Expedient Non-Malleability Notions for Hash Functions

Paul Baecher Marc Fischlin Dominique Schröder

Darmstadt University of Technology, Germany
www.minicrypt.de

Abstract. Non-malleability of a cryptographic primitive is a fundamental security property which ensures some sort of independence of cryptographic values. The notion has been extensively studied for commitments, encryption and zero-knowledge proofs, but it was not until recently that the notion—and its peculiarities—have been considered for hash functions by Boldyreva et al. (Asiacrypt 2009). They give a simulation-based definition, basically saying that for any adversary mauling hash values into related ones there is a simulator which is as successful in producing such hash values, even when not seeing the original hash values. Their notion, although following previous approaches to non-malleability, is nonetheless quite unwieldy; it is hard to achieve and, due to the existential quantification over the simulator, hard to falsify. We also note that finding an equivalent indistinguishability-based notion is still open.

Here we take a different, more handy approach to non-malleability of hash functions. Our definition avoids simulators completely and rather asks the adversary to maul the hash value and to also specify a transformation ϕ of the pre-image, taken from a fixed class Φ of admissible transformations. These transformations are usually determined by group operations and include such cases such as exclusive-ors (i.e., bit flips) and modular additions. We then simply demand that the probability of succeeding is negligible, as long as the original pre-image carries enough entropy. We continue to show that our notion is useful by proving that, for example, the strengthened Merkle-Damgård transformation meets our notion for the case of bit flips, assuming an ideal compression function. We also improve over the security result by Boldyreva et al., showing that our notion of non-malleability suffices for the security of the Bellare-Rogaway encryption scheme.

Keywords. Hash function, Non-malleability.

1 Introduction

Non-malleability, first treated formally in [DDN00] for commitments, encryption and zero-knowledge, provides some level of independence between cryptographic values. That is, a commitment or ciphertext should not help to generate another commitment or ciphertext of a related message. By this, the adversary should not be able to produce a meaningful value by flipping some bits in a commitment or ciphertext. Several subsequent works considered this notion for the aforementioned primitives [DDN00, BS99, Sah99, DIO98, FF00, DDO$^+$01, Bar02, DG03, PR05a, PR05b, CHS06, PPV08, CD08, CV09, LP09, LPTV10, LPV08, OPV10, OPV08].

Non-Malleability of Hash Functions. Non-malleability of hash functions was scrutinized only recently by Boldyreva et al. [BCFW09]. They provide a formal treatment of the primitive and discuss applications to the Bellare-Rogaway encryption scheme and client puzzles. Other applications of non-malleable hash functions include security proofs for HMAC [Fis08] and for OAEP [BF06]. The idea behind non-malleable hash functions, put forward in [BCFW09], is similar to other areas but also reveals some differences, originating from the fact that, unlike commitments and encryptions, hash functions do not have secret randomness.

The definition in [BCFW09] follows the simulation paradigm: For every adversary which is able to transform hash values into related ones, there should be a simulator which outputs such related values but which is

denied the original values. This ensures independency of cryptographic values because given the value does not facilitate the task. However, the definition in [BCFW09] comes with several deficiencies:

- The requirement is quite strong: the hash function must not for instance leak individual bits of the input, or else the adversary would gain a significant advantage over any simulator by learning the hash value. While this seems to be a desirable goal for commitments and encryption, and possibly for some hash function applications, in general leaking some bits may not do harm to the fact that one mauls a hash value into something meaningful. Hence, it would be preferable to separate the notions of non-malleability and of pre-image hiding.

- The existential quantification over the simulator makes it hard to falsify (cf. Naor's work on cryptographic assumptions [Nao03]) the property for a specific hash function: one would need to show that for some adversary no appropriate simulator whatsoever exists. This is contrast to other desirable hash function properties like collision resistance.

- The definition in [BCFW09] covers pathological hash functions like constant functions: since the hash value does not lend any additional power to the adversary—after all, it is a constant— such a function is trivially non-malleable. Intuitively, though, for such functions it is easy to find related hash values and to determine hash values of the pre-image with some bits flipped. One can rule out such pathological functions by demanding collision-resistance or unpredictability, but this would introduce another assumption (which is somewhat unrelated, as we discuss).

- As discussed in [BCFW09], finding an alternative indistinguishability-based approach as for commitments and encryption seems to be hard. Very often, though, such a notion is easier to work with for proofs where the hash functions are used within larger schemes.

Non-Malleability goes Handy. We overcome the above problems by reverting to the core idea behind non-malleability: it should be hard to modify a given hash value such that the pre-image is affected in a controlled way. In contrast, the simulation-based approach somewhat guarantees more than this, by ensuring that the hash values are absolutely useless for doing so; this is formalized by having a simulator approximating the adversary's strategy but without learning the hash value.

Typically, the simulator in non-malleability proofs for hash functions runs a copy of the adversary, creates a fake hash value and presents it to the adversary in order to produce a related output. This, however, inhibits some designs which may otherwise guarantee the required "mauling resistance". As an example, if the hash function leaks a single bit of the input, then this clearly violates the simulation technique above. Nonetheless, determining a related hash value may still be hard for the adversary, because of the large unknown input portion. In fact, since practically designed hash functions should ensure that small changes in the input affect all output bits ("avalanche effect") it is likely that most hash functions would still withstand such mauling attacks: the adversary would still need to "control" the effect on the other bits. We note that the avalanche effect, albeit seemingly necessary, is not known to provide our desired level of non-malleability.[1]

We formalize the above non-malleability approach following related-key attacks on pseudorandom functions [Knu92, Bih94, BK04, Luc04]. There, the adversary can query the pseudorandom function with secret key k on values, but also specify a transformation ϕ of the key to receive values for related key $\phi(k)$. The class Φ of admissible functions ϕ is usually restricted, else achieving security in this setting is impossible [BK03].

In our case, we hand the adversary a hash value y of an unknown pre-image x and ask her to specify a transformation $\phi \in \Phi$ from a class of admissible transformations, together with a hash value y^*. The adversary

[1] We also note that the idea of reverting to basic modification attacks does not seem to be applicable to non-malleable commitments or encryptions in a reasonable way; the latter primitives are designated to hide the messages, whereas hash functions are a-priori not meant to provide such a guarantee.

wins if $x^* = \phi(x)$ hashes to y^*; to avoid trivial copy-attacks, we also demand that $x^* \neq x$. Non-malleability now requires that the probability of the adversary winning is negligible, implying that the (adversarially chosen) distribution of the unknown pre-image x contains super-logarithmic min-entropy. We denote this notion by Φ-non-malleability.

Similar to related-key attacks on pseudorandom functions, we cannot hope to achieve our notion of Φ-non-malleability for arbitrary classes Φ of transformations. Still, it comprises a large set of interesting transformation. For example, it contains the important class of all "bit flips", $\phi_\delta(x) = x \oplus \delta$ for all δ, for which we denote the notion by $\oplus$-non-malleability. More generally, one may consider any operations over groups, like $\phi_\delta(x) = x+\delta \bmod 2^n$ to capture modular additions. For a group $(G, \odot)$ and such group-induced transformations $\phi_\delta^{\odot}(x) = x \odot \delta$ we speak of $\odot$-non-malleability.

Applying our Non-Malleability Notion. We show that our notion $\odot$-non-malleability is strictly weaker than the simulation-based approach of Boldyreva et al. [BCFW09]. This, per se, would not be an interesting result, unless we can prove that the notion is still useful or enriches the class of suitable hash functions. We show both.

Clearly, since simulation-based non-malleability implies $\odot$-non-malleability, we can immediately conclude from the construction in [BCFW09] that $\odot$-non-malleable (and thus $\oplus$-non-malleable) hash functions can be derived under standard assumptions in principle. However, one of the ideas behind our notion is to give more guidance on how to design practical hash functions, and we rather envision more practical constructions from $\oplus$-non-malleable hash functions.

As for positive results, we show that the strengthened Merkle-Damgård, achieves the notion of $\oplus$-non-malleability if we assume an ideal compression function, emphasizing that restricting the class of admissible transformations allows to derive non-trivial non-malleability statements. Some of our results are also negative, though. We show that, due to length extension attacks, (strengthened) Merkle-Damgård is not Φ-non-malleable for a large class of transformations, even if we model the compression function as a random oracle. We also indicate that the Matyas-Meyer-Oseas (MMO) construction where one adds the input message to the output of a cipher shows some weaknesses when it comes to $\oplus$-non-malleability. More precisely, our result says that, if the compression function can be an arbitrary $\oplus$-non-malleable function, then MMO may not preserve this property. However, note that MMO itself assumes that the compression function is a block cipher, hence our result does not immediately apply to hash functions built from MMO like Skein [FLS$^+$08].

We finally show that $\oplus$-non-malleability suffices to show the Bellare-Rogaway (BR) encryption scheme [BR93] to be chosen-ciphertext secure, where a message is encrypted via $(f(r), G(r) \oplus m, H(r||m))$ for trapdoor permutation f, random r and hash functions G and H. Boldyreva et al. [BCFW09] prove that (simulation-based) non-malleability of H suffices, as long as G is still treated as a random oracle. To be precise, they also require H to be collision-resistant and perfectly one-way [Can97], hiding all information about the pre-image. Here we improve over their result and show the BR scheme is chosen-ciphertext secure (for random oracle G), as long as H is $\oplus$-non-malleable and perfectly one-way. This is an example where hiding the entire pre-image is now welcome and made explicit.

Other Related Work. "Bit-flipping" attacks have been known for a long time. It is the achievement of works like [DDN00] to put them into a general and formal framework and show how to avoid them and how to apply this security property. As explained above, our definition of Φ-non-malleability follows this line. It is inspired by the notion of related-key attacks [Bih94, Knu92, BK03] and the fact that the definition of Boldyreva et al. [BCFW09] is quite bulky.

It should be mentioned that a similar notion to $\oplus$-non-malleability appeared previously in Shoup's attack on the OAEP proof [Sho01]. Shoup gives an ad-hoc definition for XOR-malleable trapdoor permutations, i.e., trapdoor permutations which succumb to such bit-flip attacks, and he shows that OAEP is insecure when

instantiated with such a permutation. His notion, besides demanding the opposite of non-malleability, is slightly different (and incomparable) to ours in the sense that the adversary runs on a given difference δ and is not allowed to choose the distribution of inputs to the permutation. Our work here provides a positive, more expedient notion of *non*-malleable hash functions.

