A preliminary version of this paper appears in the proceedings of *11th Conference on Applied Cryptography and Network Security (ACNS 2013)*.
This is a revised and updated full version, available as IACR Cryptology ePrint Archive Report 2013/210.
Version: June 26, 2014

# 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, 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 [Riv92, Nat08, AHMP10, GKM+11, Wu11, BDPA11, FLS+10]. As history has shown, with time, it is not unlikely that cryptanalysists find plausible attacks [WY05, SSA+09, SA09, WYY05, DR06, AS09, CR08] 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 $\pi$ if it guarantees to satisfy property $\pi$ 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 $\pi$ 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 [FKK11, DR08].

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 $\pi$ is usually defined via a reductionist approach. That is, the combiner is called robust for $\pi$ if there exists a reduction $\mathcal{P}$ such that if $\mathcal{P}$ is given access to any (breaking-)oracle $\mathcal{B}$ that breaks $\pi$ on the combiner with non-negligible probability, then $\mathcal{P}^{\mathcal{B},H_1,H_2}$ must in turn break $\pi$ 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-resistance 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. [FL08, FLP08].

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 [BB06]. 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 [BB06, CRS+07, Pie07, Pie08, Mit12] where the lower bound is roughly the output length achieved by the concatenation combiner.

Black-box vs. Non-black-box. Constructions in cryptography are usually fully black-box [RTV04] in that the construction (in our case the combiner) accesses the primitive (i.e., hash functions) as a black-box and, similarly, the accompanying security reduction accesses the primitive and the adversary in a black-box way. This ensures that the combiner does indeed work for any hash function and is secure in a way such that any adversary against the combiner can be transformed into an efficient adversary against the underlying hash functions.

A recent framework by Baecher et al. [BBF13] allows to give a more fine-grained characterization of reductions (and thus separation results) in terms of their "level of black-boxness". Here reductions are characterized using the CAP notation denoting whether the (C)onstruction has black-box access to the primitive and whether the reduction accesses the (A)dversary and or the (P)rimitive in a black-box manner. Each access can be either black-box or not resulting in eight possible combinations of CAP types from $\{\mathtt{N},\mathtt{B}\}^3$. A BBB-combiner would, thus, work for any pair of hash functions (it only gets black-box access) and the security reduction would need to turn any adversary into an efficient adversary against the underlying hash functions itself only having black-box access to the hash functions.

The CAP classification becomes particularly interesting when considering impossibility results such as that robust combiners for collision resistant hash functions must have long output [Pie08]. On the outset it seems to only rule out fully black-box combiners (in the terminology of [RTV04]) but on closer inspection it is, in fact, ruling out NNN-reductions (see Section 3). This, however, means that to circumvent the impossibility result, a "mere switch" to non-black-box techniques would not suffice. Rather, it seems that the definition of robustness must be changed.

For this, consider once more Cryptophia's non-black-box magical combiner. Cryptess rejected the 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. Thus, robustness seems to require a combiner to be, in some sense, stronger than the strongest ingoing function (be the access black-box or non-black-box). In terms of security, however, this clearly 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 is tailored to analyze combiners in idealized models such as the random oracle model (ROM; [BR93]) or the indifferentiability framework introduced by Maurer, Renner and Holenstein [MRH04, CDMP05] giving guarantees of the form: *the combiner has property $\pi$ 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. As many security proofs for constructions used in practice (for example, [CMPP05, BBO07, BBN$^+$09, BCFW09]) rely on idealized models in the first place, our combiner is especially interesting from a practical point of view: under the same assumptions, it yields an efficient multi-property combiner with the same or even better security guarantees than are given for traditional combiners, without having to increase its output length.

# 2 Preliminaries

## 2.1 Notation

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. Note that for the exclusive-or operation we always ensure that $X$ and $Y$ are of the same length. 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.

We model algorithms as (oracle) Turing machine where an oracle invocation is counted as a single computation step. 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, $\text{Prob}[X = x]$ denotes the probability that $X$ takes on value $x$. By $\text{H}_\infty(X)$ we denote the min-entropy of variable $X$, defined as

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

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 [ADW09]) as

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

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

## 2.2 Hash Functions and their Properties

Formally, a hash function $\mathcal{H}$ is defined as a family of functions together with a key generation algorithm $\mathsf{HKGen}$ that picks one of the functions to be used. That is, a *hash function (family)* is a pair of efficient algorithms $\mathcal{H} = (\mathsf{HKGen}, H)$ where $\mathsf{HKGen}(1^n)$ 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.

### 2.2.1 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 $\texttt{negl}$ such that:

$$\mathbf{Adv}^{\text{cr}}_{\mathcal{H}}(\mathcal{A}) := \text{Prob}\left[\begin{array}{rcl} k \leftarrow \mathsf{HKGen}(1^n); & & M \neq M' \quad \wedge \\ (M, M') \leftarrow \mathcal{A}(k) & : & H_k(M) = H_k(M') \end{array}\right] \leq \texttt{negl}(n)$$

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 [RS04] 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 the second pre-image experiment, the adversary is given a message $M$ (sampled according to some distribution $\mathcal{M}$) and has to output a second pre-image $M'$ such that $M \neq M'$ and $H_k(M) = H_k(M')$. For this experiment the adversary's advantage is defined over the choice of key $k$ and the choice of message $M$. In the *target collision experiment*[1] the first pre-image $M$ is not sampled but specified by the adversary (without knowledge of key $k$). In a second step, the adversary then gets access to the key and again has to find a second pre-image $M'$ such that $M \neq M'$ and $H_k(M) = H_k(M')$. Here the adversary's advantage is defined only over the choice of key $k$.

