Sign}(sk, z)$, output $y = (vk, z, \sigma)$. This algorithm implicitly defines a family of randomized functions $\mathcal{F}$ from X to Y indexed by s.
- $\mathsf{Vefy}(s, x, y)$: on input a function index s, x and $y = (vk, z, \sigma)$, output "1" if $z = g_{s,vk}(x) \wedge \text{OTS.}\mathsf{Vefy}(vk, z, \sigma) = 1$ and "0" otherwise.

Lemma 7.2. *$\mathcal{H}$ is a family of hardcore functions for $\mathcal{F}$.*

Proof. We proceed via a sequence of games. Let S_i be the probability that $\mathcal{A}$ wins in Game i.

Game 0. This is the standard pseudorandomness game. $\mathcal{CH}$ interacts with $\mathcal{A}$ as below.

1. Setup and challenge: $\mathcal{CH}$ generates a random function index s of $\mathcal{F}$ via $\text{ABOLF.}\mathsf{Gen}(\lambda, 0^d)$, picks $x^* \xleftarrow{R} X$, generates a fresh key pair $(vk^*, sk^*) \leftarrow \text{OTS.}\mathsf{Gen}(\lambda)$, computes $z^* \leftarrow g_{s,vk^*}(x^*)$, $\sigma^* \leftarrow \text{OTS.}\mathsf{Sign}(sk^*, z^*)$. $\mathcal{CH}$ then picks $h \xleftarrow{R} \mathcal{H}$, computes $k_0^* \leftarrow h(x^*)$, picks $k_1^* \xleftarrow{R} \{0,1\}^\ell$, $\beta \xleftarrow{R} \{0,1\}$. Finally, $\mathcal{CH}$ sends $(s, y^* = (vk^*, z^*, \sigma^*), k_\beta^*)$ to $\mathcal{A}$.
2. Guess: $\mathcal{A}$ outputs a guess β' for β and wins if $\beta = \beta'$.

According to the definition, we have:

$$\mathsf{Adv}_{\mathcal{A}} = |\Pr[S_0] - 1/2|$$

Game 1. Same as in Game 0 except that $\mathcal{CH}$ prepares $(vk^*, sk^*) \leftarrow \text{OTS.}\mathsf{Gen}(\lambda)$ at the very beginning and generate a random function index s via $\text{ABOLF.}\mathsf{Gen}(\lambda, vk^*)$. By the hidden lossy branch property of ABOLF, we conclude that:

$$|\Pr[S_1] - \Pr[S_0]| \leq \mathsf{negl}(\lambda)$$

Game 2. Same as in Game 1 except that $\mathcal{CH}$ picks $k_0^* \xleftarrow{R} \{0,1\}^\ell$ rather than setting $k_0^* \leftarrow h(x^*)$. Let $view = (s, h, y^* = (vk^*, z^*, \sigma^*))$, we have:

$$\begin{aligned} \tilde{\mathsf{H}}_\infty(x^*|view) &= \tilde{\mathsf{H}}_\infty(x^*|z^*, \sigma^*) && (2)\\ &= \tilde{\mathsf{H}}_\infty(x^*|z^*) && (3)\\ &\geq \mathsf{H}_\infty(x^*) - \tau = n - \tau && (4) \end{aligned}$$

In the above deduction, Equation (2) follows from the fact that s, h and vk^* are independent of x^*. Equation (3) follows from the fact that σ^* is derived from sk^* and z^*, while sk^* is independent of x^*. In Game 2, $g_{s,vk^*}(\cdot)$ is a lossy function whose image size is at most 2^τ. Equation (4) thus follows from Lemma 2.1 and z^* has at most 2^τ possible values. By the parameter choice $\ell \leq n - \tau - 2\log(1/\epsilon)$, $h(x^*)$ is ϵ-close to a uniform distribution over $\{0,1\}^\ell$. Therefore, $\mathcal{A}$'s views in Game 1 and Game 2 are statistically indistinguishable. We have:

$$|\Pr[S_2] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$$

In Game 2, both k_0^* and k_1^* are identically distributed. Therefore, we have:

$$\Pr[S_2] = 1/2$$

Taken all the above together, we have $|\Pr[S_0] - 1/2| = \mathsf{negl}(\lambda)$. This proves the lemma. □

Theorem 7.3. *Assuming the security of one-time signature and ABO lossy functions, $\mathcal{F}$ is hinted-Φ-non-malleable w.r.t. $\mathcal{H}$ as the family of hint functions, where $\Phi = \Phi^{\mathsf{poly}(d)} \backslash \mathsf{cf}$, $\log d \leq n - \tau - \ell - \omega(\log \lambda)$.*

Proof. We proceed via a sequence of games. Let S_i be the probability that $\mathcal{A}$ wins in Game i.