2 Preliminary Definitions

NOTATIONS. If x is a bit string then $|x|$ denotes its length and x_i the ith bit, whereas the least significant bit is at position $i = 0$; a bit position i is modulo $|x|$. By $x||y$ we denote the concatenation of two strings x and y and assume that it has the lowest evaluation priority. We assume (unless indicated otherwise) that all algorithms run in "probabilistic polynomial-time". If A is a set, we write $a \leftarrow A$ to indicate that a is picked uniformly at random from this set. The same syntax is used if A is a distribution (a is then drawn accordingly) and if $A(\cdot)$ is an algorithm, then a denotes its output. When dealing with a concatenated bit string $x_0||x_1$ where $|x_0| = |x_1|$, we often write left-hand (or right-hand) side to refer to x_0 (or x_1) and function $\mathsf{split}(b, x_0||x_1)$ returns x_b. Functions $\mathsf{poly}(\lambda)$ and $\mathsf{negl}(\lambda)$ denote any function that is polynomial in λ and negligible in λ, respectively. The security parameter is denoted by 1^λ. All logarithms are base 2.

The following definition captures a very general notion of hash functions, allowing also probabilistic schemes:

Definition 2.1 (Hash function) *A hash function $\mathcal{H} = (\mathsf{HK}, \mathsf{H}, \mathsf{HVf})$ consists of algorithms, where*

Ken Generation. $\mathsf{HK}(1^\lambda)$ *outputs a key K which implicitly defines a domain $D(K)$,*

Evaluation. H *for inputs K and $x \in D(K)$ evaluates to a value $y \in \{0,1\}^*$, and*

Verification. HVf *returns a decision bit for inputs K, x, y.*

It is required for any $K \leftarrow \mathsf{HK}(1^\lambda)$, any $x \in D(K)$, and any $y \leftarrow \mathsf{H}(K, x)$ that $\mathsf{HVf}(K, x, y)$ outputs 1.

We often require probability distributions to be non-trivial with regard to predictability. This means that it should be hard to predict which element is drawn from a given distribution when sampled, even in the presence of additional information in terms of a hint about the sample. We formalize this requirement as conditional min-entropy and resort to an equivalent formulation via predictors by [ADW09].

Definition 2.2 (Conditional min-entropy) *The conditional min-entropy of a distribution $\mathcal{X}$ conditioned on distribution $\mathcal{Z}$ is*

$$\tilde{\mathbf{H}}_\infty(\mathcal{X}|\mathcal{Z}) := -\log \max_{\mathcal{A}} \Pr\left[\mathcal{A}(\mathcal{Z}) = \mathcal{X}\right]$$

where the probability is taken over the random coins of $\mathcal{A}$, $\mathcal{Z}$ and $\mathcal{X}$.

In our case, the distribution $\mathcal{Z}$ is often dependent on $\mathcal{X}$, e.g., a function of $\mathcal{X}$.

3 Defining Non-Malleability

We first present the original definition from Boldyreva et al. [BCFW09], before introducing our notion of Φ-non-malleability and discussing their relationship.

3.1 Simulation-Based Non-Malleability

The idea of simulation-based non-malleability originates from [DDN00], where it has been defined, among others, for private key encryption. It states, intuitively, that for given ciphertext $C(x)$ of a message x, it should be computationally hard to find $C(x^*)$ where x and x^* are related in some interesting way. In their recent work Boldyreva et al. [BCFW09] provide a simulation-based definition of non-malleable hash functions. This definition constitutes the natural translation of simulation-based non-malleable encryption: For given $\mathsf{H}(x)$, it should be computationally hard to find $\mathsf{H}(x^*)$ where x and x^* are meaningfully related, defined via a relation R form a class of relations $\mathcal{R}$.

As with any simulation-based definition, the hardness of finding x and x^* is expressed over the probability that an efficient attacker is not significantly more successful than a simulator in an idealized version of the same task. That means, it is required that for any efficient attacker there exists a corresponding simulator that does just as well. In this specific case, both algorithms have to output x^* eventually, but the attacker is given $\mathsf{H}(x)$ while the simulator does not have access to $\mathsf{H}(x)$. The following definition is almost verbatim from [BCFW09] with a minor change (discussed afterwards):

Definition 3.1 (NM-Hash) *A hash function $\mathcal{H} = (\mathsf{HK}, \mathsf{H}, \mathsf{HVf})$ is called non-malleable (with respect to probabilistic algorithm hint and relation class $\mathcal{R}$) if for any PPT algorithm $\mathcal{A} = (\mathcal{A}_d, \mathcal{A}_y, \mathcal{A}_x)$ there exists a PPT algorithm $\mathcal{S} = (\mathcal{S}_d, \mathcal{S}_x)$ such that for every relation $R \in \mathcal{R}$ the difference*

$$\Pr\left[\mathbf{Exp}^{nmh\text{-}1}_{\mathcal{H},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{nmh\text{-}0}_{\mathcal{H},\mathcal{S}}(\lambda) = 1\right]$$

is negligible, where:

Experiment $\mathbf{Exp}^{nmh\text{-}1}_{\mathcal{H},\mathcal{A}}(\lambda)$	***Experiment*** $\mathbf{Exp}^{nmh\text{-}0}_{\mathcal{H},\mathcal{S}}(\lambda)$
$K \leftarrow \mathsf{HK}(1^\lambda)$	$K \leftarrow \mathsf{HK}(1^\lambda)$
$(\mathcal{X}, st_d) \leftarrow \mathcal{A}_d(K)$	$(\mathcal{X}, st_d) \leftarrow \mathcal{S}_d(K)$
$x \leftarrow \mathcal{X}(1^\lambda)$, $h_x \leftarrow \mathsf{hint}(K, x)$	$x \leftarrow \mathcal{X}(1^\lambda)$, $h_x \leftarrow \mathsf{hint}(K, x)$
$y \leftarrow \mathsf{H}(K, x)$	
$(y^*, st_y) \leftarrow \mathcal{A}_y(y, h_x, st_d)$	
$(x^*, r) \leftarrow \mathcal{A}_x(x, st_y)$	$(x^*, r) \leftarrow \mathcal{S}_x(h_x, st_d)$
Return 1 iff	*Return 1 iff*
$\quad R(\mathcal{X}, x, x^*, r)$	$\quad R(\mathcal{X}, x, x^*, r)$
$\quad \wedge\, (x, y) \neq (x^*, y^*)$	
$\quad \wedge\, \mathsf{HVf}(K, x^*, y^*) = 1$	

where $\mathcal{X}$ denotes a distribution that is chosen by the attacker $\mathcal{A}_d$ or simulator $\mathcal{S}_d$ and st stands for state information passed among the algorithms. $\mathcal{H}$ is called entropically non-malleable if the above only holds for algorithms $\mathcal{A}_d$ and $\mathcal{S}_d$ that output distributions $\mathcal{X}$ such that $\tilde{\mathbf{H}}_\infty(\mathcal{X}|\mathsf{HK}, \mathsf{hint}) \in \omega(\log \lambda)$.

A wealth of remarks is presented in the original work; we only comment on the most important details:

- Both experiments provide the algorithms with a hint h_x, which reflects information about x that might have been collected from previous actions, such as other protocol executions that involve x.
- The informal statement that x and x^* are related in a meaningful way is represented by quantification over a class of relations $\mathcal{R}$. Ideally, we would want to allow any relation, but subtle technicalities described in [BCFW09] demand that the set of all possible relations is restricted; the definition becomes unachievable otherwise. This restriction can be expressed by excluding the problematic relations from $\mathcal{R}$.

- We introduced a small change concerning the output of $\mathcal{A}_x$ and $\mathcal{S}_x$ as well as the relation. In particular, the original definition omits the value r which gives the attacker a little more control over the relation. By specifying a parameter r along with x^*, the attacker is effectively able to select a specific subset of the relation. This can be viewed as a decomposition of a corresponding relation that omits r into subsets where the resulting subsets are indexed by r. Applying this modification will turn out useful later for our analysis, and does not affect the results in [BCFW09].
- We also define entropically non-malleability as a variant of non-malleability that requires attackers not to output trivial preimage distributions. This is useful because the attacker could otherwise attempt to guess the preimage that the experiment samples from the distribution instead of attacking non-malleability. The original definition does not formalize this explicitly but mentions it as a naturally arising condition.

3.2 Game-Based Non-Malleability

We avoid the drawbacks of the simulation-based approach mentioned in the introduction, by proposing an alternative definition that does not rely on a simulator. Our goal is to provide a definition that is more compact and accessible to both cryptanalysts and practitioners while it still captures the idea of non-malleability.

The Idea. One candidate realization of our goal is captured by the following game. After selecting a preimage distribution of inputs to some hash function h, the attacker is given image $\mathsf{H}(x)$ and is required to output $\mathsf{H}(x^*)$ along with a description of how to transform x into x^* in terms of a function ϕ. Note that this does not imply that the attacker knows x or x^*. Consider the following experiment which outlines this idea.

1. Attacker $\mathcal{A}$ outputs a preimage distribution $\mathcal{X}$.
2. A random preimage x is picked according to $\mathcal{X}$. Let $y = \mathsf{H}(x)$.
3. On input y, the attacker outputs y^* along with a function ϕ which maps one preimage to another, e.g., $\phi = \phi_\delta$ may be the function which maps x to $\phi_\delta(x) = x \oplus \delta$.
4. The output of the experiment is defined to be 1 if, and only if, $\mathsf{H}(\phi(x)) = y^*$ (and we have $\phi(x) \neq x$).

As a natural consequence for a non-malleability definition, we would require the probability that this experiment outputs 1 to be negligible in the security parameter. Modeling the description of how to transform the preimage with a function also allows the attacker to output modifications like $\phi(x) := x+1$. Unfortunately, this broad definition is unachievable. Consider, for example, an attacker that simply picks x^* at random from the preimage domain and outputs the constant function $\phi(\cdot) := x^*$ and $y := \mathsf{H}(x^*)$. It succeeds unconditionally in this experiment since $\mathsf{H}(\phi(x)) = \mathsf{H}(x^*) = y^*$, regardless of the hash function in question. This problem occurs whenever the adversary can provide a function which significantly reduces the size of the resulting range of the transformation function. We thus restrict the class Φ of functions ϕ.

