A preliminary version appears at ICALP, Lecture Notes in Computer Science, Springer-Verlag, 2008.

Robust Multi-Property Combiners for Hash Functions Revisited

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

[1]Darmstadt University of Technology, Germany
www.minicrypt.de
[2]CWI, Amsterdam, Netherlands

Abstract. A robust multi-property combiner for a set of security properties merges two hash functions such that the resulting function satisfies each of the properties which at least one of the two starting functions has. Fischlin and Lehmann (TCC 2008) recently constructed a combiner which simultaneously preserves collision-resistance, target collision-resistance, message authentication, pseudorandomness and indifferentiability from a random oracle (IRO). Their combiner produces outputs of $5n$ bits, where n denotes the output length of the underlying hash functions.

In this paper we propose improved combiners with shorter outputs. By sacrificing the indifferentiability from random oracles we obtain a combiner which preserves all of the other aforementioned properties but with output length $2n$ only. This matches a lower bound for black-box combiners for collision-resistance as the only property, showing that the other properties can be achieved without penalizing the length of the hash values. We then propose a combiner which also preserves the IRO property, slightly increasing the output length to $2n + \omega(\log n)$. Finally, we show that a twist on our combiners also makes them robust for one-wayness (but at the price of a fixed input length).

1 Introduction

A black-box combiner for some cryptographic primitive, is a construction, which given black-box access to two candidate schemes, securely implements the primitive, if at least one of the two candidates securely implements it [Her05, HKN$^+$05]. Thus combiners can be used as hedge against the failure of a concrete construction, as the combiner is secure as long as at least one of the two candidates is not broken. In light of the many recent attacks on popular collision resistant hash functions [BCJ$^+$05, WYY05, WLF$^+$05, WY05, FLN07], combiners for hash-functions are of particular interest.

For many important primitives very simple combiners do exist. For example, the "concatenation combiner" $\mathsf{C}_{\|}^{H_0,H_1}(M) = H_0(M)\|H_1(M)$ for hash-functions preserves the property of being collision-resistant (CR) and target collision-resistant (TCR), 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 function H_0 or H_1 is collision-resistant, then so is the combined function.

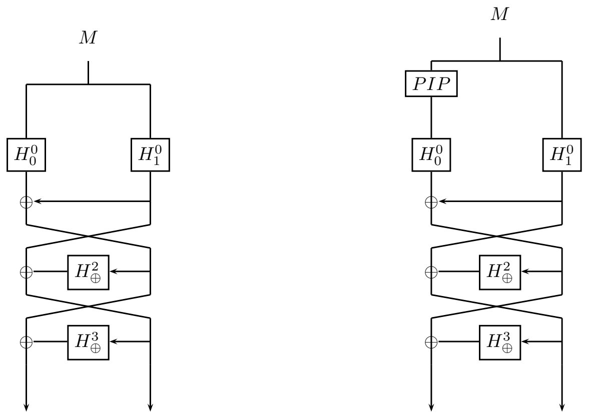

Figure 1: Illustration of the basic construction C_{4P} (left) preserving $\mathsf{CR}, \mathsf{PRF}, \mathsf{TCR}$ and MAC. Here $H_b^i(\cdot)$ denotes $H_b(\langle i \rangle_2 \,\|\cdot)$ where $\langle i \rangle_2$ is the binary representation of the integer i with two bits. $H_\oplus^i(\cdot)$ denotes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

Nowadays hash functions are often deployed in many facets, e.g., as pseudorandom functions in TLS or message authentication codes in IPSec. In some standardized protocols as RSA-OAEP [BR94] and RSA-PSS [BR96], even stronger assumptions on the underlying hash-functions are made [BF05, BF06].

While the concatenation combiner preserves the MAC property, the PRF property is in general not conserved. In contrast, the "XOR combiner" $\mathsf{C}_\oplus^{H_0,H_1}(M) = H_0(M) \oplus M_1(M)$ is robust with respect to PRF, and also for indistinguishability from a random oracle (IRO), but neither preserves the CR nor the TCR property.

Ideally, one would like to have a single combiner preserving many properties simultaneously. To this end, Fischlin and Lehmann [FL08] have introduced the notion of robust multi-property combiners for a set of security properties $\textsc{prop}$. According to their strongest notion such a combiner satisfies the property $\mathsf{P} \in \textsc{prop}$ if P is satisfied by at least one of the two candidate hash functions. Their combiner, denoted here as C_{5P}, preserves all of the discussed properties, i.e., (target) collision-resistance (TCR, CR), pseudorandomness (PRF), message authentication (MAC) and indifferentiability from a random oracle[1] (IRO). Paying tribute to the fact that several properties are conserved, the the C_{5P} combiner has a rather long output of $5n$ bits, where n denotes the output length of the underlying hash functions. This raises the question whether having such a long output is necessary for a combiner which preserves all the properties simultaneously, or if we can do better. Let us mention that we can not hope to get below $2n$ bits (except for a logarithmic additive term), as this is already a lower bound for black-box combiners preserving collision-resistance only [BB06, Pie07, CRS$^+$07, Pie08].

The Combiner C_{4P}. In this paper we first propose a combiner C_{4P} with optimal output length of $2n$ bits and which preserves all the properties of the C_{5P} combiner from [FL08], except for indifferentiability from random oracles. The basic idea of this construction is to

[1] Indifferentiability from a random oracle is sometimes also referred to as "being a pseudorandom oracle".