Game 0. This is the standard hinted-non-malleable security experiment. $\mathcal{CH}$ interacts with $\mathcal{A}$ as below.

1. Setup and challenge: $\mathcal{CH}$ generates a random function index s of $\mathcal{F}$ via ABOLF.$\mathsf{Gen}(\lambda, 0^d)$, picks $x^* \xleftarrow{\mathrm{R}} X$, generates a fresh key pair $(vk^*, sk^*) \leftarrow \mathrm{OTS}.\mathsf{Gen}(\lambda)$, computes $z^* \leftarrow g_{s,vk^*}(x^*)$, $\sigma^* \leftarrow \mathrm{OTS}.\mathsf{Sign}(sk^*, z^*)$. Finally, $\mathcal{CH}$ picks $h \xleftarrow{\mathrm{R}} \mathcal{H}$, then sends $(s, y^* = (vk^*, z^*, \sigma^*), h(x^*))$ to $\mathcal{A}$.
2. Attack: $\mathcal{A}$ outputs a pair $(\phi, y = (vk, z, \sigma))$ and wins if $\mathcal{F}.\mathsf{Vefy}(s, \phi(x^*), y) = 1$, namely $z = g_{s,vk}(\phi(x^*))$ and $\mathrm{OTS}.\mathsf{Vefy}(vk, z, \sigma) = 1$.

According to the definition, we have:

$$\mathsf{Adv}_{\mathcal{A}} = \Pr[S_0]$$

Game 1. Same as Game 0 except that $\mathcal{A}$'s solutions of the form (ϕ, y^*) are considered to be successful if and only if $\phi(x^*) = x^* \wedge \phi \in \Phi \backslash \mathsf{id}$. This change is only conceptual since $g_{s,vk^*}(\cdot)$ is an injective function and thus z^* uniquely determines its preimage. Therefore, we have:

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

Game 2. The same as Game 1 except that (vk^*, sk^*) is generated at the very beginning and s is generated via ABOLF.$\mathsf{Gen}(\lambda, vk^*)$ rather than ABOLF.$\mathsf{Gen}(\lambda, 0^d)$. Due to the hidden lossy branch property, we have:

$$|\Pr[S_2] - \Pr[S_1]| \leq \mathsf{negl}(\lambda)$$

We now analyze $\Pr[S_2]$. In Game 2, $\mathcal{A}$'s solutions (ϕ, x) are considered to be successful if one of the following three disjoint events happen (E_2 and E_3 are the sub-events of $y \neq y^*$):

- E_1: $y = y^*$ and $\phi(x^*) = x^* \wedge \phi \in \Phi \backslash \mathsf{id}$.
- E_2: $vk = vk^* \wedge (z, \sigma) \neq (z^*, \sigma^*)$ and $\mathrm{OTS}.\mathsf{Vefy}(vk^*, z, \sigma) = 1 \wedge g_{s,vk^*}(\phi(x^*)) = z \wedge \phi \in \Phi$.
- E_3: $vk \neq vk^*$ and $\mathrm{OTS}.\mathsf{Vefy}(vk^*, z, \sigma) = 1 \wedge g_{s,vk^*}(\phi(x^*)) = z \wedge \phi \in \Phi$.

Clearly, $S_2 = E_1 \vee E_2 \vee E_3$. We then bound $\Pr[E_i]$ for $1 \leq i \leq 3$ respectively.

