Cryptophia's Short Combiner for Collision-Resistant Hash Functions

Arno Mittelbach

Darmstadt University of Technology, Germany
www.cryptoplexity.de,
arno.mittelbach@cased.de

Abstract. A combiner for collision-resistant hash functions takes two functions as input and implements a hash function with the guarantee that it is collision-resistant if one of the functions is. It has been shown that such a combiner cannot have short output (Pietrzak, Crypto 2008); that is, its output length is lower bounded by roughly $2n$ if the ingoing functions output n-bit hash values. In this paper, we present two novel definitions for hash function combiners that allow to bypass the lower bound: the first is an extended semi-black-box definition. The second is a new game-based, fully black-box definition which allows to better analyze combiners in idealized settings such as the random-oracle model or indifferentiability framework (Maurer, Renner, and Holenstein, TCC 2004). We then present a new combiner which is robust for pseudorandom functions (in the traditional sense), which does not increase the output length of its underlying functions and which is collision-resistant in the indifferentiability setting. Our combiner is particularly relevant in practical scenarios, where security proofs are often given in idealized models, and our combiner, in the same idealized model, yields strong security guarantees while remaining *short*.

Keywords: hash functions, combiners, collision resistance, multi-property combiner.

1 Introduction

A Story. Once upon a time little Cryptess was walking through her favorite forest. As usual she was thinking about a hard problem and thus did not pay much attention on where she was going. It thus came that she suddenly found herself on a beautiful glade that she had never seen before. In its center she could make out what seemed to be a fairy flapping her wings in a welcoming pattern. Little Cryptess slowly approached the fairy and politely asked "Hello little one, who are you?" The fairy responded "I am the fairy Cryptophia and since you have found my magical glade, I grant you one wish." Little Cryptess did not take long to come up with a wish: "Can you build me a hash-function combiner that while being robust for collision resistance does not increase the output length of the hash functions?" "Of course I can", said the fairy. "Here it is. But beware,

M. Jacobson et al. (Eds.): ACNS 2013, LNCS 7954, pp. 136–153, 2013.

it is a magical combiner. Given access to two hash functions H_1 and H_2 and a message M it returns $H_1(M)$ if and only if H_1 is 'more' collision-resistant than H_2. Else it returns $H_2(M)$". Cryptess thought for a moment and then replied "I am sorry Cryptohia, but your combiner is utterly useless. It is not robust for collision resistance after all. Assume I give it access to two uniformly random functions $\mathcal{R}_1$ and $\mathcal{R}_2$ and I am given an oracle that computes collisions for the combiner. As the oracle will only provide collisions for $\mathcal{R}_1$ no efficient reduction can compute collisions for $\mathcal{R}_2$. This, as you should know, violates the definition of robustness and thus your combiner is useless to me." With this she turned around and went home.

Hash-Function Combiners. Hash functions are an important cryptographic primitive but, as with many primitives, efficient constructions used in practice are based on heuristics [40,34,8]. As history has shown, with time, it is not unlikely that cryptanalysists find plausible attacks [45,43,42,44,19,3,14] and it is thus a natural question to ask whether we can hedge against the failure of an implemented hash function.

A hash-function combiner is a construction which, given access to two or more hash functions, itself implements a hash function that, however, comes with certain guarantees. A combiner is called *robust* for some property π if it guarantees to satisfy property π provided that sufficiently many input functions do. The simplest version (and the one usually used in practice) is a combiner which takes two hash functions as input and hedges against the failure of one of them, i.e., it obeys π if either of the input functions does. This will also be the variant that we examine more closely in this paper. A practical example of the application of hash-function combiners are the original versions of the TLS and SSL protocols [24,20].

Assume C^{H_1,H_2} is a hash-function combiner given access to two hash functions H_1 and H_2, then robustness for property π is usually defined via a reductionist approach. That is, the combiner is called robust for π if there exists a reduction $\mathcal{P}$ such that if $\mathcal{P}$ is given access to any (breaking-)oracle $\mathcal{B}$ that breaks π on the combiner with non-negligible probability, then $\mathcal{P}^{\mathcal{B},H_1,H_2}$ must in turn break π on both input hash functions (H_1 **and** H_2) with non-negligible probability.

There are two folklore combiners for hash functions. The *concatenation combiner* $C_{\|}^{H_1,H_2}(M) := H_1(M)||H_2(M)$ is, amongst others, robust for collision resistance (it should be difficult to find two distinct messages that hash to the same value). It is easy to see that a collision on the combiner directly yields collisions for both input functions. In other words, for a message pair (M, M') with $M \neq M'$ it holds that $C^{H_1,H_2}(M) = C^{H_1,H_2}(M')$ if and only if $H_1(M) = H_1(M')$ and $H_2(M) = H_2(M')$. The concatenation combiner is, however, not robust for pseudorandomness (no efficient distinguisher that is only given black-box access should be able to distinguish between the hash function and a randomly chosen function with the same domain and codomain). On the other hand, the *exclusive-or combiner* $C_{\oplus}^{H_1,H_2}(M) := H_1(M) \oplus H_2(M)$ which computes the bitwise exclusive-or on the outputs of the two hash functions is robust for pseudorandomness if instantiated with two independent hash functions.

However, it is not robust for collision resistance, nor even collision-resistant preserving. Hash-function combiners that are robust for multiple properties, in particular for collision resistance and pseudorandomness together, have been studied by Fischlin et al. [21,22].

Short Combiners. If we assume that H_1 and H_2 take on values in $\{0,1\}^n$ then the concatenation combiner doubles the output length, whereas the exclusive-or combiner does not. Furthermore, it is a common property that all combiners robust for collision resistance share: their output length is in the order of the sum of the output lengths of the input hash functions.

This observation lead to the question whether *short* hash-function combiners (combiners with an output length significantly shorter than that of the concatenation combiner) that are robust for collision resistance exist [12]. It has been shown that this is not the case, i.e., there exists a lower bound on the output length for combiners that are robust for collision resistance as well as for related properties [12,13,36,37,30] where the lower bound is roughly the output length achieved by the concatenation combiner.[1]

Cryptophia's Magical Combiner. Cryptess rejected Cryptophia's magical combiner on the grounds that it is not *robust* for collision resistance. Indeed, she was right, as the combiner only evaluates one of the two functions a collision on the combiner cannot possibly yield information about collisions for the other function. On the other hand, the robustness definition is usually only given for black-box combiners, i.e., combiners that only get black-box access to the hash functions; Cryptophia's magical combiner is, however, clearly not black-box. Nevertheless, what this shows is that the robustness definition requires the combiner, in some sense, to be stronger than both input hash functions which in turn leads to the lower bound on the output-length of combiners for collision resistance. This, however, goes against the intuition of what a combiner should capture: it should be at least as strong as the stronger of the two functions, but not necessarily stronger.

