Robust Multi-Property Combiners for Hash Functions*

Marc Fischlin[1] Anja Lehmann[2] Krzysztof Pietrzak[3]

[1]Darmstadt University of Technology, Germany
marc.fischlin@gmail.com

[2]IBM Research – Zurich, Switzerland
anj@zurich.ibm.com

[3]Institute of Science and Technology, Austria
krzpie@gmail.com

Abstract. A robust combiner for hash functions takes two candidate implementations and constructs a hash function which is secure as long as at least one of the candidates is secure. So far, hash function combiners only aim at preserving a single property such as collision-resistance or pseudorandomness. However, when hash functions are used in protocols like TLS they are often required to provide several properties simultaneously.

We therefore put forward the notion of *robust multi-property combiners* and elaborate on different definitions for such combiners. We then propose a combiner that provably preserves (target) collision-resistance, pseudorandomness, and being a secure message authentication code. This combiner satisfies the strongest notion we propose, which requires that the combined function satisfies every security property which is satisfied by at least one of the underlying hash function. If the underlying hash functions have output length n, the combiner has output length $2n$. This basically matches a known lower bound for black-box combiners for collision-resistance only, thus the other properties can be achieved without penalizing the length of the hash values. We then propose a combiner which also preserves the property of being indifferentiable from a random oracle, slightly increasing the output length to $2n + \omega(\log n)$. Moreover, we show how to augment our constructions in order to make them also robust for the one-wayness property, but in this case require an a priory upper bound on the input length.

*This is the full version of the papers *Multi-Property Preserving Combiners for Hash Functions* (appeared in Theory of Cryptography - TCC 2008, LNCS 4948, pp. 372-389, Springer, 2008) and *Robust Multi-Property Combiners for Hash Functions Revisited* (appeared in ICALP 2008, LNCS 5125, pp. 655-667, Springer, 2008).

1 Introduction

Recent attacks on collision-resistant hash functions [31, 32, 13, 30] caused a decrease of confidence that today's candidates really have this property and have raised the question how to devise constructions that are more tolerant to cryptanalytic results. Hence, approaches like robust combiners [19, 20, 18] which "merge" several candidate functions into a single failure-tolerant one, are of great interest and have triggered a long line of research [11, 28, 12, 15, 16, 21, 29, 17]. Informally, a hash combiner takes two hash functions H_0, H_1 and combines them in such a way that the resulting function satisfies some security property, whenever one of the underlying candidates H_0 or H_1 is satisfies it. For example, the "concatenation combiner" $\mathsf{Comb}_{||}^{H_0,H_1}(M) = H_0(M)||H_1(M)$ preserves the property of being collision-resistant because a collision $M \neq M'$ for the combiner is always also a collision for both components H_0 or H_1. Thus if either of the hash functions H_0 or H_1 is collision-resistant, then so is the combined function.

However, hash functions are currently used for various tasks that require numerous properties beyond collision resistance, e.g., the HMAC construction [3] based on a keyed hash function is used (amongst others) in the IPSec and TLS protocols as a pseudorandom function and as a MAC. In the standardized protocols RSA-OAEP [6] and RSA-PSS [7] even stronger properties are required (cf. [9, 10]), prompting Coron et al. [14] to give constructions which propagate the random-oracle property from the compression function to the hash function. A further example for the need of multiple properties is given by Katz and Shin [22], where collision-resistant pseudorandom functions are required in order to protect authenticated group key exchange protocols against insider attacks.[1]

While one could in principle always employ a suitable hash combiner tailored to the individual security property needed by one particular cryptographic scheme, common practices such as code re-use, call for the design of a *single* (combiner) function satisfying as many properties as possible. On the level of hash functions this point of view has also been adopted by NIST in its on-going SHA-3 competition [27] and motivated a series of works [4, 2, 23] that, e.g., show how to lift multiple properties provided by a compression functions to a full-grown hash function.

Thus, also for hash combiners one would ideally like to have a single construction that is robust for many properties simultaneously. Combiners which preserve a single property such as collision-resistance or pseudorandomness are quite well understood. Robust multi-property combiners, on the other hand, are not covered by these strategies and require new techniques instead. As an example we discuss this issue for the case of collision-resistance and pseudorandomness.

The Problem with Multiple Properties. The simplest combiner for collision-resistance simply concatenates the outputs of both hash functions $\mathsf{Comb}_{||}(M) = H_0(M)||H_1(M)$. Obviously, the combiner is collision-resistant as long es either H_0 or H_1 has this property. Yet, it does not guarantee for example pseudorandomness (assuming that the hash functions are keyed) if only one of the underlying hash functions is pseudorandom. An

[1]Technically, they require *statistical* collision-resistance for the keys of the pseudorandom function.

adversary can immediately distinguish the concatenated output from a truly random value by simply examining the part of the insecure hash function.

An obvious approach to obtain a hash combiner that is robust for pseudorandomness is to set $\mathsf{Comb}_{\oplus}(M) = H_0(M) \oplus H_1(M)$. However, this combiner is not known to preserve collision-resistance anymore, since a collision for the combiner does not necessarily require collisions on both hash functions. In fact, this combiner also violates the conditions of [11, 28, 29] and [12], who have shown that the output of a (black-box) collision-resistant combiner cannot be significantly shorter than the concatenation of the outputs from all employed hash functions. Thus, already the attempt of combining only two properties in a robust manner indicates that finding a robust multi-property combiner is far from trivial.

Our Result. In this work we show how to build combiners that provably preserve multiple properties in a robust manner. We concentrate on the most common properties as proposed in [5], namely, collision-resistance (CR), target collision-resistance (TCR), pseudorandomness (PRF), message authentication (MAC), one-wayness (OW) and indifferentiability from a random oracle (IRO).

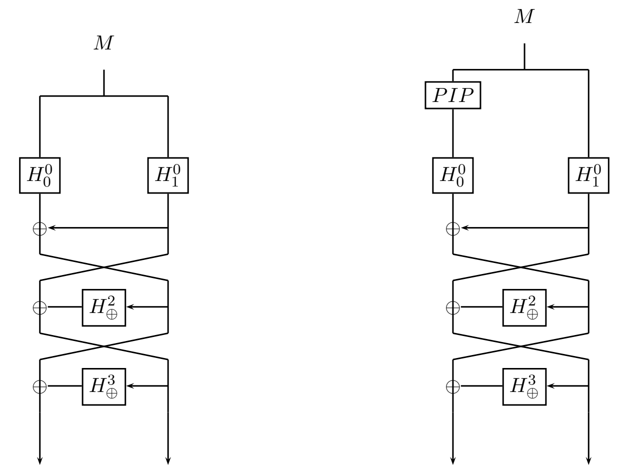

Figure 1: Illustration of the basic construction $\mathsf{Comb}_{4\mathsf{P}}$ (left) preserving CR, PRF, TCR and MAC. Here $H_b^i(\cdot)$ denotes $H_b(\langle i\rangle_2 \,\|\cdot)$ where $\langle i\rangle_2$ is the binary representation of the integer i with two bits. $H_\oplus^i(\cdot)$ denotes $H_0^i(\cdot)\oplus H_1^i(\cdot)$. By applying a pairwise independent permutation (PIP) to the input of H_0^0 we get our construction $\mathsf{Comb}_{4\mathsf{P}\&\mathsf{OW}}$ (right), which also preserves OW. Because of the PIP, the input length of the construction must now be fixed.

The Combiner $\mathsf{Comb}_{4\mathsf{P}}$. Our first construction is a combiner $\mathsf{Comb}_{4\mathsf{P}}$ which robustly preserves the four properties collision-resistance, target collision-resistance, pseudorandomness and message authentication. If the underlying hash functions have output length n bits, the combiner has output length $2n$, which basically matches known lower bounds for combiners which preserve collision-resistance only [11, 28, 29]. The idea for

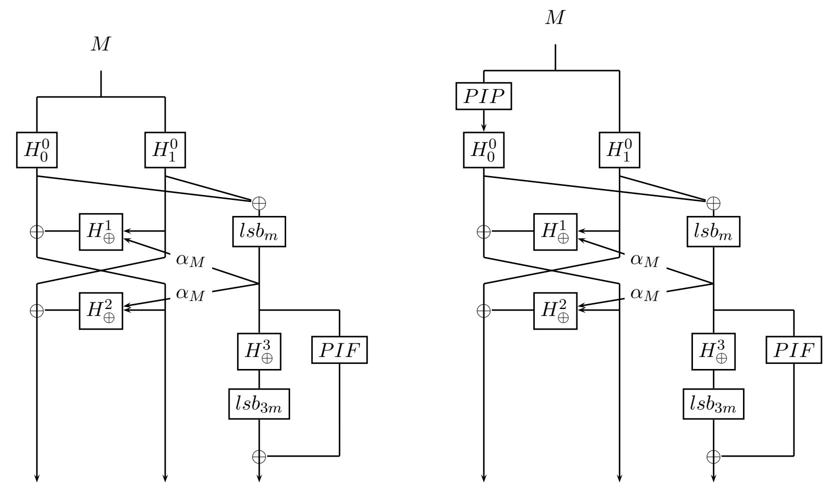

Figure 2: Illustration of the construction $\mathsf{Comb}_{\mathsf{4P\&IRO}}$ (left), which (besides the four properties preserved by $\mathsf{Comb}_{\mathsf{4P}}$) also preserves the IRO property, at the prize of an increased output length. The third branch of the construction operates on a signature value α_M depending on input M and applies a pairwise independent function. On the right side the construction $\mathsf{Comb}_{\mathsf{6P}}$ is illustrated which simultaneously preserves all six properties considered.

our combiner is to use the concatenation combiner $\mathsf{Comb}_{||}$, followed by a three-round Feistel permutation. In the first round of the Feistel permutation no round function is applied, whereas the two subsequent rounds are constructed by using the XOR-combiner $\mathsf{Comb}_{\oplus}$ (cf. Figure 1). The round functions are made somewhat independent by prepending the round number to the input.

The rationale here is that applying the Feistel (or any other) permutation to the output of $\mathsf{Comb}_{||}$ still preserves the CR, TCR and MAC properties, e.g., collisions for $\mathsf{Comb}_{||}$ are pulled through the downstream permutation and can be traced back to collisions for $\mathsf{Comb}_{||}$. At the same time, one achieves robustness for the PRF property. The latter can be seen as follows: if either H_0 or H_1 is pseudorandom, then the round functions in the Feistel network are pseudorandom as $\mathsf{Comb}_{\oplus}$ is a secure combiner for pseudorandom functions. The Luby-Rackoff [24] result now states that a three-round Feistel-network, instantiated with pseudorandom functions, is a pseudorandom permutation. We note that the formal argument also needs to take into account that finding collisions in the keyed version of the initial $\mathsf{Comb}_{||}$ computation is infeasible.

Our $\mathsf{Comb}_{\mathsf{4P}}$ combiner was recently implemented in an open source project [1].

Preserving IRO. In Section 4 we modify the $\mathsf{Comb}_{\mathsf{4P}}$ construction such that it also preserves indifferentiability from a random oracle. The obstruction of the IRO robustness in the $\mathsf{Comb}_{\mathsf{4P}}$ combiner stems from the invertibility of the Feistel permutation: an adversary trying to distinguish the output of the combiner from a random function (given access to the underlying hash functions, as opposed to the case of pseudorandom functions for example) can partly "reverse engineer" images under the combiner. Hence, we introduce a "signature" value α_M (depending on the input message M), entering the round functions in the Feistel network and basically allowing combiner computations in the forward direction only.

The description of our enhanced combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}$ is given in Figure 2. The signature α_M is taken as (a prefix of) the XOR of the output halves of the $\mathsf{Comb}_{||}$ combiner and is used as additional input parameter in the Feistel round functions, allowing us to also save one round of the Feistel structure. Note that this essentially means that different Feistel permutations may be used for different inputs M, M', because the signatures $\alpha_M, \alpha_{M'}$ may be distinct. In order to apply again the argument that the Feistel permutation does not interfere with the CR,TCR and MAC robustness of the concatenating combiner, we therefore also need to ensure that finding "bad" pairs α_M and $\alpha_{M'}$ is infeasible. To this end we introduce another output branch which basically guarantees collision-resistance of the signatures. This additional output is of length $3m$ for some $m = \omega(\log n)$, yielding an overall output length of $2n + \omega(\log n)$.

Preserving One-Wayness. Even though both our solutions are robust for an important set of properties they are not good combiners for one-wayness. Our results so far merely show that they are one-way functions making for example the potentially stronger assumption that one of the two hash functions is collision-resistant. In Section 5 we therefore show how to augment our constructions such that also the one-wayness property is preserved.

By apply a pairwise-independent permutation (PIP) to the input of H_0 (or H_1) in the concatenation combiner $\mathsf{Comb}_{||}$, we get a construction which still is a combiner for collision-resistance, but now also combines the one-wayness property. Using this extended concatenation combiner in the initial stages of our previous constructions we additionally achieve robustness for one-wayness. As the description length of a PIP is linear in its input length, now the input length of the combiner must be a priory fixed.

Weak vs. Strong Robustness. We call a multi-property combiner *strongly* robust for a set of properties, if the combined function satisfies every property which is satisfied by at least one hash function, i.e. if H_0 or H_1 has property MAC, then so does the combined function, independently of the other properties. All our constructions achieve this strongest notion. We also define weaker notions of multi-property robustness (MPR), which we denote by weak MPR and mild MPR. In the weak case the combiner only inherits a set of multiple properties if they are all provided by at least one hash function (i.e., if there is a strong candidate which has all properties at the same time). Mild MPR combiners are between strong MPR and weak MPR combiners, here we also require that all properties hold, but different hash functions may cover different properties.

We then address several questions related to the different notions of multi-property robustness. In particular, we show that strong MPR is strictly stronger than mild MPR which is strictly stronger than weak MPR. We finally discuss the case of general tree-based combiners for more than two hash functions built out of combiners for two hash functions, as suggested in a more general setting by Harnik et al. [18]. As part of this result we show that such tree-combiners inherit the weakly and strongly MPR property of two-function combiners, whereas mildly MPR two-function combiners surprisingly do not propagate their security to trees.

Organization. We start by defining the three notions of robust multi-property combiners and give definitions of the security properties considered in Section 2. In Section 3 we present the construction of our most efficient MPR combiner that is robust for CR, TCR, PRF and MAC according to our strongest notion. A combiner which additionally preserves the IRO property, slightly increasing the output length and computational costs, is then discussed in Section 4. In Section 5 we show that a twist on our combiners also makes them robust for one-wayness (but at the price of a fixed input length). In Section 6 we prove separations for the different notions of multi-property robustness. We address the problem of composing more than just two hash functions in Section 7.

2 Preliminaries

We denote by $\{0,1\}^n$ the set of bit-strings x of length $|x| = n$, and 1^n stands for n in unary encoding, i.e., the string that consist of n ones. For two strings x, y we write $x||y$ for the concatenation and $x \oplus y$ for the bitwise exclusive-or of x and y. For the latter we assume that x and y have equal length.