Note that $\mathcal{A}$'s view prior outputting (ϕ, y) is $(s, h, y^* = (vk^*, z^*, \sigma^*), h(x^*))$. We have:

$$\begin{aligned}
\tilde{\mathsf{H}}_\infty(x^*|view) &= \tilde{\mathsf{H}}_\infty(x^*|z^*, \sigma^*, h(x^*)) && (5)\\
&= \tilde{\mathsf{H}}_\infty(x^*|z^*, h(x^*)) && (6)\\
&\geq \mathsf{H}_\infty(x^*) - \tau - \ell = n - \tau - \ell && (7)
\end{aligned}$$

In the above deduction, Equation (5) follows from the fact that s, h, and vk^* are independent of x^*. Equation (6) follows from the fact that σ^* is derived from sk^* and z^* while sk^* is independent of x^*. In Game 2, $g_{s,vk^*}(\cdot)$ is a lossy function whose image size is at most 2^τ. Equation (7) thus follows from Lemma 2.1 and the fact that z^* has at most 2^τ values.

Since $\tilde{\mathsf{H}}_\infty(x^*|view) \geq n - \tau - \ell$ and the IOCR property for $\phi \in \Phi \backslash \mathsf{id}$ (cf. Lemma 4.2), we have $\Pr[E_1] \leq 1/2^{n-\tau-\ell-\log d}$.

By the strong unforgeability of OTS, we have $\Pr[E_2] \leq \mathsf{Adv}^{\mathrm{OTS}}_{\mathcal{A}} \leq \mathsf{negl}(\lambda)$.

By the HOE property for $\phi \in \Phi$ (cf. Lemma 4.2), we have $\tilde{\mathsf{H}}_\infty(\phi(x^*)|view) \geq n-\tau-\ell-\log d$. Since for all $vk \neq vk^*$, $g_{s,vk}(\cdot)$ is an injective function, the value $z = g_{s,vk}(\phi(x^*))$ has the same average min-entropy of $\phi(x^*)$. Thus, we have $\Pr[E_3] \leq 1/2^{n-\tau-\ell-\log d}$.

By the parameter choice of d, namely $\omega(\lambda) \leq n - \tau - \ell - \log d$, we have $\Pr[E_1]$ and $\Pr[E_3]$ are negligible in λ. Putting all the above together, we conclude that $\Pr[S_2]$ is negligible in λ.

The theorem immediately follows. □

8 Built-in Resilience against Non-trivial Copy Attacks

Here, we extend the idea behind the implication AOW ⇒ ANM further still to address non-trivial copy attacks in the RKA area. We begin by briefly introducing the background of RKA security and defining what it means for "copy attacks" (including trivial ones and non-trivial ones).

8.1 RKA-security Model and Copy Attacks

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, to induce 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 pseudorandom function/permutation, and their constructions were subsequently improved by [BC10, ABPP14]. So far, the study of RKA security has expands to other primitives, such as private-key encryption [AHI11], public-key encryption [Wee12], signature [BPT12], and identity-based encryption [BPT12].

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 $\mathsf{Func}(sk, c)$, where $\mathsf{Func}(sk, \cdot)$ denotes some keyed-operations (e.g., signing, decryption) and x denotes its input (e.g., message, ciphertext). A primitive is said to be RKA-secure if it remains secure when the adversary can access to an RKA oracle $\mathcal{O}_{\mathsf{rka}}(\phi, c) := \mathsf{Func}(\phi(sk), c)$. We note that constant RKA queries $\langle \phi, c \rangle$ where $\phi \in \mathsf{cf}$ are easy to handle. Thus, we can always allow $\Phi \supseteq \mathsf{cf}$.

Let c^* be the challenge in the security experiment. The RKA queries $\langle \phi, c^* \rangle$ 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., via fault injection attacks [BS97, BDL97]). However, attaining resilience against non-trivial copy attacks turns out to be difficult.

8.2 Known Techniques in Tackling Non-trivial Copy Attacks

Prior works paid a lot of effort to address 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 $sk \in SK$, $\phi(sk) \neq \phi'(sk)$. 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

already 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.[11] The third method in the context of pseudorandom functions is to rely on Φ-key-collision-security [ABPP14], which requires that for a random key sk it is impossible to find two distinct $\phi_1, \phi_2 \in \Phi$ such that $\phi_1(sk) = \phi_2(sk)$. However, such property is only known to hold w.r.t. specific Φ under concrete number-theoretic assumptions.

