Combining properties of cryptographic hash functions*

Michal Rjaško

Department of Computer Science
Faculty of Mathematics, Physics and Informatics
Comenius University
Mlynská dolina, 842 48 Bratislava, Slovak Republic
rjasko@dcs.fmph.uniba.sk

Abstract. A "strong" cryptographic hash function suitable for practical applications should simultaneously satisfy many security properties, like pseudo-randomness, collision resistance and unforgeability. This paper shows how to combine two hash function families each satisfying different security property into one hash function family, which satisfies both properties. In particular, given two hash function families H_1 and H_2, where H_1 is pseudo-random and H_2 is collision resistant, we construct a combiner which satisfies pseudo-randomness and collision resistance. We also present a combiner for collision resistance and everywhere preimage resistance. When designing a new hash function family for some particular application, we can use such combiners with existing primitives and thus combine a hash function family satisfying all needed properties.

1 Introduction

Cryptographic hash functions are used in many applications including digital signatures, message authentication and data integrity. Each application requires different set of properties, which a "strong" hash function should have simultaneously. Several methods of constructing hash function satisfying multiple properties have been proposed.

Bellare and Ristenpart [1, 2] suggest multi-property preserving (MPP) domain extension transforms. Such transforms extend domain of a "small" compression function $f : \{0,1\}^m \to \{0,1\}^n$ to a "big" hash function $F : \{0,1\}^* \to \{0,1\}^n$. Moreover MPP domain extension transforms preserve different security properties as long as they are all satisfied by the compression function f. The domain extension transforms thus reduce the problem of multi-property satisfaction from the hash function to the compression function, which can be easier to build.

Fischlin, Lehman and Pietrzak [4, 5] present robust MPP combiners. A robust combiner for two cryptographic hash functions F_1, F_2 and a property P (e.g. collision resistance) is a construction, which is secure (with respect to the property P) if at least one of the hash functions F_1 or F_2 is secure. A robust MPP combiner for two hash functions F_1, F_2 and the set of properties $\{P_1, \ldots, P_k\}$ preserves all the properties $P_1, \ldots, P_k$ if they are satisfied by at least one hash function F_1 and F_2 independently (i.e. it doesn't matter which properties are satisfied by F_1 and which by F_2). Thus robust MPP combiners make it possible to construct "fault tolerant" hash functions. If an attack against F_2 on a property P_i is found, the MPP combiner can still satisfy P_i as long as F_1 satisfies P_i. However, it was shown [3, 7, 8] that robust combiners for collision resistance must have at least twice as long output length as the partial hash functions F_1, F_2. This fact limits practical applicability of MPP combiners.

We present a different approach how to construct a hash function satisfying multiple properties. Given two hash function families H_1 and H_2, where H_1 is pseudo-random and H_2 is collision resistant, we construct a hash function family $C_1^{H_1,H_2}$, which preserves both

* Research supported by VEGA grant No. 1/0266/09 and Comenius University grant No. UK/429/2010.

collision resistance and pseudo-randomness. By the results of Rogaway-Shrimpton [10] and Rjaško [9], collision resistance implies second-preimage resistance, preimage resistance, target collision resistance and several other properties (cf. [9]). Moreover, pseudo-randomness implies unforgeability. Thus, our construction $C_1^{H_1,H_2}$ has all the mentioned properties as long as H_1 is pseudo-random and H_2 is collision resistant.

Moreover, we present a construction $C_2^{H_1,H_2}$, which is collision resistant and everywhere preimage resistant (cf. [10]) as long as H_1 is collision resistant and H_2 everywhere preimage resistant. By combining constructions C_1 and C_2 we get a construction $C_3 = C_1^{H_1,C_2^{H_2,H_3}}$, which preserves collision resistance, pseudo-randomness and everywhere preimage resistance.

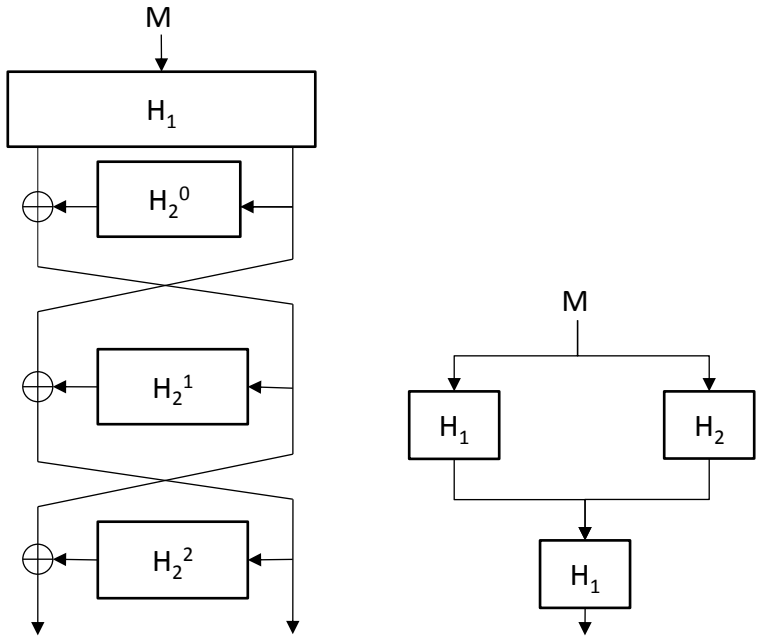

Fig. 1. Constructions combining properties Prf and Coll (left) and ePre and Coll (right).