More formally, a hash function is called second pre-image resistant if for any efficient adversary $\mathcal{A}$ there exists a negligible function $\texttt{negl}$ such that:

$$\mathbf{Adv}^{\text{spr}}_{\mathcal{H}}(\mathcal{A}) := \text{Prob}\left[\begin{array}{l} k \leftarrow \mathsf{HKGen}(1^n); \\ M \leftarrow \mathcal{M}(1^n); \\ M' \leftarrow \mathcal{A}(k, M) \end{array} : M \neq M' \wedge H_k(M) = H_k(M')\right] \leq \texttt{negl}(n)$$

where the probability is over the choice of key, message and $\mathcal{A}$'s internal coin tosses. The function family is called target collision-resistant if for any efficient adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2)$ there exists a negligible function $\texttt{negl}$ such that:

$$\mathbf{Adv}^{\text{tcr}}_{\mathcal{H}}(\mathcal{A}) := \text{Prob}\left[\begin{array}{l} (M, st) \leftarrow \mathcal{A}_1(1^n); \\ k \leftarrow \mathsf{HKGen}(1^n); \\ M' \leftarrow \mathcal{A}_2(k, M, st) \end{array} : M \neq M' \wedge H_k(M) = H_k(M')\right] \leq \texttt{negl}(n)$$

where the probability is over the choice of key and the adversary's $(\mathcal{A}_1, \mathcal{A}_2)$ internal coin tosses.

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 $H_k(M) = H_k(M')$. More formally, a hash function is *pre-image resistant*, if for any efficient adversary $\mathcal{A}$ there exists negligible function $\texttt{negl}$ such that:

$$\mathbf{Adv}^{\text{ow}}_{\mathcal{H}}(\mathcal{A}) := \text{Prob}\left[\begin{array}{l} k \leftarrow \mathsf{HKGen}(1^n); \\ M \leftarrow \mathcal{M}(1^n); \\ M' \leftarrow \mathcal{A}(k, H_k(M)) \end{array} : H_k(M) = H_k(M')\right] \leq \texttt{negl}(n)$$

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

[1] Note that target collision resistant hash functions are also known as universal one-way hash functions [NY89].

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

$$\mathbf{Adv}^{\text{prf}}_{\mathcal{A}}(\mathcal{A}) := \left|\text{Prob}_k\left[\mathcal{A}^{H_k}(1^n)=1\right] - \text{Prob}_f\left[\mathcal{A}^{f}(1^n)=1\right]\right| \leq \texttt{negl}(n)$$

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 the advantage of any efficient adversary $\mathcal{A}$ in the following security experiment is bounded by a negligible function $\texttt{negl}$:

$$\mathbf{Adv}^{\text{mac}}_{\mathcal{H}}(\mathcal{A}) := \text{Prob}\left[\begin{array}{ll} k \leftarrow \mathsf{HKGen}(1^n); & H_k(M^*) = t^* \quad \wedge \\ (t^*, M^*) \leftarrow \mathcal{A}^{H_k}(1^n) & M^* \text{ was not sent to hash oracle} \end{array}\right] \leq \texttt{negl}(n)$$

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

### 2.2.3 Random Oracles and Indifferentiability

Many security proofs are given in the random oracle model (ROM; [BR93]) where hash functions are modeled as ideal, i.e., as truly random functions (e.g., [CMPP05, BBO07, BBN$^+$09, BCFW09]). While random oracles have no structure at all hash functions, on the other hand, are usually built from a fixed-length compression function and some iteration scheme defining how arbitrarily long messages are hashed [Mer89, Dam89, Riv92, Lis06, AHMP10, GKM$^+$11, Wu11, BDPA11, FLS$^+$10].

The *indifferentiability* notion introduced by Maurer, Renner and Holenstein in [MRH04] 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. [CDMP05] 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 [NIS], including the winner Keccak [BDPA11] enjoy proofs of indifferentiability [CNY11, AMP10, MPST12, BDPV08, BMN09].

## 3 A Novel Definition of Combiners for Hash Functions

A $(k, l)$-combiner for property $\pi$ (for example, collision resistance) is a construction that, given access to $l$ hash functions, satisfies property $\pi$ as long as this is the case for at least $k$ "input" hash functions. Combiners in practice are usually $(1, 2)$-black-box-combiners, that is a construction which is given black-box access to two hash functions and which obeys property $\pi$ as long as either of the two functions does. In this paper we restrict ourselves to this "practical" class of combiners.

In this section we will examine the current definition of robust black-box combiners and explain why combiners robust under this definition must satisfy a lower bound on their output-length (Section 3.1). We then present a semi-black box extension to this definition (Sections 3.2) as well as a new notion for

analyzing combiners in idealized models (Section 3.3). The two have in common that they allow us to bypass the restriction on the output-length.

## 3.1 Black-box Combiners for Hash Functions

Combiners for hash functions are traditionally defined in the following fashion (see, for example, [BB06, Pie08] for a version of this definition for collision resistance): a hash-function combiner robust for property $\pi$ (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 $\pi$ as long as $\mathcal{H}_1$ or $\mathcal{H}_2$ obeys property $\pi$. Formally, a hash-function combiner $\mathcal{C} := (\mathsf{CKGen}, C, \mathcal{P})$, robust for property $\pi$, is a triple of efficient algorithms, where $\mathsf{CKGen}(1^n, \mathsf{HKGen}_1, \mathsf{HKGen}_2)$ 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 $C_{k_{H_1},k_{H_2},k_C}(M) := C(k_{H_1}, k_{H_2}, k_C, M)$ in target domain $\{0,1\}^n$. We will usually simply write $C^{H_1,H_2}(M)$ indicating that the combiner gets black-box access to the two hash functions. 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 $\pi$ on the combiner (for example, samples collisions) breaks property $\pi$ on both hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$. Note that $\mathcal{B}$ may be inefficient.

Folklore Combiners. 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 [FLW10, Leh10]. 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.

