Security-Amplifying Combiners for Collision-Resistant Hash Functions

Marc Fischlin and Anja Lehmann

Darmstadt University of Technology, Germany
www.minicrypt.de

Abstract. The classical combiner $\mathsf{Comb}_{\text{class}}^{H_0,H_1}(M) = H_0(M)||H_1(M)$ for hash functions H_0, H_1 provides collision-resistance as long as at least one of the two underlying hash functions is secure. This statement is complemented by the multi-collision attack of Joux (Crypto 2004) for iterated hash functions H_0, H_1 with n-bit outputs. He shows that one can break the classical combiner in $\frac{n}{2} \cdot T_0 + T_1$ steps if one can find collisions for H_0 and H_1 in time T_0 and T_1, respectively. Here we address the question if there are *security-amplifying* combiners where the security of the building blocks increases the security of the combined hash function, thus beating the bound of Joux. We discuss that one can indeed have such combiners and, somewhat surprisingly in light of results of Nandi and Stinson (ePrint 2004) and of Hoch and Shamir (FSE 2006), our solution is essentially as efficient as the classical combiner.

1 Introduction

A hash function combiner [6] takes two hash functions H_0 and H_1 and combines them into a single, failure-resistant hash function. That is, collision-resistance of the combined function is granted, given that at least one of the starting hash functions H_0, H_1 is secure. A classical example of a secure combiner is $\mathsf{Comb}_{\text{class}}^{H_0,H_1}(M) = H_0(M)||H_1(M)$, concatenating the outputs of the two hash functions. For this combiner any collision $M \neq M'$ immediately gives collisions for both hash functions H_0 and H_1.

From a more quantitative viewpoint, the classical combiner provides the following security guarantee: If breaking H_0 and H_1 requires T_0 and T_1 steps, respectively, finding a collision for the classical combiner takes at least $T_0 + T_1$ steps. This almost matches an upper bound by Joux [8], showing that for Merkle-Damgård hash functions H_0, H_1 with n-bit outputs the classical combiner can be broken in $\frac{n}{2} \cdot T_0 + T_1$ steps. This means that if the security level of each hash function is degraded only moderately through a new attack method, e.g., from 2^{80} to 2^{60}, then the classical combiner, too, merely warrants a reduced security level of $T_0 + T_1 = 2 \cdot 2^{60}$. Ideally, we would like to have a better security bound for combiners and such moderate degradations, going beyond the $T_0 + T_1$ limit and the bound due to Joux.

Our Results. Here we introduce the notion of security-amplifying combiners for collision-resistant hash functions. Such combiners guarantee a security level $\alpha \cdot (T_0 + T_1)$ for some $\alpha > 1$ and, in a sense, are therefore stronger than the sum of their components. Note that the classical combiner (and similar proposals) are *not* security amplifying according to the previous discussion, indicating that constructing such security-amplifying combiners is far from trivial.

We next discuss how to achieve security amplification. Consider two Merkle-Damgård hash functions H_0, H_1 (given by compression functions f_0, f_1) and the classical combiner, but limited to input messages $M = m_0||\ldots||m_{t-1}$ of $t < \frac{n}{4}$ blocks exactly:

$$\mathsf{Comb}^{H_0,H_1}_{\mathrm{amp},t}(M) = H_0(m_0||\ldots||m_{t-1})\,||\,H_1(m_0||\ldots||m_{t-1})$$

This is clearly a secure combiner in the traditional sense, guaranteeing collision resistance if at least one of both hash functions is collision-resistant. But we show that it is even a security-amplifying combiner, assuming that the underlying compression functions behave ideally. More precisely, we consider an attack model in which the compression functions f_0, f_1 are given by random functions, but where the adversary against the combiner can use subroutines $\mathcal{C}_0, \mathcal{C}_1$ to generate collisions for the corresponding compression function. Intuitively, these collision finder oracles implement the best known strategy to find collisions, and each time the adversary calls $\mathcal{C}_b$ to get a collision for f_b, we charge T_b steps. The adversary's task is now to turn such collisions derived through $\mathcal{C}_0, \mathcal{C}_1$ into one against the combiner.