8.3 Our Insight in Addressing Non-trivial Copy Attacks

As discussed above, non-trivial copy attacks have not been well addressed at a general level. Being aware of the conceptual 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 5.6 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 $\Phi = \Phi^{\mathsf{srs}}_{\mathsf{brs}}$, 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 an 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 contradicts the security of the starting primitives, and therefore we must have $\Pr[E] = \mathsf{negl}(\lambda)$.

Somewhat surprisingly, 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.

[11] Briefly, Φ-fingerprinting for 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^*.

9 Application to RKA-secure Authenticated KDFs

9.1 Continuous Non-Malleable KDFs, Revisited

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 source key space X, the public key space Π, and the derived key space K. (2) $\mathsf{Sample}(pp)$, on input pp, samples a random key $x \xleftarrow{R} X$ and computes public key $\pi \in \Pi$. (3) $\mathsf{Derive}(s, \pi)$, on input (x, π), outputs a derived key $k \in K$ or $\perp$ indicating that π is not a valid proof of x.

Security. 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 $k^* \leftarrow \mathsf{Derive}(x^*, \pi^*)$ from a random one, even if 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 $\underline{(\phi(x^*), \pi) = (x^*, \pi^*)}$, or $\mathsf{Derive}(\phi(x^*), \pi)$ otherwise.

Construction. Let OTLF $= (\mathsf{Gen}, \mathsf{Eval}, \mathsf{SampLossy})$ be a family of one-time lossy filters [QL13] with domain X, range Y, and branch set $B = B_c \times B_a$, OTS $= (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vefy})$ be a strongly one-time signature, and $\mathcal{H}$ be a family of pairwise independent functions from X to K.[12] The construction is as below.

- $\mathsf{Setup}(\lambda)$: run $(s, td) \leftarrow \mathrm{OTLF}.\mathsf{Gen}(\lambda)$, pick $h \xleftarrow{R} \mathcal{H}$, output $pp = (s, h)$. Precisely, pp also includes the public parameters of OTLF and OTS.
- $\mathsf{Sample}(pp)$: run $(vk, sk) \leftarrow \mathrm{OTS}.\mathsf{Gen}(\lambda)$, pick $x \xleftarrow{R} X$ and a random core branch $b_c \xleftarrow{R} B_c$; compute $y \leftarrow g_{s, b_c || vk}(x)$ by treating vk as the auxiliary branch, generate $\sigma \leftarrow \mathrm{OTS}.\mathsf{Sign}(sk, b_c || y)$, then set $\pi = (b_c, vk, y, \sigma)$, and finally output (x, π).
- $\mathsf{Derive}(x, \pi)$: parse $\pi = (b_c, vk, y, \sigma)$, output derived key $k \leftarrow h(x)$ if $g_{s, b_c || vk}(x) = y$ and $\mathrm{OTS}.\mathsf{Vefy}(vk, b_c || y, \sigma) = 1$ hold simultaneously, else output $\perp$.

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

Enhanced 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 $\phi(x^*) = x^*$ 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, non-malleability is conventionally 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.

[12] We note that a family of universal hash functions is sufficient here.

In light of this distinction, the "continuous non-malleability" is actually a form of related-key security. In the rest of this work, we use the term "RKA-secure authenticated KDFs" instead of continuous non-malleable KDFs, and our RKA security notion allows non-trivial copy attacks.

9.2 RKA-secure Authenticated KDFs

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

Definition 9.1 (AKDFs)**.** AKDFs are given by three polynomial time algorithms as follows:

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

RKA security. AKDFs are said to be Φ-RKA-secure w.r.t. transformation class Φ if for any PPT adversary $\mathcal{A}$ its advantage defined in the following experiment is negligible in λ.

$$\mathsf{Adv}^{\mathrm{rka}}_{\mathcal{A},\mathrm{AKDF}}(\lambda) = \Pr\left[\beta' = \beta : \begin{array}{l} pp \leftarrow \mathsf{Setup}(\lambda); \\ (x^*, t^*) \leftarrow \mathsf{Sample}(pp); \\ k_0^* \leftarrow \mathsf{Derive}(x^*, t^*), k_1^* \xleftarrow{\mathrm{R}} K; \\ \beta \xleftarrow{\mathrm{R}} \{0,1\}; \\ \beta' \leftarrow \mathcal{A}^{\mathcal{O}^{\Phi}_{\mathsf{derive}}(\cdot,\cdot)}(pp, t^*, k_\beta^*); \end{array}\right] - \frac{1}{2}.$$

Here $\mathcal{O}^{\Phi}_{\mathsf{derive}}(\phi, t)$ 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 illegal query (underlined as above) is not allowed. By the result in [QLY+15], one can use RKA-secure AKDFs to transform a cryptographic primitive to an 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.