Φ-Non-Malleability. Seeing that the above experiment is unachievable, the question remains if a weaker—but working—game-based definition exists that still reflects the intuition of non-malleability. We approach this question with the following idea: If two related values x and x^* exist, then there is a difference $\delta := x \oplus x^*$ that essentially describes how to modify one value in order to obtain the other. Yet, given the difference only, no information in absolute terms about x (or x^*, separately) can be inferred. We can view an attacker that outputs δ as a specialization of the experiment above by fixing $\phi(x) = x \oplus \delta$. The practical impact of this idea would be that an attacker has an understanding of how bit flips in the output affect bit flips in the input of the function (or vice versa).

It is possible to generalize this concept by allowing any group operation instead of bitwise differences. We note that a similar issue of specifying such a transformation function has been already used in the formal treatment of related-key attacks on pseudorandom functions, where the adversary is allowed to provide a key transformation. Here, Lucks [Luc04] proposes the class of group-induced transformations, a class which is neither too powerful nor too restrictive: It is the set of functions which apply the group operation to their argument and a fixed group element. Subsequent work in the area by Bellare and Cash [BC10] advocates the use of this class and strengthens our confidence that this is a "natural" choice for related key attacks. Since the requirements and issues seem to be very similar, we adopt this class in our definition of non-malleability.

We now turn this idea into a general formal definition that is similar to the simulation-based experiments. First, HK generates a key K which implicitly contains the security parameter λ. On input K, the first stage of the attacker $\mathcal{A}_d$ outputs a valid distribution of preimages $\mathcal{X}$ in the sense that it has a sufficient min-entropy. A random element x is drawn from this distribution and mapped to image y. The function hint outputs hint h_x about x. The attacker in the second stage then receives image y and hint h_x. It is required to output a modified image y^* and a preimage transformation function $\phi \in \Phi$ (where Φ is known to the adversary). The output of the experiment is defined to be 1 if, and only if, $\mathsf{HVf}(K, \phi(x), y^*) = 1$ and $\phi(x) \neq x$.

Definition 3.2 (Φ-NM-Hash) *A hash function $\mathcal{H} = (\mathsf{HK}, \mathsf{H}, \mathsf{HVf})$ is called Φ-non-malleable (with respect to probabilistic function hint) if for any PPT algorithm $\mathcal{A} = (\mathcal{A}_d, \mathcal{A}_y)$ the probability $\Pr[\mathbf{Exp}^{\Phi\mathsf{nm}}_{\mathcal{H},\mathcal{A}}(\lambda) = 1]$ is negligible in λ, where:*

Experiment $\mathbf{Exp}^{\Phi\mathsf{nm}}_{\mathcal{H},\mathcal{A}}(\lambda)$
$K \leftarrow \mathsf{HK}(1^\lambda)$
$(\mathcal{X}, \mathrm{st}) \leftarrow \mathcal{A}_d(K)$
$x \leftarrow \mathcal{X}(1^\lambda)$, $h_x \leftarrow \mathsf{hint}(K, x)$
$y \leftarrow \mathsf{H}(K, x)$
$(y^*, \phi) \leftarrow \mathcal{A}_y(y, h_x, \mathrm{st})$
Return 1 iff
$\mathsf{HVf}(K, \phi(x), y^*) = 1$
$\wedge\, \phi(x) \neq x$

where $\phi \in \Phi$. It is required that algorithm $\mathcal{A}_d$ only outputs efficiently sampleable distributions $\mathcal{X}$ such that $\tilde{\mathbf{H}}_\infty(\mathcal{X}|\mathsf{HK}, \mathsf{hint}) \in \omega(\log \lambda)$.

We require $\mathcal{A}_d$ to output a non-trivial distribution $\mathcal{X}$ in this experiment that is not predictable by demanding a conditional min-entropy strictly greater than logarithmic in λ. This requirement makes sense because, otherwise, an attacker can choose a distribution such that it succeeds to predict the most likely event of this distribution with high probability. In the context of our experiment, this would imply that the attacker guesses x with non-negligible probability and trivially succeeds.

Group-Induced Non-Malleability. Note that we let the adversary choose from a set of predetermined transformation functions. From this general version we get the aforementioned group-induced transformations by letting $\Phi = \{\phi^{\odot}_{\delta} : \delta \in G\}$ where $\phi^{\odot}_{\delta}(x) = x \odot \delta$ for some group $(G, \odot)$ for which group operations can be efficiently performed. Although this definition allows us to model even richer classes of transformation functions (e.g. many different group operations in the same experiment), one must take care to exclude cases where the definition becomes unachievable. In practice, one will likely want to work with one specific group. In fact, for this paper, we restrict ourselves to group-induced transformations. The following definition captures two special cases of group-induced transformations and the $\oplus$ operation:

Definition 3.3 ($(G,\odot)$-NM-Hash and $\oplus$-NM-Hash) *Let $(G,\odot)$ denote a group. A hash function $\mathcal{H}$ is called $(G,\odot)$-non-malleable if $\mathcal{H}$ is $\Phi^{\odot}$-non-malleable where $\Phi^{\odot} = \{\phi_\delta^{\odot} : \delta \in G\}$ and $\phi_\delta^{\odot}(x) = x \odot \delta$. We omit G if it is clear from the context. In particular, we call a $(\{0,1\}^*,\oplus)$-non-malleable function simply $\oplus$-non-malleable.*

It should be observed that Definition 3.3 is weaker than the original simulation-based definition. We discuss this in depth in the next section. Nevertheless, despite this limitation, we feel that Definition 3.3 is suitable to capture an essential aspect of non-malleability and a broad class of transformation functions. The absence of a simulator makes it relatively easy to work with and relates well to a practical view of an attacker.

3.3 Relations Between Simulation-Based and $\odot$-Non-Malleability

In this section, we show that every simulation-based non-malleable hash function is also $\odot$-non-malleable but $\odot$-non-malleable hash functions exist that are not simulation-based non-malleable.

3.3.1 Simulation-Based Non-Malleability $\Rightarrow$ $\odot$-Non-Malleability

We show by black-box reduction that every function which is not $\odot$-non-malleable is also not entropically non-malleable in the simulation-based sense.

Proposition 3.4 *Let $R_\odot(\mathcal{X}, x, x^*, r)$ denote any relation that outputs 1 if, and only if, $r = x \odot x^*$. If $\mathcal{H}$ is an entropically non-malleable hash function with respect to arbitrary hint and relation class $\mathcal{R} \ni R_\odot$, then $\mathcal{H}$ is $\odot$-non-malleable with respect to hint.*

Our proof shows the equivalent proposition that if $\mathcal{H}$ is not $\odot$-non-malleable then $\mathcal{H}$ is not entropically non-malleable.

Proof. Suppose that $\mathcal{H}$ is not $\odot$-non-malleable. Then PPT algorithm $\mathcal{A} = (\mathcal{A}_d, \mathcal{A}_y)$ exists such that $\Pr[\mathbf{Exp}^{\odot\mathsf{nm}}_{\mathcal{H},\mathcal{A}}(\lambda) = 1] = 1/\mathsf{poly}(\lambda)$. Construct $\mathcal{A}' = (\mathcal{A}'_d, \mathcal{A}'_y, \mathcal{A}'_x)$ against entropical non-malleability of $\mathcal{H}$ as follows. Let $\mathcal{A}'_d := \mathcal{A}_d$ and $\mathcal{A}'_y := \mathcal{A}_y$. On input $(x, \mathrm{st}_y) = (x, \phi)$, $\mathcal{A}'_x$ outputs $(\phi(x), x \odot \phi(x))$.

All stages of $\mathcal{A}'$ are obviously efficient. The view of algorithm $\mathcal{A}$ in this construction is identical to its view when run in $\mathbf{Exp}^{\odot\mathsf{nm}}_{\mathcal{H},\mathcal{A}}$. If, and only if, $\mathcal{A}$ succeeds in $\mathbf{Exp}^{\odot\mathsf{nm}}_{\mathcal{H},\mathcal{A}}$ we have $\phi(x) \neq x$ and $\mathsf{HVf}(K, \phi(x), y^*) = 1$. But since $\mathrm{st}_y = \phi$ and $x^* = \mathrm{st}_y(x)$ in $\mathbf{Exp}^{nmh-1}_{\mathcal{H},\mathcal{A}'}$ it follows that $(x,y) \neq (x^*, y^*)$ and $R(\mathcal{X}, x, \phi(x), x \odot \phi(x)) = R_\odot(\mathcal{X}, x, x^*, r) = 1$. Hence $\mathcal{A}'$ succeeds if and only if $\mathcal{A}$ succeeds under relation $R_\odot$, i.e.,

$$\Pr\left[\mathbf{Exp}^{nmh-1}_{\mathcal{H},\mathcal{A}'}(\lambda) = 1\right] = \Pr\left[\mathbf{Exp}^{\odot\mathsf{nm}}_{\mathcal{H},\mathcal{A}}(\lambda) = 1\right] = \frac{1}{\mathsf{poly}(\lambda)}.$$

It remains to show that any simulator $\mathcal{S}$ is successful in $\mathbf{Exp}^{nmh-1}_{\mathcal{H},\mathcal{S}}(\lambda)$ with negligible probability at most. First, note that any successful simulator in this experiment is also an algorithm that successfully predicts which value is drawn from $\mathcal{X}$. This follows from the fact that $\mathcal{S}_x$ outputs (x^*, r) and $r \odot (x^*)^{-1} = x$. However, since $\mathcal{S}$ does not have any information on x besides HK and hint, it follows from the high conditional min-entropy that $\mathcal{S}$ can only predict x with negligible probability. Thus, any simulator $\mathcal{S}$ which is successful in $\mathbf{Exp}^{nmh-1}_{\mathcal{H},\mathcal{S}}(\lambda)$ with non-negligible probability is a contradiction to this argument and cannot exist.