Contributions and Outline. In this paper we examine the current definition of robust combiners and the reason why it is necessary for combiners that are robust for collision resistance to satisfy a lower bound on their output-length (Section 3). In Section 3.2, we extend the definition (in a semi black-box way) in order to better capture the intuition: a combiner does only need to be as strong as the strongest input function and not necessarily stronger. We then present a new game-based definition for combiners (Section 3.3) which also allows to bypass the lower bounds while still being fully black-box. This second notion

[1] A recent framework by Baecher et al. [4] allows to precisely characterize reductions (and thus separation results) in terms of the level of black-boxness used by the construction and the reduction. In terms of the impossibility result for short combiners it can be shown that the ruled-out reductions are of the type `NNN` meaning that it holds even if the construction or the accompanying security reduction were using non-black-box techniques. See the full-version of this work [31] for further details.

is tailored to analyze combiners in idealized models such as the random oracle model (ROM; [7]) or the indifferentiability framework introduced by Maurer, Renner and Holenstein [28,17] giving guarantees of the form: *the combiner has property π if one of the input functions is* ideal *even if the other function is completely under the control of the adversary and possibly even based on the first function.* We go on to present a new construction for a combiner which we analyze in this new model (Section 4). The combiner does not increase the output length of its ingoing functions while guaranteeing collision resistance (and related properties) provided that one of the two input functions is indifferentiable from a random oracle (assuming ideal compression functions). Finally, we show that our combiner is robust for pseudorandomness under the "traditional" definition of robustness without needing to assume independence (as is the case for the "standard" xor-combiner). This yields the first multi-property combiner with short output length, which is robust for pseudorandomness and which gives additional guarantees about collision resistance and related properties such as pre-image resistance or target collision resistance.

2 Preliminaries

Lower-case letters, such as $n \in \mathbb{N}$, usually represent natural numbers and by 1^n we denote the unary representation of n. Upper-case letters in standard typeface, like M, stand for bit-strings which we usually call messages. By $\{0,1\}^n$ we denote the set of all bit-strings M of length $|M| = n$, while $\{0,1\}^*$ denotes the set of all bit-strings. For bit-strings $X, Y \in \{0,1\}^*$ we denote with $X||Y$ their concatenation and with $X \oplus Y$ the bit-wise exclusive-or (XOR) operation. If $\mathcal{X}$ is a set then by $M \leftarrow \mathcal{X}$ we mean that M is chosen uniformly from $\mathcal{X}$. If $\mathcal{X}$ is a distribution then $M \leftarrow \mathcal{X}$ denotes that M is chosen according to the distribution.

If $\mathcal{A}$ is an algorithm (often also called adversary) that has black-box access to one or more oracles $\mathcal{O}_1, ..., \mathcal{O}_z$ we denote this by adding them in superscript, i.e., $\mathcal{A}^{\mathcal{O}_1,...,\mathcal{O}_z}$. By $X \leftarrow \mathcal{A}(M)$ we denote that algorithm $\mathcal{A}$ on input M outputs value X. Throughout this paper we assume 1^n to be a security parameter and we call an algorithm efficient if it runs in polynomial time in the security parameter.

If X is a random variable, $\Pr[X = x]$ denotes the probability that X takes on value x. By $\mathrm{H}_\infty(X)$ we denote the min-entropy of variable X, defined as

$$\mathrm{H}_\infty(X) := \min_{x \in \mathrm{Supp}(X)} \log(1/\Pr[X = x])$$

where the probability is over X. The (average) conditional min-entropy of random variable X conditioned on variable Z is defined (in the style of [1]) as

$$\tilde{\mathrm{H}}_\infty(X|Z) := \min_{\mathcal{A}} \log(1/\Pr[X = \mathcal{A}(Z)])$$

where the probability is over X and Z and the random coins of $\mathcal{A}$ (which has no efficiency bounds).

2.1 Hash Functions and their Properties

Formally, a hash function $\mathcal{H}$ is defined as a family of functions together with a key generation algorithm **HKGen** that picks one of the functions to be used. That is, a *hash function (family)* is a pair of efficient algorithms $\mathcal{H} = (\mathbf{HKGen}, H)$ where $\mathbf{HKGen}(1^n)$ is a probabilistic algorithm that takes as input the security parameter 1^n and outputs a key k, while deterministic algorithm $H_k(M) := H(k, M)$ takes a key k and message $M \in \{0,1\}^*$ as input and outputs a hash value $H_k(M) \in \{0,1\}^n$. Note that we will drop the subscript and simply write $H(M)$ whenever the key is clear from context.

Collision Resistance and Related Properties. A hash function $\mathcal{H}$ is called *collision-resistant* (cr) if no efficient adversary can find two distinct messages (M, M') such that $H_k(M) = H_k(M')$. More formally, a hash function is called collision-resistant, if for any efficient adversary $\mathcal{A}$ there exists a negligible function $\mathtt{negl}$ such that:

$$\mathbf{Adv}^{\mathrm{cr}}_{\mathcal{H}}(\mathcal{A}) := \Pr\left[\begin{array}{c|c} k \leftarrow \mathbf{HKGen}(1^n); & M \neq M' \quad \wedge \\ (M, M') \leftarrow \mathcal{A}(k) & H_k(M) = H_k(M') \end{array}\right] \leq \mathtt{negl}(n)$$

where the probability is over the choice of key and $\mathcal{A}$'s internal coin tosses.

Two closely related properties are *second pre-image resistance* (spr) and *target collision resistance* (tcr) (see [41] for an overview of several variants of these notions). Here the adversary's task is not to find an arbitrary collision but a specific one, in case of second pre-image resistance the target message is sampled according to a distribution $\mathcal{M}$ whereas for target collision resistance the target message is specified by a first-round adversary.[2]

Finally, we consider another variant of second pre-image resistance called *pre-image resistance* (also often referred to as *one-wayness*). In the pre-image resistance experiment a message M is again chosen according to some distribution $\mathcal{M}$. Given only the resulting hash value $H_k(M)$ (and not message M) and key k, the adversary's task is to find a corresponding pre-image M', i.e., a message M' such that $H_k(M) = H_k(M')$.

Pseudorandomness and Message Authentication Codes. Besides collision resistance and its variants, hash functions are often assumed to be *pseudorandom* (or a pseudorandom function; prf) or *secure message authentication codes*. Here the adversary is not given access to the hash function's key but only to a black-box implementing the hash function, i.e., the key is kept private at all times. A hash function $\mathcal{H}$ is called *pseudorandom* if no efficient adversary can tell whether it is given black-box access to the hash function $\mathcal{H}$ or to a random

[2] Note that target collision resistant hash functions are also known as universal one-way hash functions [33].

function f with the same domain and range. More formally, for any efficient adversary $\mathcal{A}$ there exists a negligible function $\mathtt{negl}$ such that:

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathcal{A}}(\mathcal{A}) := \left|\Pr_k\left[\mathcal{A}^{H_k}(1^n)=1\right] - \Pr_f\left[\mathcal{A}^{f}(1^n)=1\right]\right| \leq \mathtt{negl}(n)$$

The probability is over the adversary's random coins and the choice of key in the first part and the choice of function in the second, respectively.