Organization. We start the Section 2 by introducing some basic notations and definitions. Then we present definitions of nine basic security properties of cryptographic hash functions. The properties were introduced/used in [1, 2, 4, 10]. In the Section 3 we introduce the combiner C_1 preserving pseudo-randomness and collision resistance. The Section 4 introduces and analyzes the combiner C_2 for collision resistance and everywhere preimage resistance. Finally, in the Section 5 we show that these two combiners can be combined into the combiner, which preserves pseudo-randomness, collision resistance and everywhere preimage resistance.

2 Preliminaries

We write $M \xleftarrow{\$} \mathcal{S}$ for the uniform random selection of M from the finite set $\mathcal{S}$. Concatenation of finite strings M_1 and M_2 is denoted by $M_1||M_2$ or simply M_1M_2, $\overline{M}$ denotes bitwise complement of string M. Let $\mathrm{Func}(D, R)$ represent the set of all functions $\rho : D \to R$ and

let $RF_{D,R}$ be a function chosen randomly from the set $\mathrm{Func}(D,R)$ (i.e. $RF_{D,R} \xleftarrow{\$} \mathrm{Func}(D,R)$). By $\mathrm{Perm}(R)$ we denote the set of all permutations $\phi : R \to R$. Let RP_R be a permutation chosen randomly from the set $\mathrm{Perm}(R)$. We sometimes write $RF_{d,r}$, $\mathrm{Func}(d,r)$ or $\mathrm{Perm}(r)$ when $D = \{0,1\}^d$ and $R = \{0,1\}^r$. If i is an integer, then $\langle i \rangle_r$ is r-bit string representation of i. By $\mathrm{Prefix}_n(M)$ we denote the n-bit prefix of string M.

Let $n \in \mathbb{N}$ be a security parameter. A hash function family is a function $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ computable in polynomial time, where $k, y \in \mathbb{N}$ are polynomially related to the security parameter n (i.e. $k = p_1(n)$ and $y = p_2(n)$ for some polynomials p_1, p_2). We will often write the first argument to H as a subscript, i.e. $H_K(M) := H(K,M)$.

A function f is negligible if for every polynomial $p(\cdot)$ there exists N such that for every $n > N$ it holds that $f(n) < \frac{1}{p(n)}$. Negligible functions are denoted as $\mathrm{negl}(\cdot)$.

An oracle Turing machine T with oracle access to Turing machines $T_1, \ldots, T_l$ is a Turing machine, which accepts inputs via input tape, performs some computation and replies via output tape. During the computation it can write on some additional "oracle" input tapes $t_1, \ldots, t_l$ and receives responses via "oracle" output tapes $t'_1, \ldots, t'_l$ – connections to the Turing machines $T_1, \ldots, T_l$. Whenever T writes some input on tape t_i, the Turing machine T_i is run on that input and T receives the output on tape t'_i. We call such a operation a query to oracle T_i. All queries are performed in unit time (i.e. computation of T_i is not counted into the running time of T). The fact that T has oracle access to $T_1, \ldots, T_l$ is denoted as $T^{T_1,\ldots,T_l}$.

An adversary is a probabilistic polynomial-time oracle Turing machine. Running time of an adversary A is the worst case running time of A plus the description size of A (hence one cannot precompute some large amount of information and store it into A's description). Running time of an adversary is polynomial in length of its inputs and the security parameter n. Without loss of generality we assume that an adversary always stop and returns some output.

Security notions. Below are definitions of the nine important security properties we consider in this work (cf. [1, 2, 4, 10]). Let $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be a hash function family and let λ be a positive integer. Let A be an adversary. We define the following advantage measures:

$$\mathbf{Adv}^{\mathrm{Coll}}_H(A) = \Pr\Big[K \xleftarrow{\$} \{0,1\}^k; (M, M') \leftarrow A(K) : (M \neq M') \wedge (H_K(M) = H_K(M'))\Big]$$

$$\mathbf{Adv}^{\mathrm{Pre}[\lambda]}_H(A) = \Pr\Big[K \xleftarrow{\$} \{0,1\}^k; M \xleftarrow{\$} \{0,1\}^\lambda; Y \leftarrow H_K(M); M' \leftarrow A(K,Y) : H_K(M') = Y\Big]$$

$$\mathbf{Adv}^{\mathrm{aPre}[\lambda]}_H(A) = \Pr\Big[(K,S) \leftarrow A; M \xleftarrow{\$} \{0,1\}^\lambda; Y \leftarrow H_K(M); M' \leftarrow A(Y,S) : H_K(M') = H_K(M)\Big]$$

$$\mathbf{Adv}^{\mathrm{ePre}}_H(A) = \Pr\Big[(Y,S) \leftarrow A; K \xleftarrow{\$} \{0,1\}^k; M' \leftarrow A(K,S) : H_K(M') = Y\Big]$$