Altogether it then follows that

$$\left|\Pr\left[\mathbf{Exp}^{nmh-1}_{\mathcal{H},\mathcal{A}'}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{nmh-1}_{\mathcal{H},\mathcal{S}}(\lambda) = 1\right]\right| = |1/\mathsf{poly}(\lambda) - \mathsf{negl}(\lambda)|$$

is not negligible in λ and thus $\mathcal{H}$ is not entropically non-malleable. □

We remark that the implication actually holds in a more general sense, beyond group-induced transformations. Namely, assume that for every $\phi \in \Phi$ there exists $\phi^{-1} \in \Phi$ such that $\phi^{-1}(\phi(x)) = x$ for all x. Now define the relation $R_{-1}(\mathcal{X}, x, x^*, r)$ (instead of $R_{\odot}$) which outputs 1 if and only if $r(x^*) = x$ for $r \in \Phi$. Then any successful adversary against Φ-non-malleability could still be turned into a successful adversary against entropical non-malleability, whereas a simulator could output (x^*, r) predicting $r(x^*) = x$ with negligible probability only. As an example consider the class $\Phi^{||}$ of concatenation transformation consisting of function $\phi_\delta(x) = x||\delta$ and such that $\phi_\delta^{-1}(x||\delta) = x$ (or equals the input, in case the input string does not end on δ). This example will be useful when showing that Merkle-Damgård does not preserve Φ-non-malleability.

3.3.2 Φ-Non-Malleability $\not\Rightarrow$ Simulation-Based Non-Malleability

We show that if Φ-non-malleable functions exist, then there is one which is not (simulation-based) non-malleable. In particular, Construction 3.5 below is one example for such a function.

Construction 3.5 *Let $\mathcal{F} = (\mathsf{FK}, \mathsf{F}, \mathsf{HVf})$ denote a hash function. Define $\mathcal{G} := (\mathsf{GK}, \mathsf{G}, \mathsf{GVf})$ where*

- $\mathsf{GK} := \mathsf{FK}$,
- $\mathsf{G}(K, x) := \mathsf{F}(K, x)||x_1$, *and*
- $\mathsf{GVf}(K, x, y) := \mathsf{HVf}(K, x, y_1 y_2 \ldots y_{n-1}) \wedge (y_n = x_1)$.

Lemma 3.6 *For any class Φ it holds that, if $\mathcal{F}$ is Φ-non-malleable, then $\mathcal{G}$ is Φ-non-malleable.*

The proof follows straightforwardly by a black-box reduction as an adversary against $\mathcal{G}$ can simply guess the extra bit and still succeed with non-negligible probability. The proof is in Appendix A.

Proposition 3.7 *Let $R_1(\mathcal{X}, x, x^*, r)$ denote any relation that outputs 1 if, and only if, $x_1 = x_1^*$ and $\mathcal{X}$ is the uniform distribution. Then Construction 3.5 is not (simulation-based) non-malleable with respect to $\mathsf{hint} = \emptyset$ and relation class $\mathcal{R} \ni R_1$.*

For the proof of Proposition 3.7 we consider the specific relation R_1 consisting of inputs x, x^* which are equal in the first bit. In addition, this relation rejects distributions if they are not the uniform distribution. Since an attacker against Construction 3.5 is given the first bit as part of the image, it succeeds unconditionally. On the other hand, the simulator does not have this information and is forced to guess, but it succeeds only with a probability that makes the difference non-negligible. To complete the picture we finally note that relation R_1 is in the set of relations for which Boldyreva et al. [BCFW09] derive (simulation-based) non-malleable hash functions.

See again Appendix A for the full proof where we also show that non-malleability is not implied by collision-resistance or one-wayness.

Finally, we point out that it is an open question to identify precisely how much weaker Definition 3.2 is in contrast to the original definition. In particular, it is unclear if a Φ-non-malleable hash function that also hides all preimage information is also non-malleable in the simulation-based sense.

4 Constructions

As explained in the introduction, the Merkle-Damgård construction does not preserve (Φ-)non-malleability because of length-extension attacks, even if the compression function is modeled as a random oracle. Clearly,

this is the case for the class of transformations $\Phi^{||}$ with $\phi_\delta(x) = x||\delta$ first appending the padding and then arbitrary data.[2] It follows that Merkle-Damgård is not simulation-based non-malleable either.

Merkle-Damgård, Random Oracles, and $\oplus$-Non-Malleability. Somewhat surprisingly, we show that Merkle-Damgård does provide $\oplus$-non-malleability (for fixed-length messages) if the compression function is modeled as random oracle. This again shows the advantage of restricting the class of transformation Φ: while Merkle-Damgård is not simulation-based non-malleable and not $\Phi^{||}$-non-malleable according to the discussion above, it is for another class of transformation which suffices to show security of the BR encryption schemes (see Section 5).

Construction 4.1 (MD) *Let $\mathsf{pad}(M)$ be a padding function which maps messages to multiples of the block length such that the final 64 bits contain the message length. Then*

$$MD^h_{iv}(M) = h^*_{iv}(M_1|| \dots ||M_k) = h(\cdots h(h(iv, M_1), M_2) \cdots)$$

where $M_1|| \dots ||M_k = \mathsf{pad}(M)$.

In the following we usually denote by y_i the value $h^*_{\text{iv}}(M_1|| \dots ||M_i)$, i.e., the i-th intermediate value when iterating h.

Proposition 4.2 *For a random oracle h the hash function MD^h in Construction 4.1 is $\oplus$-non-malleable (with respect to arbitrary hint).*

Proof. Consider an adversary $\mathcal{A}$ against $\oplus$-non-malleability. Assume that this adversary for random $M \leftarrow \mathcal{X}$ and the hash value $y = h^*_{\text{iv}}(\mathsf{pad}(M))$ eventually outputs (y^*, δ). Note that, with overwhelming probability, if the adversary has not queried *all* pairs (M^*_i, y^*_i) of $M^*_1|| \dots ||M^*_k = \mathsf{pad}(M \oplus \delta)$ and intermediate values y^*_i in the hash computations to h, then the final output y^* does not constitute a valid hash value. This holds because if the adversary simply copies $y^* = y$ this would contradict the collision-resistance for $\delta \neq 0$; in any other case the adversary would otherwise be able to predict random oracle values.

Hence, given that the adversary has queried about all intermediate values, and since they are unique with overwhelming probability, we can deduce the adversary's message M^* and together with the output δ thus the original message M. This, however, would contradict the super-logarithmic min-entropy of $\mathcal{X}$ (because the additional hash value of M can only decrease this entropy by a logarithmic term for the efficient adversary making at most polynomial many queries to h). □

We are not aware if one can show that Merkle-Damgård is $\oplus$-non-malleable under standard assumptions. However, we note that $\oplus$-non-malleability of the compression function alone is not sufficient, but that collision resistance of the compression function is necessary. Consider for example a given $\odot$-non-malleable compression function h and modify it into a compression function

$$h'(w, x) = \begin{cases} h(w, x) & \text{if } x \neq 1 \dots 1 \\ h(w, 0 \dots 0) & \text{if } x = 1 \dots 1 \end{cases}$$

Then h' inherits $\odot$-non-malleability from h because the non-trivial min-entropy of the input distribution guarantees that x hits $1 \dots 1$ and that the adversary outputs ϕ_δ with $\phi_\delta(x) = x \odot \delta = 1 \dots 1$ only with negligible probability. In any other case breaking non-malleability of h' requires breaking the non-malleability of h.

[2] Note that an attacker is not even limited to fixed-length preimage distributions, since the length is polynomial and can simply be guessed.

Now consider the MD construction based on h'. It is easy for the adversary to specify a distribution $\mathcal{X}$ which outputs λ message blocks, and each block consists of $0 \dots 0$ or $1 \dots 1$ with probability $1/2$ each. Clearly, this distribution has super-logarithmic min-entropy. But, the adversary can now output the same hash value $y^* = y$ and $\delta = (1 \dots 1)^\lambda \neq 0$ and win the $\oplus$-non-malleability game (because $x^* = x \oplus \delta$ maps to the same hash value).

The Matyas-Meyer-Oseas-Construction. A common technique to build hash functions—more precisely compressions functions—is to start from a block cipher. Preneel et al. [PGV94] discuss 64 such variants which all rely on one call to a block cipher. One of these variants, proposed earlier by Matyas et al. [MMO85], is the Matyas-Meyer-Oseas (MMO) scheme given by $h(k, m) = C(g(k), m) \oplus m$ where g an arbitrary function. Amongst others, the SHA-3 candidate hash function Skein [FLS+08] is known to adopt this scheme.

In this section, we deal with a construction called *hash-$\oplus$-composition* which is quite similar to the MMO scheme and show that it does not sustain $\oplus$-non-malleability. The difference is that, instead of using a cipher C (and function g), we substitute this part with an inner function h', i.e. $h(k, m) = h'(k, m) \oplus m$. This may be a hash function or, more general, any compression function. We then construct a $\oplus$ non-malleable h' and show that the overall construction becomes $\oplus$-malleable. The general idea is presented below; for a full formal treatment see Appendix B.

Consider a hash function u, an $\oplus$-non-malleable hash function f and define

$$h'(x_0||x_1) = (u(x_0) \oplus (f(x_0)||f(x_1)))||(x_0 \oplus x_1).$$

Function h' is then $\oplus$-non-malleable under certain assumptions, namely that (a) an attacker against $\oplus$-non-malleability outputs only uniform distributions and (b) function u is modeled as a random oracle. Intuitively, the second assumption assures that the first half of h' is padded by true randomness and the first assumption guarantees that the second half of h' is uniformly random. These two properties combined ensure that h' is $\oplus$-non-malleable. However, it becomes completely insecure in the above scheme. Since x_1 is canceled out, the attacker learns x_0 and is thus able to recompute u and f for arbitrary new values. It is easy to see that this breaks ($\oplus$-)non-malleability.

How does the above result affect hash functions constructed by the MMO paradigm such as Skein? Strictly speaking, it is not applicable, because the MMO schemes uses a cipher whereas the hash-$\oplus$-composition (and the counterexample above) assumes a hash function. Yet, we feel that it is not far-fetched to consider a cipher with unknown key and a hash function functionally similar. For example, collision resistance is implied by the necessary permutation property of a cipher. Likewise, one-wayness is closely related to the security of the cipher: If one is able to invert an "image" of the cipher without knowing its key, then the cipher is blatantly broken.