A hash function is called a *secure message authentication code* (mac) if no adversary given only black-box access to a hash oracle can find a message and corresponding hash value (without querying the oracle on the corresponding message) with noticeable probability. The probability is over the choice of key k and the adversary's internal coin tosses.

Random Oracles and Indifferentiability. Many security proofs are given in the random oracle model (ROM; [7]) where hash functions are modeled as ideal, i.e., as truly random functions (e.g., [16,5,6,11]). While random oracles have no structure at all hash functions, on the other hand, are usually built from a fixed-length compression function and some iteration scheme defining how arbitrarily long messages are hashed [29,18,40,27,8].

The *indifferentiability* notion introduced by Maurer, Renner and Holenstein in [28] can be seen as a generalization of indistinguishability that allows to better analyze constructions—such as hash functions—where internal state is publicly available. Coron et al. [17] applied the notion to hash functions and proved several hash constructions to be indifferentiable from a random oracle. The composition theorem for indifferentiability allows to reduce the security of a scheme in the random oracle model to the security of the compression function, in case the random oracle is implemented by a hash construction that is indifferentiable from a random oracle. As a compression function is a much more graspable object than a random oracle, indifferentiability has become an accepted design criterion for hash functions; indeed, many candidates to the SHA-3 competition [35], including the winner Keccak [8] enjoy proofs of indifferentiability [15,2,32,9,10].

3 A Novel Definition of Combiners for Hash Functions

3.1 Black-Box Combiners for Hash Functions

Combiners for hash functions are traditionally defined in the following fashion (see, for example, [12,37] for a version of this definition for collision resistance): a hash-function combiner robust for property π (e.g., collision resistance) is a construction that given black-box access to two hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$ implements a hash function which obeys property π as long as $\mathcal{H}_1$ or $\mathcal{H}_2$ obeys property π. Formally, a hash-function combiner $\mathcal{C} := (\mathbf{CKGen}, C, \mathcal{P})$, robust for property π, is a triple of efficient algorithms, where $\mathbf{CKGen}(1^n, \mathbf{HKGen}_1, \mathbf{HKGen}_2)$ generates keys for hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$ and possibly some additional key k_C for the combiner. Algorithm C is an efficient deterministic algorithm that on input keys k_{H_1}, k_{H_2}, k_C and $M \in \{0,1\}^*$ returns a hash value $C_{k_{H_1},k_{H_2},k_C}(M)$

in target domain $\{0,1\}^n$. We will usually simply write $C^{H_1,H_2}(M)$. Algorithm $\mathcal{P}$ is a security reduction, i.e., $\mathcal{P}$ is a probabilistic polynomial-time oracle Turing machine that given access to a (breaking-)oracle $\mathcal{B}$ that breaks property π on the combiner (for example, samples collisions) breaks property π on both hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$. Note that $\mathcal{B}$ may be inefficient.

The classical combiner for collision resistance (and related properties) is the *concatenation combiner* defined as

$$C_{||}^{H_1,H_2}(M) := H_1(M)||H_2(M) \ .$$

Obviously, any collision on the combiner $C_{||}$ directly yields collisions for hash functions H_1 and H_2. The same applies for second pre-image resistance, target collision resistance and pre-image resistance. This combiner is, however, trivially not robust for pseudorandomness. The traditional combiner for pseudorandomness is the *exclusive-or combiner*

$$C_{\oplus}^{H_1,H_2}(M) := H_1(M) \oplus H_2(M)$$

although one has to make the additional assumption that the two functions are independent. Under this assumption the combiner is robust for pseudorandomness, message authentication codes and indifferentiability [23,26]. Without this additional assumption it is, however, not even pseudorandomness preserving. Take two (keyed) random oracles $H_1, H_2 : \{0,1\}^* \to \{0,1\}^n$ where H_2 is defined as $H_2 := H_1 \oplus 1^n$. Individually, these two functions are information-theoretically indistinguishable from random functions. The XOR-combiner would, however, implement the constant 1^n-function. The exclusive-or combiner is also not robust for collision resistance, even assuming independent functions, as a collision on the combiner does not require collisions under both input functions.

Short Combiners for Collision Resistance. A crucial difference between the two classical combiners (apart from being robust for different properties) is that the concatenation combiner doubles the output length, i.e., if the two input hash functions have range $\{0,1\}^n$, then the concatenation combiner outputs hash values in $\{0,1\}^{2n}$ while the exclusive-or combiner only outputs bit-strings of length n. A natural question to ask is: can we do better? That is, *does a secure combiner for collision resistance, which has a significantly shorter output length than the concatenation combiner, exist?* This question was first posed by Boneh and Boyen in [12] and has since been answered negatively [12,13,36,37]: combiners, robust for collision resistance, with significantly shorter output length than the concatenation combiner do not exist. Recently, a similar result was proved for second pre-image resistance, target collision resistance and pre-image resistance [30].

Let us quickly sketch the proof idea for collision-resistance. Assume we have a combiner for two hash functions with range $\{0,1\}^n$. If the combiner compresses its output to below $2n$ bits, then by the pigeonhole principle, there must exist collisions that result from compression rather than from collisions on the original hash functions. This allows to show the existence of an adversary which only

samples such collisions that result from compression (note that the breaking oracle does *not* need to be efficient and can, thus, search for such a collision). Naturally, these collisions do not help any security reduction $\mathcal{P}$ in finding collisions on the input hash functions. For example, assume the input hash functions are random oracles: then, a collision on the combiner which solely results from compression does not provide any help in finding a collision for one of the random oracles. This allows to show that no security reduction can exist if the combiner compresses. Hence, combiners with short output-length do not exist.

3.2 Extending the Traditional Definition

In the introduction we saw that Cryptophia's magical combiner is not robust for collision resistance under the traditional definition of robustness. In the following we extend the traditional definition of combiners for collision-resistant hash functions such that it also captures the "magical" combiner. To this end, we need to relax the requirements on the security reduction $\mathcal{P}$ while ensuring that, in doing so, we won't label any insecure combiners "secure". The idea is to call a combiner robust for some property π if the advantage of any efficient adversary against the combiner is upper-bounded by the maximal advantage of *any* efficient adversary against any of the two input hash functions. That is, the combiner needs to be at least as *strong* as the better of the two functions, but not necessarily stronger.

To formalize the idea, we need a notion of the maximum advantage of any adversary against some property π.

Definition 1. *Let $t \in \mathbb{N}$ be a natural number and n be a security parameter. The maximum t-advantage $\boldsymbol{AdvMax}^t_\pi$ against property π on hash function $\mathcal{H}$ is defined as the maximum advantage of any adversary running in time t against property π on hash function $\mathcal{H}$:*

$$\boldsymbol{AdvMax}^t_\pi(\mathcal{H}, 1^n) := \max_{\mathcal{A}} \boldsymbol{Adv}^\pi_{\mathcal{A}}(\mathcal{H}, 1^n) \quad \textit{s.t. } \mathcal{A} \textit{ runs in time } t$$

We now present an extension to the current black-box definition of robust combiners for hash functions. We extend the original definition such that all robust combiners remain robust under the new definition but we relax the requirements on the security reduction such that the combiner does not need to be stronger than any of the input functions.