An *adversary* $\mathcal{A}$ is a probabilistic algorithm. We write $\mathcal{A}^{\mathcal{O}}(y)$ for an adversary that runs on input y and has oracle access to $\mathcal{O}$. The shorthand $x \leftarrow X$ denotes that x is sampled from the random variable X. Similarly we write $x \leftarrow \mathcal{A}(y)$ for the output of $\mathcal{A}$ for input y. We say an adversary is *efficient* if it runs in polynomial-time. That is, if there exists a polynomial $p(n)$ such that $\mathcal{A}$ takes at most $p(n)$ steps where n is the length of the input.

2.1 Hash Functions and Their Properties

A hash function $\mathcal{H} = (\mathsf{HKGen}, \mathsf{H})$ is a pair of efficient algorithms such that HKGen for input 1^n returns (the description of) a hash function H (which contains 1^n), and H for input H and $M \in \{0,1\}^*$ deterministically outputs a digest $H(M)$. We often identify the hash function with its digest values $H(\cdot)$ if the key generation algorithm is clear from the context.

In this work we consider the following six important security properties for hash functions (cf. [5]): the unkeyed properties of (target) collision-resistance and one-wayness and the keyed properties of being a pseudorandom function or a message authentication code. The final property – indifferentiability from a random oracle – is special, as one considers idealized components. In particular, there is no efficient key-generation algorithm, but rather the hash function is given directly by an oracle.

Depending on the security property we are interested in, the access of the adversary to the hash function is modeled differently. For unkeyed primitives, the description of H is given to the adversary. Whereas for keyed primitives the adversary only gets black-box access to the hash function. We could also consider a somewhat more general notion, where the key-generation algorithm outputs a pair H^p, H^s of values, which together define the hash function H, and where in the keyed setting, only H^s (but not H^p) is kept secret. For example in the HMAC construction, H^p would define the underlying compression function, and the secret key H^s would be the randomly chosen initial value IV. All our results also hold in this setting, but we avoid using such a fine-grained definition as to save on notation which would only distract from the main ideas.

collision resistance (CR): Informally, collision-resistance of a hash function H requires that it should be infeasible to find two distinct messages $M \neq M'$ that map under H to the same value $H(M) = H(M')$. For the formal treatment we consider families of hash functions and call a hash function *collision-resistant* if for any efficient adversary $\mathcal{A}$ the advantage

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{cr}}_{\mathcal{A}}(n) = \\ &\qquad \mathrm{Prob}\left[H \leftarrow \mathsf{HKGen}(1^n); (M, M') \leftarrow \mathcal{A}(H) : M \neq M' \wedge H(M) = H(M')\right]\end{aligned}$$

is negligible (as a function of n).

target collision-resistance (TCR): Target collision-resistance is a weaker security notion than collision-resistance which obliges the adversary to first commit to a target message M before getting the description $H \leftarrow \mathsf{HKGen}(1^n)$ of the hash function. For the given H the adversary must then find a second message $M' \neq M$ such that $H(M) = H(M')$.

More formally, a hash function is *target collision-resistant* if for any efficient adversary $\mathcal{A} = (\mathcal{A}^1, \mathcal{A}^2)$ the following advantage is negligible in n:

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{tcr}}_{\mathcal{A}}(n) = \\ &\qquad \mathrm{Prob}\left[\begin{array}{rcc}(M, \mathsf{st}) \leftarrow \mathcal{A}^1(1^n); H \leftarrow \mathsf{HKGen}(1^n); & & M \neq M' \wedge \\ M' \leftarrow \mathcal{A}^2(H, M, \mathsf{st}) & : & H(M) = H(M')\end{array}\right].\end{aligned}$$

one-wayness (OW) : The definition of one-wayness intuitively requires that it is infeasible to determine the preimage of a hash value. A hash function is called *one-way*, if for any efficient algorithm $\mathcal{A}$ the advantage

$$\begin{aligned}&\mathbf{Adv}^{\mathrm{owf}}_{\mathcal{A}}(n) = \\ &\quad \mathrm{Prob}\left[H \leftarrow \mathsf{HKGen}(1^n); M \leftarrow \{0,1\}^*; M' \leftarrow \mathcal{A}(H, H(M)) : H(M') = H(M)\right]\end{aligned}$$

is negligible in n.

pseudorandomness (PRF): A hash function can be used as a pseudorandom function if, e.g., the initial value IV is replaced by a randomly chosen key K of the same size. We capture such a keyed setting by granting the adversary only black-box access to the (randomly chosen) hash function $H(\cdot)$. The hash function is then called *pseudorandom*, if no efficient adversary can distinguish H from a uniformly random function f (with the same range and same domain) with noticeable advantage. More formally, we require that for any efficient adversary $\mathcal{A}$ the advantage

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{prf}}(n) = \left|\mathrm{Prob}\left[\mathcal{A}^{H(\cdot)}(1^n) = 1\right] - \mathrm{Prob}\left[\mathcal{A}^{f}(1^n) = 1\right]\right|$$

is negligible, where the probability in the first case is over $\mathcal{A}$'s coin tosses and the choice of $H \leftarrow \mathsf{HKGen}(1^n)$, and in the second case over $\mathcal{A}$'s coin tosses and the choice of the random function $f : \{0,1\}^* \to \{0,1\}^n$.

message authentication (MAC): A message authentication code is a symmetric primitive which allows a sender and receiver, both sharing a secret, to exchange information in an authenticated manner. When a hash function is used as a MAC, the description $H \leftarrow \mathsf{HKGen}(1^n)$ constitutes the shared secret, and the sender augments a message M by the tag $\tau \leftarrow H(M)$. The receiver of (M, τ) then verifies whether $\tau = H(M)$ holds.

A MAC is considered secure, if it is unforgeable under chosen message attacks, i.e., an adversary after adaptively learning several tags $(M_1, \tau_1), (M_2, \tau_2), \ldots, (M_q, \tau_q)$ should not be able to compute a forgery for a fresh message M^*. Note that the adversary has again only oracle access to $H(\cdot)$. More compactly, a hash function is called a *secure MAC*, if for any efficient adversary $\mathcal{A}$ the following advantage is negligible in n

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{mac}}(n) = \\ \mathrm{Prob}\left[H \leftarrow \mathsf{HKGen}(1^n), (M, \tau) \leftarrow \mathcal{A}^{H(\cdot)} : H(M) = \tau \wedge M \text{ not queried}\right].$$

indifferentiability from random oracles (IRO): Indifferentiability [25] is a generalization of indistinguishability allowing to consider random oracles that are used as a public component. More formally, a hash function H^f based on a random oracle f is *indifferentiable* from a random oracle $\mathcal{F}$ if for any efficient adversary $\mathcal{A}$ there exists an efficient algorithm $\mathcal{S}$ such that the advantage

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{ind}}(n) = \left|\mathrm{Prob}\left[\mathcal{A}^{H^f, f}(1^n) = 1\right] - \mathrm{Prob}\left[\mathcal{A}^{\mathcal{F}, \mathcal{S}^{\mathcal{F}}}(1^n) = 1\right]\right|$$

is negligible in n, where the probability in the first case is over $\mathcal{A}$'s coin tosses and the choice of the random function f, and in the second case over the coin tosses of $\mathcal{A}$ and $\mathcal{S}$, and over the choice of $\mathcal{F}$.

Thus, the goal of the simulator $\mathcal{S}^{\mathcal{F}}$ is to mimic the ideal compression function f, such that no adversary $\mathcal{A}$ can decide whether its interacting with H^f and f or with $\mathcal{F}$ and $\mathcal{S}^{\mathcal{F}}$. To this end, $\mathcal{S}^{\mathcal{F}}$ has to produce output that is random but consistent with the values the adversary can obtain from the random oracle $\mathcal{F}$. Note that the simulator has oracle access to $\mathcal{F}$ too, but it does *not* get to see the queries $\mathcal{A}$ issues to $\mathcal{F}$.

2.2 Robust Multi-Property Hash Combiners

A hash function combiner $\mathcal{C} = (\mathsf{CKGen}, \mathsf{Comb})$ for some security property P is a pair of algorithms which, when instantiated with two hash functions $\mathcal{H}_0, \mathcal{H}_1$, itself implements a hash function, such that the combined function satisfies P if at least one of the two candidates satisfies P.

For multiple properties $\textsc{prop} = \{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_N\}$ one can either demand that the combiner inherits the properties if one of the candidate hash functions is strong and has all the properties (weakly robust), or that for each property at least one of the two hash functions has the property (strongly robust). We also consider a notion in between but somewhat closer to the weak case, called mildly robust, in which case all properties from $\textsc{prop}$ must hold, albeit different functions may cover different properties (instead of one function as in the case of weakly robust combiners).[2] In the following, we denote by $\textsc{prop}(\mathcal{H}) \subseteq \textsc{prop}$ for a set $\textsc{prop} = \{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_N\}$ the properties which a hash function $\mathcal{H}$ has.

More formally,

Definition 2.1 (Multi-Property Robustness) *For a set* $\textsc{prop} = \{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_N\}$ *of properties a hash function combiner* $\mathcal{C} = (\mathsf{CKGen}, \mathsf{Comb})$ *for hash functions* $\mathcal{H}_0, \mathcal{H}_1$ *is called:*

weakly multi-property robust *(wMPR) for* $\textsc{prop}$ *iff*

$$\textsc{prop} = \textsc{prop}(\mathcal{H}_0) \text{ or } \textsc{prop} = \textsc{prop}(\mathcal{H}_1) \quad \Longrightarrow \quad \textsc{prop} = \textsc{prop}(\mathcal{C}),$$

mildly multi-property robust *(mMPR) for* $\textsc{prop}$ *iff*

$$\textsc{prop} = \textsc{prop}(\mathcal{H}_0) \cup \textsc{prop}(\mathcal{H}_1) \quad \Longrightarrow \quad \textsc{prop} = \textsc{prop}(\mathcal{C}),$$

strongly multi-property robust *(sMPR) for* $\textsc{prop}$ *iff for all* $\mathsf{P}_i \in \textsc{prop}$,

$$\mathsf{P}_i \in \textsc{prop}(\mathcal{H}_0) \cup \textsc{prop}(\mathcal{H}_1) \quad \Longrightarrow \quad \mathsf{P}_i \in \textsc{prop}(\mathcal{C}).$$

We remark that for weak and mild robustness all individual properties $\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_N$ from $\textsc{prop}$ are guaranteed to hold, either by a single function as in weak robustness, or possibly by different functions as in mild robustness. The combiner may therefore depend on some strong property $\mathsf{P}_i \in \textsc{prop}$ which one of the hash functions has, and which helps to implement some other property P_j in the combined hash function. But then, for a subset $\textsc{prop}' \subseteq \textsc{prop}$ which, for instance, misses this strong property P_i, the combiner may no longer preserve the properties $\textsc{prop}'$. This is in contrast to strongly robust combiners which support such subsets of properties by definition.

Note that for a singleton $\textsc{prop} = \{\mathsf{P}\}$ all notions coincide and we simply say that $\mathcal{C}$ is P-robust in this case. However, for two or more properties the notions become strictly stronger from weak to mild to strong, as we show in Section 6. Finally, we remark that our definition allows the case $\mathcal{H}_0 = \mathcal{H}_1$, which may require some care when designing combiners, especially if the hash functions are based on random oracles.

[2]One may also refine these notions further. We focus on these three "natural" cases.

3 The $\mathcal{C}_{4\mathsf{P}}$ Combiner for CR, PRF, TCR and MAC

In this section we introduce the construction of our basic combiner $\mathcal{C}_{4\mathsf{P}}$ as illustrated in Figure 1. Recall that the idea of this combiner is to apply a Feistel permutation (with quasi-independent round functions given by the XOR combiner) to the concatenating combiner to ensure CR, PRF, TCR and MAC robustness.

3.1 Our Construction

The three-round Feistel permutation P^3 over $\{0,1\}^{2n}$ is given by the round functions $H_\oplus^i(\cdot) = H_0^i(\cdot) \oplus H_1^i(\cdot)$ for $i = 2, 3$, with $H_b^i(\cdot)$ denoting the function $H_b(\langle i \rangle_2 \,||\cdot)$ where $\langle i \rangle_2$ is the binary representation of the integer i with two bits. The first round function is the identity function, which we denote for notational convenience as $H_\oplus^1(X) = X$. In the i-th round the input (L_i, R_i) is mapped to the output $(R_i, L_i \oplus H_\oplus^i(R_i))$. We occasionally denote this Feistel permutation more explicitly by $P^3 = \psi[H_\oplus^1, H_\oplus^2, H_\oplus^3](\cdot)$.

Our combiner, instantiated with hash functions $\mathcal{H}_0, \mathcal{H}_1$, is a pair of efficient algorithms $\mathcal{C}_{4\mathsf{P}} = (\mathsf{CKGen}_{4\mathsf{P}}, \mathsf{Comb}_{4\mathsf{P}})$ where the key generation algorithm $\mathsf{CKGen}_{4\mathsf{P}}(1^n)$ samples $H_0 \leftarrow \mathsf{HKGen}_0(1^n)$ and $H_1 \leftarrow \mathsf{HKGen}_1(1^n)$. The evaluation algorithm $\mathsf{Comb}_{4\mathsf{P}}^{H_0,H_1}$ for parameters H_0, H_1 and input message M outputs

$$\mathsf{Comb}_{4\mathsf{P}}^{H_0,H_1}(M) = P^3(H_0^0(M)||H_1^0(M)).$$

3.2 Multi-Property Robustness

We next show that the construction satisfies the strongest notion for robust multi-property combiners:

Theorem 3.1 *The combiner $\mathcal{C}_{4\mathsf{P}}$ is a strongly robust multi-property combiner for* $\textsc{prop} = \{\mathsf{CR}, \mathsf{PRF}, \mathsf{TCR}, \mathsf{MAC}\}$.

Recall that a strongly robust multi-property combiner inherits all properties that are provided by at least one of the underlying hash functions. Thus, we have to prove that each property CR, PRF, TCR and MAC is preserved independently.

Lemma 3.2 *The combiner $\mathcal{C}_{4\mathsf{P}}$ is CR-robust.*

Proof. Observe that any collision $M \neq M'$ for $\mathsf{Comb}_{4\mathsf{P}}^{H_0,H_1}(\cdot)$ directly gives a collision $00\|M \neq 00\|M'$ for $H_0(\cdot)$ and $H_1(\cdot)$. Thus any adversary that finds collisions for $\mathsf{Comb}_{4\mathsf{P}}$ when instantiated with H_0, H_1 with non-negligible probability, can be used to find collision (with the same probability) for H_0 and H_1 respectively: to find a collision for $H_b \leftarrow \mathsf{HKGen}_b(1^n)$ with $b \in \{0,1\}$, run $H_{\overline{b}} \leftarrow \mathsf{HKGen}_{\overline{b}}(1^n)$ and then invoke the adversary on input $H_b, H_{\overline{b}}$. If the adversary outputs a collision for $\mathsf{Comb}_{4\mathsf{P}}^{H_0,H_1}(\cdot)$, this is also a collision for $H_b(\cdot)$. □

Lemma 3.3 *The combiner $\mathcal{C}_{4\mathsf{P}}$ is TCR-robust.*

Proof. Assume towards contradiction that there exist an efficient adversary $\mathcal{A}_{\mathsf{Comb}} = (\mathcal{A}^1_{\mathsf{Comb}}, \mathcal{A}^2_{\mathsf{Comb}})$ that commits to a message M before getting H_0 and H_1 and then finds some M' such that $\mathsf{Comb}^{H_0,H_1}_{\mathsf{4P}}(M) = \mathsf{Comb}^{H_0,H_1}_{\mathsf{4P}}(M')$ with noticeable probability. Then we can use this attacker to construct a successful target-collision adversary $\mathcal{A}_b = (\mathcal{A}^1_b, \mathcal{A}^2_b)$ against the underlying hash functions H_b for $b \in \{0,1\}$ which contradicts the assumption that at least one of the two hash functions is target collision-resistant.