5 Application

In this section we revisit the proof in [BCFW09] that (simulation-based) non-malleability of the hash function H in the Bellare-Rogaway encryption [BR93] scheme $\mathsf{f}(r),\ G(r) \oplus m,\ H(r||m)$, together with some form of perfect one-wayness hiding the entire pre-image, suffices to achieve chosen-ciphertext security (as long as G is still modeled as a random oracle). Exploiting the fact that the component $G(r) \oplus m$ uses the exclusive-or we show that $\oplus$-non-malleability (and perfect one-wayness) is sufficient for H.

Preliminaries. We first recall the BR encryption scheme $\mathrm{BR}^{G,\mathcal{H}}[\mathcal{F}] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ and the instantiation of the random oracle H through a hash function $\mathcal{H} = (\mathsf{HK}, \mathsf{H}, \mathsf{HVf})$ formally. The key generation algorithm $\mathcal{K}$ of the encryption scheme outputs a random $\mathcal{F}$-instance f and its inverse f^{-1} as the public and secret key, respectively.

It also runs HK to generate a key $K \leftarrow \mathsf{HK}(1^k)$ for a hash function and puts K into both the public and secret key. The encryption algorithm $\mathcal{E}$ on inputs f, K and m picks random r in the domain of f and outputs $(\mathsf{f}(r), G(r) \oplus m, \mathsf{H}(K, r||m))$. The decryption algorithm on inputs f^{-1}, K and (y, g, h) first computes $r \leftarrow \mathsf{f}^{-1}(y)$, then $m \leftarrow g \oplus G(r)$, and outputs m iff $\mathsf{HVf}(K, r||m, h) = 1$ (and $\perp$ otherwise).

We would like to show that the encryption is IND-CCA, meaning that for any adversary with access to a decryption oracle and receiving the public key as input, the adversary cannot distinguish encryptions for chosen messages m_0, m_1 (of equal length) better than by guessing. It is understood that the adversary, after receiving the challenge ciphertext of m_b for random bit b cannot query the decryption oracle about this ciphertext.

Next, we define the hiding property of the hash function $\mathcal{H}$ formally. We start by recalling the definition of a perfectly one-way hash function [Can97]:

Definition 5.1 (POWHF) *A hash function $\mathcal{P} = (\mathsf{POWK}, \mathsf{POW}, \mathsf{POWVf})$ is called a perfectly one-way hash function (with respect to probabilistic function hint) if for any PPT algorithm $\mathcal{B} = (\mathcal{B}_d, \mathcal{B}_b)$, where $\mathcal{B}_b$ has binary output, the following random variables are computationally indistinguishable:*

$K \leftarrow \mathsf{POWK}(1^k)$	$K \leftarrow \mathsf{POWK}(1^k)$
$(\mathcal{X}, st_d) \leftarrow \mathcal{B}_d(K)\,; x \leftarrow \mathcal{X}(1^k)$	$(\mathcal{X}, st_d) \leftarrow \mathcal{B}_d(K)\,; x \leftarrow \mathcal{X}(1^k)$
	$x' \leftarrow \mathcal{X}(1^k)$
$h_x \leftarrow \mathsf{hint}(K, x)\,; y \leftarrow \mathsf{POW}(K, x)$	$h_x \leftarrow \mathsf{hint}(K, x)\,; y' \leftarrow \mathsf{POW}(K, x')$
$b \leftarrow \mathcal{B}_b(y, h_x, st_d)$	$b \leftarrow \mathcal{B}_b(y', h_x, st_d)$
return (K, x, b)	*return* (K, x, b)

Security Proof. As in [BCFW09] we also assume for technical reasons that the hash function $\mathcal{H}$ is $\oplus$-non-malleable when a random instance of $\mathcal{F}$ is included with the key output by HK. Let $\mathcal{H} = (\mathsf{HK}_{\mathcal{F}}, \mathsf{H}, \mathsf{HVf})$ denote the modified hash function for which key generation outputs a random instance of $\mathcal{F}$ together with the original hash key. Then we can include side information about the sample in terms of f, i.e., we too need that $\mathcal{H}$ is $\oplus$-non-malleable with respect to the hint function $\mathsf{hint}_{\mathsf{BR}}((K, \mathsf{f}), r||m) = \mathsf{f}(r)$.

Theorem 5.2 *Let $\mathcal{F}$ be a trapdoor one-way permutation and $\mathcal{H} = (\mathsf{HK}_{\mathcal{F}}, \mathsf{H}, \mathsf{HVf})$ be a perfectly one-way hash function with respect to hint_{BR} which is also $\oplus$-non-malleable with respect to hint_{BR}. Then $BR^{G,\mathcal{H}}[\mathcal{F}] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ for messages of length $\ell = \omega(\log \lambda)$ is IND-CCA secure (for random oracle G).*

The proof in Appendix C proceeds in game hops, starting with the original attack scenario and transforming this into a game where the adversary has no advantage. Letting (y^*, g^*, h^*) denote the challenge ciphertext of message m_b, we have:

Game$_0$: Corresponds to the original attack of the adversary $\mathcal{A}$.

Game$_1$: Reject all ciphertexts (y, g, h) submitted to the decryption oracle, where $y \neq y^*$ and $\mathcal{A}$ has never queried G about $r = \mathsf{f}^{-1}(y)$ before (including the case that y is not in the range of f). (This cannot change the adversary's success probability significantly; else the adversary would need to predict a hash value for an unknown random input, which can be shown to contradict $\oplus$-non-malleability.)

Game$_2$: Reject all ciphertexts (y, g, h) submitted to the decryption oracle before receiving the challenge ciphertext, where $y = y^*$. (Happens with negligible probability only.)

Game$_3$: Reject all ciphertexts (y, g, h) submitted to the decryption oracle, where $y = y^*$, $g = g^*$ and $h \neq h^*$ (queries after receiving the challenge ciphertext, where also $h = h^*$, are not allowed). (The hash value must be invalid then.)

Game$_4$: Reject all ciphertexts (y, g, h) submitted to the decryption oracle, where $y = y^*$ but $g \neq g^*$, and $\mathcal{A}$ has never queried G about $r = \mathsf{f}^{-1}(y)$ before. (Would contradict the $\oplus$-non-malleability, since $g \oplus g^*$ essentially determines the difference between the pre-images under H.)

Game$_5$: Replace the computation of $h^* \leftarrow \mathsf{H}(K, r||m_b)$ in the challenge ciphertext by $h^* \leftarrow \mathsf{H}(K, r'||0^{|m_b|})$. (Is a valid hop because of the perfect one-wayness.)

Game$_6$: Replace the computation of $g^* \leftarrow G(r) \oplus m_b$ in the challenge ciphertext by picking g^* uniformly at random. (Can only be detected by the adversary if she queries G about r^* at some point, which would contradict the one-wayness of f.)

Acknowledgments

We thank the anonymous reviewers for valuable comments. The authors are supported by the Emmy Noether Grant Fi 940/2-1 of the German Research Foundation (DFG).

References

[ADW09] Joël Alwen, Yevgeniy Dodis, and Daniel Wichs. Leakage-resilient public-key cryptography in the bounded-retrieval model. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 36–54, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany.

[Bar02] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *43rd Annual Symposium on Foundations of Computer Science*, pages 345–355, Vancouver, British Columbia, Canada, November 16–19, 2002. IEEE Computer Society Press.

[BC10] Mihir Bellare and David Cash. Pseudorandom functions and permutations provably secure against related-key attacks. In *Advances in Cryptology – CRYPTO 2010*, Lecture Notes in Computer Science, pages 666–684, Santa Barbara, CA, USA, August 2010. Springer, Berlin, Germany.

[BCFW09] Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi. Foundations of non-malleable hash and one-way functions. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 524–541, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany.

[BF06] Alexandra Boldyreva and Marc Fischlin. On the security of OAEP. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 210–225, Shanghai, China, December 3–7, 2006. Springer, Berlin, Germany.

[Bih94] Eli Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, 1994.

[BK03] Mihir Bellare and Tadayoshi Kohno. A theoretical treatment of related-key attacks: RKA-PRPs, RKA-PRFs, and applications. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 491–506, Warsaw, Poland, May 4–8, 2003. Springer, Berlin, Germany.

[BK04] Mihir Bellare and Tadayoshi Kohno. Hash function balance and its impact on birthday attacks. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 401–418, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In V. Ashby, editor, *ACM CCS 93: 1st Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, USA, November 3–5, 1993. ACM Press.

[BS99] Mihir Bellare and Amit Sahai. Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterization. In Michael J. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 519–536, Santa Barbara, CA, USA, August 15–19, 1999. Springer, Berlin, Germany.

[Can97] Ran Canetti. Towards realizing random oracles: Hash functions that hide all partial information. In Burton S. Kaliski Jr., editor, *Advances in Cryptology – CRYPTO'97*, volume 1294 of *Lecture Notes in Computer Science*, pages 455–469, Santa Barbara, CA, USA, August 17–21, 1997. Springer, Berlin, Germany.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008: 35th International Colloquium on Automata, Languages and Programming, Part II*, volume 5126 of *Lecture Notes in Computer Science*, pages 449–460, Reykjavik, Iceland, July 7–11, 2008. Springer, Berlin, Germany.

[CHS06] Ran Canetti, Shai Halevi, and Michael Steiner. Mitigating dictionary attacks on password-protected local storage. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 160–179, Santa Barbara, CA, USA, August 20–24, 2006. Springer, Berlin, Germany.

[CV09] Ran Canetti and Mayank Varia. Non-malleable obfuscation. In Omer Reingold, editor, *TCC 2009: 6th Theory of Cryptography Conference*, volume 5444 of *Lecture Notes in Computer Science*, pages 73–90. Springer, Berlin, Germany, March 15–17, 2009.

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

[DDO+01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 566–598, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.

[DG03] Ivan Damgård and Jens Groth. Non-interactive and reusable non-malleable commitment schemes. In *35th Annual ACM Symposium on Theory of Computing*, pages 426–437, San Diego, California, USA, June 9–11, 2003. ACM Press.

[DIO98] Giovanni Di Crescenzo, Yuval Ishai, and Rafail Ostrovsky. Non-interactive and non-malleable commitment. In *30th Annual ACM Symposium on Theory of Computing*, pages 141–150, Dallas, Texas, USA, May 23–26, 1998. ACM Press.