### 3.1.1 On the Level of Black-boxness

In a recently presented framework, Baecher et al. [BBF13] extend the notions developed by Reingold et al. [RTV04] to precisely allow capturing the level of "black-boxness" of a reduction (resp. construction). Reductions are classified according to the CAP terminology denoting whether the (C)onstruction accesses the primitive in a black-box way, and whether the reduction makes black-box use of the (A)dversary and/or the (P)rimitive. Each access can be either black-box or not resulting in eight possible combinations of CAP types from $\{\mathtt{N}, \mathtt{B}\}^3$. In terms of the definition of black-box combiners given, we have defined what would be a BBB-combiner as the primitives (i.e., the hash functions) are accessed in a black-box way by

the construction (i.e., the combiner) and the reduction has to work for any breaking-oracle while also only accessing the hash functions via black-box access. The framework is particularly useful to better classify impossibility results as it allows to pinpoint possible ways to circumvent the result. As we will see, the outlook for "short" combiners is rather grim.

### 3.1.2 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 [BB06] and has since been answered negatively [BB06, CRS+07, Pie07, Pie08]: combiners, robust for collision resistance, with significantly shorter output length than the concatenation combiner do not exist. A similar result was also proved for second pre-image resistance, target collision resistance and pre-image resistance [Rja09, Mit12].

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.

When we classify the reductions that are ruled out by the impossibility result sketched above in the framework of Beacher et al. [BBF13], then we see that not just `BBB`-constructions are ruled out, but essentially any `NNN`-construction is ruled out. For this note, that the breaking oracle is indifferent to the construction as well as indifferent to the reduction having non-black-box access to the hash functions and finally note that the breaking-oracle is universal in the sense that the reduction may depend on it. Thus, trying to find non-black-box techniques will not help in circumventing the lower bound.

## 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 $\pi$ 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 $\pi$.

**Definition 3.1** *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 $\pi$ on hash function $\mathcal{H}$ is defined as the maximum advantage of any adversary*

*running in time $t$ against property $\pi$ 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 3.2 (extension)** *Let $n$ be a security parameter. Let $\mathcal{C} := (\mathsf{CKGen}, C)$ be a combiner for hash functions $\mathcal{H}_1$ and $\mathcal{H}_2$ as defined earlier. Let $\pi$ be a property on hash functions. We say $\mathcal{C}$ is a robust combiner for property $\pi$ if $\mathcal{C}$ is robust under the original definition, or if for all $t \in \mathbb{N}$:*

$$\boldsymbol{AdvMax}^{t}_{\pi}(\mathcal{C}, 1^n) \leq \min\Big(\boldsymbol{AdvMax}^{t}_{\pi}(\mathcal{H}_1, 1^n), \boldsymbol{AdvMax}^{t}_{\pi}(\mathcal{H}_2, 1^n)\Big)$$

Note that any combiner that is robust for some property $\pi$ 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 $\pi$ 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.

Relaxing the Definition. The reduction guaranteed by the traditional definition needs to be efficient, that is, run in polynomial time. A consequence is that an adversary against the combiner induces an adversary against both input hash functions. However, the advantage of the induced adversary might be much lower while its runtime is much higher than that of the adversary against the combiner. For our extension, on the other hand, we have not allowed such a polynomial factor and require the combiner to be at least as strong as the stronger of the two functions. This might be a point we want to relax and only require that there exists a polynomial $\mathtt{poly}$ such that the advantage of an adversary against the combiner that runs in time $t$ is upper-bounded by an adversary against either $\mathcal{H}_1$ or $\mathcal{H}_2$ that runs in time $t \cdot \mathtt{poly}(n)$.

Similarly we could argue that it is sufficient if the advantage does not increase by more than a polynomial factor. Thus we would yield a definition that is attributed by two polynomials $p, p'$:

$$\mathbf{AdvMax}^{t}_{\pi,p,p'}(\mathcal{C}, 1^n) \leq \min\Big(p'(n) \cdot \mathbf{AdvMax}^{t \cdot p(n)}_{\pi}(\mathcal{H}_1, 1^n), p'(n) \cdot \mathbf{AdvMax}^{t \cdot p(n)}_{\pi}(\mathcal{H}_2, 1^n)\Big)$$

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 $\pi$ 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 $\pi$ on combiner $C$ initialized with the random oracle and the function output by $\mathcal{A}_1$: that is, $\mathcal{A}_2$ breaks property $\pi$ 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[2]

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

Figure 1: Security of Combiners in Idealized Settings

**Definition 3.3** *A combiner $C$ is called* ideally secure for property $\pi$ *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 $\pi$ 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 [MRH04]. 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 [CMPP05, BBO07, BBN$^+$09, BCFW09]) 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.** Recently, Ristenpart et al. [RSS11] 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 and we note that this applies to all the security games considered in this paper (see Figures 1 and 2).

# 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 pseudorandomness (under the traditional definition of robust

[2] 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 $C$.

combiners) without needing to assume independence of the input functions (cf. Section 3.1). Further, it is *ideally secure* (cf. Definition 3.3) 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 4.1** *Let $H_1, H_2 : \{0,1\}^* \to \{0,1\}^n$ be two hash functions and $m_1||\ldots||m_\ell := M||\mathsf{pad}(M)$ be a message from the joint domain of both hash functions padded to a multiple of the block length $n$ with some injective padding function* pad*. 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 $i = 1, ..., 6$.*

**Remark.** In the original work we claimed that the above construction is ideally secure for collision resistance, second-preimage resistance and target-collision resistance. However, Mennink and Preneel pointed out that the proof erroneously assumed injectivity of the preprocessing blocks $\tilde{m}$ and presented attacks exploiting this oversight [MP14]. Nevertheless, their observations do not invalidate the security claims on pre-image resistance, pseudorandomness and message-authentication security, and we include the original analysis for these security properties.
Mennink and Preneel also present an elegant fix to the above construction which restores its original security claims. For the purpose of completeness, we present their adapted construction in Section 4.2 and refer to [MP14] for its security analysis.