First, the adversary $\mathcal{A}^1_b(1^n)$ runs $\mathcal{A}^1_{\mathsf{Comb}}(1^n)$ to receive the target message M and some state information st. $\mathcal{A}^1_b$ then commits to $00||M$. On input H_b the adversary $\mathcal{A}^2_b$ samples the second hash function $H_{\overline{b}} \leftarrow \mathsf{HKGen}_{\overline{b}}(1^n)$ and passes $H_b, H_{\overline{b}}$ together with (M, st) to $\mathcal{A}^2_{\mathsf{Comb}}$. When $\mathcal{A}^2_{\mathsf{Comb}}$ outputs a message $M' \neq M$ with $\mathsf{Comb}^{H_0,H_1}_{\mathsf{4P}}(M) = \mathsf{Comb}^{H_0,H_1}_{\mathsf{4P}}(M')$ the adversary $\mathcal{A}^2_b$ returns $00||M'$.

Since $P^3(\cdot)$ is an invertible permutation, a collision of M, M' for the combiner can be traced back to the input of $P^3(\cdot)$ and thus we have

$$H_0(00||M)||H_1(00||M) = H_0(00||M')||H_1(00||M').$$

Hence, both adversaries $\mathcal{A}_b$ for $b = 0,1$ succeed in finding a message $00||M'$ that together with the target message $00||M$ leads to a collision under H_b with the same noticeable probability as $\mathcal{A}_{\mathsf{Comb}}$. $\square$

Lemma 3.4 *The combiner $\mathcal{C}_{\mathsf{4P}}$ is PRF-robust.*

Proof. As the XOR combiner is a good combiner for pseudorandom functions (PRFs), the round functions $H^2_\oplus, H^3_\oplus$ in the Feistel network are instantiated with PRFs, as long as at least H_0 or H_1 is a PRF. Prepending the unique prefix $\langle i \rangle_2$ for $i = 2,3$ to the input of $H^i_\oplus(\cdot) = H_\oplus(\langle i \rangle_2 \| \cdot)$ in each round ensures that the functions in different rounds are never invoked on the same input, which means they are indistinguishable from two independent random functions. The first round of our Feistel permutation, that does not apply a round function, simply prepares the input for the second round function $H^2_\oplus(\cdot)$ by xoring both input halves $H^0_0(M) \oplus H^0_1(M)$. Thus, if at least one hash function is a PRF then the input to the second round function is already a pseudorandom value, which prevents an adversary from directly choosing the inputs to the second Feistel round.

We can now apply the results due to Luby-Rackoff [24] and Naor-Reingold [26] which state that a two-round Feistel-network invoked on an unpredictable input and instantiated with independent pseudorandom functions is a pseudorandom permutation (PRP).

Further, if either H_0 or H_1 is a PRF, then the initial concatenation combiner $\mathsf{Comb}^{H_0,H_1}_{||}$ is weakly collision-resistant[3], thus the probability that the adversary will invoke the combiner on distinct inputs M, M' where a collision $H^0_0(M) \| H^0_1(M) = H^0_0(M') \| H^0_1(M')$ occurs, is negligible. So with overwhelming probability, all the adversary sees is the output of a PRP on distinct inputs. This distribution is indistinguishable

[3]Weak collision-resistance is defined similarly to collision resistance, except that here the function is keyed and the key is secret, i.e., the adversary only gets black-box access to the function.

from uniformly random (this follows from the PRP/PRF switching lemma [8]), thus $\mathcal{C}_{\mathsf{4P}}$ is PRF robust.

More precisely, from any adversary $\mathcal{A}_{\mathsf{Comb}}$ who has advantage ϵ in distinguishing $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}$ making q queries, we can construct an attacker $\mathcal{A}_b$ for $b \in \{0,1\}$, that distinguishes $H_b \leftarrow \mathsf{HKGen}_b(1^n)$ from random with advantage $\epsilon - \mathcal{O}(q^2/2^n)$. For $b = 0$ (the case $b = 1$ is symmetric) the adversary $\mathcal{A}_0$ first samples $H_1 \leftarrow \mathsf{HKGen}_1(1^n)$ and then simulates the experiment of $\mathcal{A}_{\mathsf{Comb}}$ using this knowledge of H_1 and its oracle access to H_0. Finally, $\mathcal{A}_0$ returns the output of $\mathcal{A}_{\mathsf{Comb}}$. If H_0 is a uniformly random function $f : \{0,1\}^* \to \{0,1\}^n$, then any (even computationally unbounded) adversary making q queries has advantage at most $\mathcal{O}(q^2/2^n)$ in distinguishing $\mathsf{Comb}_{\mathsf{4P}}^{f,H_1}$ from a random function (as the advantage from the PRP/PRF switching lemma and the advantage in the Luby-Rackoff result are both $\mathcal{O}(q^2/2^n)$). Thus, if $\mathcal{A}_{\mathsf{Comb}}$ distinguishes $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}$ from a truly random function $F : \{0,1\}^* \to \{0,1\}^{2n}$ with advantage ϵ, it has advantage $\epsilon - \mathcal{O}(q^2/2^n)$ to distinguish $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}$ from $\mathsf{Comb}_{\mathsf{4P}}^{f,H_1}$. The latter is by definition also $\mathcal{A}_0$'s advantage for f and H_0.

□

Lemma 3.5 *The combiner $\mathcal{C}_{\mathsf{4P}}$ is MAC-robust.*

Proof. Assume towards contradiction that an adversary $\mathcal{A}_{\mathsf{Comb}}$ with oracle access to the combiner $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}(\cdot)$ finds with non-negligible probability a valid pair (M,τ), such that $\tau = \mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}(M)$ but the message M was never queried to the MAC-oracle. Given $\mathcal{A}_{\mathsf{Comb}}$ we can construct a successful adversary $\mathcal{A}_b$ against the underlying hash function H_b for $b \in \{0,1\}$. To forge $H_b(\cdot)$, the adversary $\mathcal{A}_b$ first samples $H_{\overline{b}} \leftarrow \mathsf{HKGen}_{\overline{b}}(1^n)$, and then lets $\mathcal{A}_{\mathsf{Comb}}$ attack $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}(\cdot)$, and let $\mathcal{A}_b$ use his oracle access to $H_b(\cdot)$ and the knowledge of $H_{\overline{b}}$ to compute the answers to $\mathcal{A}_{\mathsf{Comb}}$'s oracle queries. When finally $\mathcal{A}_{\mathsf{Comb}}$ outputs (M,τ), the adversary $\mathcal{A}_b$ computes its forgery $(00||M, \tau_b)$ by inverting the permutation $P^3 = \psi[H_\oplus^1, H_\oplus^2, H_\oplus^3]$ (recall that $H_\oplus^i(\cdot) = H_0(\langle i\rangle_2\,||\cdot) \oplus H_1(\langle i\rangle_2\,||\cdot)$ for $i = 2,3$ and that the required hash function evaluations can be made with the help of the MAC oracle):

$$\tau_0||\tau_1 = P^{3^{-1}}(\tau).$$

The adversary $\mathcal{A}_b$ then outputs the message $00||M$ and τ_b. If M was not previously queried by $\mathcal{A}_{\mathcal{C}}$, then $00||M$ is distinct from all of $\mathcal{A}_b$'s previous queries, because all additional queries are prepended by $\langle i\rangle_2$ where $i \in \{2,3\}$. By construction, if (M,τ) is a valid forgery for $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}(\cdot)$, then $H_0^0(M)||H_1^0(M) = \tau_0||\tau_1$ and thus $(00\|M, \tau_b)$ is a valid forgery for $H_b(\cdot)$. □

4 Preserving Indifferentiability: the $\mathcal{C}_{\mathsf{4P\&IRO}}$ Combiner

First, we give a brief idea why our $\mathcal{C}_{\mathsf{4P}}$ combiner does not guarantee the IRO property. To be IRO-robust the combiner has to be indifferentiable from a random oracle for any efficient adversary $\mathcal{A}$, if H_b is a random oracle for some $b \in \{0,1\}$. Thereby the adversary

$\mathcal{A}$ has oracle access either to the combiner $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}$ and the random oracle H_b, or to $\mathcal{F}$ and a simulator $\mathcal{S}^{\mathcal{F}}$. The simulator's goal is to mimic H_b such that $\mathcal{A}$ cannot have a significant advantage on deciding whether its interacting with $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}$ and H_b, or with $\mathcal{F}$ and $S^{\mathcal{F}}$.

Usually, the strategy for designing such a simulator is to check if a query is a potential attempt of $\mathcal{A}$ to imitate the construction of the combiner and then to precompute further answers that are consistent with the information $\mathcal{A}$ can get from $\mathcal{F}$. However, for $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}$ the simulator may be unable to precompute those consistent values, because an adversary $\mathcal{A}$ can compute the permutation part of the combiner backwards such that $\mathcal{S}^{\mathcal{F}}$ has to commit to its round values used in the permutation P^3 before knowing the initial input M. To this end, $\mathcal{A}$ first queries the random oracle $\mathcal{F}$ on input M and uses the response $Y \leftarrow \mathcal{F}(M)$ to compute $X = P^{3^{-1}}(Y)$ with the help of $\mathcal{S}^{\mathcal{F}}$ simulating H_b and the function $H_{\overline{b}}$ which is accessible in a black-box manner. Then the answers of $\mathcal{S}^{\mathcal{F}}$, in order to be indistinguishable from those of H_b, must lead to a value $X = S(00||M)||H_1(00||M)$ if $b = 0$, and $X = H_0(00||M)||S(00||M)$ else.

While the part of X corresponding to $S(00||M)$ can simply be set as response to a further query $00||M$ by the simulator, the part of $H_{\overline{b}}(00||M)$ is determined by the oracle $H_{\overline{b}}(\cdot)$ and the message M. However, since the simulator does not know the message M when answering $\mathcal{A}$'s queries for computing $P^{3^{-1}}$, it is not able to call the $H_{\overline{b}}$ oracle about $00||M$ and to choose those answers accordingly. Thus, the probability that the responses provided by $\mathcal{S}^{\mathcal{F}}$ will lead in $P^{3^{-1}}(Y)$ to a value that is consistent with the structure of the combiner, is negligible and the adversary $\mathcal{A}$ can distinguish between $(\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}, H_b)$ and $(\mathcal{F}, \mathcal{S}^{\mathcal{F}})$ with noticeable probability.

In order to guarantee the IRO property, we modify the $\mathsf{Comb}_{\mathsf{4P}}^{H_0,H_1}$ combiner such that the adversary is forced to query the message M before he can create meaningful queries aiming to imitate the construction. By this the simulator becomes able to switch to the common strategy of preparing consistent answers in advance. As explained in the introduction, adding a signature value α_M into the computation does the job.

4.1 The Combiner $\mathcal{C}_{\mathsf{4P\&IRO}}$

In this section we consider the modified combiner $\mathcal{C}_{\mathsf{4P\&IRO}}$ as illustrated in Figure 2. The combiner $\mathcal{C}_{\mathsf{4P\&IRO}} = (\mathsf{CKGen}_{\mathsf{4P\&IRO}}, \mathsf{Comb}_{\mathsf{4P\&IRO}})$ is defined as follows: $\mathsf{CKGen}_{\mathsf{4P\&IRO}}$ first samples $H_0 \leftarrow \mathsf{HKGen}_0(1^n), H_1 \leftarrow \mathsf{HKGen}_1(1^n)$ and a pairwise independent function $g : \{0,1\}^m \rightarrow \{0,1\}^{3m}$ for some $m \leq n/3$ (the larger m, the better the security level, but the longer the output, too):

Definition 4.1 (Pairwise-Independent Function/Permutation) *A family of functions $G : A \rightarrow B$ from domain A to range B is called pairwise independent iff for all $x \neq x' \in A$ and $z \neq z' \in B$ we have* $\mathrm{Prob}_{g \in G}[g(x) = z \wedge g(x') = z'] = |B|^{-2}$.

A family of function $\Pi : A \rightarrow A$ is a pairwise independent permutation, if for $x \neq x'$ and $z \neq z' \in A$ we have $\mathrm{Prob}_{g \in G}[g(x) = z \wedge g(x') = z'] = \frac{1}{|B|(|B|-1)}$.

One gets a simple construction of a pairwise independent function (PIF) mapping $\{0,1\}^n$ to $\{0,1\}^n$, by sampling $a, b \in \{0,1\}^n$ at random, which then defines the function

$g_{(a,b)}(x) = (ax + b)$, where addition and multiplication are in the field $GF(2^n)$. For a smaller range $\{0,1\}^m$ with $m < n$, one can simply drop $n - m$ bits of the output. This construction is also a pairwise-independent *permutation* (PIP), if a is chosen at random from $\{0,1\}^n \setminus 0^n$ (instead of $\{0,1\}^n$).

The evaluation algorithm $\mathsf{Comb}^{H_0,H_1,g}_{\mathsf{4P\&IRO}}(M)$ first computes $\mathsf{Comb}^{H_0,H_1}_{||}(M) = H^0_0(M)|| H^0_1(M)$ and a value α_M – which we call the "signature of M" – as $\alpha_M = lsb_m(H^0_\oplus(M))$ where $H^0_\oplus(M) = H^0_0(M) \oplus H^0_1(M)$ and $lsb_a(x)$ denotes the a least significant bits of x. The value α_M is used as an extra prefix in the round functions of the two-round Feistel permutation $P^2_\alpha(\cdot) = \psi[H^1_\oplus(\alpha_M\|\cdot), H^2_\oplus(\alpha_M\|\cdot)]$. Applying P^2_α on $H^0_0(M)||H^0_1(M)$ then gives the first part of the combiners output.

The construction as described so far, is already a robust combiner for IRO and PRF, but not for CR and TCR. The reason is that now distinct input messages M, M' where $\alpha_M \neq \alpha_{M'}$ lead to distinct Feistel permutations $P^2_{\alpha_M} \neq P^2_{\alpha_{M'}}$, and thus we cannot compute a collision for Comb^{H_0,H_1} (and thus for H_0 and H_1) from a collision $\mathsf{Comb}^{H_0,H_1}_{||}(P^2_{\alpha_M}(M)) = \mathsf{Comb}^{H_0,H_1}_{||}(P^2_{\alpha'_M}(M'))$.