We note that the adversary against the combiner in our model is quite powerful. For each query to the collision finders the adversary can significantly bias the outcome, e.g., by presetting parts of the colliding messages. To give further support of the significance of our model, we show that we can implement the attack of Joux on the classical combiner $\mathsf{Comb}_{\mathrm{class}}$ in our model. We can also realize similar attacks for more advanced combiners like $\mathsf{Comb}^{H_0,H_1}(M) = H_0(M)||H_1(H_0(M) \oplus M)$.

Our main result is to certify the security amplification of our combiner $\mathsf{Comb}_{\mathrm{amp},t}$. The proof is basically split into two parts: one covering general statements about our model (such as pre-image resistance, even in presence of the collision finders), and the other part uses the basic facts to prove our specific combiner $\mathsf{Comb}_{\mathrm{amp},t}$ to be security-amplifying. In our security proof we show that calling each collision finder $\mathcal{C}_0, \mathcal{C}_1$ only polynomially many times does not help to find a collision for $\mathsf{Comb}_{\mathrm{amp},t}$. Therefore, successful attacks on the combiner require more than $\mathrm{poly}(n) \cdot (T_0 + T_1)$ steps.

Viewed from a different perspective we can think of our result as a supplementary lower bound to the attack of Joux. His attack breaks the classical combiner in $\frac{n}{2} \cdot T_0 + T_1$ steps if the hash functions allow to process $t \geq \frac{n}{2}$ message blocks. Our result indicates that restricting the input to $t < \frac{n}{4}$ many blocks suffices to make the combiner security-amplifying and to overcome the bound by Joux. The situation for t in between $\frac{n}{4}$ and $\frac{n}{2}$ remains open.

Finally, recall that our proposal at this point only allows to hash messages of $t < \frac{n}{4}$ blocks. To extend the combiner to handle arbitrarily long messages one can use hash trees in a straightforward way (with our combiner placed at every node of the tree). Since finding collisions in such hash trees requires to come up with collisions in one of the nodes, our security amplification result carries over instantaneously. For messages of k blocks the classical combiner takes about $2k$ applications of the compression functions, compared to roughly $\frac{t}{t-1} \cdot 2k$ applications for our tree-based combiner (but coming with the stronger security amplification guarantee).

Limitations of the Model. Our hash combiner guarantees security amplification in an idealized world where the underlying compression functions behave like random functions. In this model only generic attacks on the hash function are allowed, in the sense that the adversary cannot take advantage of weaknesses of the compression functions beyond the ability to generate collisions (albeit the collision finders are quite flexible). It remains open if similar results can be obtained in a non-idealized setting at all.

Currently, our collision finders return two values mapping to the same compression function output. A recent work of Yu and Wang [14], however, shows that very weak compression functions as in MD4 may allow K-multi-collision attacks, where one is able to find K instead of 2 simultaneous collisions for the compression functions. We expect our results to transfer to this case, when restriciting the number of message blocks further to $t < \frac{n}{4 \log_2 K}$. This will be addressed in the full version of the paper.

Related Work. The idea of cryptographic combiners has been considered explicitly by Herzberg [6]. Among others, he analyzes the classical combiner $\mathsf{Comb}_{\text{class}}$ concatenating the hash function values. As for hash function combiners, Boneh and Boyen [1] and subsequently Pietrzak [12] show that collision-resistant combiners cannot do better than the classical combiner in terms of the length, i.e., the output length of a secure combiner must essentially equal the sum of the output lengths of the hash functions (as in our construction).

Interestingly, the idea of security amplification for cryptographic combiners already appears implicitly in Yao's work [13]. He shows that the existence of weak one-way functions —where inversion may succeed with probability $1 - 1/\text{poly}(n)$— can be turned into strong one-way functions where inversion almost surely fails. The construction can be viewed as a security-amplifying self-combiner for one-way functions. See also [5] for improvements and [9] for related results.

Other relevant works are the upper bounds of Nandi and Stinson [11] and of Hoch and Shamir [7]. They extend the attack of Joux to arbitrary combiners for iterated hash functions, where each message block is possibly processed via the compression function more than once but at most a constant number of times. They also transfer their results to tree-based constructions. However, in their model the output of one compression function must not serve as an input to the other compression function, thus disallowing mixes of intermediate hash values. By this, the hash-tree based extension of our combiner circumvents their bounds.

Finally we remark that, in a concurrent work, Canetti et al. [3] also consider amplification of collision resistance. In contrast to our idealized setting they use a complexity-theoretic approach.