Definition 2 (extension). *Let n be a security parameter. Let $\mathcal{C} := (\boldsymbol{CKGen}, C)$ be a combiner for hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$ as defined earlier. Let π be a property on hash functions. We say $\mathcal{C}$ is a robust combiner for property π if $\mathcal{C}$ is robust under the original definition, or if for all $t \in \mathbb{N}$:*

$$\boldsymbol{AdvMax}^t_\pi(\mathcal{C}, 1^n) \leq \min\left(\boldsymbol{AdvMax}^t_\pi(\mathcal{H}_1, 1^n), \boldsymbol{AdvMax}^t_\pi(\mathcal{H}_2, 1^n)\right)$$

Note that any combiner that is robust for some property π under the traditional definition is also robust under our new definition. The introduced loophole, however, allows a combiner to be robust even if no security reduction $\mathcal{P}$ exists. In this

case, the combiner must guarantee that the advantage for any adversary running in time t against property π on either $\mathcal{H}_1$ or $\mathcal{H}_2$ denotes an upper-bound on the advantage of any adversary running in time t against the combiner.

Discussion. The extended definition captures the security of the "magical" (non black-box) combiner. However, being a semi-black-box notion, it seems difficult to design an actual (non-magical) combiner exploiting the loophole offered by this notion. In the following section we build upon the ideas developed so far and present a fully black-box model which also allows to circumvent the lower bound on the output length. For this, we strengthen the assumption on the "input functions" requesting that one of the functions is ideal. Knowing that one of the functions is ideal then allows us to model that the combiner should be as strong as the ideal function, while it can "ignore" the second function.

3.3 Secure Combiners in Idealized Models

In this section we use a different and more practical approach to bypass the lower bound. We present a novel game-based security notion for black-box combiners that is tailored to be used in the idealized random oracle setting. Being black-box makes it easy to design combiners for this new notion and assuming, to a certain extend, idealized functions allows us to bypass the lower bound. In short, a combiner proven secure in our new notion provides the guarantee that it has a certain property as long as one of the two functions is ideal even in case the other function is highly dependent upon the first; this is modeled by giving the adversary full control over the second function.

We say that a combiner C is *ideally secure* for some property π if no adversary can win the *ideally secure combiner game* (see Figure 1). For this we consider a two-stage adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, where $\mathcal{A}_1$ outputs some state st and a description of an efficient function that can contain special oracle gates to call a random oracle. Then a random oracle $\mathcal{R}$ and a key k for the combiner are sampled. We say the adversary wins the game if $\mathcal{A}_2$ breaks property π on combiner C initialized with the random oracle and the function output by $\mathcal{A}_1$: that is, $\mathcal{A}_2$ breaks property π on either combiner $C^{\mathcal{R},H^{\mathcal{R}}_{\mathcal{A}}}$ or on combiner $C^{H^{\mathcal{R}}_{\mathcal{A}},\mathcal{R}}$ (note the different order of oracles).

Ideally Secure Combiner$^{C}_{\mathcal{A}}$

$H^{\mathcal{R}}_{\mathcal{A}}, st \longleftarrow \mathcal{A}_1(1^n)$

$\mathcal{R}, k \longleftarrow$ sample RO and keys[3]

return $\mathcal{A}_2^{\mathcal{R},H^{\mathcal{R}}_{\mathcal{A}}}(st, k)$ breaks π for combiner $C^{\mathcal{R},H^{\mathcal{R}}_{\mathcal{A}}}$ or $C^{H^{\mathcal{R}}_{\mathcal{A}},\mathcal{R}}$

Fig. 1. Security of Combiners in Idealized Settings

[3] In the *Ideally Secure Combiner* game (and in following security games) the random oracle is sampled such that its domain and range matches allowed hash functions and the keys are sampled using the key generation algorithm of combiner C.

Definition 3. *A combiner C is called* ideally secure for property π *if no efficient adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ can win the* Ideally Secure Combiner *game (Figure 1) with non-negligible advantage.*

The security guarantees given in this model are that the combiner has property π as long as one of the two functions is a random oracle. Furthermore, security may be reduced to the security of compression functions, when analyzing the security in the indifferentiability model [28]. We find this notion particularly useful from a practical point of view as many security proofs are only given in the random oracle model (to name a few [16,5,6,11]) and a combiner proven secure under our new notion allows us to hedge against the failure of the instantiation of the random oracle in the corresponding scheme. Furthermore, while our new notion makes stronger assumptions about the ingoing hash functions it allows to bypass the restrictions given by the traditional definition. As these stronger assumptions are, however, frequently needed in security proofs for practical constructions, we do not loose anything by also applying the very same assumptions in the examinations of combiners to be used in these schemes. On the other hand, there is lots to gain.

Further note that our new notion is far from trivial to fulfill although we know that one of the two functions is ideal to begin with. Take the exclusive-or combiner (compare Section 3.1) as an example. If one of the functions can depend on the other, most, if not all properties are easily breakable. Let, for example, adversary $\mathcal{A}_1$ output function $H^{\mathcal{R}}_{\mathcal{A}}(M) := \mathcal{R}(M)$. In this setting the exclusive-or combiner would implement the constant zero function $C_{\oplus}(M) = \mathcal{R}(M) \oplus \mathcal{R}(M)$ which is, of course, not collision-resistant or pseudorandom.

Remark 1. Recently, Ristenpart et al. [39] gave the somewhat surprising result that the indifferentiability composition theorem does not hold in general but only in what they call *single-stage settings*. A game is called single-stage if we can assume a single global adversary. Note that this applies to all but one of the security games considered in this paper (see Figures 1 and 2), as we usually allow adversaries to pass on their current state without any restrictions. For the exception, Lemma 2, it can be shown that it falls into the class of *secure*-1-*pass-games* in the terminology of [25]. The authors in [25] study multi-stage games for which indifferentiability (with certain additions) suffices to allow composition. For games falling into their class of *secure*-1-*pass-games* no additions are needed and thus plain indifferentiability is sufficient to allow composition. The idea, why access to the underlying compression function does not yield any advantage is that all "interesting" random oracle evaluations (notably, $\mathcal{R}(m \oplus k_3)$, cf. Figure 3) have a block-length of exactly 1. Thus, length extension attacks via the computation of inner compression function evaluations do not yield any advantage over directly computing the full hash value.

4 A Short Multi-property Combiner for Hash Functions

In this section we present a new black-box combiner for two hash functions that does not increase the output length. The combiner is robust for pseudorandom-

ness (under the traditional definition of robust combiners) without needing to assume independence of the input functions (cf. Section 3.1). Further, it is *ideally secure* (cf. Definition 3) for collision resistance, second pre-image resistance, target collision resistance and pre-image resistance, that is, it holds these properties if one of the hash functions is instantiated with a random oracle or if one of the functions is indifferentiable from a random oracle (assuming an ideal compression function, also see remark at end of last the section).