To solve this problem, we could append the signature to the output of the combiner, which would enforce that two inputs can only collide if they have the same signature. Unfortunately, outputting the signature α directly would make the permutation P^2_α invertible, and ruin the IRO robustness of our construction again. This is why we only output a "blinded" version of the signature computed as $lsb_{3m}(H^3_\oplus(\alpha_M)) \oplus g(\alpha_M)$. This way the signature α_M gets not leaked when H_0 or H_1 is a random oracle, which is necessary for the combiner to be IRO robust. Moreover with high probability (over the choice of the pairwise-independent function g) the blinding, which maps $\{0,1\}^m$ to $\{0,1\}^{3m}$, will be injective (i.e., contain no collisions), which as explained before is necessary to get robustness for CR and TCR.

Overall, the combiner – as illustrated in Figure 2 – computes for input message M and its corresponding signature $\alpha_M = lsb_m(H^0_\oplus(M))$ the following output:

$$\mathsf{Comb}^{H_0,H_1,g}_{\mathsf{4P\&IRO}}(M) = P^2_\alpha(H^0_0(M)||H^0_1(M)) \ || \ lsb_{3m}(H^3_\oplus(\alpha_M)) \oplus g(\alpha_M).$$

4.2 $\mathcal{C}_{\mathsf{4P\&IRO}}$ is IRO-Robust

We show that our combiner is indifferentiable from a random oracle when instantiated with two functions H_0, H_1, where one of them is a random oracle (we refer to it as $H_b, b \in \{0,1\}$), and the other function $H_{\overline{b}}$ is arbitrary[4]. Like the random oracle H_b, also $H_{\overline{b}}$ is given as an oracle and accessible by all parties. The pairwise independent function g that comes up in this construction is only needed to prove that $\mathcal{C}_{\mathsf{4P\&IRO}}$ still preserves the CR and TCR properties; for the IRO property this function can be arbitrary.

Lemma 4.2 *The combiner $\mathcal{C}_{\mathsf{4P\&IRO}}$ is IRO-robust.*

[4]There is a small caveat here. Our definition of combiners allows to use the same hash function $H_0 = H_1$, albeit our combiner samples independent instances of the hash functions then. In this sense, it is understood that, if the hash function H_b is given by a random oracle, then in case $H_b = H_{\overline{b}}$ the other hash function instance uses an independent random oracle.

Remark. Note that the security of $\mathsf{Comb}_{\mathsf{4P\&IRO}}$ as a random oracle combiner depends on m, and thus on the output length, which is $2n + 3m$. This can be slightly improved to $2n + 2m + m'$ for some $m' < m$ (by simply replacing $3m$ with $2m + m'$ in Figure 2), though m' should not be too small, as $\mathcal{C}_{\mathsf{4P\&IRO}}$ is a good combiner for the CR and TCR with probability $2^{-m'}$ (this probability is over the choice of the PIF, as we explain later in Section 4.3).

Proof. For the proof we assume that $b = 0$, i.e., the hash function $H_0 : \{0,1\}^* \to \{0,1\}^n$ is a random oracle. The case $b = 1$ is proved analogously. The adversary $\mathcal{A}$ has then access either to the combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}$ and H_0 or to a random oracle $\mathcal{F} : \{0,1\}^* \to \{0,1\}^{2n+3m}$ and a simulator $\mathcal{S}^{\mathcal{F}}$. Our combiner is indifferentiable from a random oracle $\mathcal{F}$ if there exists a simulator $\mathcal{S}^{\mathcal{F}}$, such that the adversary $\mathcal{A}$ can distinguish between $\mathsf{Comb}_{\mathsf{4P\&IRO}}, H_0$ and $\mathcal{F}, \mathcal{S}^{\mathcal{F}}$ only with negligible probability. The proof consists of two parts: we first provide the description of our simulator $\mathcal{S}^{\mathcal{F}}$ and then we show that $\mathcal{A}$ has only negligible advantage in distinguishing the ideal setting (with $\mathcal{S}^{\mathcal{F}}$) and the real setting.

The simulator keeps as state the function table of a (partially defined) function $\hat{H}_0 : \{0,1\}^* \to \{0,1\}^n$, which initially is empty, i.e., $\hat{H}_0(X) = \perp$ for all X. We define $\hat{H}_0^i(M) = \hat{H}_0(\langle i \rangle_2 \| M)$ to mimic the notion used in Figure 2. The goal of $\mathcal{S}^{\mathcal{F}}$ is to define $\hat{H}_0$ in such a way that, from $\mathcal{A}$'s point of view, $(\mathcal{F}, \hat{H}_0)$ look like $(\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}, H_0)$, i.e., the output of $\hat{H}_0$ has to be random and consistent with what the distinguisher can obtain from $\mathcal{F}$. Therefore, our simulator $\mathcal{S}^{\mathcal{F}}$ parses each query X it is invoked on as $X = \langle i \rangle_2 \| M$ and proceeds as follows:

```
**Simulator** $\mathcal{S}^{\mathcal{F}}(X)$:
on query $X$ check if some entry $Y \leftarrow \hat{H}_0(X)$ already exists
    if $Y = \perp$    //no entry so far
        if $X = \langle 0 \rangle_2 \| M$ for some $M$
            set $\hat{H}_0^0(M) = y_0$ where $y_0$ is randomly chosen from $\{0,1\}^n$    $(*)$
            get $y_1 \leftarrow H_1^0(M)$ and compute $\alpha_M = lsb_m(y_0 \oplus y_1)$
            get $U \leftarrow \mathcal{F}(M)$ for query $M$ and parse $U$ as $U_1 \| U_2 \| U_3$    $(*)$
                where $|U_1| = |U_2| = n$ and $|U_3| = 3m$.
            set $\hat{H}_0^1(\alpha_M \| y_1) = U_2 \oplus y_0 \oplus H_1^1(\alpha_M \| y_1)$
            set $\hat{H}_0^2(\alpha_M \| U_2) = U_1 \oplus y_1 \oplus H_1^2(\alpha_M \| U_2)$
            set $\hat{H}_0^3(\alpha_M) = (U_3 \| z) \oplus (g(\alpha_M) \| 0^{n-3m}) \oplus H_1^3(\alpha_M)$
                where $z$ is randomly chosen from $\{0,1\}^{n-3m}$
        if $X \neq \langle 0 \rangle_2 \| M$, choose a random $Y \in \{0,1\}^n$    $(*)$
                and save the value by setting $\hat{H}_0(X) = Y$
output $Y \leftarrow \hat{H}_0(X)$
```

Figure 3: Description of the Simulator

Whenever $\mathcal{S}^{\mathcal{F}}$ is invoked on a query X where $\hat{H}_0(X) \neq \perp$, $\mathcal{S}^{\mathcal{F}}$ simply outputs $\hat{H}_0(M)$. Thus from now on we only consider queries X where $\hat{H}_0(X) = \perp$. In this case, $\mathcal{S}^{\mathcal{F}}$ will define the output of $\hat{H}_0(X)$, and in some cases also on some additional inputs. On a query $X = \langle i \rangle_2 || M$ where $\hat{H}_0^i(M) = \perp$ and $i \neq 0$, the simulator samples a random $Y \in \{0,1\}^n$, sets $\hat{H}_0^i(M) = Y$ and outputs Y.

The interesting queries are the queries of the form $X = \langle 0 \rangle_2 || M$ which could be an attempt of $\mathcal{A}$ to simulate the construction of the combiner, such that the simulator has to compute in addition consistent answers to potential subsequent queries of $\mathcal{A}$. The simulator starts by sampling a random $y_0 \in \{0,1\}^n$ and sets $\hat{H}_0^0(M) = y_0$. To define the "signature" α_M of M, $\mathcal{S}^{\mathcal{F}}$ queries its oracle H_1 on $\langle 0 \rangle_2 || M$ and uses the answer $y_1 = H_1^0(M)$ to compute $\alpha_M = lsb_m(y_0 \oplus y_1)$. The simulator then defines the outputs of the intermediate functions $\hat{H}_0^1, \hat{H}_0^2$ and $\hat{H}_0^3$ such that $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{\hat{H}_0,H_1,g}(M) = \mathcal{F}(M)$. Therefore $\mathcal{S}^{\mathcal{F}}$ invokes its random oracle $\mathcal{F}$ on input M and computes the corresponding outputs of $\hat{H}_0$ by retracing the combiners construction as defined in the simulators description. Note that this is possible in a unique way, except for the $n - 3m$ last bits of $\hat{H}_0^3(\alpha_M)$, which must be chosen uniformly at random.

We now prove that from $\mathcal{A}$'s point of view $(\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}, H_0)$ and $(\mathcal{F}, \mathcal{S}^{\mathcal{F}})$ are indistinguishable, when making at most q queries to each oracle. To this end we consider a sequence of hybrid games, starting with a game where $\mathcal{A}$ interacts with $(\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}, H_0)$ and ending in the ideal setting where the distinguisher has access to $(\mathcal{F}, \mathcal{S}^{\mathcal{F}})$. The game structure of this proof is depicted in Figure 4.

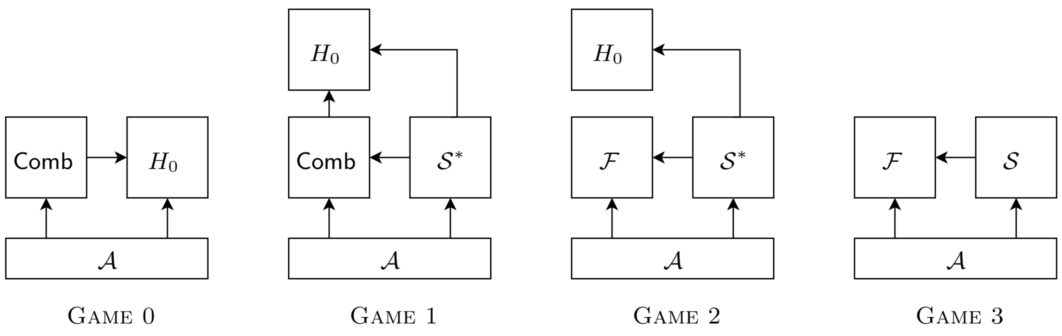

Figure 4: Games used in the Indifferentiability Proof

Game 0: The adversary interacts with $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ and H_0.

Game 1: We change the way $\mathcal{A}$'s queries to H_0 are answered, by giving $\mathcal{A}$ access to an algorithm $\mathcal{S}^*$ instead of direct access to the random oracle. The algorithm $\mathcal{S}^*$ works as our simulator $\mathcal{S}$, except that it queries H_0 instead of simulating it via lazy sampling, and it calls $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}(M)$ instead of $\mathcal{F}(M)$. Thus, $\mathcal{S}^*$ basically relays all queries of $\mathcal{A}$ to H_0 but also keeps a table of answered values. For all queries of the form $X = \langle 0 \rangle_2 || M$ the algorithm additionally precomputes further values as described in Figure 3 (the lines where $\mathcal{S}^*$ deviates from $\mathcal{S}$ are marked with $*$). Note that $\mathcal{S}^*$'s answers and stored values are (with one exception) exactly the same as the values one would obtain directly from H_0. In particular, the values $\mathcal{S}^*$ defines for $\hat{H}_0^1, \hat{H}_0^2$ by querying $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ are identical to the real values of H_0^1, H_0^2. The only difference occurs when in the precomputations of $\mathcal{S}^*$ a value for $\hat{H}_0^3(\alpha_M)$ is set, since only the first $3m$ bits will equal the value of $H_0^3(\alpha_M)$. However, the final $n-3m$ bits are set to random such that also $\hat{H}_0^3(\alpha_M)$ is a truly random string. Thus, the only way for $\mathcal{A}$ to recognize the discrepancy to the real $H_0^3(\alpha_M)$ value, is by querying $\langle 3\rangle_2 \,||\alpha_M$ *before* sending a query $\langle 0\rangle_2 \,||M$ that will lead to α_M. We denote this event by Bad_1. As all signature values α_M that originate from a query to H_0^0 are uniform random values of length m and $\mathcal{A}$ makes at most q queries to its $\mathcal{S}^*$ oracle, this event happens with overall probability at most $q^2 \cdot 2^{-m}$. Unless $\mathcal{A}$ provokes Bad_1, Game 0 and Game 1 are identical (we denote by Game $i \Rightarrow 1$ the event that $\mathcal{A}$ outputs 1 in Game i):

$$|\,\mathrm{Prob}[\,\text{Game } 1 \Rightarrow 1] - \mathrm{Prob}[\,\text{Game } 0 \Rightarrow 1]\,| \leq \mathrm{Prob}[\mathsf{Bad}_1] \leq q^2 \cdot 2^{-m}$$

Game 2: In our second game we replace the combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ with the random oracle $\mathcal{F}$. Due to that change, the algorithm $\mathcal{S}^*$ now obtains $\mathcal{F}(M)$ instead of $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}(M)$ when doing its precomputations. Thus, the additional values that $\mathcal{S}^*$ stores in $\hat{H}_0^i$ for $i \in \{1,2,3\}$ when responding to a $\langle 0\rangle_2 \,||M$ query, are now consistent with $\mathcal{F}(M)$ and thereby with high probability different from the real values of H_0^i for $i \in \{1,2,3\}$. Again, this only matters if $\mathcal{A}$ manages to first issue a query $\langle i\rangle_2 \,||\alpha_M||*$ and subsequently invokes $\mathcal{S}^*$ on $\langle 0\rangle_2 \,||M$ that will lead to α_M. Otherwise, all $\mathcal{A}$ gets to see from $\mathcal{S}^*$ are random and consistent answers. To capture that case where $\mathcal{S}^*$ "fails", we consider by Bad_2 the event that the function $\hat{H}_0^i$ for $i \in \{1,2\}$ is already defined on any input of the form $\alpha_M||*$ when $\mathcal{S}^*$ wants to set a value in the course of a precomputation. (Note that the case for $i=3$ is already handled by Bad_1 in Game 1.) As $\alpha_M \in \{0,1\}^m$ is uniformly random, the probability that Bad_2 occurs in the q-th query is at most $q \cdot 2^{-m}$ (as each $\hat{H}_0^i$ for $i \in \{1,2\}$ is defined on at most $q-1$ inputs). Then the overall probability that Bad_2 in any of $\mathcal{A}$'s queries happens is at most $2q^2 \cdot 2^{-m}$.

Furthermore, the outputs provided by $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ are indistinguishable from $\mathcal{F}$, as long as no collision on the signature values occurs, i.e., $M \neq M'$ but $\alpha_M = \alpha'_M$ (we omit a formal proof, as it follows the argumentation of Lemma 4.4 closely). Since $\mathcal{A}$ sends at most q queries to $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$, such a collision occurs with probability at most $q^2 \cdot 2^{-m}$. By adding the probabilities of both events we obtain

$$|\,\mathrm{Prob}[\,\text{Game } 2 \Rightarrow 1] - \mathrm{Prob}[\,\text{Game } 1 \Rightarrow 1]\,| \leq 3q^2 \cdot 2^{-m}$$

Game 3: In the final game the adversary interacts with $\mathcal{F}$ and $\mathcal{S}^{\mathcal{F}}$. That is, Game 2 and Game 3 only differ in the fact that $\mathcal{S}^{\mathcal{F}}$ simulates the random responses from H_0 by using lazy sampling instead of querying H_0. Thus, from $\mathcal{A}$'s viewpoint both games are identical:

$$\mathrm{Prob}[\,\text{Game } 3 \Rightarrow 1] = \mathrm{Prob}[\,\text{Game } 2 \Rightarrow 1]$$

Overall, we have

$$|\,\mathrm{Prob}[\,\text{Game } 3 \Rightarrow 1] - \mathrm{Prob}[\,\text{Game } 0 \Rightarrow 1]\,| \leq 4q^2 \cdot 2^{-m}.$$

Hence, the advantage of $\mathcal{A}$ in distinguishing $(\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}, H_0)$ from $(\mathcal{F}, \mathcal{S}^{\mathcal{F}})$ is negligible. This proves our claim.