Organization. We start by defining our model and security amplifying combiners (Section 2). Next, in Section 3, we discuss that the classical combiner and similar proposals are not security amplifying. Section 4 present some general conclusions in our model. The main result appears in Section 5 and the proof of this result is given in Section 6. Some proofs in this version have been moved to the Appendix.

2 Preliminaries

2.1 Hash Functions and Combiners

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

Definition 1. *A hash function combiner* Comb *for hash functions* $\mathcal{H}_0, \mathcal{H}_1$ *is an efficient deterministic algorithm such that, for input* $H_0 \leftarrow \mathsf{HKGen}_0(1^n)$, $H_1 \leftarrow \mathsf{HKGen}_1(1^n)$ *and* $M \in \{0,1\}^*$, *it returns a digest* $\mathsf{Comb}(H_0, H_1, M)$. *In addition, the pair* $(\mathsf{CKGen}, \mathsf{Comb})$, *where* $\mathsf{CKGen}(1^n)$ *generates* $H_0 \leftarrow \mathsf{HKGen}_0(1^n)$ *and* $H_1 \leftarrow \mathsf{HKGen}_1(1^n)$ *and outputs* (H_0, H_1), *is a collision-resistant hash function as long as* $\mathcal{H}_0$ *or* $\mathcal{H}_1$ *is collision-resistant.*

The popular Merkle-Damgård construction [10,4] of a hash function takes any collision-resistant compression function $f : \{0,1\}^{l+n} \to \{0,1\}^n$ and an initial vector IV. To compute a digest one divides (and possibly pads) the message $M = m_0 m_1 \dots m_{k-1}$ into blocks m_i of l bits and computes the digest $H(M) = \mathrm{iv}_k$ as

$$\mathrm{iv}_0 = \mathrm{IV}, \qquad \mathrm{iv}_{i+1} = f(\mathrm{iv}_i, m_i) \qquad \text{for } i = 0, 1, \dots, k-1.$$

In this case the description of the hash function simply consists of the pair (f, IV). We note that, in order to make this construction collision-resistant for messages of arbitrary length, one still needs to apply the compression function once more to the bit length of the message.

In the *idealized* Merkle-Damgård construction we assume that the compression function f behaves like a random function (drawn from the set of all functions mapping $(l+n)$-bit strings to n-bit strings). In particular, if an algorithm now gets as input the description of such an idealized MD-hash function then it is understood that this algorithms gets IV as input string and oracle access to the random function f. This holds also for a combiner Comb of such idealized MD hash function, i.e., Comb gets oracle access to f_0, f_1 and receives the strings $\mathrm{IV}_0, \mathrm{IV}_1$ as input. We then often write $\mathsf{Comb}^{H_0,H_1}(\cdot)$ instead of $\mathsf{Comb}^{f_0,f_1}(\mathrm{IV}_0, \mathrm{IV}_1, \cdot)$. *We emphasize that the combiner may assemble a solution from the compression functions and the initial vectors which is not necessarily an iterated hash function.*

2.2 Our Model

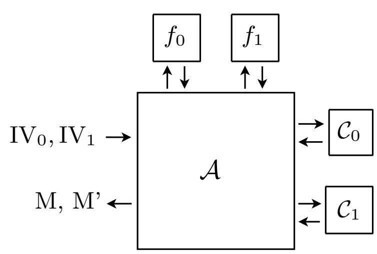
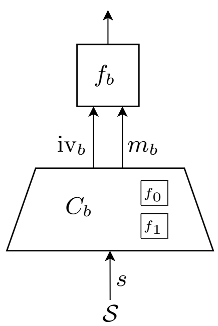