$$\mathbf{Adv}^{\mathrm{Sec}[\lambda]}_H(A) = \Pr\Big[K \xleftarrow{\$} \{0,1\}^k; M \xleftarrow{\$} \{0,1\}^\lambda; M' \leftarrow A(K,M) : (M \neq M') \wedge (H_K(M) = H_K(M'))\Big]$$

$$\mathbf{Adv}_H^{\mathrm{aSec}[\lambda]}(A) = \Pr\Big[(K,S) \leftarrow A; M \xleftarrow{\$} \{0,1\}^\lambda; M' \leftarrow A(M,S) : (M \neq M') \wedge (H_K(M) = H_K(M'))\Big]$$

$$\mathbf{Adv}_H^{\mathrm{eSec}}(A) = \Pr\Big[(M,S) \leftarrow A; K \xleftarrow{\$} \{0,1\}^k; M' \leftarrow A(K,S) : (M \neq M') \wedge (H_K(M) = H_K(M'))\Big]$$

$$\mathbf{Adv}_H^{\mathrm{Prf}}(A) = \Big|\Pr\Big[K \xleftarrow{\$} \{0,1\}^k; 1 \leftarrow A^{H_K}\Big] - \Pr\Big[\mathcal{F} \xleftarrow{\$} \mathrm{Func}(*,y); 1 \leftarrow A^{\mathcal{F}}\Big]\Big|$$

$$\mathbf{Adv}_H^{\mathrm{MAC}}(A) = \Pr\Big[K \xleftarrow{\$} \mathcal{K}; (M,Y) \leftarrow A^{H_K} : H_K(M) = Y \wedge M \text{ not queried}\Big]$$

We say that H is xxx secure (or H is xxx) for xxx $\in$ {Pre, aPre, Sec, aSec} if for all λ and any polynomial adversary A there exists a negligible function negl, such that

$$\mathbf{Adv}_H^{\mathrm{xxx}[\lambda]}(A) \leq \mathrm{negl}(n).$$

For yyy $\in$ {eSec, ePre, Coll, Prf} we say that H is yyy secure if for any polynomial adversary A there exists a negligible function negl, such that

$$\mathbf{Adv}_H^{\mathrm{yyy}}(A) \leq \mathrm{negl}(n).$$

Pseudo-random permutation. We say function $f : \{0,1\}^k \times \{0,1\}^y \to \{0,1\}^y$ is keyed permutation if $f_K(\cdot)$ is bijective for all $K \in \{0,1\}^k$. A keyed permutation f is pseudo-random if for any polynomial adversary A there exists a negligible function negl, such that

$$\Big|\Pr\Big[K \xleftarrow{\$} \{0,1\}^k; 1 \leftarrow A^{f_K}\Big] - \Pr\Big[\mathcal{P} \xleftarrow{\$} \mathrm{Perm}(y); 1 \leftarrow A^{\mathcal{P}}\Big]\Big| \leq \mathrm{negl}(n).$$

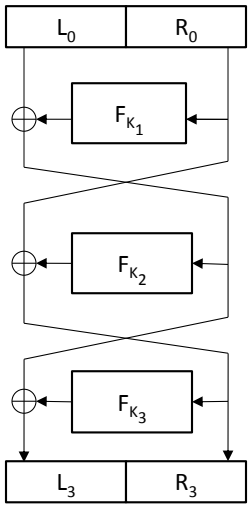

Fig. 2. Three round Feistel network $\mathrm{Feistel}_{F_{K_1},F_{K_2},F_{K_3}}$ is pseudo-random permutation if F is a pseudo-random function.

Feistel permutation. A Feistel network is a way of constructing invertible functions from possibly non-invertible ones. It operates in series of rounds. Input into the round i is divided into two halves L_{i-1} and R_{i-1}. Output of the round i is defined as

$$L_i := R_{i-1} \qquad \text{and} \qquad R_i = L_{i-1} \oplus f_i(R_{i-1}),$$

where $f_i : \{0,1\}^n \to \{0,1\}^n$ is a round function. By $\text{Feistel}_{f_1,\ldots,f_r}$ we denote the r-round Feistel network with round functions $f_1, \ldots, f_r$. Hence $\text{Feistel}_{f_1,\ldots,f_r}(L_0, R_0)$ outputs $2n$ bit string (L_r, R_r).

It is easy to see that Feistel network is invertible. Moreover, it can be shown that if a pseudo-random function $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is used instead of round functions, then the 3-round $\text{Feistel}_{F_{K_1},F_{K_2},F_{K_3}}$ is pseudo-random permutation.

Proposition 1 ([6]). *If $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3 \xleftarrow{\$} \text{Func}(n,n)$ are three independent random functions, then*

$$\text{Feistel}_{\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_3} : \{0,1\}^{2n} \to \{0,1\}^{2n}$$

is indistinguishable from a random permutation. If $F : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a pseudo-random function then the function

$$F^{(3)}_{K_1,K_2,K_3} := \text{Feistel}_{F_{K_1},F_{K_2},F_{K_3}}$$

is a pseudo-random permutation.