9.3 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 source key x^*. As we recalled in Section 8, 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 the 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 transformation 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 x^*, no PPT adversary is able to generate a valid key derivation query (ϕ, t) (here valid means that t is a valid tag of $\phi(x^*)$). In this way, the reduction can handle all key derivation queries without knowing x^*, 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 hinted NMFs w.r.t. $\mathcal{H}$, where $\mathcal{H}$ is a family of hardcore functions for $\mathcal{F}$. The Setup algorithm randomly picks f from $\mathcal{F}$ and h from $\mathcal{H}$. To generate a tag for a random source key x, one simply computes $t \leftarrow f(x)$. Intuitively, t serves as a non-malleable tag of x. To get a derived key from (x, t), one first checks if $f(x) = t$ and then outputs $k \leftarrow h(x)$ if so. Due to the hinted non-malleability of NMFs, all key derivation queries can be safely rejected, and thus the pseudorandomness of the derived key can be reduced to the pseudorandomness of h. We present our generic construction and formal security proof in details as below.

Our construction. Let $\mathcal{F} = (\mathsf{Gen}, \mathsf{Eval}, \mathsf{Vefy})$ be a family of Φ-hinted-NMF described above and $\mathcal{H}$ be a family of associated hardcore functions that maps X to K. We show how to build Φ'-RKA-secure AKDFs from it, where $\Phi' = \Phi \cup \mathsf{cf}$.[13]

- $\mathsf{Setup}(\lambda)$: run $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, $h \leftarrow \mathcal{H}$, output $pp = (f, h)$.
- $\mathsf{Sample}(pp)$: pick $x \xleftarrow{\mathrm{R}} X$, compute $t \leftarrow f(x)$, output (x, t).
- $\mathsf{Derive}(x, t)$: if $\mathcal{F}.\mathsf{Vefy}(f, x, t) = 0$, output $\perp$; otherwise output $k \leftarrow h(x)$.

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

Theorem 9.1. *The above construction of AKDFs is Φ'-RKA-secure if $\mathcal{F}$ is Φ-hinted-non-malleable w.r.t. $\mathsf{hint} = h$, where $\Phi' = \Phi \cup \mathsf{cf}$.*

Proof. We prove this theorem via a sequence of games.

Game 0 (return k_0^*): $\mathcal{CH}$ 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 $pp = (f, h)$; picks $x^* \xleftarrow{\mathrm{R}} X$, computes $t^* \leftarrow f(s^*)$, $k_0^* \leftarrow h(x^*)$, sends (pp, t^*, k_0^*) to $\mathcal{A}$ as the challenge.
2. Upon receiving an 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 $h(\phi(x^*))$ if $\mathcal{F}.\mathsf{Vefy}(f, \phi(x^*), t) = 1$ or $\perp$ otherwise.

Game 1 (handle constant queries without x^*): Same as Game 0 except that in step 2 $\mathcal{CH}$ handles all constant queries $\langle \phi, t \rangle$ without s^*. Here the term "constant" means all queries where $\phi \in \mathsf{cf}$.

- $\phi \in \mathsf{cf}$ and all t: suppose ϕ is a constant transform that maps all its inputs to a constant c, return $h(c)$ if $\mathcal{F}.\mathsf{Vefy}(f, c, t) = 1$ and $\perp$ otherwise.

These modifications are purely conceptual and hence Game 0 and Game 1 are identical in any PPT adversary's view.

Game 2 (handle non-constant queries without x^*): Same as Game 1 except $\mathcal{CH}$ directly returns $\perp$ for all non-constant queries $\langle \phi, t \rangle$. Here the term "non-constant" means $\phi \in \Phi$. Let E be the event that $\mathcal{A}$ issues a non-constant query $\langle \phi, t \rangle$ such that $\mathcal{F}.\mathsf{Vefy}(f, \phi(x^*), t) = 1$. 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 the views in Game 1 and Game 2 are identical if E never occurs. We show that event E happens with negligible probability, assuming Φ-hinted-non-malleability of $\mathcal{F}$.