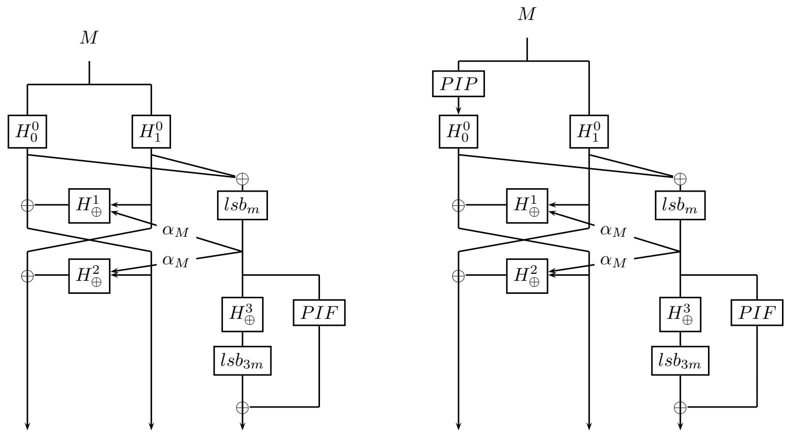

Figure 2: Illustration of the construction $\mathsf{C}_{\mathsf{4P\&IRO}}$ (left), which (besides the four properties preserved by C_{4P}) also preserves the IRO property, at the prize of an increased output length. The third branch of the construction operates on a signature value α_M depending on input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

use the concatenation combiner $\mathsf{C}_{\|}$, and to apply a three-round Feistel permutation to its output. 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 $\mathsf{C}_{\oplus}$ (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 $\mathsf{C}_{\|}$ still preserves the CR, TCR and MAC properties, e.g., collisions for $\mathsf{C}_{\|}$ are pulled through the downstream permutation and can be traced back to collisions for $\mathsf{C}_{\|}$. 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 $H_\oplus$ is a secure combiner for pseudorandom functions. The Luby-Rackoff [LR88] result now states that a three-round Feistel-network, instantiated with quasi independent 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 $\mathsf{C}_{\|}$ computation is infeasible.

Preserving IRO. In Section 4.2 we modify the C_{4P} construction such that it also preserves indifferentiability from a random oracle. The obstruction of the IRO robustness in the C_{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 $\mathsf{C}_{\mathsf{4P\&IRO}}$ is given in Figure 2. The signature α_M is taken as (a prefix of) the XOR of the output halves of the $\mathsf{C}_{\|}$ 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 interfer 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 still not known to be 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-resistance. In Section 5 we therefore show how to augment our constructions such they also preserves the one-wayness property.

The idea is that applying a pairwise-independent permutation (PIP) to the input of H_0 (or H_1) in the concatenation combiner $\mathsf{C}_{\|}$ makes this combiner also robust for one-wayness. Then we can use this modified concatenation combiner in the initial stages of our previous constructions, noting again the subsequent Feistel permutations do not interfere with this property either. Yet, as the description length of a PIP is linear in its input length, the input length of the derived combiners must be fixed, too, giving one-wayness as an additional property.

More Related Work. Although for most basic primitives black-box combiners are easily seen to exist, there are a few primitives for which combiners are not known, and there is strong evidence that they might not exist. Most notably commitment schemes [Her05] and oblivious transfer [HKN$^+$05, HIKN08, MPW07, MP06]. Note that those are primitives where the security notion is defined for two parties, e.g. for commitments, we have a hiding property for the committer, and a binding property for the receiver.

"Multi-property preservation" is not only interesting for combiners, but also for reductions between primitives. In particular Bellare and Ristenpart [BR06a] show how to construct a hash-function (taking as inputs messages of arbitrary length) from a fixed-input length compression function, while preserving multiple properties. One can use the construction of [BR06a] with the combiners from this paper in order to construct a hash function from two candidate compression functions, where the hash functions enjoys any (of the preserved) security properties, which is satisfied by at least one of the compression functions.

2 Preliminaries

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, and H for input H and $M \in \{0,1\}^*$ deterministically outputs the hash value $H(M) \in \{0,1\}^n$.

Depending on the security property we are interested in, the access of the adversary to the hash function is modeled differently. For unkeyed primitives like (target) collision-resistance or one-wayness, the description of H is given to the adversary. Whereas for keyed primitives like pseudorandomness or the MAC property, the adversary only gets black-box access to H.

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.

Below are formal definitions of the six important security properties for hash functions we consider in this work: the unkeyed properties of (target) collision-resistance and one-wayness and the keyed properties of being a PRF or a MAC. The final property – indifferentiability from a random oracle – is a bit special, as one considers idealized components. In particular, there's no efficient key-generation algorithm, but rather the hash function is given directly by an oracle.

collision resistance (CR): The hash function is called *collision-resistant* if for any efficient adversary $\mathcal{A}$ the probability that for $H \leftarrow \mathsf{HKGen}(1^n)$ and $(M, M') \leftarrow \mathcal{A}(H)$ we have $M \neq M'$ but $H(M) = H(M')$ is negligible (as a function of n).

target collision-resistance (TCR): A hash function is called *target collision-resistant* if any adversary defined by two efficient algorithms $(\mathcal{A}^1, \mathcal{A}^2)$, has negligible success probability of winning the following experiment. Let $\mathcal{A}^1(1^n)$ first generate the target message M and possibly some additional state information st. Then, a hash function $H \leftarrow \mathsf{HKGen}(1^n)$ is chosen and $\mathcal{A}^2$ on input (H, M, st) tries to compute a colliding message M'. The adversary wins if $M \neq M'$ but $H(M) = H(M')$.

one-wayness (OW) : A hash function is called *one-way* if for any efficient algorithm $\mathcal{A}$ the probability that for $H \leftarrow \mathsf{HKGen}(1^n)$ and for random M (chosen from some domain which is clear from the context) the probability that $\mathcal{A}(H, H(M))$ returns M' with $H(M') = H(M)$, is negligible.

pseudorandomness (PRF): A hash function is called *pseudorandom*, if for any efficient adversary $\mathcal{D}$ the advantage $|\,\mathsf{Pr}\left[\mathcal{D}^H(1^n) = 1\right] - \mathsf{Pr}\left[\mathcal{D}^f(1^n) = 1\right]\,|$ is negligible, where $H \leftarrow \mathsf{HKGen}(1^n)$ and f is a random function $f : \{0,1\}^* \rightarrow \{0,1\}^n$.

message authentication (MAC): A hash function is a *secure MAC* (where "secure" means unforgeable under a chosen message attack), if for any efficient adversary $\mathcal{A}$ the probability that for $H \leftarrow \mathsf{HKGen}(1^n)$ the adversary $\mathcal{A}^H$ (having oracle access to H) outputs (M, τ) where $\tau = H(M)$ and $\mathcal{A}$ did not make the oracle query M, is negligible.

indifferentiability from random oracles (IRO): Indifferentiability [MRH04, CDMP05] 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 $\mathcal{D}$ there exists an efficient algorithm $\mathcal{S}$ such that the advantage $\mathsf{Pr}\left[\mathcal{D}^{H^f, f}(H) = 1\right] - \mathsf{Pr}\left[\mathcal{D}^{\mathcal{F}, \mathcal{S}^{\mathcal{F}}(H)}(H) = 1\right]$ is negligible in n, where the probability in the first case is over $\mathcal{D}$'s coin tosses, $H \leftarrow \mathsf{HKGen}(1^n)$ and the choice of the random function f, and in the second case over the coin tosses of $\mathcal{D}$ and $\mathcal{S}$, and $H \leftarrow \mathsf{HKGen}(1^n)$ and over the choice of $\mathcal{F}$.