3 Construction C_1 for Coll and Prf

In this section we introduce a construction C_1 illustrated in the Figure 1. It utilizes ideas of Fischlin and Lehman [5] for the $C_{4\text{P}}$ robust MPP combiner, which is robust for Coll, Prf, MAC and eSec.

Let $H_1, H_2 : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be two hash function families. We will assume that H_1 is Prf secure and H_2 is Coll secure. Let C_1 be defined as:

$$C_1^{H_1,H_2}(K_1, K_2, M) := \text{Feistel}_{H_1^{(1)},H_1^{(2)},H_1^{(3)}}(H_2(K_2, M)),$$

where

$$H_1^{(i)}(M) := \text{Prefix}_{y/2}(H_1(K_1, \langle i \rangle_2 || M)).$$

Idea behind the construction is to apply a pseudo-random permutation over the collision resistant hash function family. This leads to a hash function family which is pseudo-random and collision resistant.

Lemma 1. *Let $H : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be a collision resistant hash function family and $f : \{0,1\}^k \times \{0,1\}^y \to \{0,1\}^y$ be a pseudo-random permutation. Then a hash function family $C : \{0,1\}^{2k} \times \{0,1\}^* \to \{0,1\}^y$ defined as*

$$C_{K_1,K_2}(M) := f_{K_1}(H_{K_2}(M))$$

is collision resistant and pseudo-random.

Proof. Fix some security parameter n. Since f_{K_1} is permutation for all $K_1 \in \{0,1\}^k$, any collision (M, M') in C_{K_1,K_2} also collides in H_{K_2}:

$$H_{K_2}(M) = f_{K_1}^{-1}(C_{K_1,K_2}(M)) = f_{K_1}^{-1}(C_{K_1,K_2}(M')) = H_{K_2}(M').$$

Hence, if H is Coll secure, then also C is.

Let A be a polynomial adversary and let

$$\varepsilon(n) := \mathbf{Adv}_C^{\mathrm{Prf}}(A).$$

The adversary A has oracle access either to C_{K_1,K_2} for $K_1, K_2 \xleftarrow{\$} \{0,1\}^k$ or to a random function $\mathcal{F} \xleftarrow{\$} \mathrm{Func}(*, y)$. Now consider the following adversary D:

Adversary D

D has oracle access either to f_{K_1} for $K_1 \xleftarrow{\$} \{0,1\}^k$ or to $\mathcal{P} \xleftarrow{\$} \mathrm{Perm}(y)$. Let $\mathcal{O}$ denote D's current oracle.

1. Choose $K_2 \xleftarrow{\$} \{0,1\}^k$.
2. Simulate an adversary A. When A asks its oracle a query M, answer $\mathcal{O}(H_{K_2}(M))$.
3. When A outputs a bit $b \in \{0,1\}$, output b and end.

It is clear that D runs in a polynomial time. When D's oracle is f_{K_1}, then view of A in the simulation is the same as in the Prf experiment with oracle C_{K_1,K_2}. Hence,

$$\Pr\left[K_1 \xleftarrow{\$} \{0,1\}^k; 1 \leftarrow D^{f_{K_1}}\right] = \Pr\left[K_1, K_2 \xleftarrow{\$} \{0,1\}^k; 1 \leftarrow A^{C_{K_1,K_2}}\right].$$

Consider the part when D has oracle access to the random permutation $\mathcal{P}$. The adversary A cannot find a difference between oracles $\mathcal{P}(H_{K_2}(\cdot))$ and $\mathcal{F}(\cdot)$, unless it asks two distinct queries M, M' for which it gets the same answer (otherwise outputs of both $\mathcal{P}$ and $\mathcal{F}$ are uniformly random). This case can occur only with negligible probability if A's oracle is $\mathcal{F}$. If A's oracle is $\mathcal{P}(H_{K_2}(\cdot))$, then M, M' also collides in H_{K_2}. Since H_{K_2} is collision resistant, A can find such M, M' only with negligible probability. Thus, there exists a negligible function negl for which

$$\left|\Pr\left[\mathcal{P} \xleftarrow{\$} \mathrm{Func}(y, y); 1 \leftarrow D^{\mathcal{P}}\right] - \Pr\left[\mathcal{F} \xleftarrow{\$} \mathrm{Func}(*, y); 1 \leftarrow A^{\mathcal{F}}\right]\right| \leq \mathrm{negl}(n).$$

Hence,

$$\begin{aligned}
&\left|\Pr\left[K_1 \xleftarrow{\$} \{0,1\}^k; 1 \leftarrow D^{f_{K_1}}\right] - \Pr\left[\mathcal{P} \xleftarrow{\$} \mathrm{Func}(y, y); 1 \leftarrow D^{\mathcal{P}}\right]\right| \\
&\quad \geq \left|\Pr\left[K_1, K_2 \xleftarrow{\$} \{0,1\}^k; 1 \leftarrow A^{C_{K_1,K_2}}\right] - \Pr\left[\mathcal{F} \xleftarrow{\$} \mathrm{Func}(*, y); 1 \leftarrow A^{\mathcal{F}}\right]\right| + \mathrm{negl}(n) \\
&\quad = \varepsilon(n) + \mathrm{negl}(n).
\end{aligned}$$

Since f is a pseudo-random permutation, we conclude that $\varepsilon(n)$ must be negligible. □

Lemma 2. *If $F : \{0,1\}^k \times \{0,1\}^y \to \{0,1\}^n$ is a pseudo-random function, then the function*

$$F'(K, M) := \mathit{Prefix}_l(F(K, M)),$$

where $l \leq n$, is pseudo-random too.