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

Lemma 9.2. $\Pr[E]$ *is negligible in* λ *assuming the hinted* Φ*-non-malleability of* $\mathcal{F}$*.*

Proof. Suppose $\mathcal{B}$ is an adversary against hinted Φ-non-malleability of $\mathcal{F}$ w.r.t. $\mathcal{H}$. Given $(f, h, y^*, h(x^*))$, where $f \leftarrow \mathcal{F}.\mathsf{Gen}(\lambda)$, $h \xleftarrow{R} \mathcal{H}$, $y^* \leftarrow f(x^*)$ for $x^* \xleftarrow{R} X$, $\mathcal{B}$ simulates $\mathcal{A}$'s challenger in Game 2 as below: set $pp = (f, h)$, $t^* = y^*$, $k^* \leftarrow h(x^*)$, then send (pp, t^*, k^*) to $\mathcal{A}$. Here x^* is unknown to $\mathcal{B}$. Nevertheless, this is not a problem because $\mathcal{B}$ does not need x^* to handle RKA queries according to the definition of Game 2. Let L be the list of all non-constant 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, $\mathcal{B}$'s advantage is at least $\Pr[E]/\mathsf{poly}(\lambda)$. If $\Pr[E]$ is non-negligible, $\mathcal{B}$'s advanatge is also non-negligible. This contradicts the assumed hinted Φ-non-malleability of $\mathcal{F}$. The lemma immediately follows. □

Game 3 (return k_1^*): Same as in Game 2 except that $\mathcal{CH}$ returns $k_1^* \xleftarrow{R} K$. It is easy to see the views in Game 2 and Game 3 are computationally indistinguishable assuming the pseudo-randomness of hardcore function.

Game 4 (handle non-constant queries normally): This change is the mirror image of that between Game 1 and Game 2. By a similar argument in the proof of Lemma 9.2, the adversary's views in Game 3 and Game 4 are computationally indistinguishable by the Φ-non-malleability of $\mathcal{F}$, which is implied by the hinted Φ-non-malleability.

Game 5 (handle constant queries normally): This change is the mirror image of that between Game 0 and Game 1, which is purely conceptual. Thereby, adversary's view in Game 4 and Game 5 are identical.

Putting it all together, Game 0 and Game 5 are computationally indistinguishable. Note that Game 0 corresponds to the experiment of AKDF that returns k_0^*, and Game 5 corresponds to the experiment of AKDF that returns k_1^*. The theorem immediately follows. □

Instantiations. By instantiating the above generic construction with the deterministic NMFs presented in Section 7.1, we obtain deterministic AKDF (for each derivation key, there exists only one valid tag) which is RKA-secure w.r.t. $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id} \cup \mathsf{cf}$. By instantiating the above generic construction with the randomized NMFs presented in Section 7.2, we obtain randomized AKDF (for each derivation key, there exists possibly many valid tag) which is RKA-secure w.r.t. $\Phi^{\mathsf{poly}(d)}$. Particularly, our second instantiation of AKDFs simplifies and clarifies the construction due to Qin et al. [QLY+15]. In details, Qin et al.'s AKDF construction is based on OTLF, OTS and a family of pairwise independent hash functions, while our second instantiation of AKDF actually indicates that OTLF can be replaced by ABOLF and pairwise independent hash function can be replaced by universal hash functions. Note that OTLF is arguably stronger than ABOLF. To date, the only known construction of OTLF is based on ABOLF and chameleon hash functions. Besides, our second instantiation of AKDF is based on NMFs. This indicates Qin et al.'s construction of AKDFs is essentially building randomized NMFs.

9.4 Optimizations

Increasing the length of derived key. 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 pseudorandom generator.