2.2 Robust Multi-Property Combiners

We now give a formal definition of robust multi-property combiners. A hash function combiner $\mathcal{C} = (\mathsf{CKGen}, \mathsf{C})$ 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. The concept of combiners for multiple properties $\textsc{prop} = \{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_N\}$ has been introduced in [FL08] and distinguishes between different levels of robustness. In the weakest case the combiner inherits a set of multiple properties if one of the hash functions is strong and has all the properties (weakly robust), whereas the strongest notion only requires that each property individually is provided by at least one of the two candidates (strongly robust). In between, there are mildly robust combiners for which one property may support the implementation of another property. In this paper we only consider strongly robust multi-property combiners. 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 hash function $\mathcal{H}$ has.

Definition 2.1 (Multi-Property Robustness) *A hash function combiner* $\mathcal{C} = (\mathsf{CKGen}, \mathsf{C})$ *is* strongly multi-property-robust *(sMPR) for a set* $\textsc{prop} = \{\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_N\}$ *of properties, if for any hash functions* $\mathcal{H}_0, \mathcal{H}_1$ *we have* $\mathsf{P}_i \in \textsc{prop}(\mathcal{H}_0) \cup \textsc{prop}(\mathcal{H}_1) \Longrightarrow \mathsf{P}_i \in \textsc{prop}(\mathcal{C}^{\mathcal{H}_0,\mathcal{H}_1})$.

In our construction the key-generation procedure CKGen of the combiner calls the key-generation procedure HKGen of $\mathcal{H}_0$ and $\mathcal{H}_1$, and possibly samples some more random variable h which will define a pairwise independent function or permutation. The sampled functions H_0, H_1 and h are then used in the evaluation procedure $\mathsf{C}^{H_0,H_1,h}$ as "black-boxes". For the IRO property we assume that the evaluation procedure is given access to the oracles directly. The security property then requires that $\mathsf{C}^{H_0,H_1,h}$ is indifferentiable from a random oracle if H_0 or H_1 is a random oracle, and the other oracle is arbitrary (but independent of the random oracle).

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 $H^i_\oplus(\cdot) = H^i_0(\cdot) \oplus H^i_1(\cdot)$ for $i = 2, 3$, with $H^i_b(\cdot)$ 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, i.e., $H^1_\oplus(X) = X$. In the i-th round the input (L_i, R_i) is mapped to the output $(R_i, L_i \oplus H^i_\oplus(R_i))$. We occasionally denote this Feistel permutation more explicitly by $\psi[H^1_\oplus, H^2_\oplus, H^3_\oplus](\cdot)$.

Our combiner, instantiated with hash functions $\mathcal{H}_0, \mathcal{H}_1$, is a pair of efficient algorithms $\mathcal{C}_{4\mathsf{P}} = (\mathsf{CKGen}_{4\mathsf{P}}, \mathsf{C}_{4\mathsf{P}})$ 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 $\mathsf{C}^{H_0,H_1}_{4\mathsf{P}}$ for parameters H_0, H_1 and input message M outputs

$$\mathsf{C}^{H_0,H_1}_{4\mathsf{P}}(M) = P^3(H^0_0(M) \| H^0_1(M)).$$

3.2 Multi-Property Robustness

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

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

Recall that a strong 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 $\mathsf{CR}, \mathsf{PRF}, \mathsf{TCR}$ and MAC is preserved independently.

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

Proof. Observe that any collision $M \neq M'$ for $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}(\cdot)$ directly gives a collision $00\|M \neq 00\|M'$ for for $H_0(\cdot)$ and $H_1(\cdot)$. Thus any adversary that finds collisions for $\mathcal{C}_{\mathsf{4P}}$ when instantiated with $\mathcal{H}_0, \mathcal{H}_1$ with non-negligible probability, can be used to find collision (with the same probability) for $\mathcal{H}_0$ and $\mathcal{H}_1$ 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 $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}(\cdot)$, this is also a collision for $H_b(\cdot)$. □

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

Proof. The proof is by contradiction. Assume an adversary $\mathcal{A}_{\mathsf{C}} = (\mathcal{A}_{\mathsf{C}}^1, \mathcal{A}_{\mathsf{C}}^2)$ that commits to a message M before getting H_0 and H_1 and then finds some M' such that $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}(M) = \mathsf{C}_{\mathsf{4P}}^{H_0,H_1}(M')$ with noticeable probability. Then we can use this attacker to construct a successful target-collision adversary $\mathcal{A}_b = (\mathcal{A}_b^1, \mathcal{A}_b^2)$ 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 $\mathcal{A}_b^1(1^n)$ runs $\mathcal{A}_{\mathsf{C}}^1(1^n)$ to receive the target message M and some state information st. $\mathcal{A}_b^1$ then commits to $00\|M$. On input H_b the adversary $\mathcal{A}_b^2$ 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 $\mathcal{A}_{\mathsf{C}}^2$. When $\mathcal{A}_{\mathsf{C}}^2$ outputs a message $M' \neq M$ with $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}(M) = \mathsf{C}_{\mathsf{4P}}^{H_0,H_1}(M')$ the adversary $\mathcal{A}_b^2$ returns $00\|M'$.