□

4.3 $\mathcal{C}_{\mathsf{4P\&IRO}}$ is Robust for $\mathsf{CR}, \mathsf{TCR}, \mathsf{MAC}, \mathsf{PRF}$

We now prove that, like the $\mathcal{C}_{\mathsf{4P}}$ combiner, $\mathcal{C}_{\mathsf{4P\&IRO}}$ also preserves the CR, TCR, MAC and PRF property in a robust manner.

Lemma 4.3 *The combiner $\mathcal{C}_{\mathsf{4P\&IRO}}$ is CR- and TCR-robust.*

Proof. We will prove that for any H_0, H_1, with probability $1 - 2^{-m}$ over the choice of the pairwise independent function g, any collision for $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ is simultaneously a collision for H_0^0 and H_1^0. To this end, let $M \neq M'$ be a collision for $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ and let α_M and $\alpha_{M'}$ denote their signatures. Let $Y \| Y' = \mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}(M)$ where $Y \in \{0,1\}^{2n}$ and $Y' \in \{0,1\}^{3m}$.

If $\alpha_M = \alpha_{M'}$, then M, M' must be a collision for H_0^0 and H_1^0, as we have

$$H_0^0(M) \| H_1^0(M) = P_\alpha^{2^{-1}}(Y) = P_{\alpha'}^{2^{-1}}(Y) = H_0^0(M') \| H_1^0(M') \tag{1}$$

and the Feistel permutations $P_\alpha^2, P_{\alpha'}^2$ are identical if $\alpha_M = \alpha_{M'}$.

For M, M' where $\alpha_M \neq \alpha_{M'}$, a collision on the combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}(M) = \mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}(M')$ does not imply (1), and thus will in general not be a collision for H_0 and H_1. Yet, as with probability $1 - 2^{-m}$ over the choice of the pairwise independent function $g : \{0,1\}^m \to \{0,1\}^{3m}$, there does not exist a collision M, M' for $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ where $\alpha_M \neq \alpha_{M'}$. Note that for this it is sufficient to prove that for any two potential signatures $\alpha \neq \alpha' \in \{0,1\}^m$, we have

$$lsb_{3m}(H_\oplus^3(\alpha)) \oplus g(\alpha) \neq lsb_{3m}(H_\oplus^3(\alpha')) \oplus g(\alpha') \tag{2}$$

as this implies that the final outputs are distinct for any two messages with different signatures. As g is pairwise independent, for any particular $\alpha \neq \alpha'$, equation (2) holds with probability $1 - 2^{-3m}$. Taking the union bound over all $2^m(2^m - 1)/2 < 2^{2m}$ distinct values $\alpha \neq \alpha'$, we get that the probability that there exists some $\alpha \neq \alpha'$ not satisfying (2) is at most $2^{2m}/2^{3m} = 2^{-m}$.

The proof of TCR-robustness follows a similar argumentation. A collision $M \neq M'$ on the combiner implies with overwhelming probability a collision $H_0^0(M)||H_1^0(M) = H_0^0(M')||H_1^0(M')$ on the first evaluation of both hash functions. Thus, given an adversary $\mathcal{A}_{\mathsf{Comb}}$ against the combiner that commits to a target message M and later outputs a colliding message M', one can build an adversary $\mathcal{A}_b$ against hash function H_b that commits to $00||M$ and outputs in the second stage $00||M'$. □

Lemma 4.4 *The combiner $\mathcal{C}_{\mathsf{4P\&IRO}}$ is PRF-robust.*

Remark. To compute the first part of the output, our combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ applies a two-round Feistel network, which in general does not preserve the (pseudo)-randomness from an underlying round function $H_{\oplus}^{i}$, because it maps an input (L_0, R_0) to (L_2, R_2) where $R_2 = H_{\oplus}^{1}(R_0) \oplus L_0$ depends only on the given input values. When evaluating the Feistel network with two distinct inputs (L_0, R_0) and (L_0', R_0), the difference $L_0 \oplus L_0'$ then propagates to the outputs, i.e., $L_0 \oplus L_0' = R_2 \oplus R_2'$, which can be exploited by an adversary. In our construction we destroy this dependence by prepending the value α_M to the input of each round function, where $\alpha_M = lsb_m(H_{\oplus}^{0}(M))$ is a uniformly random value if $H_b, b \in \{0,1\}$ is a uniformly random function. Thus we have $R_2 = H_{\oplus}^{1}(\alpha_M||R_0) \oplus L_0$ with $L_0 = H_0^0(M)$ and $R_0 = H_1^0(M)$ such that for two distinct inputs $M \neq M'$, the probability for $R_2 \oplus R_2' = H_0^0(M) \oplus H_0^0(M')$ is $\text{Prob}[\alpha_M = \alpha_{M'}] = 2^{-m}$.

Proof. Assume that the hash function H_0 is a pseudorandom function, but the combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ is not (the proof for H_1 can be done analogously). Hence, there exists a successful adversary $\mathcal{A}_{\mathsf{Comb}}$ which can distinguish the construction $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ from a truly random function $F : \{0,1\}^* \to \{0,1\}^{2n+3m}$ with non-negligible probability. We show that this allows to construct an adversary $\mathcal{A}_0$ that can distinguish H_0 from a random function $f : \{0,1\}^* \to \{0,1\}^n$.

Algorithm $\mathcal{A}_0$ simulates the oracle of $\mathcal{A}_{\mathsf{Comb}}$, which is either $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ or F, with his own oracle and the knowledge of $H_1 \leftarrow \mathsf{HKGen}_1$ and g that he samples accordingly. For each query of $\mathcal{A}_{\mathsf{Comb}}$, the adversary $\mathcal{A}_0$ computes an answer by emulating the combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}$ using $H_1(\cdot), g$ and his oracle which serves as H_0.

For the analysis recall that the underlying oracle of $\mathcal{A}_0$ is either a random function f or the hash function $H_0(\cdot)$. In the latter case $\mathcal{A}_0$ provides outputs that are identically distributed to the values $\mathcal{A}_{\mathsf{Comb}}$ would obtain from $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$. Hence, we have

$$\text{Prob}\left[\mathcal{A}_0^{H_0}(1^n) = 1\right] = \text{Prob}\left[\mathcal{A}_{\mathsf{Comb}}^{\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}}(1^n) = 1\right].$$

If the underlying oracle is the random function f, then the computed answers of $\mathcal{A}_0$ have to look like a truly random function as well. We show that this is true if, for q queries $M_1 \dots M_q$ and for all $i \neq j$, we have $\alpha_{M_i} \neq \alpha_{M_j}$. The probability of this not being the case is at most $q^2 \cdot 2^{-m}$, since $\alpha_M = lsb_m(H_{\oplus}^{0}(M))$ is a random value when H_0 gets replaced by the random function f.

Hence, with high probability $\mathcal{A}_0$ will create for each query M_i of $\mathcal{A}_{\mathsf{Comb}}$ a fresh signature α_{M_i}. To analyze the corresponding output of $\mathcal{A}_0$ we parse his answer in three parts, namely $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{f,H_1,g}(M_i) = U_1||U_2||U_3$ with $|U_1| = |U_2| = n$ and $|U_3| = 3m$. The last part U_3 results from the computation $lsb_{3m}(f(\langle 3\rangle_2\,||\alpha_{M_i}) \oplus H_1^3(\alpha_{M_i})) \oplus g(\alpha_{M_i})$. Since α_{M_i} is uniformly distributed and gets extended by the unique prefix $\langle 3\rangle_2$, the input value of $f(\langle 3\rangle_2\,||\alpha_{M_i})$ is distinct from all other queries to f during the $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{f,H_1,g}(M_i)$ computation, and hence the corresponding output is an independently and uniformly distributed value. As xor-ing is a good combiner for random functions, the randomness of f gets preserved in the computation of U_3. For the second part U_2 we just consider the final calculation, i.e., $U_2 = f(\langle 0\rangle_2\,||M_i) \oplus f(\langle 1\rangle_2\,||\alpha_{M_i}||Y) \oplus H_1^1(\alpha_{M_i}||Y)$ for some

$Y \in \{0,1\}^n$. Here we prepend the bits $\langle 1 \rangle_2$ to the random value α_{M_i}, such that we have again distinct evaluations of f which gives us uniformly random images. A similar argumentation holds for $U_1 = Y' \oplus f(\langle 2 \rangle_2 || \alpha_{M_i} || Y'') \oplus H_1^2(\alpha_{M_i} || Y'')$ for $Y', Y'' \in \{0,1\}^n$, where we use the unique prefix $\langle 2 \rangle_2$ when querying f in order to obtain values that are independently and uniformly distributed. Thus, if for all queried messages $M_i \neq M_j$ of $\mathcal{A}_{\mathsf{Comb}}$ there occurs no collision on the signatures, i.e., $\alpha_{M_i} \neq \alpha_{M_j}$, the values $U_1 || U_2 || U_3$ are independent random strings.

Overall, the output distribution of $\mathcal{A}_{\mathsf{Comb}}$ satisfies

$$\text{Prob}\left[\mathcal{A}_0^f(1^n) = 1\right] \leq \text{Prob}\left[\mathcal{A}_{\mathsf{Comb}}^F(1^n) = 1\right] + q^2 \cdot 2^{-m}.$$

Thus, the probability that $\mathcal{A}_0$ can distinguish H_0 from f is not negligible, which contradicts the assumption that H_0 is a pseudorandom function. □

Lemma 4.5 *The combiner $\mathcal{C}_{4\mathsf{P\&}\mathit{IRO}}$ is MAC-robust.*

Proof. The proof is by contradiction. Assume that an adversary $\mathcal{A}_{\mathsf{Comb}}$ with oracle access to the combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ outputs with noticeable probability a valid pair (M, τ) where $\tau = \mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}(M)$ and M is distinct from all previous queries to the MAC-oracle. This allows to construct an adversary $\mathcal{A}_b$ against the hash function H_b for $b \in \{0,1\}$.

Adversary $\mathcal{A}_b$ first samples $H_{\overline{b}} \leftarrow \mathsf{HKGen}_1$ that it uses together with its own oracle $H_b(\cdot)$ to answer all queries by $\mathcal{A}_{\mathsf{Comb}}$ in a black-box simulation. When $\mathcal{A}_{\mathsf{Comb}}$ returns a valid forgery (M, τ), where $M \neq M_1, M_2 \ldots M_q$, the adversary $\mathcal{A}_b$ flips a coin $c \leftarrow \{0,1\}$ and proceeds as follows:

- If $c = 0$, then $\mathcal{A}_b$ randomly chooses an index k between 1 and q and looks up the corresponding signature value α_{M_k}. It then computes $\tau_0 || \tau_1 = P_\alpha^{2^{-1}}(lsb_{2n}(\tau))$ using α_{M_k} and stops with the output $(\langle 0 \rangle_2 || M, \tau_b)$.
- If $c = 1$, then $\mathcal{A}_b$ queries its oracle about $\langle 0 \rangle_2 || M$ to receive an answer y_0 and computes $\alpha_M = y_0 \oplus y_1$ with $y_1 = H_1^0(M)$. It then calculates the first round of the Feistel permutation, i.e., until the evaluation of $H_\oplus^2$ where $x = y_0 \oplus H_0^1(\alpha_M || y_1)$ would be used as input to this function. It outputs as forgery the message $(\langle 2 \rangle_2 || \alpha_M || x)$ with tag $\tau' = \tau_b \oplus H_{\overline{b}}(\langle 2 \rangle_2 || \alpha_M || x) \oplus y_1$ where $\tau_0 || \tau_1 = lsb_n(\tau)$.

For the analysis we have to consider two cases of a successful adversary $\mathcal{A}_{\mathsf{Comb}}$. In the first case, $\mathcal{A}_{\mathsf{Comb}}$ returns a pair (M, τ), such that $\alpha_M = \alpha_{M_j}$ for some $j = 1, 2, \ldots, q$, i.e., the signature value of M has already been computed for another message $M_j \neq M$ during A_b's process of simulating the combiner. Then, if $c = 0$, the adversary $\mathcal{A}_b$ obtains a valid forgery $(\langle 0 \rangle_2 || M, \tau_b)$ if it guesses the index j correctly and then inverts the Feistel step for input $lsb_{2n}(\tau)$ and α_{M_j}. The message $\langle 0 \rangle_2 || M$ is distinct from all of $\mathcal{A}_b$'s queries, because $\langle 0 \rangle_2 || M$ is distinct from all $\langle 0 \rangle_2 || M_i$ and the additional queries of $\mathcal{A}_b$ start with a prefix $\langle i \rangle_2$ where $i \in 1, 2, 3$. Hence, if $\mathcal{A}_{\mathsf{Comb}}$ forges such a MAC with non-negligible probability ϵ, then $\mathcal{A}_b$ succeeds with probability $\epsilon / 2q$.

In the second case, $\mathcal{A}_{\mathsf{Comb}}$ outputs (M,τ) where α_M has not occurred in $\mathcal{A}_b$'s computations, i.e., $\alpha_M \neq \alpha_{M_j}$ for all $j = 1,2,\dots,q$. In this case, we have $c = 1$ with probability 1/2 where $\mathcal{A}_b$ starts its forgery by computing the first round of the Feistel permutation for input $H_0^0(M)||H_1^0(M)$ and $\alpha_M = lsb_m(H_\oplus^0(M))$, which requires a further oracle query about $00||M$. The left part of the computed Feistel output is then $x = H_0^0(M) \oplus H_0^1(\alpha_M||H_1^0(M))$ and would serve as input for $H_\oplus^2$. The adversary uses this value together with the fresh signature α_M as its output message $(\langle 2\rangle_2 ||\alpha_M||x)$ and reconstructs the corresponding tag with the knowledge about the other parameters. Since α_M is distinct from all α_{M_j}, the message $(\langle 2\rangle_2 ||\alpha_M||x)$ was never queried by $\mathcal{A}_b$ before.

In both cases a successful attack against the combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}$ allows successful attacks on H_0 and H_1, contradicting the assumption that at least one hash function is a secure MAC. $\square$

5 Preserving One-Wayness and the $\mathcal{C}_{\mathsf{4P\&OW}}$ Combiner

In this section we first propose a combiner which is simultaneously a combiner for CR and OW. At the end of this section we discuss how to plug this combiner into our combiners $\mathsf{Comb}_{\mathsf{4P}}$ and $\mathsf{Comb}_{\mathsf{4P\&IRO}}$ to derive our constructions $\mathsf{Comb}_{\mathsf{4P\&OW}}$ (cf. Figure 1) and $\mathsf{Comb}_{\mathsf{6P}}$ (cf. Figure 2), respectively.