[FF00] Marc Fischlin and Roger Fischlin. Efficient non-malleable commitment schemes. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 413–431, Santa Barbara, CA, USA, August 20–24, 2000. Springer, Berlin, Germany.

[Fis08] Marc Fischlin. Security of NMAC and HMAC based on non-malleability. In Tal Malkin, editor, *Topics in Cryptology – CT-RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 138–154, San Francisco, CA, USA, April 7–11, 2008. Springer, Berlin, Germany.

[FLS+08] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The skein hash function family, 2008.

[Knu92] Lars R. Knudsen. Cryptanalysis of LOKI91. In Jennifer Seberry and Yuliang Zheng, editors, *Advances in Cryptology – AUSCRYPT'92*, volume 718 of *Lecture Notes in Computer Science*, pages 196–208, Gold Coast, Queensland, Australia, December 13–16, 1992. Springer, Berlin, Germany.

[LP09] Huijia Lin and Rafael Pass. Non-malleability amplification. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 189–198, Bethesda, Maryland, USA, May 31 – June 2, 2009. ACM Press.

[LPTV10] Huijia Lin, Rafael Pass, Wei-Lung Dustin Tseng, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable zero knowledge proofs. In *Advances in Cryptology – CRYPTO 2010*, Lecture Notes in Computer Science, pages 429–446, Santa Barbara, CA, USA, August 2010. Springer, Berlin, Germany.

[LPV08] Huijia Lin, Rafael Pass, and Muthuramakrishnan Venkitasubramaniam. Concurrent non-malleable commitments from any one-way function. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 571–588, San Francisco, CA, USA, March 19–21, 2008. Springer, Berlin, Germany.

[Luc04] Stefan Lucks. Ciphers secure against related-key attacks. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 359–370, New Delhi, India, February 5–7, 2004. Springer, Berlin, Germany.

[MMO85] Stephen Matyas, Carl Meyer, and Jonathan Oseas. Generating strong one-way functions with cryptographic algorithms. *IBM Technical Bulletin*, 1985.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109, Santa Barbara, CA, USA, August 17–21, 2003. Springer, Berlin, Germany.

[OPV08] Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Constant-round concurrent non-malleable zero knowledge in the bare public-key model. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008: 35th International Colloquium on Automata, Languages and Programming, Part II*, volume 5126 of *Lecture Notes in Computer Science*, pages 548–559, Reykjavik, Iceland, July 7–11, 2008. Springer, Berlin, Germany.

[OPV10] Rafail Ostrovsky, Omkant Pandey, and Ivan Visconti. Efficiency preserving transformations for concurrent non-malleable zero knowledge. In Daniele Micciancio, editor, *TCC 2010: 7th Theory of Cryptography Conference*, volume 5978 of *Lecture Notes in Computer Science*, pages 535–552, Zurich, Switzerland, February 9–11, 2010. Springer, Berlin, Germany.

[PGV94] Bart Preneel, René Govaerts, and Joos Vandewalle. Hash functions based on block ciphers: A synthetic approach. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 368–378, Santa Barbara, CA, USA, August 22–26, 1994. Springer, Berlin, Germany.

[PPV08] Omkant Pandey, Rafael Pass, and Vinod Vaikuntanathan. Adaptive one-way functions and applications. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 57–74, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany.

[PR05a] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *46th Annual Symposium on Foundations of Computer Science*, pages 563–572, Pittsburgh, PA, USA, October 23–25, 2005. IEEE Computer Society Press.

[PR05b] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In Harold N. Gabow and Ronald Fagin, editors, *37th Annual ACM Symposium on Theory of Computing*, pages 533–542, Baltimore, Maryland, USA, May 22–24, 2005. ACM Press.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553, New York, New York, USA, October 17–19, 1999. IEEE Computer Society Press.

[Sho01] Victor Shoup. OAEP reconsidered. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 239–259, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Berlin, Germany.

A Φ-Non-Malleability ⇏ Simulation-Based Non-Malleability

A.1 Proofs

Recall that we would like to show that non-malleability of $\mathcal{F}$ implies non-malleability of $\mathcal{G}$ which leaks one input bit:

Proof. Assume $\mathcal{F}$ is Φ-non-malleable, but $\mathcal{G}$ is not Φ-non-malleable, then some PPT algorithm $\mathcal{A} = (\mathcal{A}_d, \mathcal{A}_y)$ exists such that $\Pr[\mathbf{Exp}^{\Phi\mathsf{nm}}_{\mathcal{G},\mathcal{A}}(\lambda) = 1] = 1/\mathsf{poly}(\lambda)$. We construct $\mathcal{A}' = (\mathcal{A}'_d, \mathcal{A}'_y)$ by $\mathcal{A}'_d := \mathcal{A}_d$ and $\mathcal{A}'_y$ as follows. On input (y, h_x, st), it runs $\mathcal{A}_y(y||b, h_x, \mathrm{st})$ as a black-box, where $b \in \{0,1\}$ is chosen uniformly at random. The output of $\mathcal{A}'_y$ is defined to be the output of $\mathcal{A}_y$ with the difference that the rightmost bit of y^* is truncated.

Both stages of $\mathcal{A}'$ are clearly efficient since they run PPT algorithms $\mathcal{A}$ plus one coin toss. Independently of distribution $\mathcal{X}$, the probability that $b = x_1$ is exactly $\frac{1}{2}$. In this case, $y||b$ is in the range of G under $\mathcal{X}$, which is precisely what $\mathcal{A}_y$ expects. Furthermore, since $\mathsf{HVf}(K, x, y) = \mathsf{HVf}(K, x, y) \wedge (b = x_1) = \mathsf{GVf}(K, x, y||b)$ it follows that $\mathbf{Exp}^{\Phi\mathsf{nm}}_{\mathcal{F},\mathcal{A}'}(\lambda) = \mathbf{Exp}^{\Phi\mathsf{nm}}_{\mathcal{G},\mathcal{A}}(\lambda)$ in this case. Hence, we have

$$\Pr\left[\mathbf{Exp}^{\Phi\mathsf{nm}}_{\mathcal{F},\mathcal{A}'}(\lambda)\right] \geq \frac{1}{2} \cdot \Pr\left[\mathbf{Exp}^{\Phi\mathsf{nm}}_{\mathcal{G},\mathcal{A}}(\lambda) = 1\right] = \frac{1}{2 \cdot \mathsf{poly}(\lambda)}.$$

This contradicts our assumption that $\mathcal{F}$ is Φ-non-malleable, thus $\mathcal{A}$ cannot exist and $\mathcal{G}$ must be Φ-non-malleable. □

The proof of Proposition 3.7 now shows that $\mathcal{G}$ can be used to separate Φ-non-malleability from simulation-based non-malleability.

Proof. Fix some relation $R_1 \in \mathcal{R}$ as in the proposition and construct $\mathcal{A} = (\mathcal{A}_d, \mathcal{A}_y, \mathcal{A}_x)$ as follows. Algorithm $\mathcal{A}_d$ on input K outputs the uniform distribution $\mathcal{X}$ and the empty state. On input (y, h_x, st_d) algorithm $\mathcal{A}_y$ samples $x^* \leftarrow \mathcal{X}$ and manipulates the first bit such that $x_1^* = y_n$. It then eventually outputs $(\mathsf{G}(K, x^*), x^*) = (y^*, \mathrm{st}_y)$. Finally, algorithm $\mathcal{A}_x$ outputs $(\mathrm{st}_y, r) = (x^*, 0)$ for input (x, st_y).

Algorithm $\mathcal{A}$ is obviously efficient. We now analyze the probability that $\mathbf{Exp}^{nmh-1}_{\mathcal{G},\mathcal{A}}(\lambda)$ returns 1. It is easy to see that $R_1(\mathcal{X}, x, x^*, r)$ holds since we have $x_1 = y_n = x_1^*$ and $\mathcal{A}_d$ always outputs the uniform distribution; the value r can be ignored for this relation. The verification, $\mathsf{GVf}(K, x^*, y^*) = \mathsf{GVf}(K, x^*, \mathsf{G}(K, x^*)) = 1$, holds by correctness since $\mathcal{A}_y$ calculates y^* by evaluating the hash function. Lastly, the probability that $(x, y) = (x^*, y^*)$ is negligible in λ since $\lambda - 1$ bits of both x and x^* are selected independently and uniformly at random. Hence, it follows that $\Pr[\mathbf{Exp}^{nmh-1}_{\mathcal{G},\mathcal{A}}(\lambda) = 1] = 1 - \mathsf{negl}(\lambda)$.

Consider now any simulator $\mathcal{S}$ for attacker $\mathcal{A}$. Since the simulator does not have any information about x, its success to find a valid x^* (in terms of relation R_1) depends solely on the ability that $\mathcal{S}_x$ correctly guesses x_1. But since $\mathcal{X}$ is required to be the uniform distribution, this probability is exactly $\frac{1}{2}$ and thus $\Pr[\mathbf{Exp}^{nmh-0}_{\mathcal{G},\mathcal{S}}(\lambda) = 1] = \frac{1}{2}$. For the difference we then have

$$\left|\Pr\left[\mathbf{Exp}^{nmh-1}_{\mathcal{G},\mathcal{A}}(\lambda) = 1\right] - \Pr\left[\mathbf{Exp}^{nmh-0}_{\mathcal{G},\mathcal{S}}(\lambda) = 1\right]\right| = \left|\frac{1}{2} - \mathsf{negl}(\lambda)\right|$$

which is not negligible. Hence $\mathcal{G}$ is not non-malleable. □

A.2 Collision Resistance and One-Wayness

As mentioned in the introduction, collision resistance does not necessarily imply $\oplus$-non-malleability. More precisely, if $\oplus$-non-malleable hash functions exist, then there exists a function which is $\oplus$-non-malleable but not collision resistant. For example, one such function is $\mathsf{G}(x||b) := \mathsf{H}(x)||b$ for $b \in \{0, 1\}$. Intuitively, this function leaks one bit of the input which it does not use for the actual evaluation of H.

It is easy to verify that if H is a collision-resistant hash function, then G is still collision-resistant. Assume towards contradiction that G is not collision resistant and an attacker exists which is able to generate a collision with preimages $x_1||b \neq x_2||b$. Note that last bit is identical since it is part of the output. Hence we have $\mathsf{H}(x_1) = \mathsf{H}(x_2)$, but this is a collision under H. This contradicts our assumption that H is collision resistant and thus the attacker cannot exist.