Due to the permutation a collision of M, M' for the combiner can be traced back to the input of $P(\cdot)$, i.e., $H_0(00\|M)\|H_1(00\|M) = H_0(00\|M')\|H_1(00\|M')$. 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 $\mathcal{A}_{\mathsf{C}}$. □

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 $H_\oplus^2, H_\oplus^3$ in the Feistel network $P^3 = \psi[H_\oplus^1, H_\oplus^2, H_\oplus^3]$ are instantiated with PRFs, as long as at least $\mathcal{H}_0$ or $\mathcal{H}_1$ is a PRF. Prepending the unique prefix $\langle i \rangle_2$ for $i = 2, 3$ to the input of $H_\oplus^i(\cdot) = H_\oplus(\langle i \rangle_2 \|\cdot)$ 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 $H_\oplus^2(\cdot)$ by xoring both input halves $H_0^0(M) \oplus H_1^0(M)$. 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 [LR88] and Naor-Reingold [NR99] 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 $\mathcal{H}_0$ or $\mathcal{H}_1$ is a PRF, then the initial concatenation combiner $\mathsf{C}_{\|}^{H_0,H_1}$ is weakly collision resistant[2], thus the probability that the adversary will invoke the combiner on distinct inputs M, M' where $H_0^0(M)\|H_1^0(M) = H_0^0(M')\|H_1^0(M')$, is negligible. So with overwhelming probability, all the adversary sees is the output of a PRP on distinct inputs. This distribution is indistinguishable from uniformly random (this follows from the PRP/PRF switching lemma [BR06b]), thus $\mathcal{C}_{4\mathsf{P}}$ is PRF robust.

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

□

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

Proof. Assume towards contradiction that an adversary $\mathcal{A}_{\mathsf{C}}$ with oracle access to the combiner $\mathsf{C}_{4\mathsf{P}}^{H_0,H_1}(\cdot)$ finds with non-negligible probability a valid pair (M,τ), such that $\tau = \mathsf{C}_{4\mathsf{P}}^{H_0,H_1}(M)$ but the message M was never queried to the MAC-oracle. Given $\mathcal{A}_{\mathsf{C}}$ 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 $\mathcal{A}_{\mathsf{C}}$ attack $\mathsf{C}_{4\mathsf{P}}^{H_0,H_1}(\cdot)$, 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 $\mathcal{A}_{\mathsf{C}}$'s oracle queries. When finally $\mathcal{A}_{\mathsf{C}}$ 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 $H_\oplus^i(\cdot) = H_0(\langle i\rangle_2 \|\cdot) \oplus H_1(\langle i\rangle_2 \|\cdot)$ 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 $i \in \{1,2,3\}$. By construction, if (M,τ) is a valid forgery for $\mathsf{C}_{4\mathsf{P}}^{H_0,H_1}(\cdot)$, then $H_0^0(M)\|H_1^0(M) = \tau_0\|\tau_1$ and thus $(00\|M, \tau_b)$ is a valid forgery for $H_b(\cdot)$. □

Why $\mathcal{C}_{4\mathsf{P}}$ is not a combiner for pseudorandom oracles (IRO). Finally, we give a brief idea why our combiner does not preserve IRO, unlike the robust multi-property combiner proposed in [FL08]. To be IRO-robust our $\mathsf{C}_{4\mathsf{P}}^{H_0,H_1}$ has to be indifferentiable from a random oracle for any efficient adversary $\mathcal{D}$, if H_b is a random oracle for some $b \in \{0,1\}$. Thereby

[2] 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.

the adversary $\mathcal{D}$ has oracle access either to the combiner $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}$ 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 $\mathcal{D}$ cannot have a significant advantage on deciding whether its interacting with $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}$ 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 $\mathcal{D}$ to simulate the construction of the combiner and then to precompute further answers that are consistent with the information $\mathcal{D}$ can get from $\mathcal{F}$. However, for $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}$ the simulator may be unable to precompute those consistent values, because an adversary $\mathcal{D}$ 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. That is, $\mathcal{D}$ 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 $\mathcal{D}$'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 $\mathcal{D}$ can distinguish between $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}$, H_b and $\mathcal{F}$, $\mathcal{S}^{\mathcal{F}}$ with noticeable probability.

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

In order to guarantee the IRO property, we modify the $\mathsf{C}_{\mathsf{4P}}^{H_0,H_1}$ 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 $\mathcal{C}_{\mathsf{4P\&IRO}} = (\mathsf{CKGen}_{\mathsf{4P\&IRO}}, \mathsf{C}_{\mathsf{4P\&IRO}})$ 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 $\mathsf{Pr}_{g \in G}[g(x) = z \wedge g(x') = z'] = |B|^{-2}$.*

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 $\mathsf{Pr}_{g \in G}[g(x) = z \wedge g(x') = z'] = \frac{1}{|B|(|B|-1)}$.

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 $GF(2^n)$ (if we want a smaller range $\{0,1\}^m, m < n$, 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 $\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,h}(M)$ first computes $C_{\|}^{H_0,H_1}(M) = H_0^0(M) \| H_1^0(M)$ and a value α_M – which we call the "signature of M" – as $\alpha_M = lsb_m(H_{\oplus}^0(M))$ where $H_{\oplus}^0(M) = H_0^0(M) \oplus H_1^0(M)$ 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 $P_{\alpha}^2(\cdot) = \psi[H_{\oplus}^1(\alpha_M \| \cdot), H_{\oplus}^2(\alpha_M \| \cdot)]$. Applying P_{α}^2 on $H_0^0(M) \| H_1^0(M)$ 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 $P_{\alpha_M}^2 \neq P_{\alpha_{M'}}^2$, and thus we cannot compute a collision for $C_{\|}^{H_0,H_1}$ (and thus for H_0 and H_1) from a collision $C_{\|}^{H_0,H_1}(P_{\alpha_M}^2(M)) = C_{\|}^{H_0,H_1}(P_{\alpha'_M}^2(M'))$.

To solve this problem, we could append the signature to the output of the combiner, and this way enforce that two inputs can only collide if they have the same signature. Unfortunately, outputting the signature α directly would make the permutation P_{α}^2 invertible, and ruin the IRO robustness of our construction. This is why we only output a "blinded" version of the signature computed as $lsb_{3m}(H_{\oplus}^3(\alpha_M)) \oplus g(\alpha_M)$. 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 $\alpha_M = lsb_m(H_{\oplus}^0(M))$ the following output:

$$\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,g}(M) = P_{\alpha}^2(H_0^0(M) \| H_1^0(M)) \parallel lsb_{3m}(H_{\oplus}^3(\alpha_M)) \oplus g(\alpha_M).$$

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. Like the random oracle H_b, also $H_{\overline{b}}$ is given as an oracle and accessible by all parties. The pairwise independent function h 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 $\mathsf{C}_{\mathsf{4P\&}\mathit{IRO}}$ is IRO-robust.*

Remark. Note that the security of $\mathsf{C}_{\mathsf{4P\&IRO}}$ 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 $\mathcal{D}$ has then access either to the combiner $\mathsf{C}_{\mathsf{4P\&IRO}}$ 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 $\mathcal{D}$ can distinguish between $\mathsf{C}_{\mathsf{4P\&IRO}}, H_0$ and $\mathcal{F}, \mathcal{S}^{\mathcal{F}}$ only with negligible probability.