Recall that the concatenation combiner

$$\mathsf{Comb}_{||}^{H_0,H_1}(M) = H_0(M)\|H_1(M)$$

is a robust combiner for the CR property, but its not hard to see that this combiner is not robust for the one-wayness property OW.[5] On the other hand, the following combiner

$$\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}(M_L\|M_R) = H_0(M_L)\|H_1(M_R)$$

is robust for the OW property, i.e. $\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}(M_L\|M_R)$ is hard to invert on a random input from $\{0,1\}^{2m}$, if either H_0 or H_1 is hard to invert on $\{0,1\}^m$. Unfortunately, this combiner is not robust for CR.[6]

The basic idea to construct a combiner which is robust for CR and OW is to use the $\mathsf{Comb}_{||}^{H_0,H_1}$ combiner, and to apply a pairwise independent permutation (PIP) to the input of one of the two components. As the length of a description of a PIP is twice its input length, we have to assume an upper bound on the input length of the components. We fix the domain of H_0 and H_1 to $\{0,1\}^{5n}$, but remark that any longer input length $kn, k > 5$ would work, but then also the $2kn$-bits key for the PIP grows accordingly. Allowing (much) shorter input length kn for some $k < 5$ is not possible, as we use the fact that the input is (at least) $5n$ bits in the proof.

[5]Consider e.g. the case where $H_0(\cdot)$ is a one-way function on the first n bits of its input (and ignoring the rest), and $H_1(\cdot)$ outputs the first n bits of its input.

[6]Consider e.g. the case where $H_0(\cdot)$ is collision resistant, and $H_1(\cdot)$ outputs a constant, then any pair $M_L\|M_R, M_L\|M_R'$ gives a collision.

5.1 A Combiner for CR and OW

We define the combiner $\mathcal{C}_{\mathsf{CR\&OW}}$ for preserving collision-resistance and one-wayness in a robust manner as follows. The key generation algorithm of the combiner $\mathsf{CKGen}_{\mathsf{CR\&OW}}(1^n)$ generates $H_0 \leftarrow \mathsf{HKGen}_0(1^n)$ and $H_1 \leftarrow \mathsf{HKGen}_1(1^n)$ and picks a permutation π from a family Π of pairwise independent permutations over $\{0,1\}^{5n}$. It outputs (H_0, H_1, π). The evaluation algorithm $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}$ on input $M \in \{0,1\}^{5n}$ returns $H_0(\pi(M))||H_1(M)$. By the following theorem $\mathcal{C}_{\mathsf{CR\&OW}}$ preserves the properties of $\mathsf{Comb}_{||}$ and $\mathsf{Comb}_{\mathsf{OW}}$ simultaneously.

Theorem 5.1 *The combiner $\mathcal{C}_{\mathsf{CR\&OW}}$ is a strongly robust multi-property combiner for* $\textsc{prop} = \{$*CR, TCR, MAC, OW*$\}$.

The proof is again split into lemmas for the individual properties.

Lemma 5.2 *The combiner $\mathcal{C}_{\mathsf{CR\&OW}}$ is CR-, TCR- and MAC-robust.*

Proof. As for the CR and TCR properties, note that given any collision $M \neq M'$ for $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}$, we get a collision M, M' for H_1 and a collision $\pi(M), \pi(M')$ for H_0. Note that $\pi(M) \neq \pi(M')$ as π is a permutation.

To see that the MAC property is preserved, observe that given any forgery (M, τ) for $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}$, we get a forgery $(\pi(M), \tau_0)$ for H_0 and a forgery (M, τ_1) for H_1 where $\tau_0 \| \tau_1 = \tau$. $\square$

Lemma 5.3 *The combiner $\mathcal{C}_{\mathsf{CR\&OW}}$ is OW-robust.*

Technically, we show the following. Let $H_0, H_1 : \{0,1\}^{5n} \to \{0,1\}^n$ be any hash functions, $T = T(n)$ be arbitrary and Π be a family of pairwise independent permutations over $\{0,1\}^{5n}$. Then with probability $1 - 2/T$ over the choice of $\pi \leftarrow \Pi$ the following holds: any adversary $\mathcal{A}$ that inverts $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(\cdot)$ on a random output with probability at least $2/T$, can be used to invert $\mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}(\cdot)$ with probability $2/T^3$. Thus, if we assume that the advantage of every efficient adversary in inverting $\mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}(\cdot)$ is some negligible value $\epsilon = 2/T^3$, then for an overwhelming $1 - 2/T$ fraction of the π's, the advantage of every adversary inverting $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(\cdot)$ is bounded by a negligible term $2/T$.

To get some intuition for our proof, assume there is a "perfect" adversary $\mathcal{A}$ who inverts $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(\cdot)$ on every value in its range. We can simply invoke this algorithm $\mathcal{A}$ on an output $y_0 \| y_1$ generated by the "plain" combiner $\mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}(\cdot) = H_0(\cdot)||H_1(\cdot)$, and if now $\mathcal{A}$ gives us a preimage M with $y_0 \| y_1 = \mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(M)$ for our advanced combiner, then this gives us a preimage $\pi(M) \| M$ with $y_0 \| y_1 = \mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}(\pi(M) \| M)$ for the plain combiner. There is one caveat here, namely, even such an ideal $\mathcal{A}$ will not invert an output $y_0 \| y_1 = \mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}(M)$ if $y_0 \| y_1$ is not in the range of $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(\cdot)$. Using the fact that the H_0, H_1 are shrinking and π is a PIP, this can be shown to happen with only small probability (over the choice of π, M).

The general case where $\mathcal{A}$ only inverts $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(\cdot)$ with some small probability ϵ is more tricky as here $\mathcal{A}$ might invert exactly on these outputs $y_0 \| y_1$ which are much

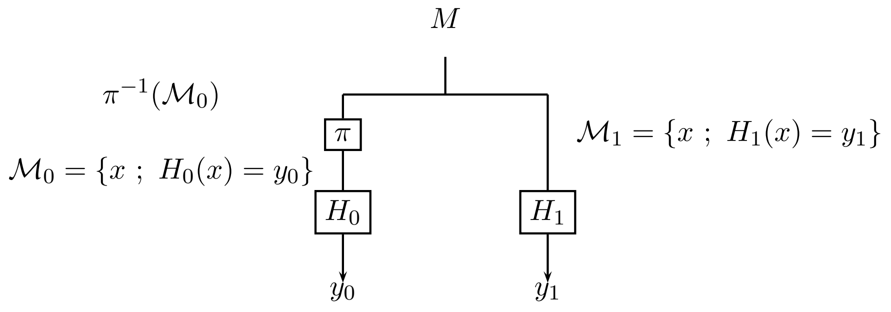
M
$\pi^{-1}(\mathcal{M}_0)$
π
$\mathcal{M}_1 = \{x \; ; \; H_1(x) = y_1\}$
$\mathcal{M}_0 = \{x \; ; \; H_0(x) = y_0\}$
H_0
H_1
y_0
y_1

Figure 5: Illustration of $\mathsf{Comb}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$ and the sets defined in the proof of Claim 1.

more likely to be outputs of $\mathsf{Comb}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}(\cdot)$ than of $\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}(\cdot)$, and thus would be of limited use for inverting $\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}(\cdot)$. In the proof below we show that for most choices of π, $\mathcal{A}$ has still success probability $\Omega(\epsilon^3)$ in inverting $\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}(\cdot)$ even if $\mathcal{A}$ shows such worst-case behavior.

Proof (of Lemma 5.3). Let H_0, H_1 be fixed, we prove that for a $1 - 2/T$ fraction of the $\pi \in \Pi$ the following holds: If an adversary $\mathcal{A}$ inverts $\mathsf{Comb}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}(\cdot)$ with probability $2/T$, this adversary also inverts the one-way combiner $\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}(\cdot)$ (and thus also H_0 and H_1) with probability at least $1/2T^3$.

We first relate the output distribution of the combiner $\mathsf{Comb}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$ with the output distribution of the one-wayness combiner $\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}$. Call a tuple $(\pi, y_0||y_1)$ bad if it is more than $2T^2$ times more likely to be an output of $\mathsf{Comb}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$ than of $\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}(\cdot) = H_0(\cdot)||H_1(\cdot)$. That is, $(\pi, y_0||y_1)$ is called *bad* iff

$$\begin{aligned}&\mathrm{Prob}_M[\mathsf{Comb}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}(M) = y_0||y_1] \\ &\qquad \geq \quad 2 \cdot T^2 \cdot \mathrm{Prob}_{M_0,M_1}[\mathsf{Comb}_{\mathsf{OW}}^{H_0,H_1}(M_0||M_1) = y_0||y_1] \qquad (3)\end{aligned}$$

which, by definition, means

$$\begin{aligned}&\mathrm{Prob}_M[H_0(\pi(M)) = y_0 \wedge H_1(M) = y_1] \\ &\qquad \geq \quad 2 \cdot T^2 \cdot \mathrm{Prob}_{M_0,M_1}[H_0(M_0) = y_0 \wedge H_1(M_1) = y_1]. \qquad (4)\end{aligned}$$

The following Claim bounds the probability of a tuple (sampled in a particular way) to be bad:

Claim 1:

$$\mathrm{Prob}_{\pi,M}[(\pi, \mathsf{Comb}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}(M)) \text{ is bad}] \leq 2/T^2$$

Proof. To save on notation, we let τ denote the probability space defined by the following process:

$$\pi \leftarrow \Pi \ , \ M \leftarrow \{0,1\}^{5n} \ , \ y_0 := H_0(\pi(M)) \ , \ y_1 := H_1(M) \qquad (5)$$

With this, we can write the statement of the claim as

$$\mathrm{Prob}_\tau[(\pi, y_0||y_1) \text{ is bad}] \leq 2/T^2 \qquad (6)$$

Plugging the definition (4) of "being bad" into (6) we get

$$\begin{aligned} \mathrm{Prob}_\tau \quad & [\ \mathrm{Prob}_{M'}[H_0(\pi(M')) = y_0 \wedge H_1(M') = y_1] \\ & \quad \geq 2 \cdot T^2 \cdot \mathrm{Prob}_{M_0,M_1}[H_0(M_0) = y_0 \wedge H_1(M_1) = y_1]\ ] \\ \leq \quad & 2/T^2 \end{aligned} \tag{7}$$

For fixed $y_0, y_1 \in \{0,1\}^n, \pi \in \Pi$ we let $\mathcal{M}_b$ denote the preimages of y_b under H_b, that is,

$$\mathcal{M}_0 \stackrel{\text{def}}{=} \{x \ : \ H_0(x) = y_0\}, \quad \mathcal{M}_1 \stackrel{\text{def}}{=} \{x \ : \ H_1(x) = y_1\}, \quad \pi^{-1}(\mathcal{M}_0) \stackrel{\text{def}}{=} \{x \ : \ \pi(x) \in \mathcal{M}_0\}.$$

We can now express the terms in (7) as:

$$\mathrm{Prob}_{M_0,M_1}[H_0(M_0) = y_0 \wedge H_1(M_1) = y_1] \quad = \quad \frac{|\mathcal{M}_0|}{2^{5n}} \frac{|\mathcal{M}_1|}{2^{5n}} \tag{8}$$

$$\mathrm{Prob}_{M'}[H_0(\pi(M')) = y_0 \wedge H_1(M') = y_1] \quad = \quad \frac{|\pi^{-1}(\mathcal{M}_0) \cap \mathcal{M}_1)|}{2^{5n}} \tag{9}$$

Equation (8) is simply the probability that the random $M_0 \leftarrow \{0,1\}^{5n}$ falls into $\mathcal{M}_0$ and that $M_1 \leftarrow \{0,1\}^{5n}$ falls into $\mathcal{M}_1$. To see eq.(9) note that $H_0(m(M'))\|H_1(M') = y_0\|y_1$ if and only if $M' \in \pi^{-1}(\mathcal{M}_0) \cap \mathcal{M}_1$.

Plugging (8),(9) into (7) (and multiplying with 2^{5n}) we can rewrite the statement of the claim as

$$\mathrm{Prob}_\tau \left[|\pi^{-1}(\mathcal{M}_0) \cap \mathcal{M}_1| \geq T^2 \cdot 2 \frac{|\mathcal{M}_0||\mathcal{M}_1|}{2^{5n}} \right] \leq \frac{2}{T^2}. \tag{10}$$

We claim that $\mathbb{E}_\tau \left[|\pi^{-1}(\mathcal{M}_0) \cap \mathcal{M}_1| \right]$, the expected number of preimages of $y_0\|y_1$ (sampled according to τ) under $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}$, can be expressed as

$$\mathbb{E}_\tau \left[|\pi^{-1}(\mathcal{M}_0) \cap \mathcal{M}_1| \right] = \mathbb{E}_\tau \left[1 + \frac{(|\mathcal{M}_0| - 1)(|\mathcal{M}_1| - 1)}{2^{5n} - 1} \right] \tag{11}$$

This can be seen as follows. A way to sample a variable with the same expectation as $|\pi^{-1}(\mathcal{M}_0) \cap \mathcal{M}_1|$ is to sample $M, \pi(M)$ (i.e., only the output of π on input M, but not the entire π) which defines $y_0\|y_1 := H_0(\pi(M))\|H_1(M)$. Now, one preimage of $y_0\|y_1$ is M (this is accounted for by the term "1+" above), and for every of the $|\mathcal{M}_1| - 1$ other $M' \in \mathcal{M}_1$ we have another preimage if $\pi(M') \in \mathcal{M}_0$. As π is pairwise independent, $\pi(M')$ is uniform in $\mathcal{M}_0 \setminus \pi(M)$, thus this happens with probability $(|\mathcal{M}_0| - 1)/(2^{5n} - 1)$.

To get some intuition, assume for the moment that $\mathcal{M}_0$ and $\mathcal{M}_1$ are "large", say of size at least 2^{3n}. Then we can essentially ignore the "1+" and "-1" terms in (11) and the statement becomes roughly

$$\mathbb{E}_\tau \left[|\pi^{-1}(\mathcal{M}_0) \cap \mathcal{M}_1| \right] \approx \mathbb{E}_\tau \left[|\mathcal{M}_0||\mathcal{M}_1|/2^{5n} \right]$$

This would allow us to relate the probabilities (8) and (9), and applying Markov's inequality would prove the claim. Thus to prove the claim we first show that $\mathcal{M}_0, \mathcal{M}_1$ are indeed very large with high probability, and then we will work out the details of the outlined intuition.