On the other hand, G is trivially malleable. Given any image, an attacker may simply flip the last bit, knowing that this also flips the last bit of the according preimage. In the case of $\oplus$-non-malleability this translates to an attacker that outputs $(y^*, \delta) = (y \oplus 00\ldots01, 00\ldots01)$ and succeeds unconditionally in the experiment.

A similar argument applies to the one-way property of hash functions. Using the same idea as above, it is possible to show that one-way hash functions are not necessarily non-malleable.

B Results for Matyas-Meyer-Oseas

The section presents the formal analysis of the counterexample claim in Section 4 about MMO not necessarily providing $\oplus$-non-malleability. We first define hash-$\oplus$-composition and give the technical construction of h'.

Construction B.1 (Hash-$\oplus$-Composition) *If $\mathcal{F} = (\mathsf{FK}, \mathsf{F}, \mathsf{FVf})$ is a hash function, then the* hash-$\oplus$-composition *(with respect to $\mathcal{F}$) is constructed by $\mathcal{F}^{\oplus} := (\mathsf{FK}, \mathsf{F}^{\oplus}, \mathsf{FVf}^{\oplus})$ where $\mathsf{F}^{\oplus}(K, x) := \mathsf{F}(K, x) \oplus x$ and $\mathsf{FVf}^{\oplus}(K, x, y) := \mathsf{FVf}(K, x, y \oplus x)$.*

Construction B.2 *Let $\mathcal{F} = (\mathsf{FK}, \mathsf{F}, \mathsf{FVf})$ and $\mathcal{U} = (\mathsf{UK}, \mathsf{U}, \mathsf{UVf})$ denote hash functions where $\mathsf{F} : \{0,1\}^{2n} \to \{0,1\}^n$ and $\mathsf{U} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ for some $n \in \mathbb{N}$. Construct $\mathcal{G} = (\mathsf{GK}, \mathsf{G}, \mathsf{GVf})$ as follows.*

- GK *on input* 1^λ *outputs* $\mathsf{FK}(1^\lambda)||\mathsf{UK}(1^\lambda)$.
- G *parses input* (K, x) *as* $(K_\mathsf{F}||K_\mathsf{U}, x_0||x_1)$ *where* $|x_0| = |x_1|$ *and outputs* $\mathsf{U}(K_U, x_0) \oplus (\mathsf{F}(K_F, x_0)||\mathsf{F}(K_F, x_1))||x_0 \oplus x_1$.
- GVf *parses input* (K, x, y) *as* $(K_\mathsf{F}||K_\mathsf{U}, x_0||x_1, y_0||y_1||y_2)$ *where* $|x_0| = |x_1|$ *and* $2|y_0| = 2|y_1| = |y_3|$. *It outputs* 1 *if, and only if,*

$$\begin{aligned} & \mathsf{UVf}(K_\mathsf{U}, x_0, y_0||y_1 \oplus \mathsf{F}(K_\mathsf{F}, x_0)||\mathsf{F}(K_\mathsf{F}, x_1)) \\ \wedge\ & \mathsf{FVf}(K_\mathsf{F}, x_0, \mathsf{split}(0, \mathsf{U}(K_\mathsf{U}, x_0) \oplus y_0)) \\ \wedge\ & \mathsf{FVf}(K_\mathsf{F}, x_1, \mathsf{split}(1, \mathsf{U}(K_\mathsf{U}, x_0) \oplus y_1)) \\ \wedge\ & y_3 = (x_0 \oplus x_1). \end{aligned}$$

This construction is $\oplus$-non-malleable under certain assumptions, most notably if the distribution of preimages is uniform. We capture this by defining a class of *uniform attackers* that, when prompted to output a distribution, always output the uniform distribution. We furthermore assume that all algorithms of the $\oplus$-non-malleability experiment have access to the Random Oracle used in the proposition.

Proposition B.3 *For a uniform attacker the following holds. If* $\mathcal{F}$ *is a* $\oplus$*-non-malleable hash function with respect to* $\mathsf{hint} = \emptyset$ *and* $\mathcal{U}$ *is modeled as a Random Oracle, then Construction B.2 is* $\oplus$*-non-malleable with respect to* $\mathsf{hint} = \emptyset$.

Proof. Assume towards contradiction that an efficient uniform attacker $\mathcal{A}$ against $\mathcal{G}$ (as in Construction B.2) exists such that $\Pr[\mathbf{Exp}_{\mathcal{G},\mathcal{A}}(\lambda) = 1]$ is non-negligible in λ. Construct $\mathcal{A}' = (\mathcal{A}'_d, \mathcal{A}'_y)$ as follows.

Algorithm $\mathcal{A}'_d$ on input K runs PPT algorithm $\mathcal{A}_d(K||\mathsf{UK}(1^\lambda))$ and processes its output $(\mathcal{X}, \mathrm{st}_d)$ by discarding distribution $\mathcal{X}$. $\mathcal{A}'_d$ then outputs $(\mathcal{X}_u, \mathrm{st}_d)$, where $\mathcal{X}_u$ denotes the uniform distribution over the domain of F, i.e. over $\{0,1\}^{2n}$.

On the second stage of the attack, $\mathcal{A}'_y$ receives an image y of F and hint h_x. It samples two values x'_0, x'_1 from $\mathcal{X}_u$ and calculates y' in the following way. A decision bit $b \in \{0,1\}$ is picked uniformly at random. If $b = 0$, let $y' = \mathsf{U}(K_U, x'_0) \oplus (y||\mathsf{F}(K_F, x'_1))||x'_0 \oplus x'_1$, otherwise, if $b = 1$, let $\mathsf{U}(K_U, x'_0) \oplus (\mathsf{F}(K_F, x'_0)||y)||x'_0 \oplus x'_1$. $\mathcal{A}'_y$ then runs $\mathcal{A}_y(y', h_x, \mathrm{st}_d)$ to obtain (y^*, δ) and finally outputs $(\mathsf{split}(b, y^*), \mathsf{split}(b, \delta))$.

It is evident that both stages of $\mathcal{A}'$ run in probabilistic polynomial time since they run one stage of PPT algorithm $\mathcal{A}$ and execute efficient operations only. Since we assume that the preimages of G and F are both uniformly distributed (but different in size), $\mathcal{A}'_d$ may simply discard $\mathcal{X}$ over $\{0,1\}^{4n}$ and output the uniform distribution $\mathcal{X}_u$ over $\{0,1\}^{2n}$.

For $\mathcal{A}_y$, image y' is distributed exactly as it expects: The first half of y' is padded by oracle U and thus truly random unless given x'_0; the second half, $x'_0 \oplus x'_1$, is sampled from the uniform distribution. Since the first half of y' contains $\mathsf{F}(K_F, x_0) = y$ instead of $\mathsf{F}(K_F, x'_0)$, algorithm $\mathcal{A}_y$ may fail if it guesses the padding. But since U is a random oracle, this only happens if $\mathcal{A}_y$ queries the oracle for x'_0. The probability for this event is negligible since $\mathcal{A}_y$ learns nothing about x'_0 from $x'_0 \oplus x'_1$ given that x'_1 is picked uniformly at random.

Furthermore, $\mathcal{A}'_y$ may fail to output a valid difference $\delta' = \mathsf{split}(b, \delta)$ if all 1-bits are located in the split-off half. In this case we have $\delta' = 0^\lambda$ and $\mathcal{A}'_y$ cannot succeed because the experiment rejects the trivial empty difference. We provide a conservative worst case bound for the probability of this event. Assume that the hamming weight of δ is 1, i.e., the difference is given by one bit only. This reflects a worst case in the sense that it is the minimum number of 1-bits in the difference; adding more 1-bits can only sustain or increase the probability that these bits spread over both halves of δ. Let left denote the event that the difference is contained in the left half of δ (and $\neg\mathsf{left}$ in the right half). Algorithm $\mathcal{A}'_y$ receives a usable difference with probability $\Pr[\mathsf{left}|b = 0] + \Pr[\neg\mathsf{left}|b = 1]$ from $\mathcal{A}_y$. However, $\mathcal{A}_y$ does neither know b nor is it able to deduce

its value from image y': value b only affects the left half of y', but it is padded by oracle U. Thus we may view this probability unconditional and have $\Pr[\mathsf{left}]\Pr[b=0]+\Pr[\neg\mathsf{left}]\Pr[b=1]=\frac{1}{2}$. The overall lower bound on the probability for $\mathcal{A}'$ to succeed is then

$$\Pr\left[\mathbf{Exp}_{\mathcal{F},\mathcal{A}'}(\lambda)=1\right]\geq\frac{1}{2}\Pr\left[\mathbf{Exp}_{\mathcal{G},\mathcal{A}}(\lambda)=1\right],$$

which is not negligible in λ. But this contradicts the assumption that $\mathcal{F}$ is $\oplus$-non-malleable.

To complete our example, it remains to show that $\mathcal{G}$ becomes $\oplus$-malleable once used as the inner function of the hash-$\oplus$-composition. We omit a formal proof here, since $\mathcal{G}$ is constructed in a way that makes it rather trivial.

Proposition B.4 *Let $\mathcal{G}$, as defined in Construction B.2, denote an $\oplus$-non-malleable function. If $\mathcal{G}$ is used within the hash-$\oplus$-composition, then the resulting function $\mathcal{H}$ is not $\oplus$-non-malleable.*

To see why this proposition holds, consider that the images of $\mathcal{H}$ are of the following form:

$$\begin{aligned} y &= \mathsf{H}(K_H,x)\\ &= \mathsf{G}(K_G,x)\oplus x\\ &= x_0\oplus\mathsf{U}(K_U,x_0)\oplus(\mathsf{F}(K_F,x_0)||\mathsf{F}(K_F,x_1))||x_0\oplus x_1\oplus x_1 \end{aligned}$$

This design is clearly insecure, because x_1 on the right-hand side is canceled out and thus x_0 is disclosed. While this disclosure is not a problem with regard to collision resistance or one-wayness in a theoretical sense (x_1 is still unknown and "secured" by $\mathcal{F}$), it is fatal for $\oplus$-non-malleability. Since the second stage $\mathcal{A}_y$ of an attacker against $\oplus$-non-malleability learns x_0 from y in this scheme, it may also cancel out x_0 in the left half and evaluate U. Then, $\mathcal{A}_y$ is able to calculate y' by discarding $\mathsf{F}(K_F,x_0)$ and replacing it by $\mathsf{F}(K_F,x_0\oplus\delta)$ for some arbitrary $\delta\neq 0^\lambda$. Adding U and $x_0\oplus\delta$ again yields

$$y'=x_0\oplus\delta\oplus\mathsf{U}(K_U,x_0\oplus\delta)\oplus(\mathsf{F}(K_F,x_0\oplus\delta)||\mathsf{F}(K_F,x_1))||x_0\oplus\delta.$$

An attacker then simply outputs $\mathcal{A}_y$ outputs $(y',\delta||0^{2n})$ in the $\oplus$-non-malleability game and succeeds unconditionally. The function $\mathcal{H}$ is therefore not $\oplus$-non-malleable even though it is built from a $\oplus$-non-malleable function.