Fig. 1. Attack Model

To analyze the security amplification of a combiner for two idealized MD hash functions (f_0, IV_0) and (f_1, IV_1) we consider an adversary $\mathcal{A}$ with oracle access to f_0, f_1 and input $\mathrm{IV}_0, \mathrm{IV}_1$. The task of this algorithm is to find a collision for the combiner. Since finding collisions for the random compression function directly is restricted to the birthday attack, we allow $\mathcal{A}$ oracle access to two *collision finder oracles* $\mathcal{C}_0$, $\mathcal{C}_1$ generating collisions for each compression function (both oracles themselves have access to f_0, f_1). These collision finders can be viewed as the best known algorithm to generate collision for the compression function. See Figure 1. In its most simple form algorithm $\mathcal{A}$ can query the collision finder $\mathcal{C}_b$ by forwarding values $\mathrm{iv}_b, \mathrm{iv}'_b$ and getting a collision (m_b, m'_b) with $f_b(\mathrm{iv}_b, m_b) = f_b(\mathrm{iv}'_b, m'_b)$ from $\mathcal{C}_b$. More generally, the adversary may want to influence the colliding messages or enforce dependencies between the initial values $\mathrm{iv}_b, \mathrm{iv}'_b$ and the messages m_b, m'_b. To model such advanced collision finding strategies we allow the adversary to pass (the description of) a circuit $C_b : \{0,1\}^i \to \{0,1\}^{l+n}$ (possibly containing f_0- and f_1-gates) to $\mathcal{C}_b$ instead of $\mathrm{iv}_b, \mathrm{iv}'_b$ only. The collision finder then applies an internal stateful source $\mathcal{S} = \mathcal{S}(C_b)$ to continuously generate i-bit strings $s \leftarrow \mathcal{S}$ and successively provides each s as input to the circuit C_b. See Figure 2(a).[1]



(a)

$\textsc{samples}_b(C_b)$ contains all tested pairs $(C_b(s), f_b(C_b(s)))$ in $\mathcal{C}_b$'s collision search for input circuit C_b

$\textsc{cval}_b$ contains all collisions returned by collision finder $\mathcal{C}_b$

$\textsc{fval}_b$ contains all pairs $(x, f_b(x))$ appearing in direct f_b-box queries of $\mathcal{A}$ or in an evaluation of a circuit C_b

(b)

Fig. 2. Operation of collision finder $\mathcal{C}_b$ (a), Sets of function values (b)

For the circuit's output $(\mathrm{iv}_b, m_b) = C_b(s)$ to the next input value s the finder computes $f_b(\mathrm{iv}_b, m_b)$ and checks if for some previously computed value (iv'_b, m'_b) a collision $f_b(\mathrm{iv}_b, m_b) = f_b(\mathrm{iv}'_b, m'_b)$ occurs. If so, $\mathcal{C}_b$ immediately stops and outputs the collision $((\mathrm{iv}_b, m_b), f_b(\mathrm{iv}_b, m_b), s)$ and $((\mathrm{iv}'_b, m'_b), f_b(\mathrm{iv}'_b, m'_b), s')$. Otherwise it stores the new triple $((\mathrm{iv}_b, m_b), f_b(\mathrm{iv}_b, m_b), s)$ and continues its computations. If $\mathcal{C}_b$ does not find a collision among all i-bit inputs s to the circuit it returns $\perp$. We assume that the adversary implicitly gets to know all consulted input values s, gathered in an ordered set $\textsc{sval}(C_b)$. Note that we leave it essentially up to the adversary and his choice for C_b to minimize the likelihood of undefined outputs or trivial collisions (i.e., for the same pre-image).

2.3 Lucky Collisions

The collision finders should be the only possibility to derive collisions, i.e., we exclude accidental collisions (say, $\mathcal{A}$ ignoring the collision finders and finding an f_0-collision by querying the f_0-oracle many times). To capture such *lucky collisions* we assume that each answer $((iv_b, m_b), f_b(\mathrm{iv}_b, m_b), s)$, $((\mathrm{iv}'_b, m'_b), f_b(\mathrm{iv}'_b, m'_b), s')$ of $\mathcal{C}_b$ is augmented by all pre-image/image pairs (x, y) of f_0- and f_1-gate evaluations in the circuit computations during the search. We stress that this excludes all samples $(C_b(s), f_b(C_b(s)))$ which the collision finder probes to find the collision, unless the sample also appears in one of the circuit evaluations (see also the discussion below).

For a query C_b to $\mathcal{C}_b$ we denote the set of the pre-image/image pairs returned to $\mathcal{A}$ by $\textsc{fval}_b^{\mathrm{cf}}(C_b)$ and by $\textsc{fval}_b^{\mathrm{cf}}$ we denote the union of $\textsc{fval}_b^{\mathrm{cf}}(C_b)$ over all queries C_b made to $\mathcal{C}_b$ during $\mathcal{A}$'s computation. Here we assume that the set $\textsc{fval}_b^{\mathrm{cf}}$ is updated immediately after each function gate evaluation during a circuit evaluation. Similarly, $\textsc{fval}_b^{\mathrm{box}}$ stands for the pre-image/image pairs generated by $\mathcal{A}$ as queries and answers to the f_b-box directly. We now set $\textsc{fval}$ as the union of $\textsc{fval}_b^{\mathrm{cf}}$ and $\textsc{fval}_b^{\mathrm{box}}$ for both $b = 0, 1$.