The simulator keeps as state the function table of a (partially defined) function $\hat{H}_0 : \{0,1\}^* :\rightarrow \{0,1\}^n$, 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 $\mathcal{D}$'s point of view, $(\mathcal{F}, \hat{H}_0)$ look like $(\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,h}, H_0)$, i.e., the output of $\hat{H}_0$ has to be random and consistent to 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 $\mathcal{S}^{\mathcal{F}}_{H_1,f}(X)$:
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 $\hat{H}_0^0(X) = y_0$ 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 $\hat{H}_0^3(\alpha_M) = (U_3 \| z) \oplus (h(\alpha_M) \| 0^{n-3m}) \oplus H_1^3(\alpha_M)$
          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 $Y \leftarrow \hat{H}_0(X)$
```

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 $\mathcal{D}$ to simulate the construction of the combiner, such that the simulator has to compute in addition consistent answers to potential subsequent queries of $\mathcal{D}$. 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 $\hat{H}_0^1, \hat{H}_0^2$ and $\hat{H}_0^3$ such that $\mathsf{C}_{\mathsf{4P\&IRO}}^{\hat{H}_0,H_1,h}(M) = \mathcal{F}(M)$. 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 say the simulator "loses" if, for some $i \in \{1,2,3\}$, the function $\hat{H}_0^i$ is already defined on any input of the form $\alpha_M \| *$, such that $\mathcal{S}^{\mathcal{F}}$ cannot define all $\hat{H}_0^i$ values in order to provide consistent outputs.

As $\alpha_M \in \{0,1\}^m$ is uniformly random, the probability that the simulator loses in the q-th query is at most $3q \cdot 2^{-m}$ (as each $\hat{H}_0^i$ for $i \in \{1,2,3\}$ is defined on at most $q-1$ inputs). Let $\mathcal{E}$ denote the event that the simulator loses in any of its q queries, then the overall probability that $\mathcal{E}$ happens is at most $3q^2 \cdot 2^{-m}$. If $\mathcal{E}$ does not occur, the replies of $\mathcal{S}^{\mathcal{F}}$ are consistent with $\mathcal{F}$ and random, since $\mathcal{S}^{\mathcal{F}}$ answers are determined by its random choices and the replies of $\mathcal{F}$. Hence, the advantage of the adversary $\mathcal{D}$ in distinguishing $(\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,h}, H_0)$ from $(\mathcal{F}, \mathcal{S}^{\mathcal{F}})$ is at most the probability that event $\mathcal{E}$ happens, which is by $\Pr[\mathcal{E}] = 3q^2 \cdot 2^{-m}$ negligible. □

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. We often merely sketch the proofs since they are similar to the proofs for $\mathcal{C}_{\mathsf{4P}}$.

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 h, any collision for $\mathsf{C}^{H_0,H_1,f}_{\mathsf{4P\&IRO}}$ is simultaneously a collision for H_0^0 and H_1^0. To this end, let $M \neq M'$ be a collision for $\mathsf{C}^{H_0,H_1,h}_{\mathsf{4P\&IRO}}$ and let α_M and $\alpha_{M'}$ denote their signatures. Let $Y \| Y' = \mathsf{C}^{H_0,H_1,h}_{\mathsf{4P\&IRO}}(M)$ 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 $\mathsf{C}^{H_0,H_1,h}_{\mathsf{4P\&IRO}}(M) = \mathsf{C}^{H_0,H_1,h}_{\mathsf{4P\&IRO}}(M')$ 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 $h : \{0,1\}^m \to \{0,1\}^{3m}$, there does not exist a collision M, M' for $\mathsf{C}^{H_0,H_1,h}_{\mathsf{4P\&IRO}}$ 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

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

as this implies that the final outputs are distinct for any two messages with different signatures. As h 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 similiar 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 $\mathcal{A}_{\mathcal{C}}$ against the combiner that commits to a target message M and later outputs a colliding message M', one can build an adversary 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 ouput, our combiner $\mathsf{C}^{H_0,H_1,h}_{\mathsf{4P\&IRO}}$ applies a two-round Feistel network, which in general does not preserve (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 $\mathsf{Pr}[\alpha_M = \alpha_{M'}] = 2^{-m}$.

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

Algorithm $\mathcal{D}_0$ simulates the oracle of $\mathcal{D}_\mathcal{C}$, which is either $\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,h}$ or F, with his own oracle and the knowledge of $H_1 \leftarrow \mathsf{HKGen}_1, K_1$ and h that he samples accordingly. For each query of $\mathcal{D}_\mathcal{C}$, the adversary $\mathcal{D}_0$ computes an answer by emulating the combiner $\mathsf{C}_{\mathsf{4P\&IRO}}$ using $H_1(K_1, \cdot), h$ and his oracle which serves as H_0.

For the analysis recall that the underlying oracle of $\mathcal{D}_0$ is either a random function f or the hash function $H_0(K_0, \cdot)$. In the latter case $\mathcal{D}_0$ provides outputs that are identically distributed to the values $\mathcal{D}_\mathcal{C}$ would obtain from $\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,h}$. Hence, we have

$$\mathsf{Pr}[\mathcal{D}_0^{H_0}(H_0) = 1] = \mathsf{Pr}[\mathcal{D}_\mathcal{C}^{\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,h}}(H_0, H_1, h) = 1].$$