Relaxation on NMFs. We observe that in the above construction, NMFs can be relaxed to non-malleable relations (NMRs). In NMRs, instead of requiring f to be efficiently computable, we only require that the distribution $(x, f(x))$ for a random x is efficiently sampleable. Analogous to the constructions we presented in Section 7.1 and Section 7.2, NMRs can be constructed from either adaptive trapdoor relations (ATDRs) [Wee10] or all-but-one lossy relations (ABOLRs) and one-time signature. Due to the relaxation on functionality, ATDRs (resp. ABOLRs) generally permit more efficient realizations than ATDFs (resp. ABOLFs) from a variety of standard assumptions. This finally provides us with 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 an 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 x.[14] AKDFs satisfying such strong RKA notion can be constructed from adaptive NMFs. We note our deterministic construction from ATDFs in Section 7.1 is already adaptively non-malleable, while our randomized construction from ABOLF and OTS can achieve adaptive non-malleability by replacing ABOLF with ABOLTF. This not only justifies the utility of the adaptive non-malleability notion, but also supports the view of Kiltz et al. [KMO10] that "ATDFs may be useful in the general context of black-box constructions of cryptographic primitives secure against adaptive attacks" and finds new application of ABOLTFs.

Acknowledgments

We particularly thank Zongyang Zhang for bringing up the work [BFS11] to our attention, and thank Shuai Han and Shengli Liu for insightful comments. We are grateful to Yu Yu, Qiong Huang and Marc Fischlin for helpful discussions and advice. We also thank the anonymous reviewers of PKC 2016 for their useful comments.

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.

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

[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.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How

to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[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.

[DY13] Yevgeniy Dodis and Yu Yu. Overcoming weak expectations. In *Theory of Cryptography - 10th Theory of Cryptography Conference, TCC 2013*, volume 7785 of *LNCS*, pages 1–22. Springer, 2013.

[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.

[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 *Pro-*

ceedings 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.

[QL14] Baodong Qin and Shengli Liu. Leakage-flexible cca-secure public-key encryption: Simple construction and free of pairing. In *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography*, volume 8383 of *LNCS*, pages 19–36. Springer, 2014.

[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.

[RS09] Alon Rosen and Gil Segev. Chosen-ciphertext security via correlated products. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009*, volume 5444 of *LNCS*, pages 419–436. Springer, 2009.

[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.

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.[15] The strengthened Merkle-Damgård transformation MD is defined as:

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

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

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 $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$-non-malleable.

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 $\Phi^{\mathrm{srs}}_{\mathrm{brs}} \cup \mathsf{id}$-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 $y^* = \mathsf{MD}^h_{iv}(x^*)$.*

[15] 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.

Proof. We prove this lemma by showing that if there exists a PPT adversary $\mathcal{A}$ that has non-negligible advantage against $\Phi^{\mathsf{srs}}_{\mathsf{brs}}$-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 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 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 ϵ, then $\Pr[\wedge^k_{i=1} x_i = x'_i] \geq \epsilon$. 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. Therefore, we must have $y'_j = y_j$ and $x'_j = x_j$ for each $j \in [k]$. This proves the above claim.

2. $\mathcal{B}$ then recovers $x' \in \{0,1\}^{n(\lambda)}$ from $(x'_1, \ldots, x'_k)$. We first observe that conditioned on $\mathcal{A}$ succeeds, we must have $\phi \neq \mathsf{id}$ because $\mathsf{MD}^h_{iv}(\cdot)$ is deterministic. According to the above claim, if $\mathcal{A}$ succeeds with non-negligible probability ϵ, then $\Pr[\wedge^k_{i=1} x'_i = x_i] \geq \epsilon$ 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 $\Pr[x' = \phi(x^*)] \geq \epsilon$ and the BRS & SRS properties of $\Phi^{\mathsf{srs}}_{\mathsf{brs}}$, 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. □

B Missing Cryptographic Primitives

B.1 Universal Hash Functions

A family of functions $\mathcal{H} = \{h_i : X \to Y\}$ from a domain X to range Y is said to be universal if, for every distinct $x, x' \in X$, $\Pr_{h \leftarrow \mathcal{H}}[h(x) = h(x')] = 1/|X|$.

B.2 Strongly Unforgeable One-Time Signatures

A signature scheme consists of three polynomial time algorithms as follows:

- $\mathsf{Gen}(\lambda)$: on input a security parameter λ, output a verification key vk and a signing key sk.
- $\mathsf{Sign}(sk, m)$: on input a signing key sk and a message $m \in M$ (where M is some fixed message space, possibly depending on λ), outputs a signature σ.
- $\mathsf{Vefy}(vk, m, \sigma)$: on input a verification key vk, a message $m \in M$ and a signature σ, outputs either 0 to indicate σ is invalid or 1 to indicate σ is valid.