Definition 2 (Lucky Collision). *A pair (x, x') is called a* lucky collision *if for an execution we have $x \neq x'$ and $(x, y), (x', y) \in \textsc{fval}$ for some y.*

---

[1] The source $\mathcal{S}$ can be thought of the collision finder's strategy to generate collisions for the input circuit, and is possibly even known by $\mathcal{A}$. Since we will later quantify over all collision finders we do not specify this distribution; the reader may for now think of $\mathcal{S}$ sequentially outputting the values $0, 1, 2, \ldots$ in binary.

In the definition below $\mathcal{A}$ will not be considered successful if a lucky collision occurs during an execution. It therefore lies in $\mathcal{A}$'s responsibility to prevent lucky collisions when querying f-boxes or the collision finders.

For notational convenience we collect the pre-image/image pairs of collisions generated by the collision-finders in the set CVAL, which is the union of all answers $\text{CVAL}_b(C_b)$ of collision-finder $\mathcal{C}_b$ for query C_b, over all queries C_b and $b = 0, 1$. We also let $\text{SAMPLES}_b(C_b)$ denote all samples $(C_b(s), f_b(C_b(s)))$ which the collision finder $\mathcal{C}_b$ collects to find a collision for query C_b, and SAMPLES stands for the union over all $\text{SAMPLES}_b(C_b)$ for all queries C_b and $b \in \{0, 1\}$. Clearly, $\text{CVAL}_b(C_b) \subseteq \text{SAMPLES}_b(C_b)$. An informal overview about the sets is given in Figure 2(b).

We remark that we do not include the pairs $(C_b(s), f_b(C_b(s)))$ which the collision finder probes in FVAL_b (unless they appear in the circuit's evaluations). This is in order to not punish the adversary for the collision finder's search and strengthens the model, as lucky collisions become less likely. However, for an answer of the collision finder the adversary $\mathcal{A}$ can re-compute all or some of those values by browsing through the ordered set $\text{SVAL}(C_b)$, containing all inspected s-values, and submitting $C_b(s)$ to the f_b-oracle. This value is then added to the set FVAL_b, of course.

2.4 Security Amplification

As for the costs of each oracle call to collision finder $\mathcal{C}_b$ we charge the adversary $\mathcal{A}$ a predetermined number T_b of steps for each call (e.g., $T_b = 2^{n/2}$ if $\mathcal{C}_b$ implements the birthday attack, ignoring the fact that the collision finder may even fail with some probability in this case). We do not charge the adversary for other steps than these calls. In the definition below we make no restriction on the number of calls to the collision finders, yet one might often want to limit this number in some non-trivial way, e.g., for our main result we assume that the adversary makes at most a polynomial number of calls.

Definition 3. *A hash function combiner* Comb *for idealized Merkle-Damgård hash functions* $\mathcal{H}_0, \mathcal{H}_1$ *is called* $\alpha(n)$-security amplifying *if for any oracles* $\mathcal{C}_0, \mathcal{C}_1$ *(with running times* $T_0(n)$ *and* $T_1(n)$*, respectively) and any algorithm* $\mathcal{A}$ *making at most* $\alpha(n) \cdot (T_0(n) + \cdot T_1(n))$ *steps we have*

$$\text{Prob}\left[\mathbf{Exp}^{amp,\mathsf{Comb}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n) = 1\right] \approx 0$$

where

> ***Experiment*** $\mathbf{Exp}^{amp,\mathsf{Comb}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$*:*
> *initialize* $(f_0, IV_0) \leftarrow \mathsf{HKGen}_0(1^n)$, $(f_1, IV_1) \leftarrow \mathsf{HKGen}_1(1^n)$
> *let* $(M, M') \leftarrow \mathcal{A}^{f_0,f_1,\mathcal{C}_0,\mathcal{C}_1}(IV_0, IV_1)$
> *output* 1 *iff*
> $M \neq M'$*, and*
> $\mathsf{Comb}^{f_0,f_1}(IV_0, IV_1, M) = \mathsf{Comb}^{f_0,f_1}(IV_0, IV_1, M')$*, and*
> *no lucky collisions during* $\mathcal{A}$*'s computation occured.*