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 $m_1||\ldots||m_\ell := M||\mathsf{pad}(M)$ consisted of $\ell$ blocks, then the preprocessed input also consists of $\ell$ 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.,

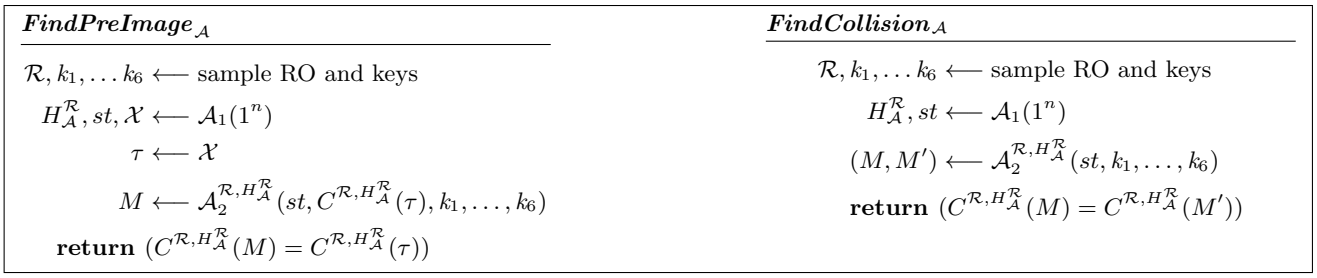


Figure 2: Security Games

$H_2(\cdot)$. 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 [FL08, FLP08] it is completely parallelizable as each block is preprocessed independently of others.

### 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 4.2** *Construction 4.1 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*

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

*where $q_{\mathcal{A}}$ denotes an upper-bound on the number of combiner evaluations and $q_H$ denotes a bound on the number of $\mathcal{R}$ evaluations in function $H^{\mathcal{R}}_{\mathcal{A}}$ as output by the first stage adversary.*

**Remark.** We note that the above bound is a worst case bound for a function $H^{\mathcal{R}}_{\mathcal{A}}$ designed by an adversarial entity. In any practical instantiation it should thus be safe to assume that $q_H$ is constant (i.e., $q_H \in \mathcal{O}(1)$) in which case the above bound is reduced to $\mathbf{Adv}^{FindPreImage}_{\mathcal{A}}(1^n) \leq q_{\mathcal{A}} \cdot 2^{-\mathrm{H}_\infty(\mathcal{X})}$.[3]

We prove Proposition 4.2 via an intermediate result about the preprocessed message blocks $\tilde{m}^b_j$ (cf. Construction 4.1). 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.

[3] In an earlier version of this work the term $\mathcal{O}(\log q_H)$ was missing from the proposition statement as pointed out by [MP14].

**Lemma 4.3** *The preprocessed blocks $\tilde{m}^{H_1,H_2}_{k_1,k_2,k_3}(\cdot)$ in Construction 4.1 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 4.1 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 4.3 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.

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

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

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

Figure 3: Adversarially Controlled Distribution

*Proof (of Lemma 4.3).* In the adversarial distribution (Figure 3), the adversary can be regarded as the adversarially created function $H^{\mathcal{R}}_{\mathcal{A}}(\cdot)$ in Construction 4.1.[4] Thus, we have that the min-entropy of the adversarial distribution is an upper bound for the min-entropy of $\tilde{m}^{H_1,H_2}_{k_1,k_2,k_3}$:

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

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 $\mathrm{Dist}^{\mathcal{R},k_1,k_2k_3}_{\mathcal{A}}(\cdot)$ 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 adversary $\mathcal{A}$ 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

[4] Note that although $H^{\mathcal{R}}_{\mathcal{A}}(\cdot)$ is deterministic we analyze the adversarial distribution allowing the adversary to be probabilistic.

($k_2$ 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,

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

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

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

It immediately follows that adversary $\mathcal{A}$ cannot output $m'$ such that the entropy in equation (2) is reduced. □

We would like to argue 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. For this note, that in equation (5) message $m$ is not necessarily independent of $k_1$ and $k_2$ any longer as it is chosen by adversary $\mathcal{A}$. This slightly complicates the argument as we show next.

**Lemma 4.4** *Let the setup be as in Lemma 4.3. 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.*

*Proof.* Let the setup be as in the previous proof for Lemma 4.3 up-to equation (5). Let us denote the "inner" adversary in $\mathrm{Dist}_{\mathcal{A}_2}$ by $\mathcal{A}_2$ and the "outer" adversary generating message $m$ by $\mathcal{A}_1$. Note that both adversaries are efficient. Let $q$ be an upper bound on the number of hash queries by the two adversaries.

As message $m$ is now chosen by an efficient adversary $\mathcal{A}_1$ that has access to keys $k_1$ and $k_2$ and the random oracle, the values $\mathcal{R}(m \oplus k_1)$ and $k_2$ are not necessarily independent any longer. Thus, we have to deploy a slightly more complex argument to estimate the entropy (cf. equations (3) and (4)):

$$\tilde{\mathrm{H}}_\infty\left(\mathcal{R}(m \oplus k_1) \oplus k_2 | m, \mathtt{qry}(\mathcal{A}_2^{\mathcal{R}}(m \oplus k_3))\right)$$

That is, now we consider the following setup:

$m \leftarrow \mathcal{A}_1^{\mathcal{R}}(k_1, k_2)$
$Q \leftarrow \mathtt{qry}^{\mathcal{R}}(\mathcal{A}_2^{\mathcal{R}}(m))$

First $m$ is chosen, then a set of queries to $\mathcal{R}$ are sampled by invoking $\mathcal{A}_2^{\mathcal{R}}(m)$ and now we need to estimate

$$\tilde{\mathrm{H}}_\infty(\mathcal{R}(m \oplus k_1) \oplus k_2 | m, Q) \ .$$

It is easily seen that

$$\tilde{\mathrm{H}}_\infty(k_1, k_2 | m, Q) = \tilde{\mathrm{H}}_\infty(k_1, k_2 | m) \geq 2n - |m| = n \ . \tag{8}$$