For any function $f : \{0,1\}^{5n} \to \{0,1\}^n$, the probability that a random image has less than 2^{3n} preimages is at most 2^{-n}, i.e.

$$\mathrm{Prob}_{x \leftarrow \{0,1\}^{5n}} \left[|\{x' \ : \ f(x') = f(x)\}| < 2^{3n} \right] \leq 2^{-n} \tag{12}$$

This holds as at most $2^n - 1$ (i.e. all except one) values in the range $\{0,1\}^n$ of f can have $< 2^{3n}$ preimages. So at most $(2^n - 1)(2^{3n} - 1) < 2^{4n}$ values in $\{0,1\}^{5n}$ are mapped by f to an image with $< 2^{3n}$ preimages under f. The probability that a random $x \leftarrow \{0,1\}^{5n}$ falls into this set is $\leq 2^{5n}/2^{4n} = 2^{-n}$. Using (12) and the definitions of $\mathcal{M}_0, \mathcal{M}_1$, we conclude

$$\mathrm{Prob}_\tau[|\mathcal{M}_0| < 2^{3n}] \leq 2^{-n} \quad \text{and} \quad \mathrm{Prob}_\tau[|\mathcal{M}_1| < 2^{3n}] \leq 2^{-n}.$$

Applying the union bound we get an upper bound on the probability that either set is small as

$$\mathrm{Prob}_\tau[|\mathcal{M}_0| < 2^{3n} \vee |\mathcal{M}_1| < 2^{3n}] \leq 2 \cdot 2^{-n}. \tag{13}$$

Hence, except with probability $2 \cdot 2^{-n}$ (which becomes smaller than $1/T^2$ for sufficiently large n's), we have $|\mathcal{M}_0| \geq 2^{3n}$ and $|\mathcal{M}_1| \geq 2^{3n}$, let us call this event $\mathcal{E}$. A short calculation shows that in this case

$$\mathcal{E} \quad \text{implies} \quad 1 + \frac{(|\mathcal{M}_0| - 1)(|\mathcal{M}_1| - 1)}{2^{5n} - 1} \leq 2\frac{|\mathcal{M}_0||\mathcal{M}_1|}{2^{5n}}, \tag{14}$$

We can now prove (10). For this, let $Z = |\pi^{-1}(\mathcal{M}_0) \cap \mathcal{M}_1|$. In the second step below we use the fact, established by equations (11) to (14), that conditioned on $\mathcal{E}$ we have $\mathbb{E}_\tau[Z] \leq \mathbb{E}_\tau[2|\mathcal{M}_0||\mathcal{M}_1|/2^{5n}]$. We use Markov's inequality and equation (13) in the third step below.

$$\begin{aligned}
\mathrm{Prob}_\tau\left[Z \geq T^2 \cdot 2\frac{|\mathcal{M}_0||\mathcal{M}_1|}{2^{5n}}\right] &\leq \mathrm{Prob}_\tau[\mathcal{E}]\mathrm{Prob}_\tau\left[Z \geq T^2 \cdot 2\frac{|\mathcal{M}_0||\mathcal{M}_1|}{2^{5n}} \mid \mathcal{E}\right] \\
&\quad + \mathrm{Prob}_\tau[\neg\mathcal{E}]\mathrm{Prob}_\tau\left[Z \geq T^2 \cdot 2\frac{|\mathcal{M}_0||\mathcal{M}_1|}{2^{5n}} \mid \neg\mathcal{E}\right] \\
&\leq \mathrm{Prob}_\tau[Z \geq T^2 \cdot \mathbb{E}[Z]|\mathcal{E}] + \mathrm{Prob}_\tau[\neg\mathcal{E}] \\
&\leq 1/T^2 + 2 \cdot 2^{-n} \leq 2/T^2
\end{aligned}$$

This concludes the proof of the claim. □

Using Markov's inequality once more the claim implies

$$\mathrm{Prob}_\pi[\mathrm{Prob}_M[(\pi, \mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(M)) \text{ is bad}] \leq 1/T] \geq 1 - 2/T. \tag{15}$$

We will now show that with probability $1 - 2/T$ over the choice of $\pi \leftarrow \Pi$ the following holds: any adversary $\mathcal{A}$ that inverts $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(\cdot)$ on a random output with probability at least $2/T$, can be used to invert $\mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}(\cdot)$ with probability $2/T^3$. This will imply the Lemma as explained in the paragraph after the statement of the Lemma.

The $1-2/T$ fraction of π's we will conisder are all $\pi \in \Pi$ where

$$\begin{aligned} \mathrm{Prob}_M[(\pi, \mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(M)) \text{ is bad}] &= \\ \mathrm{Prob}_M[(\pi, H_0(\pi(M)\|H_1(M)) \text{ is bad}] &\le 1/T \end{aligned} \tag{16}$$

by eq.(15) a random $\pi \leftarrow \Pi$ will indeed satisfy this with probability $1-2/T$.

Now consider any π satisfying (16) and any adversary $\mathcal{A}$ who inverts a random output $y_0\|y_1 = \mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(M)$ with probability $2/T$ (wlog. we assume $\mathcal{A}$ is deterministic). As by eq.(16) such a random output $(\pi, y_0\|y_1)$ is bad with probability at most $1/T$, it follows that such a random output simultaneously is good and $\mathcal{A}$ finds a preimage with probability at least $2/T - 1/T = 1/T$. Let $\mathcal{Y} \subset \{0,1\}^{2n}$ denote this set, i.e.

$$\mathcal{Y} = \{y_0\|y_0 \;:\; (\pi, y_0\|y_1) \text{ is good and } \mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(\mathcal{A}^{H_0,H_1}(\pi, y_0\|y_1)) = y_0\|y_1\}$$

As explained above

$$\mathrm{Prob}_M[\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(M) \in \mathcal{Y}] \ge 1/T$$

As $\mathcal{Y}$ only contains good outputs, by definition (cf. eq.(3)) the probability that the output of the one-way combiner $\mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}(\cdot)$ on a random input falls into $\mathcal{Y}$ is at most $2T^2$ times smaller than the probability that the output of $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(\cdot)$ on a random input falls into $\mathcal{Y}$, which as just shown is at least $1/T$, thus

$$\mathrm{Prob}_{M_0,M_1}[\mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}(M_0||M_1) \in \mathcal{Y}] \ge \frac{1}{T} \cdot \frac{1}{2T^2} = \frac{1}{2T^3}$$

As by definition of $\mathcal{Y}$ the adversary $\mathcal{A}$ inverts all values in $\mathcal{Y}$, $\mathcal{A}$ on input a random output $y_0\|y_1$ of $\mathsf{Comb}^{H_0,H_1}_{\mathsf{OW}}$ will find a preimage M' for this output with probability at least $1/2T^3$. Although $\mathcal{A}$ outputs a preimage M' for the wrong combiner $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(M') = y_0\|y_1$, from this we can easily compute a preimage $M'_0\|M'_1 := \pi(M')\|M'_1$ for the one-way combiner $\mathsf{Comb}_{\mathsf{OW}}(M'_0\|M'_1) = y_0\|y_1$. $\square$

5.2 Combining Things

We can now plug the combiner $\mathcal{C}_{\mathsf{CR\&OW}}$ into the initial computation of our combiner $\mathcal{C}_{\mathsf{4P}}$. That is, we replace the initial computation $H^0_0(M)||H^0_1(M)$ in our original combiner by $H^0_0(\pi(M))||H^0_1(M)$ for messages of $5n$ bits. Note that if $H_b(\cdot)$ is one way on inputs of length $5n+2$, then also $H^0_b(\cdot)$ is one-way on inputs of length $5n$, and we only lose a factor of 4 in the security.

More formally, in our combiner $\mathcal{C}_{\mathsf{4P\&OW}} = (\mathsf{CKGen}_{\mathsf{4P\&OW}}, \mathsf{Comb}_{\mathsf{4P\&OW}})$ for functions $\mathcal{H}_0, \mathcal{H}_1$ the key generation algorithm generates a tuple (π, H_0, H_1) consisting of a pairwise independent permutation π (over $\{0,1\}^{5n}$) and two hash functions $H_0 \leftarrow \mathsf{HKGen}_0(1^n)$ and $H_1 \leftarrow \mathsf{HKGen}_1(1^n)$. The evaluation algorithm $\mathsf{Comb}^{H_0,H_1,\pi}_{\mathsf{4P\&OW}}$ for input $M \in \{0,1\}^{5n}$ computes $P^3(H^0_0(\pi(M))||H^0_1(M))$ where P^3 is the Feistel permutation $P^3 = \psi[H^1_\oplus, H^2_\oplus, H^3_\oplus]$. Note that applying a permutation to the output of a one-way function does not violate the one-way property. We have already proved that the other three properties CR,TCR,MAC which are preserved by $\mathcal{C}_{\mathsf{CR\&OW}}$ are not affected by applying a permutation in Section 3.

Theorem 5.4 *The combiner $\mathcal{C}_{\mathsf{4P\&OW}}$ is a strongly robust multi-property combiner for* $\textsc{prop} = \{\mathit{CR}, \mathit{PRF}, \mathit{TCR}, \mathit{MAC}, \mathit{OW}\}$.

Applying the modifications from Section 5 and the combiner $\mathsf{Comb}_{\mathsf{4P\&IRO}}$ from Section 4 together, we derive our construction $\mathcal{C}_{\mathsf{6P}}$ (cf. Figure 2). This construction is defined like $\mathcal{C}_{\mathsf{4P\&IRO}}$, where one additionally applies a pairwise-independent permutation over $\{0,1\}^{kn}$ (with $k \geq 5$) to the input of H_0^0.

Theorem 5.5 *The combiner $\mathcal{C}_{\mathsf{6P}}$ is a strongly robust multi-property combiner for* $\textsc{prop} = \{\mathit{CR}, \mathit{TCR}, \mathit{PRF}, \mathit{MAC}, \mathit{OW}, \mathit{IRO}\}$.

6 Weak vs. Mild vs. Strong Robustness

In this section we revert to our different notions of multi-property robustness as introduced in Section 2.2, and analyze the relations among the three variants. The first proposition shows that strong robustness implies mild robustness which, in turn, implies weak robustness. The proof is straightforward and given only for sake of completeness:

Proposition 6.1 *Let* $\textsc{prop}$ *be a set of properties. Then any strongly robust multi-property combiner for* $\textsc{prop}$ *is also mildly robust for* $\textsc{prop}$*, and any mildly robust combiner for* $\textsc{prop}$ *is also weakly robust for* $\textsc{prop}$.

Proof. Assume that the combiner is sMPR for $\textsc{prop}$. Suppose further that $\textsc{prop}(\mathcal{C}) \not\supseteq \textsc{prop}$ such that there is some property $\mathsf{P}_i \in \textsc{prop} - \textsc{prop}(\mathcal{C})$. Then, since the combiner is sMPR, we must also have $\mathsf{P}_i \notin \textsc{prop}(\mathcal{H}_0) \cup \textsc{prop}(\mathcal{H}_1)$, else we derive a contradiction to the strong robustness. We therefore have $\textsc{prop} \not\subseteq \textsc{prop}(\mathcal{H}_0) \cup \textsc{prop}(\mathcal{H}_1)$, implying mild robustness via the contrapositive statement.

Now consider an mMPR combiner and assume $\textsc{prop} = \textsc{prop}(\mathcal{H}_0)$ or $\textsc{prop} = \textsc{prop}(\mathcal{H}_1)$. Then, in particular, $\textsc{prop} = \textsc{prop}(\mathcal{H}_0) \cup \textsc{prop}(\mathcal{H}_1)$ and the mMPR property says that also $\textsc{prop} = \textsc{prop}(\mathcal{C})$. This proves sMPR. $\square$

To separate the notions we consider the collision-resistance property CR and the property NZ (*n*on-*z*ero output) that the hash function should return $0\cdots 0$ with small probability only. This may be for example required if the hash value should be inverted in a field:

non-zero output (NZ): A hash function $\mathcal{H}$ has property NZ if for any efficient adversary $\mathcal{A}$ the probability that for $H \leftarrow \mathsf{HKGen}(1^n)$ and $M \leftarrow \mathcal{A}(H)$ we have $H(M) = 0\cdots 0$, is negligible.

Lemma 6.2 *Let* $\textsc{prop} = \{\mathit{CR}, \mathit{NZ}\}$ *and assume that collision-resistant hash functions exist. Then there is a hash function combiner which is weakly multi-property robust for* $\textsc{prop}$*, but not mildly multi-property robust for* $\textsc{prop}$.

Proof. Consider the following combiner (with the standard key generation, $(H_0, H_1) \leftarrow \mathsf{CKGen}(1^n)$ for $H_0 \leftarrow \mathsf{HKGen}_0(1^n)$ and $H_1 \leftarrow \mathsf{HKGen}_1(1^n)$):

> The combiner for input M first checks that the length of M is even, and if so, divides $M = L||R$ into halves L and R, and checks
>
> - that $H_0(L) \neq H_0(R)$ if $L \neq R$, and that $H_0(M) \neq 0 \cdots 0$,
> - that $H_1(L) \neq H_1(R)$ if $L \neq R$, and that $H_1(M) \neq 0 \cdots 0$.
>
> If the length of M is odd or any of the two properties above holds, then the combiner outputs $H_0(M)||H_1(M)$. In any other case, it returns 0^{2n}.

We first show that the combiner is weakly robust. For this assume that the hash function H_b for $b \in \{0,1\}$ has both properties. Then the combiner returns the exceptional output 0^{2n} only with negligible probability, namely, if one finds an input with a non-trivial collision under H_b which also refutes property NZ. In any other case, the combiner's output $H_0(M)||H_1(M)$ inherits the properties CR and NZ from hash function H_b.

Next we show that the combiner is not mMPR. Let H_1' be a collision-resistant hash function with $n-1$ bits output (and let H_1 include a description of H_1'). Define the following hash functions:

$$H_0(M) = 1^n, \qquad H_1(M) = \begin{cases} 0^n & \text{if } M = 0^n1^n \\ 1||H_1'(M) & \text{else} \end{cases}.$$

Clearly, H_0 has property NZ but is not collision-resistant. On the other hand, H_1 obeys CR but not NZ, as 0^n1^n is mapped to zeros. But then we have $\textsc{prop} = \{\mathsf{CR}, \mathsf{NZ}\} = \textsc{prop}(H_0) \cup \textsc{prop}(H_1)$ and mild robustness now demands that the combiner, too, has these two properties. Yet, for input $M = 0^n1^n$ the combiner returns 0^{2n} since the length of M is even, but $L = 0^n$ and $R = 1^n$ collide under H_0, and M is thrown to 0^n under H_1. This means that the combiner does not obey property NZ. □

Lemma 6.3 *Let* $\textsc{prop} = \{\mathsf{CR}, \mathsf{NZ}\}$*. Then there exists a hash function combiner which is mildly multi-property robust for* $\textsc{prop}$*, but not strongly multi-property robust for* $\textsc{prop}$*.*

Proof. Consider the following combiner (again with standard key generation):

> The combiner for input M first checks that the length of M is even, and if so, divides $M = L||R$ into halves L and R and then verifies that $H_0(L) \neq H_0(R)$ or $H_1(L) \neq H_1(R)$ or $L = R$. If any of the latter conditions holds, or the length of M is odd, then the combiner outputs $H_0(M)||H_1(M)$. In any other case it returns 0^{2n}.

We first prove that the combiner is mMPR. Given that $\textsc{prop} \subseteq \textsc{prop}(H_0) \cup \textsc{prop}(H_1)$ at least one of the two hash functions is collision-resistant. Hence, even for $M = L||R$ with even length and $L \neq R$, the hash values only collide with negligible probability. In other words, the combiner outputs $H_0(M)||H_1(M)$ with overwhelming probability, implying that the combiner too has properties CR and NZ.