Our construction is based on the exclusive-or combiner where each message block is preprocessed. To ease on notation, we will not explicitly model the key generation stage for hash functions but implicitly assume that the functions are chosen from a family of functions (i.e., the key is implicit in the hash function).

Construction 1. *Let* $H_1, H_2 : \{0,1\}^* \to \{0,1\}^n$ *be two hash functions and* $m_1||\ldots||m_\ell := M||pad(M)$ *be a message from the joint domain of both hash functions padded to a multiple of the block length* n*. The combiner is given by*

$$C^{H_1,H_2}(M) := G_1^{H_1,H_2}(M) \oplus G_2^{H_1,H_2}(M)$$

where G_1 *and* G_2 *are stateless and deterministic constructions given by*

$$G_1^{H_1,H_2}(M) := H_1\left(\tilde{m}_1^1 \| \ \ldots \ \| \ \tilde{m}_\ell^1\right) \qquad G_2^{H_1,H_2}(M) := H_2\left(\tilde{m}_1^2 \ \| \ \ldots \ \| \ \tilde{m}_\ell^2\right)$$

with preprocessed blocks

$$\tilde{m}_j^1 := H_2(1 \ \| \ m_j \oplus k_1) \oplus m_j \oplus k_2 \oplus H_1(1 \ \| \ m_j \oplus k_3)$$
$$\tilde{m}_j^2 := H_1(0 \ \| \ m_j \oplus k_4) \oplus m_j \oplus k_5 \oplus H_2(0 \ \| \ m_j \oplus k_6)$$

for $j := 1, \ldots, \ell$ *and for independently chosen keys* $k_i \in \{0,1\}^n$ *for* $i = 1, ..., 6$.

Let us examine the combiner more closely before proving its security. First notice that the combiner is symmetric, that is, it makes no difference if functions H_1 and H_2 are interchanged. Function $G_1(M)$ can be thought of as simply calling hash function H_1 on some preprocessed input. If the original input $m_1||\ldots||m_\ell := M||pad(M)$ consisted of ℓ blocks, then the preprocessed input also consists of ℓ blocks. Each block m_i is preprocessed independently and becomes

$$H_2(1 \ \| \ m_i \oplus k_1) \oplus m_i \oplus k_2 \oplus H_1(1 \ \| \ m_i \oplus k_3) \ .$$

The idea behind this construction is that the outer most hash function in G_1 (i.e., $H_1(\cdot)$) cannot, given its input, guess (or rather compute) the input that is going into the outer most hash function in G_2, i.e., $H_2(\cdot)$. This will become more evident when we prove security for various properties. Furthermore, note that we achieve domain separation between the calls to functions within G_1 and G_2 (i.e., calls to H_1 and H_2 are prefixed by 1 for G_1 and by 0 for G_2).

Finally, we want to note that the combiner can be efficiently implemented. If we take as measure the number of hash block evaluations then the combiner increases the number of evaluations by a factor of 3. However, in contrast to other multi-property combiners [21,22] it is completely parallelizable as each block is preprocessed independently of others.

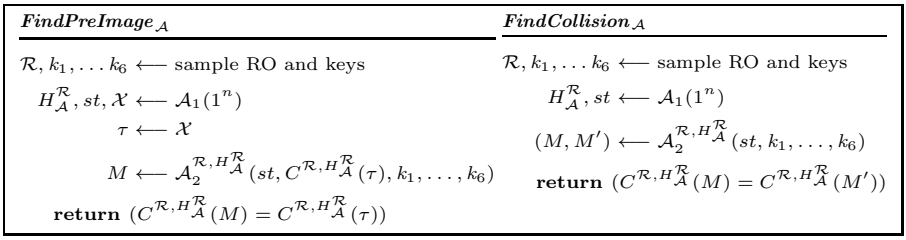

Fig. 2. Security Games

4.1 Security Analysis

We will first show that the combiner is pre-image resistant if one of its input functions is a random oracle. Remember that the basic XOR-combiner is not necessarily pre-image resistant even if instantiated with two random oracles (see Sections 3.1 and 3.3). We give the security experiments necessary for the following proofs in Figure 2.

Proposition 1. *Construction 1 is ideally secure for pre-image resistance (ow). That is, for any efficient adversary $\mathcal{A}$ which outputs efficiently sampleable distributions $\mathcal{X}$ with super-logarithmic min-entropy ($\mathrm{H}_\infty(\mathcal{X}) \in \omega(n)$) it holds that its advantage in the* FindPreImage *game is bound by*

$$\boldsymbol{Adv}^{\mathrm{FindPreImage}}_{\mathcal{A}}(1^n) \leq q_{\mathcal{A}} \cdot 2^{-\mathrm{H}_\infty(\mathcal{X})}$$

where $q_{\mathcal{A}}$ denotes an upper-bound on the number of combiner evaluations.

We prove Proposition 1 via an intermediate result about the preprocessed message blocks $\tilde{m}^b_j$ (cf. Construction 1). These we regard as "preprocessing functions" of the form $\{0,1\}^n \to \{0,1\}^n$ with oracle access to hash functions H_1 and H_2, parameterized by keys k_1, k_2, k_3, taking message blocks $m \in \{0,1\}^n$ as input and outputting a preprocessed message block; we write $\tilde{m}^{H_1,H_2}_{k_1,k_2,k_3}(m)$. We show that these pre-processed message blocks are, in fact, random variables with min-entropy n bits over the choice of random oracle and keys k_1, k_2, k_3. By applying the union bound, we can then argue that if an efficient adversary with access to the random oracle and keys $k_1, \ldots, k_3$ can choose message m it can at most reduce the entropy to $n - \mathcal{O}(\log n)$ bits, where the logarithmic reduction is bound by the number of random oracle evaluations.

Lemma 1. *The preprocessed blocks $\tilde{m}^{H_1,H_2}_{k_1,k_2,k_3}(\cdot)$ in Construction 1 are random variables with min-entropy n; that is, if $H_b := \mathcal{R}$ for $b \in \{1,2\}$ is a random oracle, then it holds for all message blocks $m \in \{0,1\}^n$ and functions H_{2-b+1} with restrictions as in Construction 1 that*

$$\tilde{\mathrm{H}}_\infty\left(\tilde{m}^{H_1,H_2}_{k_1,k_2,k_3}(m) | m, k_1, k_2, k_3\right) = n \tag{1}$$

where the probability is over the choice of random oracle $\mathcal{R}$ and keys $k_1, \ldots, k_3$.

To prove Lemma 1 we consider the following distribution (see Figure 3). The distribution is parameterized by an (efficient) algorithm $\mathcal{A}$, a random oracle from the function space $\{0,1\}^* \to \{0,1\}^n$ and uniformly and independently chosen keys k_1, k_2, k_3 from $\{0,1\}^n$. To compute the mapping for message m, adversary $\mathcal{A}$ receives value $m \oplus k_3$ and outputs a message m'. Value $\mathcal{R}(m \oplus k_1) \oplus m \oplus k_2 \oplus m'$ is returned as sample.