The keys are independently chosen bit-strings each of length $n$ bits. Furthermore, for the sampling of $Q$ the only information about the keys available is $m$ and thus $Q$ (given $m$) cannot further reduce the

entropy of $k_1$ and $k_2$. Now assume that an unbounded adversary is indeed able to predict $\mathcal{R}(m \oplus k_1) \oplus k_2$ with probability greater than $2^{-n+\mathcal{O}(\log q)}$. We can distinguish two cases: either $(m \oplus k_1) \in Q$, that is $Q$ contains the query $m \oplus k_1$ or not. If it does not contain the query, then we can immediately deduce that

$$\tilde{\mathrm{H}}_\infty(\mathcal{R}(m \oplus k_1) \oplus k_2 | m, Q) \geq n - \log q$$

as by $q$ many random oracle queries by $\mathcal{A}_1$, adversary $\mathcal{A}_1$ can choose $m$ such that the entropy is reduced by $\log q$ many bits. If $Q$ on the other hand contains query $m \oplus k_1$ (with more than negligible probability), then it must hold that

$$\tilde{\mathrm{H}}_\infty(m \oplus k_1 | m) \leq \mathcal{O}(\log q)$$

as $m$ is the only information given to $\mathcal{A}_2^{\mathcal{R}}(\cdot)$ to generate query set $Q$. As $k_1$ is a uniformly random string this, however, binds $n - \mathcal{O}(\log q)$ many bits of message $m$ as chosen by adversary $\mathcal{A}_1$. From this and with equation (8) it follows that then

$$\tilde{\mathrm{H}}_\infty(k_2 | m, Q) \geq n - \mathcal{O}(\log q)$$

and thus again

$$\tilde{\mathrm{H}}_\infty(\mathcal{R}(m \oplus k_1) \oplus k_2 | m, Q) \geq n - \mathcal{O}(\log q)$$

Thus, we have seen that adversary $\mathcal{A}_2$ in the adversarial distribution $\mathrm{Dist}_{\mathcal{A}_2}$ has an uncertainty of $n - \mathcal{O}(\log q)$ bits about value $\mathcal{R}(m \oplus k_1) \oplus k_2$ and can thus, by choosing $m'$ not reduce the min-entropy of (2) below $n - \mathcal{O}(\log q)$ bits which concludes the proof. □

**Remark.** We have examined Lemmas 4.3 and 4.4 in the random oracle model using the information theoretic min-entropy notion. We can, however, 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; we give an introduction to computational entropy in Appendix A. The proof works analogously.

We now prove Proposition 4.2 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 $H_{\mathcal{A}}^{\mathcal{R}}$ 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 $H_{\mathcal{A}}^{\mathcal{R}}$ and that distribution $\mathcal{X}$ must have super-logarithmic min-entropy given state $st$). In a next step a target message $\tau$ is sampled from distribution $\mathcal{X}$. Then, adversary $\mathcal{A}_2$ is given keys $k_1, \ldots, k_6$ and and hash value $C^{\mathcal{R}, H_{\mathcal{A}}^{\mathcal{R}}}(\tau)$ and is given oracle access to $\mathcal{R}$ and $H_{\mathcal{A}}^{\mathcal{R}}$. It wins if it outputs a message $M$ which, under the combiner, yields value $C^{\mathcal{R}, H_{\mathcal{A}}^{\mathcal{R}}}(\tau)$, i.e.: $C^{\mathcal{R}, H_{\mathcal{A}}^{\mathcal{R}}}(M) = C^{\mathcal{R}, H_{\mathcal{A}}^{\mathcal{R}}}(\tau)$.

*Proof (Proposition 4.2).* Let us examine the preprocessed message blocks going into $G_2^{\mathcal{R}, H_{\mathcal{A}}^{\mathcal{R}}}$ (cf. Construction 4.1) for some message $m_1, \ldots, m_\ell := M \| \mathsf{pad}(M)$. Each block is of the form

$$\mathcal{R}(0 \parallel m_i \oplus k_4) \oplus m_i \oplus k_5 \oplus H_{\mathcal{A}}^{\mathcal{R}}(0 \parallel m_i \oplus k_6) \tag{9}$$

By Lemma 4.4 we can assume each of these blocks to be a random variable with min-entropy $n - \mathcal{O}(\log q_H)$ bits where $q_H$ denotes the number of random oracle evaluations within function $H_{\mathcal{A}}^{\mathcal{R}}$. Thus the combined blocks (via concatenation) to be a random variable of also at least $n - \mathcal{O}(\log q_H)$ bits. The same necessarily holds 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 $2^{-n+\mathcal{O}(\log q_H)}$. As one possible pre-image (namely $\tau$) 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

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

□

Pseudorandomnessand Message Authentication Codes. Next, 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). For message authentication codes we only give the trivial statement that, if the combiner is instantiated with one pseudorandom function, then the combiner yields a secure MAC. We leave as open question whether the combiner can be proved robust also for message authentication codes.

**Proposition 4.5** *The combiner given in construction 4.1 is robust for pseudorandomness.*

*Proof (sketch).* We have already argued that we can analyze Lemma 4.3 and Lemma 4.4 also in the standard model, using computational analogues of entropy (see remark following Lemma 4.4). Thus, assuming that $H_1$ is pseudorandom, Lemma 4.4 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. □

**Corollary 4.6** *The combiner given in construction 4.1 is a secure message authentication code if either $H_1$ or $H_2$ is a pseudorandom function.*

*Proof.* Follows with Proposition 4.5 and the fact that a pseudorandom function is a secure message authentication code. □

## 4.2 Collision Resistance

In the original work we claimed that our combiner from Construction 4.1 is also collision resistant (and second pre-image/target collision resistant). As pointed out by Mennink and Preneel there was a gap in the proof as we falsely assumed that the preprocessing of the messages is injective [MP14]. Mennink and Preneel provide a nice fix to our construction which ensures that preprocessed blocks $\tilde{m}$ and $\tilde{m}'$ are different whenever $m$ and $m'$ are different (except with negligible probability) which then allows to prove

collision resistance (and second pre-image resistance/target collision resistance). We present the adapted construction next and refer to [MP14] for a security proof of the adapted construction and for further information on the attack on Construction 4.1. For this additional keys $l_1$ and $l_2$ are introduced which are prepended to the function inputs. The addition of keys $l_1$ and $l_2$ also make what where originally keys $k_2, k_5$ superfluous, that is the keys that were added outside of the hash computation. We next present the adapted construction which essentially only differs in the preprocessing.