Correctness. For any $(vk, sk) \leftarrow \mathsf{Gen}(\lambda)$ and any $m \in M$, $\mathsf{Vefy}(vk, m, \mathsf{Sign}(sk, m)) = 1$.

Strong unforgeability. Let $\mathcal{A}$ be an adversary against signature and define its advantage in the following experiment:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[\begin{array}{cc} \mathsf{Vefy}(vk, m, \sigma) = 1 & : \quad (vk, sk) \leftarrow \mathsf{Gen}(\lambda); \\ \wedge (m, \sigma) \notin \mathcal{Q} & \quad\; (m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{sign}}(\cdot)}(vk); \end{array}\right] \leq \mathsf{negl}(\lambda)$$

Here $\mathcal{O}_{\mathsf{sign}}(\cdot)$ is a signing oracle that on input m returns $\sigma \leftarrow \mathsf{Sign}(sk, m)$. The set $\mathcal{Q}$ contains pairs of queries to $\mathcal{O}_{\mathsf{sign}}(\cdot)$ and their associated responses. A signature is said to be strongly unforgeable under one-time chosen message attack if no PPT adversary has non-negligible advantage in above experiment by accessing $\mathcal{O}_{\mathsf{sign}}(\cdot)$ once.

B.3 All-But-One Lossy Functions

Qin and Liu [QL14] introduced the notion of all-but-one lossy functions (ABOLFs), which could be viewed as the trapdoor-free version of all-but-one lossy trapdoor functions [PW08]. Formally, a collection of (X, Y, τ)-ABOLFs with branch set B consists of two polynomial time algorithms satisfying the following properties:

- $\mathsf{Gen}(\lambda, b^*)$: on input a security parameter λ and any $b^* \in B$, output a function index s. For any $b \neq b^*$, $g_{s,b}(\cdot)$ is an injective function from X to Y, while $g_{s,b^*}(\cdot)$ is a lossy function from X to Y whose image has size at most 2^{τ}.
- $\mathsf{Eval}(s, b, x)$: on input a function index i and a branch $b \in B$ and an element $x \in X$, output $y \leftarrow g_{s,b}(x)$.

Hidden lossy branch. For any $b_0^*, b_1^* \in B \times B$, the output s_0 of $\mathsf{Gen}(\lambda, b_0^*)$ and the output s_1 of $\mathsf{Gen}(\lambda, b_1^*)$ are computationally indistinguishable.

B.4 One-Time Lossy Filters

Qin and Liu [QL13] introduced the notion of one-time lossy filters (OTLFs), in which a lossy branch could be generated on-the-fly in a somewhat semi-customized (or adversary-dependent) manner. A collection of (X, Y, τ)-OTLFs with branch set $B = B_c \times B_a$ (where B_c is the core branch set and B_a is the auxiliary branch set) consists of three polynomial time algorithms satisfying the following properties:

- $\mathsf{Gen}(\lambda)$: on input a security parameter λ, output a function index s and a trapdoor td. B contains two disjoint subsets, the subset of injective branches B_{inj} and the subset of lossy branches B_{lossy}. For any $b \in B_{\mathrm{inj}}$, $g_{ek,b}(\cdot)$ determines an injective function from X to Y. For any $b \in B_{\mathrm{lossy}}$, $g_{ek,b}(\cdot)$ determines a lossy function from X to Y whose image has size at most 2^{τ}.

- $\mathsf{Eval}(s, b, x)$: on input a function index s, $b \in B$ and an element $x \in X$, output $y \leftarrow g_{ek,b}(x)$.
- $\mathsf{SampLossy}(td, b_a)$: on input a trapdoor td and an auxiliary branch b_a, output a core branch b_c such that $b = (b_c, b_a)$ is lossy branch from B_{lossy}.

Indistinguishability. For any auxiliary branch $b_a \in B_a$, a random lossy core branch $b_c \leftarrow \mathsf{SampLossy}(td, b_a)$ and a random core branch $b_c \xleftarrow{R} B_c$ are computationally indistinguishable.

Evasiveness. For any PPT adversary, it is hard to generate a new lossy branch even given a lossy branch.