Proof. (sketch) The proof is straightforward. If some adversary A can distinguish F' from a random function, then it can do the same for F by looking on the corresponding part of its output. Hence, truncating output bits of a pseudo-random function does not affect its pseudo-randomness. □

The following theorem is an easy consequence of Lemma 1, 2 and Proposition 1.

Theorem 1. *Let $H_1 : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be a collision resistant hash function family and $H_2 : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be a pseudo-random hash function family. Then the hash function family C_1 defined as*

$$C_1^{H_1,H_2}(K_1, K_2, M) := \text{Feistel}_{H_1^{(1)},H_1^{(2)},H_1^{(3)}}(H_2(K_2, M)),$$

where

$$H_1^{(i)}(M) := \mathit{Prefix}_{y/2}(H_1(K_1, \langle i\rangle_2||M)), \qquad i = 1, 2, 3$$

is collision resistant and pseudo-random.

Proof. Prepending the round prefix $\langle i\rangle_2$, $i = 1, 2, 3$, to the input of pseudo-random function H_1 ensures that $H_1(K_1, \cdot)$ is never invoked on the same input in different rounds. This means the functions $H_1(K_1, \langle i\rangle_2||\cdot)$ are indistinguishable from three independent random functions. Hence, by Lemma 2 functions

$$H_1^{(i)}(M) := \text{Prefix}_{y/2}(H_1(K_1, \langle i\rangle_2||M)),$$

are indistinguishable from three independent random functions. By Proposition 1 it means that $\text{Feistel}_{H_1^{(1)},H_1^{(2)},H_1^{(3)}}$ is a pseudo-random permutation. Using the Lemma 1 we conclude that C_1 is collision resistant and pseudo-random. □

Remark 1. In [9, 10] it was proven, that if some hash function family H is Coll secure, it is also Pre, Sec and eSec secure. Similarly, if H is Prf, then it is also MAC secure. Hence, our construction $C_1^{H_1,H_2}$ preserves all these properties, as long as H_1 is Prf and H_2 is Coll.

Remark 2. We apply the pseudo-random permutation $\text{Feistel}_{H_1^{(1)},H_1^{(2)},H_1^{(3)}}$ over the collision resistant hash function H_2 to ensure that collisions in the combiner are also collisions in the collision resistant hash function H_2. Consider that we apply a pseudo-random function instead of the pseudo-random permutation, i.e. $C_1' = H_1(K_1, H_2(K_2, M))$ where H_1 is Prf and H_2 is Coll secure. Collisions in such a combiner can not be directly transformed into collisions in the hash function H_2, since H_1 is not a permutation. On the other hand, this combiner is much less complicated and it seems, that pseudo-randomness of H_1 ensures that collisions in the combiner are still hard to find, if H_2 is collision resistant. However, formal proof that such a combiner is collision resistant remains an open problem.

4 Construction C_2 for Coll and ePre

In this section we introduce the construction $C_2^{H_1,H_2}$ for two hash function families H_1 and H_2. We show that if H_1 is Coll and H_2 is ePre secure, then $C_2^{H_1,H_2}$ is both Coll and ePre secure. The construction is illustrated in Figure 1.

Let $H_1, H_2 : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be two hash function families. We will assume that H_1 is Coll secure and H_2 is ePre secure. Let C_2 be defined as:

$$C_2^{H_1,H_2}(K_1, K_2, M) := H_1(K_1, H_1(K_1, M)||H_2(K_2, M)).$$

We show that C_2 is Coll and ePre secure.

Theorem 2. *Let $H_1, H_2 : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be two hash function families. If H_1 is Coll secure, then $C_2^{H_1,H_2}$ is Coll secure.*

Proof. Fix some security parameter n. Let A be a polynomial adversary and let

$$\varepsilon(n) := \mathbf{Adv}_{C_2}^{\text{Coll}}(A).$$

From A we construct an adversary B attacking H_1 in Coll sense.

Adversary B

B is given on input a key $K \xleftarrow{\$} \{0,1\}^k$

1. Choose $K_2 \xleftarrow{\$} \{0,1\}^k$.
2. Simulate $A(K, K_2)$. At the end of its execution, A outputs a pair M, M'.
3. If $H_1(K, M) = H_1(K, M')$, output M, M'.
   Otherwise output $H_1(K, M)||H_2(K_2, M), H_1(K, M')||H_2(K_2, M')$.