Now consider the constant hash functions $H_0(M) = H_1(M) = 1^n$ for all M. Clearly, both hash functions obey property $\mathsf{NZ} \in \textsc{prop}(H_0) \cup \textsc{prop}(H_1)$. Yet, for input 0^n1^n

the combiner returns 0^{2n} such that $\mathsf{NZ} \notin \textsc{prop}(\mathcal{C})$, implying that the combiner is not strongly robust. $\square$

The proof indicates how mildly (or weakly) robust combiners may take advantage of further properties to implement other properties. It remains open if one can find similar separations for the popular properties like CR and PRF, or for CR and IRO.

7 Multiple Hash Functions and Tree-Based Composition of Combiners

So far we have considered combiners for two hash functions. The multi-property robustness definition extends to the case of more hash functions as follows:

Definition 7.1 *For a set* $\textsc{prop} = \{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_N\}$ *of properties an* m*-function combiner* $\mathcal{C} = (\mathsf{CKGen}, \mathsf{Comb})$ *for hash functions* $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_{m-1}$ *is called*

weakly multi-property robust *(wMPR) for* $\textsc{prop}$ *iff*

$$\exists j \in \{0,1,\ldots,m-1\} \text{ s.t. } \textsc{prop} = \textsc{prop}(\mathcal{H}_j) \quad \Longrightarrow \quad \textsc{prop} = \textsc{prop}(\mathcal{C}),$$

mildly multi-property robust *(mMPR) for* $\textsc{prop}$ *iff*

$$\textsc{prop} = \bigcup_{j=0}^{m-1} \textsc{prop}(\mathcal{H}_j) \quad \Longrightarrow \quad \textsc{prop} = \textsc{prop}(\mathcal{C}),$$

and strongly multi-property robust *(sMPR) for* $\textsc{prop}$ *iff for all* $\mathsf{P}_i \in \textsc{prop}$,

$$\mathsf{P}_i \in \bigcup_{j=0}^{m-1} \textsc{prop}(\mathcal{H}_j) \quad \Longrightarrow \quad \mathsf{P}_i \in \textsc{prop}(\mathcal{C}).$$

For the above definitions we still have that sMPR implies mMPR and mMPR implies wMPR. The proof is a straightforward adaption of the case of two hash functions.

Given a combiner for two hash functions one can build a combiner for three or more hash functions by considering the two-function combiner itself as a hash function and applying it recursively. For instance, to combine three hash functions $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2$ one may define the "cascaded" combiner by $\mathcal{C}_2(\mathcal{C}_2(\mathcal{H}_0, \mathcal{H}_1), \mathcal{H}_2)$, where we assume that the output of $\mathcal{C}_2$ allows to be used again as input to the combiner on the next level.

More generally, given m hash functions and a two-function combiner $\mathcal{C}_2$ we define an m-function combiner $\mathcal{C}_{\text{multi}}$ as a binary tree, as suggested for general combiners by [18]. Each leaf is labeled by one of the m hash functions (different leaves may be labeled by the same hash function). Each inner node, including the root, with two descendants labeled by $\mathcal{F}_0$ and $\mathcal{F}_1$, is labeled by $\mathcal{C}_2(\mathcal{F}_0, \mathcal{F}_1)$.

The key generation algorithm for this tree-based combiner now runs the key generation algorithm for the label at each node (each run independent of the others, even if two nodes contain the same label). To evaluate the multi-hash function combiner one inputs M into each leaf and computes the functions outputs recursively up to the root. The output of the root node is then the output of $\mathcal{C}_{\text{multi}}$. We call this a *combiner tree for* $\mathcal{C}_2$ *and* $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_{m-1}$.

For efficiency reasons we assume that there are at most polynomially many combiner evaluations in a combiner tree. Also, to make the output dependent on all hash functions we assume that each hash function appears in (at least) one of the leaves. If a combiner tree obeys these properties, we call it an *admissible combiner tree for* $\mathcal{C}_2$ *and* $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_{m-1}$.

We first show that weak MPR and strong MPR preserve their properties for admissible combiner trees:

Proposition 7.2 *Let* $\mathcal{C}_2$ *be a weakly (resp. strongly) multi-property robust two-function combiner for* PROP*. Then any admissible combiner tree for* $\mathcal{C}_2$ *and functions* $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_{m-1}$ *for* $m \geq 2$ *is also weakly (resp. strongly) multi-property robust for* PROP*.*

Proof. We give the proof by induction for the depth of the tree. For depth $d = 1$ we have $m = 2$ and $\mathcal{C}_{\text{multi}}(\mathcal{H}_0, \mathcal{H}_1) = \mathcal{C}_2(\mathcal{H}_0, \mathcal{H}_1)$ or $\mathcal{C}_{\text{multi}}(\mathcal{H}_0, \mathcal{H}_1) = \mathcal{C}_2(\mathcal{H}_1, \mathcal{H}_0)$ and the claim follows straightforwardly for both cases.

Now assume $d > 1$ and that combiner $\mathcal{C}_2$ is wMPR. Then the root node applies $\mathcal{C}_2$ to two nodes N_0 and N_1, labeled by $\mathcal{F}_0$ and $\mathcal{F}_1$. Note that by the wMPR prerequisite we assume that there exists one hash function $\mathcal{H}_j$ which has all properties in PROP. Since this hash functions appears in at least one of the subtrees under N_0 or N_1, it follows by induction that at least one of the functions $\mathcal{F}_0$ and $\mathcal{F}_1$, too, has properties PROP. But then the combiner application in the root node also inherits these properties from its descendants.

Now consider $d > 1$ and the case of strong MPR. It follows analogoulsy to the previous case that for each property $\mathsf{P}_i \in$ PROP, one of the hash functions in the subtrees rooted at N_0 and N_1 must have property P_i as well. This carries over to the combiners at nodes N_0 or N_1 by induction, and therefore to the root combiner. $\square$

Somewhat surprisingly, mild MPR in general does not propagate security for tree combiners, as we show by a counter-example below. Note that we still obtain, via the previous proposition, that the mMPR combiner is also wMPR and that the resulting tree combiner is thus also wMPR. Yet, it loses its mMPR property.

Proposition 7.3 *Let* PROP $= \{\mathsf{CR}, \mathsf{NZ}\}$ *and assume that there are collision-resistant hash functions. Then there exists a two-function mildly robust multi-property combiner* $\mathcal{C}_2$ *for* PROP*, and hash functions* $\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2$ *such that the admissible tree combiner for* $\mathcal{C}_2$ *is not mildly multi-property robust for* PROP*.*

Proof. Consider the following two-function combiner $\mathcal{C}_2$ for hash functions $\mathcal{H}_0, \mathcal{H}_1$ (again with standard key generation):

For input M check that the length of M is even and, if so, divide $M = L||R$ into halves L and R. If $H_0(L) = H_0(R)$ and $H_1(L) = H_1(R)$ and $L \neq R$, or

we have $H_0(M) = 0 \cdots 0$ and $H_1(M) = 0 \cdots 0$, then output $0^{|H_0(M)|+|H_1(M)|}$.
Else, or if the length of M is odd, return $H_0(M)||H_1(M)$.

It is easy to verify that this is an mMPR two-function combiner for PROP. Now consider the following hash functions, where H_2' is a collision-resistant hash function with $n-1$ bits output:

$$H_0(M) = 1^n, \qquad H_1(M) = 1^n, \qquad H_2(M) = \begin{cases} 0^n & \text{if } M = 0^n1^n \\ 1||H_2'(M) & \text{else} \end{cases}.$$

Then $\textsc{prop}(\mathcal{H}_0) = \textsc{prop}(\mathcal{H}_1) = \{\mathsf{NZ}\}$ and $\textsc{prop}(\mathcal{H}_2) = \{\mathsf{CR}\}$ such that $\textsc{prop} = \bigcup \textsc{prop}(\mathcal{H}_j)$.

Consider the following tree combiner defined through $\mathcal{C}(\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2) = \mathcal{C}_2(\mathcal{C}_2(\mathcal{H}_0, \mathcal{H}_1), \mathcal{H}_2)$, i.e., which cascades the three hash functions. Then the inner application of $\mathcal{C}_2$ yields a hash function which returns 0^{2n} for message $M = 0^n1^n$. Since this message also causes H_2 to return 0^n the tree combiner runs into the exception case and returns 0^{3n} for this input. Hence, the tree combiner does not have property NZ. $\square$

Note that the cascading combiner can also be applied to all our proposed MPR combiners to compose three or more hash functions. The derived combiner, however, is less efficient than the direct construction sketched there.

Acknowledgments

We thank the anonymous reviewers for valuable comments. Most of this work was done while the second author was at Darmstadt University of Technology. The first and second author were supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG). The first author is also supported by a Heisenberg grant Fi 940/3-1 of the German Research Foundation (DFG).

References

[1] Open source implementation of the Com4P combiner. `http://files.randombit.net/botan/doxygen/html/classBotan_1_1Comb4P.html`, 2010.

[2] Elena Andreeva, Gregory Neven, Bart Preneel, and Thomas Shrimpton. Seven-property-preserving iterated hashing: ROX. In *Advances in Cryptology — Asiacrypt 2007*, volume 4833 of *LNCS*, pages 130–146. Springer-Verlag, 2007.

[3] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In *Advances in Cryptology — Crypto 1996*, volume 96 of *LNCS*, pages 1–15. Springer-Verlag, 1996.

[4] Mihir Bellare and Thomas Ristenpart. Multi-property preserving hash domain extensions and the EMD transform. In *Advances in Cryptology — Asiacrypt 2006*, volume 4284 of *LNCS*, pages 299–314. Springer-Verlag, 2006.

[5] Mihir Bellare and Thomas Ristenpart. Hash functions in the dedicated-key setting: Design choices and MPP transforms. In *International Colloquium on Automata, Languages, and Programming (ICALP) 2007*, volume 4596 of *LNCS*, pages 399–410. Springer-Verlag, 2007.

[6] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption — how to encrypt with RSA. In *Advances in Cryptology — Eurocrypt 1994*, volume 950 of *LNCS*, pages 92–111. Springer-Verlag, 1994.

[7] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures — how to sign with RSA and Rabin. In *Advances in Cryptology — Eurocrypt 1996*, volume 1070 of *LNCS*, pages 399–416. Springer-Verlag, 1996.

[8] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *Advances in Cryptology — Eurocrypt 2006*, volume 4004 of *LNCS*, pages 409–426. Springer-Verlag, 2006.

[9] Alexandra Boldyreva and Marc Fischlin. Analysis of random oracle instantiation scenarios for OAEP and other practical schemes. In *Advances in Cryptology — Crypto 2005*, volume 3621 of *LNCS*, pages 412–429. Springer-Verlag, 2005.

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

[11] Dan Boneh and Xavier Boyen. On the impossibility of efficiently combining collision resistant hash functions. In *Advances in Cryptology — Crypto 2006*, volume 4117 of *LNCS*, pages 570–583. Springer-Verlag, 2006.

[12] Ran Canetti, Ronald L. Rivest, Madhu Sudan, Luca Trevisan, Salil P. Vadhan, and Hoeteck Wee. Amplifying collision resistance: A complexity-theoretic treatment. In *Advances in Cryptology — Crypto 2007*, volume 4622 of *LNCS*, pages 264–283. Springer-Verlag, 2007.

[13] Christophe De Cannière and Christian Rechberger. Preimages for reduced SHA-0 and SHA-1. In *Advances in Cryptology — Crypto 2008*, volume 5157 of *LNCS*, pages 179–202. Springer-Verlag, 2008.

[14] Jean-Sebastien Coron, Yevgeniy Dodis, Cecile Malinaud, and Prashant Puniya. Merkle-Damgård revisited: How to construct a hash function. In *Advances in Cryptology — Crypto 2005*, volume 3621 of *LNCS*, pages 430–448. Springer-Verlag, 2005.

[15] Marc Fischlin and Anja Lehmann. Security-amplifying combiners for hash functions. In *Advances in Cryptology — Crypto 2007*, volume 4622 of *LNCS*, pages 224–243. Springer-Verlag, 2007.

[16] Marc Fischlin and Anja Lehmann. Multi-property preserving combiners for hash functions. In *Theory of Cryptography Conference (TCC) 2008*, volume 4948 of *LNCS*, pages 375–392. Springer-Verlag, 2008.

[17] Marc Fischlin, Anja Lehmann, and Krzysztof Pietrzak. Robust multi-property combiners for hash functions revisited. In *International Colloquium on Automata, Languages, and Programming (ICALP) 2008*, volume 5126 of *LNCS*, pages 655–666. Springer-Verlag, 2008.

[18] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *LNCS*, pages 96–113. Springer-Verlag, 2005.

[19] Amir Herzberg. On tolerant cryptographic constructions. In *Topics in Cryptology — Cryptographer's Track, RSA Conference (CT-RSA) 2005*, volume 3376 of *LNCS*, pages 172–190. Springer-Verlag, 2005.

[20] Amir Herzberg. Folklore, practice and theory of robust combiners. *Journal of Computer Security*, 17(2):159–189, 2009.

[21] Jonathan Hoch and Adi Shamir. On the strength of the concatenated hash combiner when all the hash functions are weak. In *International Colloquium on Automata, Languages, and Programming (ICALP) 2008*, volume 5126 of *LNCS*, pages 616–630. Springer-Verlag, 2008.

[22] Jonathan Katz and Ji Sun Shin. Modeling insider attacks on group key-exchange protocols. In *Proceedings of the Annual Conference on Computer and Communications Security (CCS).* ACM Press, 2005.

[23] Anja Lehmann and Stefano Tessaro. A modular design for hash functions: Towards making the mix-compress-mix approach practical. In *Advances in Cryptology — Asiacrypt 2009*, volume 5912 of *LNCS*, pages 364–381. Springer-Verlag, 2009.

[24] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[25] Ueli Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *Theory of Cryptography Conference (TCC) 2004*, volume 2951 of *LNCS*, pages 21–39. Springer-Verlag, 2004.

[26] Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999.

[27] NIST. National institute of standards and technology: SHA-3 competition. `http://csrc.nist.gov/groups/ST/hash/sha-3/`.

[28] Krzysztof Pietrzak. Non-trivial black-box combiners for collision-resistant hash-functions don't exist. In *Advances in Cryptology — Eurocrypt 2007*, volume 4515 of *LNCS*, pages 23–33. Springer-Verlag, 2007.

[29] Krzysztof Pietrzak. Compression from collisions, or why CRHF combiners have a long output. In *Advances in Cryptology — Crypto 2008*, volume 5157 of *LNCS*, pages 413–432. Springer-Verlag, 2008.

[30] Marc Stevens, Alexander Sotirov, Jacob Appelbaum, Arjen 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 *Advances in Cryptology — Crypto 2009*, volume 5677 of *LNCS*, pages 55–73. Springer-Verlag, 2009.

[31] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full SHA-1. In *Advances in Cryptology — Crypto 2005*, volume 3621 of *LNCS*, pages 17–36. Springer-Verlag, 2005.

[32] Xiaoyun Wang and Hongbo Yu. How to break MD5 and other hash functions. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *LNCS*, pages 19–35. Springer-Verlag, 2005.