**Construction 4.7 ([MP14])** *Let $H_1, H_2 : \{0,1\}^* \to \{0,1\}^n$ be two hash functions and $m_1||\ldots||m_\ell := M||\mathsf{pad}(M)$ be a message from the joint domain of both hash functions padded to a multiple of the block length $n$ with some injective padding function* pad*. 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*

$$\tilde{m}_j^1 := H_1(0 \ \| \ l_1 \ \| \ m_j \oplus k_1) \oplus H_2(0 \ \| \ l_2 \ \| \ m_j \oplus k_2)$$
$$\tilde{m}_j^2 := H_1(1 \ \| \ l_1 \ \| \ m_j \oplus k_1) \oplus H_2(1 \ \| \ l_2 \ \| \ m_j \oplus k_2)$$

*for $j := 1, \ldots, \ell$ and for independently chosen keys $k_i, l_i \in \{0,1\}^n$ for $i = 1, 2$.*

## Acknowledgments

I thank the anonymous reviewers for their valuable comments. In particular I'd like to thank Bart Mennink and Bart Preneel for spotting the gap in the original work and for providing a nice patch [MP14]. This work was supported by CASED (`www.cased.de`).

## 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. (Cited on page 4.)

[AHMP10] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W. Phan. SHA-3 proposal BLAKE. Submission to NIST (Round 3), 2010. (Cited on pages 2 and 6.)

[AMP10] Elena Andreeva, Bart Mennink, and Bart Preneel. On the indifferentiability of the Grøstl hash function. In Juan A. Garay and Roberto De Prisco, editors, *SCN 10: 7th International Conference on Security in Communication Networks*, volume 6280 of *Lecture Notes in Computer Science*, pages 88–105, Amalfi, Italy, September 13–15, 2010. Springer, Berlin, Germany. (Cited on page 6.)

[AS09] Kazumaro Aoki and Yu Sasaki. Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 70–89, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany. (Cited on page 2.)

[BB06] Dan Boneh and Xavier Boyen. On the impossibility of efficiently combining collision resistant hash functions. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 570–583, Santa Barbara, CA, USA, August 20–24, 2006. Springer, Berlin, Germany. (Cited on pages 2, 3, 7, and 8.)

[BBF13] Paul Baecher, Christina Brzuska, and Marc Fischlin. Notions of black-box reductions, revisited. In Kazue Sako and Palash Sarkar, editors, *Advances in Cryptology – ASIACRYPT 2013, Part I*, volume 8269 of *Lecture Notes in Computer Science*, pages 296–315, Bengalore, India, December 1–5, 2013. Springer, Berlin, Germany. (Cited on pages 3, 7, and 8.)

[BBN+09] Mihir Bellare, Zvika Brakerski, Moni Naor, Thomas Ristenpart, Gil Segev, Hovav Shacham, and Scott Yilek. Hedged public-key encryption: How to protect against bad randomness. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 232–249, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany. (Cited on pages 4, 6, and 10.)

[BBO07] Mihir Bellare, Alexandra Boldyreva, and Adam O'Neill. Deterministic and efficiently searchable encryption. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 535–552, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Berlin, Germany. (Cited on pages 4, 6, and 10.)

[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. (Cited on pages 4, 6, and 10.)

[BDPA11] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. The keccak SHA-3 submission. Submission to NIST (Round 3), 2011. (Cited on pages 2 and 6.)

[BDPV08] Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. On the indifferentiability of the sponge construction. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany. (Cited on page 6.)

[BMN09] Rishiraj Bhattacharyya, Avradip Mandal, and Mridul Nandi. Indifferentiability characterization of hash functions and optimal bounds of popular domain extensions. In Bimal K. Roy and Nicolas Sendrier, editors, *Progress in Cryptology - INDOCRYPT 2009: 10th International Conference in Cryptology in India*, volume 5922 of *Lecture Notes in Computer Science*, pages 199–218, New Delhi, India, December 13–16, 2009. Springer, Berlin, Germany. (Cited on page 6.)

[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. (Cited on pages 3 and 6.)

[BSW03] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In *RANDOM-APPROX*, pages 200–215, 2003. (Cited on page 23.)

[CDMP05] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In Victor Shoup, editor, *Advances*

*in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 430–448, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany. (Cited on pages 3 and 6.)

[CMPP05] Benoît Chevallier-Mames, Duong Hieu Phan, and David Pointcheval. Optimal asymmetric encryption and signature paddings. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *ACNS 05: 3rd International Conference on Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 254–268, New York, NY, USA, June 7–10, 2005. Springer, Berlin, Germany. (Cited on pages 4, 6, and 10.)

[CNY11] Donghoon Chang, Mridul Nandi, and Moti Yung. Indifferentiability of the hash algorithm BLAKE. Cryptology ePrint Archive, Report 2011/623, 2011. `http://eprint.iacr.org/2011/623`. (Cited on page 6.)

[CR08] Christophe De Cannière and Christian Rechberger. Preimages for reduced SHA-0 and SHA-1. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 179–202, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany. (Cited on page 2.)

[CRS+07] Ran Canetti, Ronald L. Rivest, Madhu Sudan, Luca Trevisan, Salil P. Vadhan, and Hoeteck Wee. Amplifying collision resistance: A complexity-theoretic treatment. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 264–283, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Berlin, Germany. (Cited on pages 3 and 8.)

[Dam89] Ivan Damgård. A design principle for hash functions. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427, Santa Barbara, CA, USA, August 20–24, 1989. Springer, Berlin, Germany. (Cited on page 6.)

[DP08] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *49th Annual Symposium on Foundations of Computer Science*, pages 293–302, Philadelphia, Pennsylvania, USA, October 25–28, 2008. IEEE Computer Society Press. (Cited on page 23.)

[DR06] Christophe De Cannière and Christian Rechberger. Finding SHA-1 characteristics: General results and applications. In Xuejia Lai and Kefei Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 1–20, Shanghai, China, December 3–7, 2006. Springer, Berlin, Germany. (Cited on page 2.)