It is clear that B runs in a polynomial time. Consider that the pair M, M', which A outputs at the end of its simulation collides for $C_2^{H_1,H_2}$. That is $M \neq M'$ and

$$H_1(K, H_1(K, M)||H_2(K_2, M)) = H_1(K, H_1(K, M')||H_2(K_2, M')).$$

If $H_1(K, M)||H_2(K_2, M) \neq H_1(K, M')||H_2(K_2, M')$, then the pair

$$(H_1(K, M)||H_2(K_2, M), H_1(K, M')||H_2(K_2, M'))$$

is a collision for H_1. Otherwise it must hold, that $H_1(K, M) = H_1(K, M')$. This means M, M' collides also for H_1. Hence, if A finds a collision for $C_2^{H_1,H_2}$, then B finds a collision for H_1. Thus,

$$\varepsilon(n) \leq \mathbf{Adv}_{H_1}^{\text{Coll}}(B).$$

Since H_1 is collision resistant, we conclude that $\varepsilon(n)$ is negligible. □

Theorem 3. *Let $H_1, H_2 : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be two hash function families. If H_1 is Coll and H_2 is ePre secure, then $C_2^{H_1,H_2}$ is ePre secure.*

Proof. We will use an equivalent "one stage" definition of the ePre advantage measure, we denote this definition as ePre2:

$$\mathbf{Adv}_H^{\text{ePre2}}(A) = \max_{Y \in \{0,1\}^y} \left\{ \Pr\left[K \xleftarrow{\$} \{0,1\}^k; M \leftarrow A(K) : H_K(M) = Y\right] \right\}.$$

For the proof of equivalence between ePre and ePre2 see [10]. Fix some security parameter n. Let A be a polynomial adversary and let

$$\varepsilon(n) := \mathbf{Adv}_{C_2}^{\text{ePre2}}(A).$$

Let Y' be the image for which A has the maximum probability of success, i.e.

$$\Pr\left[K_1, K_2 \xleftarrow{\$} \{0,1\}^k; M \leftarrow A(K_1, K_2) : C_2^{H_1,H_2}(K_1, K_2, M) = Y'\right] = \varepsilon(n)$$

To make our presentation more succinct, let $\mathrm{AWins}(K_1, K_2)$ be the shortcut for the event that A wins given keys K_1 and K_2, i.e.

$$\mathrm{AWins}(K_1, K_2) \Leftrightarrow M \leftarrow A(K_1, K_2) \wedge C_2^{H_1,H_2}(K_1, K_2, M) = Y'$$

. Consider the following adversary B, which attacks H_2 in the ePre sense.

Adversary B
[1st stage]

1. Choose $K_1', K_2' \xleftarrow{\$} \{0,1\}^k$.
2. Simulate $M' \leftarrow A(K_1', K_2')$.
3. Compute $Y := H_2(K_2', M')$ and output (Y, K_1').

[2nd stage]

B is given on input a key $K_2 \xleftarrow{\$} \{0,1\}^k$ and "state" variable from the previous stage K_1'.

1. Simulate $M \leftarrow A(K_1', K_2)$.
2. Output M.

It is clear that B runs in a polynomial time. Consider that A outputs a valid preimage for Y' in both simulations, i.e. $C_2^{H_1,H_2}(K_1', K_2', M') = C_2^{H_1,H_2}(K_1', K_2, M) = Y'$. From the definition of C_2 we have

$$Y' = H_1(K_1', H_1(K_1', M')||H_2(K_2', M')) = H_1(K_1', H_1(K_1', M)||H_2(K_2, M)).$$

If $H_2(K_2', M') = H_2(K_2, M)$, then M is a valid preimage for Y, i.e. B wins. Otherwise, the pair

$$H_1(K_1', M')||H_2(K_2', M'), H_1(K_1', M)||H_2(K_2, M)$$

collides for $H_1(K_1', \cdot)$. Since H_1 is collision resistant, this case can occur only with negligible probability $\mathrm{negl}_1(n)$. Let $\mathcal{E}_1$ denote the event $Y' = C_2^{H_1,H_2}(K_1', K_2', M')$ (i.e. A wins in the first simulation) and $\mathcal{E}_2$ be the event $Y' = C_2^{H_1,H_2}(K_1', K_2, M)$ (i.e. A wins in the second simulation). Hence,

$$\begin{aligned}
\mathbf{Adv}_{H_2}^{\mathrm{ePre}}(B) &\geq \Pr[K_1, K_2, K_2' \xleftarrow{\$} \{0,1\}^k; \mathcal{E}_1 \wedge \mathcal{E}_2 \wedge H_2(K_2', M') = H_2(K_2, M)] \\
&= \Pr[K_1, K_2, K_2' \xleftarrow{\$} \{0,1\}^k; \mathcal{E}_1 \wedge \mathcal{E}_2] \\
&\qquad - \Pr[K_1, K_2, K_2' \xleftarrow{\$} \{0,1\}^k; \mathcal{E}_1 \wedge \mathcal{E}_2 \wedge H_2(K_2', M') \neq H_2(K_2, M)] \\
&\geq \Pr[K_1, K_2, K_2' \xleftarrow{\$} \{0,1\}^k; \mathcal{E}_1 \wedge \mathcal{E}_2] - \mathrm{negl}_1(n) \qquad (1)
\end{aligned}$$