The combiner is called security amplifying *if it is* $\alpha(n)$*-security amplifying for some function* $\alpha(n)$ *with* $\alpha(n) > 1$ *for all sufficiently large* n*'s.*

Our definition allows $\alpha(n)$ to converge to 1 rapidly, e.g., $\alpha(n) = 1 + 2^{-n}$. We do not exclude such cases explicitly, but merely remark that, as long as $T_0(n)$ and $T_1(n)$ are polynomially related and the combiner is security-amplifying, one can always find a suitable function $\alpha(n)$ bounded away from 1 by a polynomial fraction. For simplicity we have defined compression functions f_0, f_1 of equal output length n (which is also the security parameter). We remark that all our definitions and results remain valid for different output lengths n_0, n_1 by considering $n = \min\{n_0, n_1\}$.

3 Warming Up: Attack on the Classical Combiner

In this section, to get accustomed to our model, we first present the attack of Joux on the classical combiner, showing that this one is not security amplifying (even though it is a secure combiner in the traditional sense). This also proves that finding such security-amplifying is far from trivial. Recall that the classical combiner is given by

$$\mathsf{Comb}_{\text{class}}^{H_0 H_1}(M) := H_0(M)||H_1(M)$$

for idealized Merkle-Damgård hash functions. Obviously this combiner is collision-resistant as long as at least one of the hash functions has this property. Yet, it does not have the desired security-amplification property, because an adversary $\mathcal{A}$ can use the strategy of Joux [8] to find a collision rapidly. The idea is to build a multi-collision set of size $2^{\frac{n}{2}}$ for H_0 by calling $\mathcal{C}_0$ only $\frac{n}{2}$ times, and then to let $\mathcal{C}_1$ search for a pair among those messages in the multi-collision set which also constitutes a collision under H_1.

Adversary $\mathcal{A}^{f_0,f_1,\mathcal{C}_0,\mathcal{C}_1}(\text{IV}_0, \text{IV}_1)$:
for $i = 0, 1, \ldots, k := \frac{n}{2} - 1$:
  let $C_{0,i} : \{0,1\}^l \rightarrow \{0,1\}^{l+n}$ be the circuit $C_{0,i}(s) = (\text{iv}_{0,i}, s)$, where $\text{iv}_{0,0} = \text{IV}_0$
  get $((\text{iv}_{0,i}, m_i), y_i, s), ((\text{iv}_{0,i}, m_i'), y_i, s') \leftarrow \mathcal{C}_0(C_{0,i})$
    where $m_i \neq m_i'$ by the choice of $C_{0,i}$
  set $\text{iv}_{0,i+1} = y_i$
end of for

construct circuit $C_1 : \{0,1\}^{n/2} \rightarrow \{0,1\}^{l+n}$, containing all received collisions (m_i, m_i') from the first stage, as follows:
  for $i = 0, 1, \ldots, k = \frac{n}{2} - 1$:
    for the i-th input bit s_i let $\widehat{m}_i = m_i$ if $s_i = 0$, and $\widehat{m}_i = m_i'$ otherwise
    except for the last round, compute $\text{iv}_{1,i+1} = f_1(\text{iv}_{1,i}, \widehat{m}_i)$, where $\text{iv}_{1,0} = \text{IV}_1$
  end of for
  let the circuit output $(\text{iv}_{1,k}, \widehat{m}_k)$

get $((\text{iv}_{1,k}, \widehat{m}_k), y_k, s), ((\text{iv}_{1,k}', \widehat{m}_k'), y_k, s') \leftarrow \mathcal{C}_1(C_1)$

reconstruct the successful combination M, M' of $\mathcal{C}_1$ by using the values s, s'
for the pairs (m_i, m_i') as above, and output M, M'