$\mathbf{Dist}_{\mathcal{A}}^{\mathcal{R},k_1,k_2,k_3}(m)$

$m' \longleftarrow \mathcal{A}^{\mathcal{R}}(1^n, m \oplus k_3)$

return $\mathcal{R}(m \oplus k_1) \oplus m \oplus k_2 \oplus m'$

Fig. 3. Adv. Controlled Distribution

Proof (of Lemma 1). In the adversarial distribution (Figure 3), the adversary can be regarded as the adversarially created function $H_{\mathcal{A}}^{\mathcal{R}}(\cdot)$ in Construction 1. Thus, we have that the min-entropy of the adversarial distribution (instantiated with any efficient adversary $\mathcal{A}$) is an upper bound for the min-entropy of $\tilde{m}_{k_1,k_2,k_3}^{H_1,H_2}$:

$$\tilde{\mathrm{H}}_\infty\left(\tilde{m}_{k_1,k_2,k_3}^{H_1,H_2}(m)|m,k_1,k_2,k_3\right) \geq \tilde{\mathrm{H}}_\infty\left(\mathrm{Dist}_{\mathcal{A}}^{\mathcal{R},k_1,k_2k_3}(m)|m,k_1,k_2,k_3\right)$$

As the keys are chosen uniformly at random from $\{0,1\}^n$ and in particular independently of the random oracle, we know that for every message m value $\mathcal{R}(m \oplus k_1)$ is uniformly distributed and thus:

$$\tilde{\mathrm{H}}_\infty\left(\mathcal{R}(m \oplus k_1) \oplus m \oplus k_2 | m, k_1, k_2, k_3\right) = n$$

To estimate the min-entropy of distribution $\mathrm{Dist}_{\mathcal{A}}^{\mathcal{R},k_1,k_2k_3}(\cdot)$ we thus need to analyze the effect of value m' as output by adversary $\mathcal{A}$ on input $m \oplus k_3$. In order to output m' such that the min-entropy of

$$\tilde{\mathrm{H}}_\infty\left(\mathcal{R}(m \oplus k_1) \oplus m \oplus k_2 \oplus m' | m, k_1, k_2, k_3\right) \tag{2}$$

is less than n bits, adversary $\mathcal{A}$ itself must have sufficient information on $\mathcal{R}(m \oplus k_1) \oplus m \oplus k_2$ given its sole input $m \oplus k_3$. To model, that $\mathcal{A}$ has access to the random oracle, we add its list of queries to the conditions. Let $\mathtt{qry}(\mathcal{A}^{\mathcal{R}}(m \oplus k_3))$ denote the query-answer pairs of $\mathcal{A}$ to the random oracle on input $m \oplus k_3$. Note that this is a random variable over the coins of $\mathcal{A}$ and the random oracle $\mathcal{R}$. Then, we can formalize the uncertainty of $\mathcal{A}$ about value $\mathcal{R}(m \oplus k_1) \oplus m \oplus k_2$ by

$$\tilde{\mathrm{H}}_\infty\left(\mathcal{R}(m \oplus k_1) \oplus m \oplus k_2 | m \oplus k_3, \mathtt{qry}(\mathcal{A}^{\mathcal{R}}(m \oplus k_3))\right) \tag{3}$$

It is easily seen that this denotes an upper bound for

$$\tilde{\mathrm{H}}_\infty\left(\mathcal{R}(m \oplus k_1) \oplus k_2 | m, \mathtt{qry}(\mathcal{A}^{\mathcal{R}}(m))\right) \tag{4}$$

where we removed the distortion of m by k_3 on the conditions, which in turn allows us to remove message m from the conditioned side. Note that values k_2 and $\mathcal{R}(m \oplus k_1)$ are uniformly distributed and independent (k_2 is chosen independently of $\mathcal{R}$ and similarly m and k_1 are chosen independently of $\mathcal{R}$). Thus we can analyze the two terms going into the exclusive-or operation individually; that is,

$$\begin{aligned}\tilde{\mathrm{H}}_\infty\left(k_2 \oplus \mathcal{R}(m \oplus k_1) | m, \mathtt{qry}(\mathcal{A}^{\mathcal{R}}(m))\right) \geq \\ \max\left(\tilde{\mathrm{H}}_\infty\left(k_2 | m, \mathtt{qry}(\mathcal{A}^{\mathcal{R}}(m))\right), \tilde{\mathrm{H}}_\infty\left(\mathcal{R}(m \oplus k_1) | m, \mathtt{qry}(\mathcal{A}^{\mathcal{R}}(m))\right)\right)\end{aligned} \tag{5}$$

As m is independent of keys k_1 and k_2 we have that both terms in the max-operation have n bits of entropy and thus

$$\tilde{\mathrm{H}}_\infty\left(k_2 \oplus \mathcal{R}(m \oplus k_1) | m, \mathtt{qry}(\mathcal{A}^{\mathcal{R}}(m))\right) = n. \tag{6}$$

Thus, adversary $\mathcal{A}$ cannot output m' such that the entropy in (2) is reduced. □

By an application of the union bound it follows that for any message m that is generated by an efficient adversary $\mathcal{A}^{H_1,H_2}(k_1,k_2,k_3)$ which is given the keys and that has oracle access to the hash functions, the min-entropy of $\tilde{m}^{H_1,H_2}_{k_1,k_2,k_3}(m)$ is at most reduced by logarithmically (in n) many bits (see full version [31] for details).

Lemma 2. *Let the setup be as in Lemma 1. Then, for all efficient adversaries $\mathcal{A}$ it holds that*

$$\tilde{\mathrm{H}}_\infty\left(\tilde{m}^{H_1,H_2}_{k_1,k_2,k_3}(m) | m \leftarrow \mathcal{A}^{H_1,H_2}(1^n,k_1,k_2,k_3),k_1,k_2,k_3\right) \geq n - \mathcal{O}(\log q) \tag{7}$$

where q is an upper bound on random oracle evaluations by H_{2-b+1} and adversary $\mathcal{A}$. The probability is over the choice of keys k_1,k_2,k_3, random oracle $\mathcal{R}$ and $\mathcal{A}$'s internal coin tosses.

Remark 2. We have examined Lemma 1 in the random oracle model using the information theoretic min-entropy notion. We can also analyze it in the privately keyed standard model assuming a pseudorandom function instead of a random oracle. For this we need to switch to a computational version of entropy such as HILL entropy (see [38] for an introduction). The proof works analogously.