We find the lower bound for the first member of the equation (1). The events $\mathcal{E}_1$ and $\mathcal{E}_2$ share the same randomly chosen key K_1, hence they are not independent. However,

$$\begin{aligned}
&\Pr[K_1, K_2, K_2' \xleftarrow{\$} \{0,1\}^k; \mathcal{E}_1 \wedge \mathcal{E}_2] = \\
&\quad = \frac{1}{2^k} \sum_{K_1 \in \{0,1\}^k} \Pr[K_2, K_2' \xleftarrow{\$} \{0,1\}^k; \mathcal{E}_1 \wedge \mathcal{E}_2] && (2)\\
&\quad = \frac{1}{2^k} \sum_{K_1 \in \{0,1\}^k} \Pr[K_2 \xleftarrow{\$} \{0,1\}^k; \mathcal{E}_1] \cdot \Pr[K_2' \xleftarrow{\$} \{0,1\}^k; \mathcal{E}_2] && (3)\\
&\quad = \frac{1}{2^k} \sum_{K_1 \in \{0,1\}^k} \Big(\Pr\left[K_2 \xleftarrow{\$} \{0,1\}^k; \mathrm{AWins}(K_1, K_2)\right] \Big)^2, && (4)
\end{aligned}$$

where equation (3) is given by the fact, that the events $\mathcal{E}_1$ and $\mathcal{E}_2$ are independent if the key K_1 is fixed.

Let $\mathrm{GOOD} \subseteq \{0,1\}^k$ denote the set of keys K_1 for which the probability that A wins is at least $\varepsilon(n)/2$. That is

$$\forall K_1 \in \mathrm{GOOD} : \Pr\left[K_2 \xleftarrow{\$} \{0,1\}^k; M \leftarrow A(K_1, K_2) : C_2^{H_1,H_2}(K_1, K_2, M) = Y'\right] \geq \frac{\varepsilon(n)}{2}.$$

Let BAD be the set of all other keys, i.e. $\mathrm{BAD} = \{0,1\}^k - \mathrm{GOOD}$. The left hand side of the equation (4) can be bounded from below as follows:

$$\begin{aligned}
&\frac{1}{2^k} \sum_{K_1 \in \{0,1\}^k} \Big(\Pr\left[K_2 \xleftarrow{\$} \{0,1\}^k; \mathrm{AWins}(K_1, K_2)\right] \Big)^2 = \\
&\quad = \frac{1}{2^k} \sum_{K_1 \in \mathrm{GOOD}} \Big(\Pr\left[K_2 \xleftarrow{\$} \{0,1\}^k; \mathrm{AWins}(K_1, K_2)\right] \Big)^2 + \\
&\qquad + \frac{1}{2^k} \sum_{K_1 \in \mathrm{BAD}} \Big(\Pr\left[K_2 \xleftarrow{\$} \{0,1\}^k; \mathrm{AWins}(K_1, K_2)\right] \Big)^2 \\
&\quad \geq \frac{1}{2^k} \sum_{K_1 \in \mathrm{GOOD}} \frac{\varepsilon(n)^2}{4}. && (5)
\end{aligned}$$

On the other hand, we know that

$$\begin{aligned}
\varepsilon(n) &= \frac{1}{2^k} \sum_{K_1 \in \mathrm{GOOD}} \Pr\left[K_2 \xleftarrow{\$} \{0,1\}^k; \mathrm{AWins}(K_1, K_2)\right] + \\
&\quad + \frac{1}{2^k} \sum_{K_1 \in \mathrm{BAD}} \Pr\left[K_2 \xleftarrow{\$} \{0,1\}^k; \mathrm{AWins}(K_1, K_2)\right] \\
&\leq \frac{1}{2^k} \sum_{K_1 \in \mathrm{GOOD}} 1 + \frac{1}{2^k} \sum_{K_1 \in \mathrm{BAD}} \frac{\varepsilon(n)}{2} \\
&= \frac{1}{2^k} |\mathrm{GOOD}| + \frac{\varepsilon(n)}{2^{k+1}} |\mathrm{BAD}| \\
&= \frac{1}{2^k} |\mathrm{GOOD}| + \frac{\varepsilon(n)}{2^{k+1}} (2^k - |\mathrm{GOOD}|) \\
&= \frac{2 - \varepsilon(n)}{2^{k+1}} |\mathrm{GOOD}| + \frac{\varepsilon(n)}{2} && (6)
\end{aligned}$$

Thus

$$|\text{GOOD}| \geq 2^k \frac{\varepsilon(n)}{2-\varepsilon(n)}. \tag{7}$$

By combining equations (1),(4),(5),(7) we have

$$\begin{aligned}\mathbf{Adv}^{\text{ePre}}_{H_2}(B) &\geq \frac{1}{2^k} 2^k \frac{\varepsilon(n)}{2-\varepsilon(n)} \frac{\varepsilon(n)^2}{4} - \text{negl}_1(n) \\ &= \frac{\varepsilon(n)^3}{4(2-\varepsilon(n))} - \text{negl}_1(n) \\ &> \frac{\varepsilon(n)^3}{8} - \text{negl}_1(n).\end{aligned}$$

Since H_2 is ePre secure, we conclude that $\varepsilon(n)$ must be negligible. □

5 Combining combiners

It is easy to see, that the construction $C_1^{H_1,H_2}$ defined as

$$C_1^{H_1,H_2}(K_1,K_2,M) := \text{Feistel}_{H_1^{(1)},H_1^{(2)},H_1^{(3)}}(H_2(K_2,M)),$$

where

$$H_1^{(i)}(M) := \text{Prefix}_{y/2}(H_1(K_1,\langle i\rangle_2||M)), \qquad i=1,2,3$$

is ePre secure, if H_2 is ePre secure. The formal proof follows.