If the underlying oracle is the random function f, then the computed answers of $\mathcal{D}_0$ have to look like a truly random function as well. We show that this is true if, for q queries $M_1 \ldots M_q$ 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 $\mathcal{D}_0$ will create for each query M_i of $\mathcal{D}_\mathcal{C}$ a fresh signature α_{M_i}. To analyze the corresponding output of $\mathcal{D}_0$ we parse his answer in three parts, namely $\mathsf{C}_{\mathsf{4P\&IRO}}^{f,H_1,h}(M_i) = U_1\|U_2\|U_3$ with $|U_1| = |U_2| = n$ and $|U_3| = 3m$. The last part U_3 results from the computation $lsb_{3m}(f(11\|\alpha_{M_i}) \oplus H_1^3(\alpha_{M_i})) \oplus h(\alpha_{M_i})$. Since α_{M_i} is uniformly distributed and gets extended by the unique prefix 11, the input value of $f(11\|\alpha_{M_i})$ is distinct from all other queries to f during the $\mathsf{C}_{\mathsf{4P\&IRO}}^{f,H_1,h}(M_i)$ computation, and hence the corresponding output is an independently and uniformly distributed value. As xoring 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., $U_2 = f(00\|\alpha_{M_i}\|M_i) \oplus f(01\|\alpha_{M_i}\|Y) \oplus H_1^1(\alpha_{M_i}\|Y)$ for some $Y \in \{0,1\}^n$. Here we prepend the bits 00 and 01 respectively 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 $U_1 = Y' \oplus f(10\|\alpha_{M_i}\|Y'') \oplus H_1^2(\alpha_{M_i}\|Y'')$ for $Y', Y'' \in \{0,1\}^n$, where we use the unique prefix 10 when quering f in order to obtain values that are inpedendently and uniformly distributed. Thus, if for all queried messages $M_i \neq M_j$ of $\mathcal{D}_\mathcal{C}$ 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 $\mathcal{D}_\mathcal{C}$ satisfies

$$\mathsf{Pr}[\mathcal{D}_0^f(H_0) = 1] \leq \mathsf{Pr}[\mathcal{D}_\mathcal{C}^F(H_0, H_1, h) = 1] + q^2 \cdot 2^{-m}.$$

Thus, the probability that $\mathcal{D}_0$ 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 $\mathcal{C}_{\mathsf{4P\&IRO}}$ is MAC-robust.*

Proof. The proof is by contradiction. Assume that an adversary $\mathcal{A}_\mathcal{C}$ with oracle access to the combiner $\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,h}$ outputs with noticeable probability a valid pair (M, τ) where $\tau = \mathsf{C}_{\mathsf{4P\&IRO}}^{H_0,H_1,h}(M)$ 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 $H_{\overline{b}} \leftarrow \mathsf{HKGen}_1$ and chooses a random $K_{\overline{b}}$ that it uses together with its own oracle $H_b(K_b, \cdot)$ to answer all queries by $\mathcal{A}_\mathcal{C}$ in a black-box simulation. When $\mathcal{A}_\mathcal{C}$ returns a valid forgery (M, τ), where $M \neq M_1, M_2 \dots M_q$, 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 $(00\|M, \tau_b)$.
- If $c = 1$, then $\mathcal{A}_b$ queries its oracle about $00\|M$ 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 an successful adversary $\mathcal{A}_\mathcal{C}$. In the first case, $\mathcal{A}_\mathcal{C}$ returns a pair (M, τ), such that $\alpha_M = \alpha_{M_j}$ for some $j = 1, 2, \dots, q$, 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 $(00\|M, \tau_b)$ if it guesses the index j correctly and then inverts the Feistel step for input $lsb_{2n}(\tau)$ and α_{M_j}. The message $00\|M$ is distinct from all of $\mathcal{A}_b$'s queries, because $00\|M$ is distinct from all $00\|M_i$ and the additional queries of $\mathcal{A}_b$ start with a prefix $\langle i \rangle_2$ where $i \in 1, 2, 3$. Hence, if $\mathcal{A}_\mathcal{C}$ forges such a MAC with non-negligible probability ϵ, then $\mathcal{A}_b$ succeeds with probability $\epsilon/2q$.

In the second case, $\mathcal{A}_\mathcal{C}$ outputs (M, τ) where α_M has not occured 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 $\mathsf{C}_{\mathsf{4P\&IRO}}^{H_0, H_1, h}$ allows successful attacks on H_0 and H_1, contradicting the assumption that at least one hash function is a secure MAC. □

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

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

Recall that the concatenation combiner

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

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. On the other hand, the following combiner

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

is robust for tho OW property, i.e. $\mathsf{C}_{\mathsf{OW}}^{H_0,H_1}(M_L \| M_R)$ 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.

The basic idea to construct a combiner which is robust for CR and OW is to use the $\mathsf{C}_{\|}^{H_0,H_1}$ combiner, but 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 let us mention that any longer input length $kn, k > 5$ will work too (but then we'll also need $2kn$ bits for the description of P). Allowing shorter input length $kn, k < 5$ is not possible, as we use the fact that the input is (at least) $5n$ bits in the proof.

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 $\mathsf{CKGen}_{\mathsf{CR\&OW}}(1^n)$ generates $H_0 \leftarrow \mathsf{HKGen}_0(1^n)$ and $H_1 \leftarrow \mathsf{HKGen}_1(1^n)$ and picks a pairwise independent permutation $\pi : \{0,1\}^{5n} \rightarrow \{0,1\}^{5n}$. It outputs (H_0, H_1, π). The evaluation algorithm $\mathsf{C}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$ 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 $\mathcal{C}_{\|}$ and $\mathcal{C}_{\mathsf{OW}}$ simultaneously.

Theorem 5.1 *The combiner $\mathcal{C}_{\mathsf{CR\&OW}}$ is a strongly robust multi-property combiner for* 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 $\mathsf{C}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$, 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 $\mathsf{C}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$, we get a forgery $(\pi(M), \tau_0)$ for H_0 and a forgery (M, τ_1) for H_1 where $\tau_0 \| \tau_1 = \tau$. □

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

More precisely, we show that for any functions H_0, H_1 and any $T = T(n)$, the following is true for all but a $1/2T$ fraction of the π's: an adversary who inverts $\mathsf{C}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$ with probability $1/2T$, can be used to invert H_0 and H_1 with probability $1/2T^3$.[3]