We now prove Proposition 1 by showing that the advantage of any adversary in winning the *FindPreImage* game is bounded by $q_{\mathcal{A}} \cdot 2^{-\mathrm{H}_\infty(\mathcal{X})}$ where $q_{\mathcal{A}}$ is the number of combiner evaluations. Let us first examine the *FindPreImage* game. In a first step, a random oracle $\mathcal{R}$ is sampled from the space of all functions of the form $\{0,1\}^* \to \{0,1\}^n$ together with keys $k_1,\ldots,k_6$. Adversary $\mathcal{A}_1$ is then given the security parameter and it outputs a target distribution $\mathcal{X}$, some state st, and a description of a hash function $H^{\mathcal{R}}_{\mathcal{A}}$ which can contain special gates to evaluate random oracle $\mathcal{R}$ (note that $\mathcal{A}_1$ does not get access to $\mathcal{R}$ while constructing $H^{\mathcal{R}}_{\mathcal{A}}$ and that distribution $\mathcal{X}$ must have super-logarithmic min-entropy given state st). In a next step a target message τ is sampled from distribution $\mathcal{X}$. Then, adversary $\mathcal{A}_2$ is given keys $k_1,\ldots,k_6$ and and hash value $C^{\mathcal{R},H^{\mathcal{R}}_{\mathcal{A}}}(\tau)$ and is given oracle access to $\mathcal{R}$ and $H^{\mathcal{R}}_{\mathcal{A}}$. It wins if it outputs a message M which, under the combiner, yields value $C^{\mathcal{R},H^{\mathcal{R}}_{\mathcal{A}}}(\tau)$, i.e.: $C^{\mathcal{R},H^{\mathcal{R}}_{\mathcal{A}}}(M) = C^{\mathcal{R},H^{\mathcal{R}}_{\mathcal{A}}}(\tau)$.

Proof (Proposition 1). Let us examine the preprocessed message blocks going into $G_2^{\mathcal{R},H^{\mathcal{R}}_{\mathcal{A}}}$ (cf. Construction 1) for some message $m_1,\ldots,m_\ell := M\|PAD(M)$. Each block is of the form

$$\mathcal{R}(0 \parallel m_i \oplus k_4) \oplus m_i \oplus k_5 \oplus H^{\mathcal{R}}_{\mathcal{A}}(0 \parallel m_i \oplus k_6) \tag{8}$$

By Lemma 2 we can assume each of these blocks to be a random variable with min-entropy n bits (note that the factor $\log(q)$ of Lemma 2 is implicit in the number of random oracle queries by the adversary) and thus the combined blocks (via concatenation) to be a random variable of also at least n bits. The same necessarily holds for for the blocks going into $G_1^{\mathcal{R},H_{\mathcal{A}}^{\mathcal{R}}}$. Furthermore, by achieving domain separation for the random oracle calls (prefixing the input with 0 and 1, respectively) within $G_1^{\mathcal{R},H_{\mathcal{A}}^{\mathcal{R}}}(\cdot)$ and $G_2^{\mathcal{R},H_{\mathcal{A}}^{\mathcal{R}}}(\cdot)$, we can assume the random variables for blocks of G_1 to be independent of those for blocks of G_2.

If U_n and U'_n are independent random variables from the message space to $\{0,1\}^n$ with min-entropy n bits, then we can write the combiner $C^{\mathcal{R},H_{\mathcal{A}}^{\mathcal{R}}}$ as

$$C^{\mathcal{R},H_{\mathcal{A}}^{\mathcal{R}}}(M) := \mathcal{R}(U_n(M)) \oplus H_{\mathcal{A}}^{\mathcal{R}}(U'_n(M))$$

Hence, the probability for any message M to be mapped to $C^{\mathcal{R},H_{\mathcal{A}}^{\mathcal{R}}}(\tau)$ under the combiner is 2^{-n}. As one possible pre-image (namely τ) is contained in the support of distribution $\mathcal{X}$, the best strategy for an adversary is to sample messages from $\mathcal{X}$, which allows us to upper bound the advantage of an adversary winning in the *FindPreImage* game by

$$\mathbf{Adv}_{\mathcal{A}}^{FindPreImage}(1^n) \leq q_{\mathcal{A}} \cdot 2^{-\mathrm{H}_\infty(\mathcal{X})}$$

where $q_{\mathcal{A}}$ denotes the number of combiner queries by adversary $\mathcal{A}_2$. □

For second pre-image and target collision resistance it suffices to slightly change the *FindPreImage* game to adapt it to the specifics in the examined property.

Collision resistance is examined using the *FindCollision* game (see Figure 2). We show that the advantage of any efficient adversary is bound by $q_{\mathcal{A}}^2 \cdot 2^{-(n+1)}$ where $q_{\mathcal{A}}$ denotes the number of combiner evaluations. In short we show that as the inputs to the outer hash functions in G_1 and G_2 have entropy at least $n - \mathcal{O}(\log q)$ bits, the problem of finding collisions can be rewritten as finding collisions for

$$\mathcal{R}(U_n(M)) \oplus \mathcal{R}(U_n(M')) = H_{\mathcal{A}}^{\mathcal{R}}(U'_n(M)) \oplus H_{\mathcal{A}}^{\mathcal{R}}(U'_n(M))$$

where U_n and U'_n are again independent random variables mapping from $\{0,1\}^*$ to $\{0,1\}^n$ and having n bits of min-entropy (again the logarithmic factor is hidden in the number of U_n evaluations by the adversary). We refer to the full version [31] for details.

4.2 Pseudorandomness

Finally, we show that our combiner is robust for pseudorandomness and ideally secure for message authentication codes. For pseudorandomness we can directly show robustness in the standard model (that is, without assuming a random oracle). We want to stress that, in contrast to the exclusive-or combiner, we do not need to assume that the two ingoing functions H_1 and H_2 are independent (cf. Section 3.1).

Proposition 2. *The combiner given in construction 1 is robust for pseudorandomness.*

Proof (sketch). We have already argued that we can analyze Lemma 1 and Lemma 2 also in the standard model, using computational analogues of entropy (see remark following Lemma 2). Thus, assuming that H_1 is pseudorandom, Lemma 2 yields that the input to $G_1^{H_1,H_2}(M) := H_1(\tilde{M})$ has sufficiently high computational min-entropy and hence G_1 is pseudorandom. Due to the symmetric design of the combiner, this also yields that $G_2^{H_1,H_2}$ is pseudorandom if H_2 is pseudorandom. Note, that due to the domain separation, the inputs to the outer hash evaluations in G_1 and G_2 are independent and thus the further analysis can be reduced to the analysis of the exclusive-or combiner which we know to be robust for pseudorandom functions assuming independent inputs. □

Acknowledgments. I thank the anonymous reviewers for their valuable comments. This work was supported by CASED (`www.cased.de`).