Theorem 4. *Let $H_1, H_2 : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^y$ be two hash function families. If H_2 is ePre secure, then the construction $C_1^{H_1,H_2}$ is ePre secure.*

Proof. Let A be an adversary and let

$$\varepsilon(n) := \mathbf{Adv}^{ePre}_{C_1}(A).$$

Consider the following adversary B attacking H_2 in the ePre sense.

Adversary B
[1st stage]
1. Simulate $(Y', S) \leftarrow A()$.
2. Compute $Y := \text{Feistel}^{-1}_{H_1^{(1)},H_1^{(2)},H_1^{(3)}}$
3. Output (Y, S).

[2nd stage]

B is given on input a key $K \xleftarrow{\$} \{0,1\}^k$ and "state" variable from the previous stage S.
1. Simulate $M \leftarrow A(K, S)$.
2. Output M.

Clearly, B runs in a polynomial time. If A finds a valid preimage for Y', then B finds a valid preimage for Y. Since A's view in the simulation above is the same as in the ePre experiment against $C_1^{H_1,H_2}$, we have

$$\mathbf{Adv}^{\text{ePre}}_{H_2}(B) \geq \varepsilon(n).$$

Since H_2 is ePre secure, we conclude that $\varepsilon(n)$ must be negligible. □

Thus, if H_1 is Prf and H_2 is Coll and ePre secure, then $C_1^{H_1,H_2}$ is Prf, Coll and ePre secure. By replacing H_2 with the combiner C_2, we get a construction $C_1^{H_1,C_2^{H_2,H_3}}$, which is Prf, Coll and ePre secure, if H_1 is Prf, H_2 is Coll and H_3 is ePre secure.

6 Conclusion

In this paper we introduced two combiners for properties of cryptographic hash functions. The combiner $C_1^{H_1,H_2}$ is collision resistant and pseudo-random, if H_1 is pseudo-random and H_2 is collision resistant. The combiner $C_2^{H_1,H_2}$ is collision resistant and everywhere preimage resistant, if H_1 is collision resistant and H_2 is everywhere preimage resistant. We showed, that these two combiners can be used together so that the resulting combiner $C_3 := C_1^{C_2^{H_1,H_2},H_3}$ is collision resistant, pseudo-random and everywhere preimage resistant. Collision resistance implies preimage resistance and 2nd-preimage resistance [10], pseudo-randomness implies unforgeability, the combiner C_3 thus satisfies all mentioned properties.

Construction of the combiner satisfying aPre and aSec, i.e. always versions of preimage resistance and 2nd-preimage resistance, remains an open problem. Another open problem is analysis of the candidate combiner $C_1' := H_1(K_1, H_2(K_2, M))$ for pseudo-randomness (H_1) and collision resistance (H_2). This combiner is more efficient than C_1, on the other hand, a collision in the combiner C_1' cannot be directly transformed to a collision in the hash function H_2.

References

1. M. Bellare and T. Ristenpart. Hash Functions in the Dedicated-Key Setting: Design Choices and MPP Transforms. In *International Colloquim on Automata, Languages, and Progamming, LNCS vol. 4596*, pages 399–410. Springer, 2006.
2. M. Bellare and T. Ristenpart. Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In *Advances in Cryptology - ASIACRYPT 2006, LNCS vol. 4284*, pages 299–314. Springer, 2006.
3. D. Boneh and X. Boyen. On the Impossibility of Efficiently Combining Collision Resistant Hash Functions. In *Advances in Cryptology - CRYPTO 2006, LNCS vol. 4117*, pages 570–583. Springer, 2006.
4. M. Fischlin and A. Lehman. Multi-property Preserving Combiners for Hash Functions. In *Theory of Cryptography, LNCS vol. 4948*, pages 375–392. Springer, 2008.
5. M. Fischlin, A. Lehmann, and K. Pietrzak. Robust Multi-property Combiners for Hash Functions Revisited. In *Automata, Languages and Programming, LNCS vol. 5126*, pages 655–666, 2009.
6. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. In *SIAM Journal on Computing*, volume 17, pages 373–386, 1988.
7. K. Pietrzak. Non-Trivial Black-Box Combiners for Collision-Resistant Hash-Functions don't Exist. In *Advances in Cryptology - EUROCRYPT 2007, LNCS vol. 4515*, pages 23–33. Springer, 2007.
8. K. Pietrzak. Compression from Collisions, or Why CRHF Combiners Have a Long Output. In *Advances in Cryptology 2008, LNCS vol. 5157*, pages 413–432. Springer, 2008.
9. M. Rjasko. Properties of Cryptographic Hash Functions. *Cryptology ePrint Archive, Report 2008/527*, 2008.
10. P. Rogaway and T. Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In *Fast Software Encryption, LNCS vol. 3017*, pages 371–388. Springer, 2004.