[DR08] T. Dierks and E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard), August 2008. Updated by RFCs 5746, 5878, 6176. (Cited on page 2.)

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540, Interlaken, Switzerland, May 2–6, 2004. Springer, Berlin, Germany. (Cited on page 23.)

[FKK11] A. Freier, P. Karlton, and P. Kocher. The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101 (Historic), August 2011. (Cited on page 2.)

[FL08] Marc Fischlin and Anja Lehmann. Multi-property preserving combiners for hash functions. In Ran Canetti, editor, *TCC 2008: 5th Theory of Cryptography Conference*, volume 4948 of

*Lecture Notes in Computer Science*, pages 375–392, San Francisco, CA, USA, March 19–21, 2008. Springer, Berlin, Germany. (Cited on pages 2 and 12.)

[FLP08] Marc Fischlin, Anja Lehmann, and Krzysztof Pietrzak. Robust multi-property combiners for hash functions revisited. 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 655–666, Reykjavik, Iceland, July 7–11, 2008. Springer, Berlin, Germany. (Cited on pages 2 and 12.)

[FLS+10] Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker. The skein hash function family. Submission to NIST (Round 3), 2010. (Cited on pages 2 and 6.)

[FLW10] Marc Fischlin, Anja Lehmann, and Daniel Wagner. Hash function combiners in TLS and SSL. In Josef Pieprzyk, editor, *Topics in Cryptology – CT-RSA 2010*, volume 5985 of *Lecture Notes in Computer Science*, pages 268–283, San Francisco, CA, USA, March 1–5, 2010. Springer, Berlin, Germany. (Cited on page 7.)

[FOR12] Benjamin Fuller, Adam O'Neill, and Leonid Reyzin. A unified approach to deterministic encryption: New constructions and a connection to computational entropy. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 582–599, Taormina, Sicily, Italy, March 19–21, 2012. Springer, Berlin, Germany. (Cited on page 23.)

[GKM+11] Praveen Gauravaram, Lars R. Knudsen, Krystian Matusiewicz, Florian Mendel, Christian Rechberger, Martin Schlffer, and Sren S. Thomsen. Grstl – a SHA-3 candidate. Submission to NIST (Round 3), 2011. (Cited on pages 2 and 6.)

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. (Cited on page 22.)

[HLR07] Chun-Yuan Hsiao, Chi-Jen Lu, and Leonid Reyzin. Conditional computational entropy, or toward separating pseudoentropy from compressibility. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 169–186, Barcelona, Spain, May 20–24, 2007. Springer, Berlin, Germany. (Cited on page 23.)

[Leh10] Anja Lehmann. *On the Security of Hash Function Combiners*. PhD thesis, TU Darmstadt, März 2010. (Cited on page 7.)

[Lis06] Moses Liskov. Constructing an ideal hash function from weak ideal compression functions. In Eli Biham and Amr M. Youssef, editors, *SAC 2006: 13th Annual International Workshop on Selected Areas in Cryptography*, volume 4356 of *Lecture Notes in Computer Science*, pages 358–375, Montreal, Canada, August 17–18, 2006. Springer, Berlin, Germany. (Cited on page 6.)

[Mer89] Ralph C. Merkle. One way hash functions and DES. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446, Santa Barbara, CA, USA, August 20–24, 1989. Springer, Berlin, Germany. (Cited on page 6.)

[Mit12] Arno Mittelbach. Hash combiners for second pre-image resistance, target collision resistance and pre-image resistance have long output. In Ivan Visconti and Roberto De Prisco, editors,

*SCN 12: 8th International Conference on Security in Communication Networks*, volume 7485 of *Lecture Notes in Computer Science*, pages 522–539, Amalfi, Italy, September 5–7, 2012. Springer, Berlin, Germany. (Cited on pages 3 and 8.)

[MP14] Bart Mennink and Bart Preneel. Breaking and fixing Cryptophia's short combiner. To appear, 2014. (Cited on pages 11, 12, 16, and 17.)

[MPST12] Dustin Moody, Souradyuti Paul, and Daniel Smith-Tone. Improved indifferentiability security bound for the JH mode. Cryptology ePrint Archive, Report 2012/278, 2012. `http://eprint.iacr.org/2012/278`. (Cited on page 6.)

[MRH04] Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany. (Cited on pages 3, 6, and 10.)

[Nat08] National Institute of Standards and Technology. FIPS 180-3, Secure Hash Standard, Federal Information Processing Standard (FIPS), Publication 180-3. Technical report, Department of Commerce, August 2008. (Cited on page 2.)

[NIS] NIST. NIST SHA-3 Competition. (Cited on page 6.)

[NY89] Moni Naor and Moti Yung. Universal one-way hash functions and their cryptographic applications. In *21st Annual ACM Symposium on Theory of Computing*, pages 33–43, Seattle, Washington, USA, May 15–17, 1989. ACM Press. (Cited on page 5.)

[Pie07] Krzysztof Pietrzak. Non-trivial black-box combiners for collision-resistant hash-functions don't exist. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 23–33, Barcelona, Spain, May 20–24, 2007. Springer, Berlin, Germany. (Cited on pages 3 and 8.)

[Pie08] Krzysztof Pietrzak. Compression from collisions, or why CRHF combiners have a long output. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 413–432, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Berlin, Germany. (Cited on pages 3, 7, and 8.)

[Rey11] Leonid Reyzin. Some notions of entropy for cryptography, 2011. (Cited on page 22.)

[Riv92] R. Rivest. The MD5 Message-Digest Algorithm. RFC 1321 (Informational), April 1992. Updated by RFC 6151. (Cited on pages 2 and 6.)

[Rja09] Michal Rjasko. On existence of robust combiners for cryptographic hash functions. In *ITAT*, pages 71–76, 2009. (Cited on page 8.)

[RS04] Phillip Rogaway and Thomas Shrimpton. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 371–388, New Delhi, India, February 5–7, 2004. Springer, Berlin, Germany. (Cited on page 5.)