References

1. Alwen, J., Dodis, Y., Wichs, D.: Leakage-resilient public-key cryptography in the bounded-retrieval model. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 36–54. Springer, Heidelberg (2009)
2. Andreeva, E., Mennink, B., Preneel, B.: On the indifferentiability of the Grøstl hash function. In: Garay, J.A., De Prisco, R. (eds.) SCN 2010. LNCS, vol. 6280, pp. 88–105. Springer, Heidelberg (2010)
3. Aoki, K., Sasaki, Y.: Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 70–89. Springer, Heidelberg (2009)
4. Baecher, P., Brzuska, C., Fischlin, M.: Notions of black-box reductions, revisited. Cryptology ePrint Archive, Report 2013/101 (2013), `http://eprint.iacr.org/`
5. Bellare, M., Boldyreva, A., O'Neill, A.: Deterministic and efficiently searchable encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 535–552. Springer, Heidelberg (2007)
6. Bellare, M., Brakerski, Z., Naor, M., Ristenpart, T., Segev, G., Shacham, H., Yilek, S.: Hedged public-key encryption: How to protect against bad randomness. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 232–249. Springer, Heidelberg (2009)
7. Bellare, M., Rogaway, P.: Random oracles are practical: A paradigm for designing efficient protocols. In: Ashby, V. (ed.) ACM CCS 1993, pp. 62–73. ACM Press (November 1993)
8. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: The keccak SHA-3 submission. Submission to NIST (Round 3) (2011), `http://keccak.noekeon.org/Keccak-submission-3.pdf`
9. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: On the indifferentiability of the sponge construction. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 181–197. Springer, Heidelberg (2008)

10. Bhattacharyya, R., Mandal, A., Nandi, M.: Indifferentiability characterization of hash functions and optimal bounds of popular domain extensions. In: Roy, B., Sendrier, N. (eds.) INDOCRYPT 2009. LNCS, vol. 5922, pp. 199–218. Springer, Heidelberg (2009)
11. Boldyreva, A., Cash, D., Fischlin, M., Warinschi, B.: Foundations of non-malleable hash and one-way functions. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 524–541. Springer, Heidelberg (2009)
12. Boneh, D., Boyen, X.: On the impossibility of efficiently combining collision resistant hash functions. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 570–583. Springer, Heidelberg (2006)
13. Canetti, R., Rivest, R., Sudan, M., Trevisan, L., Vadhan, S.P., Wee, H.M.: Amplifying collision resistance: A complexity-theoretic treatment. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 264–283. Springer, Heidelberg (2007)
14. De Cannière, C., Rechberger, C.: Preimages for reduced SHA-0 and SHA-1. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 179–202. Springer, Heidelberg (2008)
15. Chang, D., Nandi, M., Yung, M.: Indifferentiability of the hash algorithm BLAKE. Cryptology ePrint Archive, Report 2011/623 (2011), http://eprint.iacr.org/
16. Chevallier-Mames, B., Phan, D.H., Pointcheval, D.: Optimal asymmetric encryption and signature paddings. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 254–268. Springer, Heidelberg (2005)
17. Coron, J.-S., Dodis, Y., Malinaud, C., Puniya, P.: Merkle-damgård revisited: How to construct a hash function. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 430–448. Springer, Heidelberg (2005)
18. Damgård, I.: A design principle for hash functions. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 416–427. Springer, Heidelberg (1990)
19. De Cannière, C., Rechberger, C.: Finding SHA-1 characteristics: General results and applications. In: Lai, X., Chen, K. (eds.) ASIACRYPT 2006. LNCS, vol. 4284, pp. 1–20. Springer, Heidelberg (2006)
20. Dierks, T., Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard) (August 2008), http://www.ietf.org/rfc/rfc5246.txt, updated by RFCs 5746, 5878, 6176
21. Fischlin, M., Lehmann, A.: Multi-property preserving combiners for hash functions. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 375–392. Springer, Heidelberg (2008)
22. Fischlin, M., Lehmann, A., Pietrzak, K.: Robust multi-property combiners for hash functions revisited. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 655–666. Springer, Heidelberg (2008)
23. Fischlin, M., Lehmann, A., Wagner, D.: Hash function combiners in TLS and SSL. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 268–283. Springer, Heidelberg (2010)
24. Freier, A., Karlton, P., Kocher, P.: The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101 (Historic) (August 2011), http://www.ietf.org/rfc/rfc6101.txt
25. In Submission: Salvaging indifferentiability in a multi-stage setting (2013)
26. Lehmann, A.: On the Security of Hash Function Combiners. Ph.D. thesis, TU Darmstadt (März 2010), http://tuprints.ulb.tu-darmstadt.de/2094/
27. Liskov, M.: Constructing an ideal hash function from weak ideal compression functions. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 358–375. Springer, Heidelberg (2007)

28. Maurer, U., Renner, R., Holenstein, C.: Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 21–39. Springer, Heidelberg (2004)
29. Merkle, R.C.: One way hash functions and DES. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 428–446. Springer, Heidelberg (1990)
30. Mittelbach, A.: Hash combiners for second pre-image resistance, target collision resistance and pre-image resistance have long output. In: Visconti, I., De Prisco, R. (eds.) SCN 2012. LNCS, vol. 7485, pp. 522–539. Springer, Heidelberg (2012)
31. Mittelbach, A.: Cryptophia's short combiner for collision-resistant hash functions. Cryptology ePrint Archive, Report 2013/210 (2013), http://eprint.iacr.org/
32. Moody, D., Paul, S., Smith-Tone, D.: Improved indifferentiability security bound for the JH mode. Cryptology ePrint Archive, Report 2012/278 (2012), http://eprint.iacr.org/
33. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: 21st ACM STOC, pp. 33–43. ACM Press (May 1989)
34. National Institute of Standards and Technology: FIPS 180-3, Secure Hash Standard, Federal Information Processing Standard (FIPS), Publication 180-3. Tech. rep., Department of Commerce (August 2008)
35. NIST: NIST SHA-3 Competition, http://csrc.nist.gov/groups/ST/hash/sha-3/index.html
36. Pietrzak, K.: Non-trivial black-box combiners for collision-resistant hash-functions don't exist. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 23–33. Springer, Heidelberg (2007)
37. Pietrzak, K.: Compression from collisions, or why CRHF combiners have a long output. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 413–432. Springer, Heidelberg (2008)
38. Reyzin, L.: Some notions of entropy for cryptography (2011), http://www.cs.bu.edu/~reyzin/papers/entropy-survey.pdf
39. Ristenpart, T., Shacham, H., Shrimpton, T.: Careful with composition: Limitations of the indifferentiability framework. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 487–506. Springer, Heidelberg (2011)
40. Rivest, R.: The MD5 Message-Digest Algorithm. RFC 1321 (Informational) (April 1992), http://www.ietf.org/rfc/rfc1321.txt, updated by RFC 6151
41. Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: Roy, B., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 371–388. Springer, Heidelberg (2004)
42. Sasaki, Y., Aoki, K.: Finding preimages in full MD5 faster than exhaustive search. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 134–152. Springer, Heidelberg (2009)
43. Stevens, M., Sotirov, A., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D.A., de Weger, B.: Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 55–69. Springer, Heidelberg (2009)
44. Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full SHA-1. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 17–36. Springer, Heidelberg (2005)
45. Wang, X., Yu, H.: How to break MD5 and other hash functions. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 19–35. Springer, Heidelberg (2005)