C Proof of Theorem 5.2 (BR Encryption)

Proof. Fix an arbitrary adversary $\mathcal{A}$ against IND-CCA of the scheme. We consider a sequence of games $\textsc{Game}_0, \textsc{Game}_1, \ldots$, starting with the original attack in $\textsc{Game}_0$, and slightly changing the adversary's environment in each game hop such that the adversary's output behavior does not change significantly. We eventually reach a game where the adversary cannot perform better than by guessing the secret bit b, concluding that the adversary cannot have a significant advantage in the original attack.

In all games we denote by (y^*,g^*,h^*) the components of the challenge ciphertext returned to $\mathcal{A}$ on submitting two equal-length messages m_0,m_1 (even if the ciphertext is not computed according to the scheme's description in some games). The games are then given as follows:

$\textsc{Game}_0$: Corresponds to the original attack of the adversary $\mathcal{A}$.

$\textsc{Game}_1$: Reject all ciphertexts (y,g,h) submitted to the decryption oracle, where $y\neq y^*$ and $\mathcal{A}$ has never queried G about $r=\mathsf{f}^{-1}(y)$ before (including the case that y is not in the range of f).

Game$_2$: Reject all ciphertexts (y, g, h) submitted to the decryption oracle before receiving the challenge ciphertext, where $y = y^*$.

Game$_3$: Reject all ciphertexts (y, g, h) submitted to the decryption oracle, where $y = y^*$, $g = g^*$ and $h \neq h^*$ (queries after receiving the challenge ciphertext, where also $h = h^*$, are not allowed).

Game$_4$: Reject all ciphertexts (y, g, h) submitted to the decryption oracle, where $y = y^*$ but $g \neq g^*$, and $\mathcal{A}$ has never queried G about $r = \mathsf{f}^{-1}(y)$ before.

Note: At this point we can simulate decryption queries by table-lookups in $\mathcal{A}$'s queries to oracle G.

Game$_5$: Replace the computation of $h^* \leftarrow \mathsf{H}(K, r||m_b)$ in the challenge ciphertext by $h^* \leftarrow \mathsf{H}(K, r'||0^{|m_b|})$.

Game$_6$: Replace the computation of $g^* \leftarrow G(r) \oplus m_b$ in the challenge ciphertext by picking g^* uniformly at random.

Next we show that each transition does not change the adversary's output behavior significantly.

From Game$_0$ to Game$_1$. The only difference in the games can origin from a valid ciphertext which the decryption oracle would decrypt correctly in the attack (and where in particular $r = \mathsf{f}^{-1}(y)$ exists), which we now reject in GAME$_1$.

Note that this cannot decrease the adversary's advantage significantly, because the probability that for such queries h is a valid hash value is negligible. This can be seen as follows. If the probability was not negligible, we could devise an algorithm against $\oplus$-non-malleability. This algorithms initially receives a key K of the hash function, including some f' which it ignores, and instead picks a new f. With the help of the self generated f^{-1} it runs a simulation of the adversary up to point where the adversary submits the first of such decryption queries. Then our algorithm recovers r and outputs the distribution $\mathcal{X}$ returning $r||m$ for random $m \in \{0,1\}^\ell$ with super-logarithmic min-entropy. Ignore $\mathsf{hint}_{\mathrm{BR}} = \mathsf{f}'(r)$ and instead output the hash value h in the decryption query together with $\delta = 0^{|r|}||z$ for random $z \leftarrow \{0,1\}^\ell$. Note that, since the adversary has not queried G about r so far, it follows that $m = G(r) \oplus g$ is random, and so is the m-part in $m \oplus \delta$. It follows that the non-malleability adversary here wins with non-negligible probability, by assumption about the encryption adversary making such a valid decryption query with non-negligible probability.

From Game$_1$ to Game$_2$. The next transition is easily shown to essentially preserve the adversary's behavior. The probability that any of the at most polynomial decryption queries before the challenge ciphertext already contains y^* is negligible. Else inverting y^* under f for random input would succeed with non-negligible probability.

From Game$_2$ to Game$_3$. This change is rather syntactical. Since we assume the hash function verification to recompute the has value and to compare it to a given value, any ciphertext with $h \neq h^*$ for the same $m = m^*, r = r^*$ (because $y = y^*, g = g^*$) must be invalid.

From Game$_3$ to Game$_4$. Suppose now that the adversary in GAME$_3$ submits a valid ciphertext with $y = y^*$ but $g \neq g^*$ without having queried G about r before. Let $m = G(r) \oplus g$. In particular, it must then hold that $\mathsf{HVf}(K, r||m, h) = 1$. We construct a successful adversary $\mathcal{B}$ against $\oplus$-non-malleability as follows. In the first stage $\mathcal{B}_d$ receives a key K (and a description f) as input. It runs adversary $\mathcal{A}$ in GAME$_3$ up to the point where the adversary outputs m_0, m_1. Algorithm $\mathcal{B}_d$ uses lazy sampling to simulate G and answers all decryption queries by looking up the right values r in the list of G-queries; if for some query algorithm $\mathcal{B}_d$ does not find

an appropriate entry then it rejects the ciphertext (as $\textsc{Game}_3$ would; also note that, if there is an entry, then it must be unique because y uniquely determines it).

Algorithm $\mathcal{B}_d$ then picks $b \leftarrow \{0,1\}$ and outputs a distribution $\mathcal{X}$ which picks r^* at random and returns $r^*||m_b$. Algorithm $\mathcal{B}_y$ then receives the hash value h^* and the auxiliary information $y^* = \mathsf{f}(r^*)$, and continues the simulation by implicitly defining $G(r^*)$ as a random value (note that we can easily keep up a consistent simulation by checking for G-queries r if they match $y^* = \mathsf{f}(r)$ and return this random value). Algorithm $\mathcal{B}_y$ sets $g^* = G(r^*) \oplus m_b$ for the challenge ciphertext (y^*, g^*, h^*) returned to $\mathcal{A}$. It also tries to predict the first query of $\mathcal{A}$ to the (simulated) decryption oracle for a valid ciphertext for which $y = y^*$ but $g \neq g^*$. This can be done by picking an index between 1 and the number of queries at random. When $\mathcal{A}$ later submits a ciphertext (y, g, h) with $y = y^*$ and $g \neq g^*$ to the (simulated) decryption oracle without having queries G about r before, then $\mathcal{B}_y$ rejects if it does not match the predicted query. Else it computes $\delta = g \oplus g^* \neq 0^\ell$ and outputs $(h, 0^k||\delta)$.

The simulation of $\mathcal{A}$'s attack is perfect up to the point when $\mathcal{A}$ submits its first decryption query for a valid ciphertext $y = y^*$ and $g \neq g^*$ without having queries G about $r = r^*$ before (or if $\mathcal{B}_y$ stops before). In this case $\mathcal{B}_y$ has correctly guesses this query within a polynomial factor, and now succeeds in outputting a related image: Since $y = y^*$ it must hold $r = r^*$ and $\delta = g \oplus g^* = G(r) \oplus m \oplus G(r^*) \oplus m_b = m \oplus m_b$. For a valid ciphertext it must also hold that $\mathsf{HVf}(K, r||m, h) = 1$ and thus that h is a valid hash value of $r||m_b||0^k||\delta$. In other words, if the probability that $\mathcal{A}$'s output behavior changes noticeably when hopping from $\textsc{Game}_3$ to $\textsc{Game}_4$, then we would obtain a successful attacker against the $\oplus$-non-malleability (losing a polynomial factor in the difference for the guessing).

From $\textsc{Game}_4$ to $\textsc{Game}_5$. As mentioned before we can now replace the decryption oracle by a procedure which simply checks for entries in the G-list and answers accordingly. Hence, we can now show that we can replace the hash value h of $r^*||m_b$ in the challenge ciphertext by a hash value of independent values $r'||0^\ell$. That is, we construct an attacker against perfect one-wayness similar to the previous attacker $\mathcal{B}$ against non-malleability (but simulating the decryption oracle completely with table-lookups). Actually, this step somewhat combines two game hops where, in the first hop, we have the adversary against perfect one-wayness output distributions $\mathcal{X}$ returning $r||m_b$ for the challenge message m_b, and in the second step moving from there to distributions of the form $r||0^\ell$. It follows analogously to the previous case that any noticeable change in the adversary's output behavior would yield a successful attacker (for each hop) against the perfect one-wayness.

From $\textsc{Game}_5$ to $\textsc{Game}_6$. In the final game we replace g^* in the challenge ciphertext by an independent random value, making $\mathcal{A}$'s view independent of bit b. We note that this does not change $\mathcal{A}$'s output distribution unless $\mathcal{A}$ at some point queries G about r^*. However, this would straightforwardly contradict the one-wayness of f because one could easily simulate the attack and inject y^* into the challenge ciphertext (and define $G(r^*)$ implicitly at random), simulate the decryption oracle via table-lookups, and wait for $\mathcal{A}$ to make the g-query in order to invert y^* with non-negligible probability.

Conclusion. We now have that the adversary wins in $\textsc{Game}_6$ with probability at most $1/2$. Since each transition from $\textsc{Game}_0$ to $\textsc{Game}_6$ decreases the adversary's success only negligibly, it follows that the success probability in the original attack cannot exceed $1/2$ significantly. □