[RSS11] Thomas Ristenpart, Hovav Shacham, and Thomas Shrimpton. Careful with composition: Limitations of the indifferentiability framework. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 487–506, Tallinn, Estonia, May 15–19, 2011. Springer, Berlin, Germany. (Cited on page 10.)

[RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *TCC 2004: 1st Theory of Cryptography Conference*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20, Cambridge, MA, USA, February 19–21, 2004. Springer, Berlin, Germany. (Cited on pages 3 and 7.)

[SA09] Yu Sasaki and Kazumaro Aoki. Finding preimages in full MD5 faster than exhaustive search. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 134–152, Cologne, Germany, April 26–30, 2009. Springer, Berlin, Germany. (Cited on page 2.)

[SSA+09] Marc Stevens, Alexander Sotirov, Jacob Appelbaum, Arjen K. Lenstra, David Molnar, Dag Arne Osvik, and Benne de Weger. Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 55–69, Santa Barbara, CA, USA, August 16–20, 2009. Springer, Berlin, Germany. (Cited on page 2.)

[Wu11] Hongjun Wu. The hash function JH. Submission to NIST (round 3), 2011. (Cited on pages 2 and 6.)

[WY05] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35, Aarhus, Denmark, May 22–26, 2005. Springer, Berlin, Germany. (Cited on page 2.)

[WYY05] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36, Santa Barbara, CA, USA, August 14–18, 2005. Springer, Berlin, Germany. (Cited on page 2.)

# A On Computational Analogues of Entropy

A good introduction to various notions of (computational) entropy is given by Reyzin [Rey11]. We here give a very brief introduction.

The statistical distance for two distributions $X$ and $Y$ is given by

$$\delta(X,Y) := \frac{1}{2}\sum_x \left|\mathrm{Prob}[\,X=x] - \mathrm{Prob}[\,Y=x]\right| \ .$$

Let $\mathcal{D}_s^{\mathrm{prob},\{0,1\}}$ be the set of all probabilistic circuits of size $s$ with binary output and $\mathcal{D}_s^{\mathrm{det},[0,1]}$ be the set of all deterministic circuits of size $s$ with range $[0,1]$. Given a distinguisher (circuit) $D$ we define the computational distance for two distributions $X$ and $Y$ as

$$\delta^D(X,Y) := \Big|\,\mathbb{E}[\,D(X)] - \mathbb{E}[\,D(Y)]\Big|$$

This allows us to define a computational analogue of entropy introduced by Håstad et al. [HILL99].

**Definition A.1** *A distribution $X$ has* HILL entropy *at least $k$ denoted by $\mathrm{H}^{HILL}_{\epsilon,\mathrm{s}}(X) \geq k$, if there exists a distribution $Y$ with min-entropy at least $k$, i.e., $\mathrm{H}_\infty(Y) \geq k$ such that for all circuits $D \in \mathcal{D}_s^{prob,\{0,1\}}$ it holds that $\delta^D(X,Y) \leq \epsilon$.*

If we exchange the order of quantifiers we get the so called *Metric* type entropy. If we only consider deterministic distinguishers with range $[0,1]$ than we arrive at *Metric* entropy* introduced by Dziembowski and Pietrzak in [DP08].

**Definition A.2** *A distribution $X$ has* Metric* entropy *at least $k$ denoted by $\mathrm{H}^{Metric^*}_{\epsilon,\mathrm{s}}(X) \geq k$, if for all deterministic distinguishers $D \in \mathcal{D}_s^{det,[0,1]}$ there exists a distribution $Y$ with min-entropy at least $k$, i.e., $\mathrm{H}_\infty(Y) \geq k$ such that $\delta^D(X,Y) \leq \epsilon$.*

A conditional version of HILL entropy and Metric entropy is presented in [HLR07]. For Metric* entropy simply exchange the quantifiers and only allow distinguishers in $\mathcal{D}_s^{\mathrm{det},[0,1]}$ in the following definition.

**Definition A.3** *Let $(X,Y)$ be a pair of distributions. We say $X$ has* conditional HILL entropy *at least $k$ conditioned on $Y$ and denoted by $\tilde{\mathrm{H}}^{HILL}_{\epsilon,\mathrm{s}}(X|Y) \geq k$, if there exists a collection of distributions $Z_y$ for each $y \in Y$, giving rise to a joint distribution $(Z,Y)$, such that $Z$ has conditional min-entropy at least $k$, i.e., $\tilde{\mathrm{H}}_\infty(Z|Y) \geq k$ and for all $D \in \mathcal{D}_s^{prob,\{0,1\}}$ it holds that $\delta^D((X,Y),(Z,Y)) \leq \epsilon$.*

The entropy notions HILL, Metric and Metric* can be converted into one another [BSW03, FOR12]. HILL entropy can losslessly be converted into Metric entropy and further into Metric* entropy. Metric* entropy can be converted back to HILL entropy with only a small loss in $s$ and $\epsilon$.

For regular, conditional min-entropy the following statement holds [DRS04] which yields a lower estimate:

**Lemma A.4** *For distributions $X$ and $Z$ it holds:*

$$\tilde{\mathrm{H}}_\infty(X|Z) \geq \mathrm{H}_\infty(X,Z) - \log|Z|$$

*where $|Z|$ denotes the number of elements in $Z$.*

For the computational HILL Entropy Lemma A.4 does not directly hold. However, Fuller, O'Neill and Reyzin present in [FOR12] an analogue for Metric* entropy:

**Lemma A.5** *For distributions $X$ and $Z$ it holds:*

$$\mathrm{H}^{Metric^*}_{\epsilon|\mathrm{Y}|,\mathrm{s}}(X|Z) \geq \mathrm{H}^{Metric^*}_{\epsilon,\mathrm{s}}(X,Z) - \log|Z|$$

*where $|Z|$ denotes the number of elements in $Z$.*

Together with the transformation between HILL and Metric* entropy this then yields an analogous statement for HILL entropy.