Proof. We first need to relate the output of our combiner $\mathsf{C}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$ to the one of $\mathsf{C}_{\mathsf{OW}}^{H_0,H_1}$, depending on T. For this we call a tuple $(\pi_0, y_0 \| y_1)$ bad if it is more than $2T^2$ times more likely to be a key/output pair of $\mathsf{C}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi_0}$, compared to the combiner $\mathsf{C}_{\mathsf{OW}}^{H_0,H_1}(.) = H_0(.) \| H_1(.)$ and random permutation π. That is, $(\pi, y_0 \| y_1)$ is called *bad* iff

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

[3] Note that this statement implies that if either H_0 or H_1 is one-way and π is chosen at random, then $\mathsf{C}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}$ is one-way with overwhelming probability.

Equivalently,

$$\begin{aligned}&\Pr_{M}[H_0(\pi(M)) = y_0 | H_1(M) = y_1] \\ &\qquad \geq \quad 2 \cdot T^2 \cdot \Pr_{M_0,M_1}[H_0(M_0) = y_0 | H_1(M_1) = y_1]. \end{aligned} \tag{3}$$

We next bound the likelihood of a tuple to be bad in terms of the adversary's success probability (and running time):

Claim 1: $\Pr_{\pi,M}[(\pi, \mathsf{C}_{\mathsf{CR\&OW}}^{H_0,H_1,\pi}(M))$ is bad$] \leq 2/T^2$, where the probability is over the choice of the PIP $\pi : \{0,1\}^{5n} \to \{0,1\}^{5n}$ and $M \in \{0,1\}^{5n}$.

Proof. Letting $\mathcal{M}_0 = H_0^{-1}(y_0)$ and $\mathcal{M}_1 = H_1^{-1}(y_1)$ denote the pre-images of y_0 and y_1 under H_0 and H_1, respectively, and $\pi(\mathcal{M}_0)$ be the set of all $\pi(x)$ for $x \in \mathcal{M}_0$, we can bound the terms in (3) as:

$$\Pr_{M_0,M_1}[H_0(M_0) = y_0 | H_1(M_1) = y_1] \quad = \quad \frac{|\mathcal{M}_0|}{2^{5n}} \tag{4}$$

$$\Pr_{M}[H_0(\pi(M)) = y_0 | H_1(M) = y_1] \quad = \quad \frac{|\mathcal{M}_0 \cap \pi(\mathcal{M}_1)|}{|\mathcal{M}_1|} \tag{5}$$

The former equation is clear as we hit a pre-image of y_0 for the random M_0 with the given probability, and the latter follows as each of the possible pre-images of y_1 must be mapped via π to a pre-image of y_0.[4]

With equations (4),(5) and (3) we can rewrite the statement of the claim as

$$\Pr_{\pi,M}\left[\frac{|\mathcal{M}_0 \cap \pi(\mathcal{M}_1)|}{|\mathcal{M}_1|} \geq T^2 \cdot 2\frac{|\mathcal{M}_0|}{2^{5n}}\right] \leq \frac{2}{T^2}. \tag{6}$$

In order to prove this we consider for any $\mathcal{M}_0, \mathcal{M}_1$ and $\pi(M)$ the expected size of $|\mathcal{M}_0 \cap \pi(\mathcal{M}_1)|/|\mathcal{M}_1|$ (over the choice of π). First note that at least one element, namely $\pi(M)$, lies in $\mathcal{M}_0 \cap \pi(\mathcal{M}_1)$. For any other of the $|\mathcal{M}_1| - 1$ possible values $M' \in \mathcal{M}_1, M' \neq M$, the value $\pi(M')$ is uniformly distributed in $\{0,1\}^{5n} \setminus \pi(M)$, because π is a pairwise independent permutation. So the probability that $\pi(M')$ hits $\mathcal{M}_0$ is $(|\mathcal{M}_0| - 1)/(2^{5n} - 1)$ (observe that the term $|\mathcal{M}_0| - 1$ comes from the fact that $\pi(M) \in \mathcal{M}_0$ cannot be hit). Hence,

$$\mathbb{E}\left[\frac{|\mathcal{M}_0 \cap \pi(\mathcal{M}_1)|}{|\mathcal{M}_1|}\right] = \frac{1}{|\mathcal{M}_1|}\left(1 + \frac{(|\mathcal{M}_0| - 1)(|\mathcal{M}_1| - 1)}{2^{5n} - 1}\right). \tag{7}$$

For large $\mathcal{M}_0$ and $\mathcal{M}_1$ the right hand side of the previous equation converges towards (4). We are therefore interested in the probability that $\mathcal{M}_0$ and $\mathcal{M}_1$ are large. To derive this probability first note that for any function $f : \{0,1\}^{5n} \to \{0,1\}^n$ there are at most 2^n images y with $|f^{-1}(y)| \leq 2^{3n}$, and a random input M falls into such a bad set with probability at most $2^{4n}/2^{5n} = 2^{-n}$. As M and $\pi(M)$ are uniformly distributed, it follows that

$$\Pr[|\mathcal{M}_0| < 2^{3n} \vee |\mathcal{M}_1| < 2^{3n}] \leq 2 \cdot 2^{-n}. \tag{8}$$

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 $\mathcal{E}$. In this case

$$\frac{1}{|\mathcal{M}_1|}\left(1 + \frac{(|\mathcal{M}_0| - 1)(|\mathcal{M}_1| - 1)}{2^{5n} - 1}\right) \leq 2\frac{|\mathcal{M}_0|}{2^{5n}}, \tag{9}$$

[4] Note that $\mathcal{M}_1$ contains at least the element $H_1(M)$, so division by 0 cannot occur.

We can now prove (6) as (below $Z = |\mathcal{M}_0 \cap \pi(\mathcal{M}_1)|/|\mathcal{M}_1|$)

$$\Pr\left[Z \geq T^2 \cdot 2\frac{|\mathcal{M}_0|}{2^{5n}}\right] \leq \Pr[Z \geq T^2 \cdot \mathbb{E}[Z]|\mathcal{E}] + \Pr[\neg\mathcal{E}] \leq 1/T^2 + 2 \cdot 2^{-n} \leq 2/T^2$$

where we used (7)-(9) in the first and Markov's inequality in the second step. □

Using Markov's inequality once more the claim implies

$$\Pr_{\pi}[\Pr_{M}[(\pi, \mathcal{C}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(M)) \text{ is bad}] \leq 1/T] \geq 1 - 2/T. \tag{10}$$

We say that the permutation π is *good* if $\Pr_M[(\pi, \mathcal{C}_{\mathsf{CR\&OW}}(\pi, M))$ is bad$] \leq 1/T]$, thus by the above equation, a random π is good with probability at least $1 - 2/T$.

To conclude the proof, assume there exists an adversary $\mathcal{A}$ which inverts $\mathsf{C}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(.)$ with noticeable probability $\epsilon = 2/T$ for more than a $2/T$ fraction of the π's. Thus by equation (10), this must be the case for at least one good π. For this π, the output of $\mathsf{C}^{H_0,H_1,\pi}_{\mathsf{CR\&OW}}(.)$ is bad with probability at most $1/T$, thus $\mathcal{A}$ must invert with probability at least $\epsilon - 1/T$ even on outputs that are not bad. But then, by equation (3), it must also invert of $\mathsf{C}^{H_0,H_1}_{\mathsf{OW}}(M_0\|M_1)$ for random $M_0\|M_1$ with probability at least $(\epsilon - 1/T)/2T^2 = 1/2T^3$. □

5.2 Combining Things

We can now plug in 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 $\mathcal{C}_{\mathsf{4P\&OW}} = (\mathsf{CKGen}_{\mathsf{4P\&OW}}, \mathsf{C}_{\mathsf{4P\&OW}})$ 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 $\mathsf{C}^{\pi,H_0,H_1}_{\mathsf{4P\&OW}}$ 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* PROP $= \{\mathit{CR}, \mathit{PRF}, \mathit{TCR}, \mathit{MAC}, \mathit{OW}\}$.

When we apply the modifications from Section 5 and the combiner $\mathsf{C}_{\mathsf{4P\&IRO}}$ from Section 4 together, we get 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* PROP $= \{\mathit{CR}, \mathit{TCR}, \mathit{PRF}, \mathit{MAC}, \mathit{OW}, \mathit{IRO}\}$.

Acknowledgments

We thank Thomas Holenstein for a helpful conversation on one-way functions and Stefan Lucks for the suggestion of omitting the first round function in the C_{4P} combiner. We also thank the anonymous reviewers for valuable comments. The first two authors are supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG).

References

[BB06] Dan Boneh and Xavier Boyen. On the impossibility of efficiently combining collision resistant hash functions. In *Advances in Cryptology — Crypto 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 570–583. Springer-Verlag, 2006.

[BCJ+05] Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, and William Jalby. Collisions of SHA-0 and reduced SHA-1. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 36–57. Springer-Verlag, Berlin, Germany, May 2005.

[BF05] 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 *Lecture Notes in Computer Science*, pages 412–429. Springer-Verlag, 2005.

[BF06] Alexandra Boldyreva and Marc Fischlin. On the security of oaep. In *Advances in Cryptology — Asiacrypt 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 210–225. Springer-Verlag, 2006.

[BR94] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption — how to encrypt with rsa. In *Advances in Cryptology — Eurocrypt'94*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer-Verlag, 1994.

[BR96] Mihir Bellare and Phillip Rogaway. The exact security of digital signatures — how to sign with rsa and rabin. In *Advances in Cryptology — Eurocrypt'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer-Verlag, 1996.

[BR06a] Mihir Bellare and Thomas Ristenpart. Multi-property preserving hash domain extensions and the emd transform. In *Advances in Cryptology — Asiacrypt'06*, volume 4284 of *Lecture Notes in Computer Science*, pages 299–314. Springer-Verlag, 2006.

[BR06b] 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 *Lecture Notes in Computer Science*, pages 409–426. Springer-Verlag, 2006.

[CDMP05] Jean-Sebastien Coron, Yevgeniy Dodis, Cecile Malinaud, and Prashant Puniya. Merkle-damgard revisited: How to construct a hash function. In *Advances in Cryptology — Crypto 2005*, volume 3621 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[CRS$^+$07] Ran Canetti, Ronald L. Rivest, Madhu Sudan, Luca Trevisan, Salil P. Vadhan, and Hoeteck Wee. Amplifying collision resistance: A complexity-theoretic treatment. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 264–283. Springer-Verlag, Berlin, Germany, August 2007.

[FL08] Marc Fischlin and Anja Lehmann. Multi-property preserving combiners for hash functions. In *Theory of Cryptography Conference (TCC) 2008*, volume 4948 of *Lecture Notes in Computer Science*, pages 372–389. Springer-Verlag, 2008.

[FLN07] Pierre-Alain Fouque, Gaëtan Leurent, and Phong Q. Nguyen. Full key-recovery attacks on HMAC/NMAC-MD4 and NMAC-MD5. In Alfred Menezes, editor, *CRYPTO 2007*, volume 4622 of *LNCS*, pages 13–30. Springer-Verlag, Berlin, Germany, August 2007.

[Her05] Amir Herzberg. On tolerant cryptographic constructions. In *Topics in Cryptology — Cryptographer's Track, RSA Conference (CT-RSA) 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 172–190. Springer-Verlag, 2005.

[HIKN08] Danny Harnik, Yuval Ishai, Eyal Kushilevitz, and Jesper Buus Nielsen. OT-combiners via secure computation. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 393–411. Springer-Verlag, Berlin, Germany, March 2008.

[HKN$^+$05] 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 *Lecture Notes in Computer Science*, pages 96–113. Springer-Verlag, 2005.

[LR88] Michael Luby and Charles Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, 1988.

[MP06] Remo Meier and Bartosz Przydatek. On robust combiners for private information retrieval and other primitives. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 555–569. Springer-Verlag, Berlin, Germany, August 2006.

[MPW07] Remo Meier, Bartosz Przydatek, and Jürg Wullschleger. Robuster combiners for oblivious transfer. In Salil P. Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 404–418. Springer-Verlag, Berlin, Germany, February 2007.

[MRH04] 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 *Lecture Notes in Computer Science*, pages 21–39. Springer-Verlag, 2004.

[NR99] Moni Naor and Omer Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999.

[Pie07] Krzysztof Pietrzak. Non-trivial black-box combiners for collision-resistant hash-functions don't exist. In *Advances in Cryptology — Eurocrypt 2007*, Lecture Notes in Computer Science. Springer-Verlag, 2007.

[Pie08] Krzysztof Pietrzak. Compression from collisions, or why crhf combiners have a long output. In *Advances in Cryptology — Crypto 2008*, Lecture Notes in Computer Science. Springer-Verlag, 2008.

[WLF+05] Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the hash functions md4 and ripemd. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 2005.

[WY05] Xiaoyun Wang and Hongbo Yu. How to break md5 and other hash functions. In *Advances in Cryptology — Eurocrypt 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer-Verlag, 2005.

[WYY05] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full sha-1. In *Advances in Cryptology — Crypto 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer-Verlag, 2005.