First, the collision finder $\mathcal{C}_0$ is called $\frac{n}{2}$ times by the adversary to derive $\frac{n}{2}$ pairs of colliding message blocks (m_i, m_i') where $f_0(\text{iv}_{0,i}, m_i) = f_0(\text{iv}_{0,i}, m_i')$ for $i = 0, 1, \ldots, k$. Since the circuit $C_{0,i}$ passed to $\mathcal{C}_0$ does not evaluate the functions f_0, f_1, no lucky collision can occur in this stage. The query to collision finder $\mathcal{C}_1$ then requires $\frac{n}{2}$ compression function evaluations in the circuit C_1 for each input $s \in \{0,1\}^{n/2}$, which selects one of the $2^{\frac{n}{2}}$ multi-collisions derived from $\mathcal{C}_0$'s answers. Yet, for each common prefix of the s-values the same function evaluations are repeated, and the set $\text{FVAL}_1^{\text{cf}}$ therefore contains at most $2^{\frac{n}{2}}$ pre-image/image pairs (x, y) from the circuit evaluations. This implies that the probability for a lucky collision is at most $\frac{1}{2}$.

On the other hand, given that no collision in $\textsc{fval}_1$ occurs, all circuit outputs are distinct and the set of probed values of the collision finder is at least $2^{\frac{n}{2}}$. But then, $\mathcal{C}_0$ will find a collision among the values with constant probability (which is roughly equal to $1-e^{-1/2}$ for the Euler constant e). Hence, the adversary succeeds with constant probability, taking only $\frac{n}{2} \cdot T_0(n) + T_1(n)$ steps. This implies that the classical combiner is *not* security amplifying, because no appropriate function $\alpha(n) > 1$ exists.

Our model allows to implement attacks on more sophisticated hash combiners such as $\mathsf{Comb}^{H_0,H_1}(M) = H_0(M)||H_1(H_0(M) \oplus M)$, which may seem to be more secure than the classical combiner at first glance due to the dependency of both hash functions. However, by using the circuit C_1 to compute valid inputs for H_1 we can realize a similiar attack as the one for $\mathsf{Comb}_{\text{class}}$.

4 Basic Conclusions

In this section we provide some basic conclusions in our model, e.g., that the functions f_0, f_1 are still pre-image resistant in presence of the collision finders. These results will also be useful when proving our combiner to be security amplifying.

The first lemma basically restates the well-known birthday paradox that, if the adversary $\mathcal{A}$ in experiment $\mathbf{Exp}^{\text{amp},\mathsf{Comb}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ makes too many f_0- and f_1-queries (either directly or through the collision-finders), then most likely a lucky collision will occur and $\mathcal{A}$ cannot succeed anymore. This result —like all results in this section— hold for arbitrary combiners (based on the idealized Merkle-Damgård model):

Lemma 1 (Birthday Paradox). *Consider experiment* $\mathbf{Exp}^{amp,\mathsf{Comb}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ *and assume that* $|\textsc{fval}_b| > 2^{dn}$ *for* $b \in \{0,1\}$ *and a constant* $d > \frac{1}{2}$*. Then the probability that no lucky collisions occur is negligible (and, in particular, the probability that the experiment returns* 1 *is negligible, too).*

Proof. Suppose $|\textsc{fval}_b| > 2^{dn}$ for some b. Then the birthday paradox implies that with probability at most $\exp(-\binom{2^{dn}+1}{2}/2^n) \leq \exp(-2^{(2d-1)n-1})$ there would be *no* lucky collision. Since $d > \frac{1}{2}$ the term $2^{(2d-1)n-1}$ grows exponentially in n. But if a lucky collision occurs, then the experiment outputs 0. □

We next show that the images of sample values $\textsc{samples} \setminus \textsc{cval}$ appearing during the search of the collision finder (but which are not returned to $\mathcal{A}$) are essentially uniformly distributed from $\mathcal{A}$'s viewpoint (i.e., given the sets $\textsc{fval}, \textsc{cval}$). This holds at any point in the execution and even if $\mathcal{A}$ does not win:

Lemma 2 (Image Uncertainty). *Assume that* $\mathcal{A}$ *in experiment* $\mathbf{Exp}^{amp,\mathsf{Comb}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ *makes at most* 2^{cn} *calls to each collision-finder* $\mathcal{C}_0, \mathcal{C}_1$ *and that* $\textsc{fval}_0, \textsc{fval}_1$ *each contain at most* 2^{cn} *elements for a constant* $c < 1$*. Then for any* $(iv, m), y$ *and* $b \in \{0,1\}$ *such that* $((iv, m), f_b(iv, m)) \notin \textsc{fval}_b \cup \textsc{cval}_b$*, we have* $\text{Prob}[\, f_b(iv, m) = y \mid \textsc{fval}, \textsc{cval}] \leq 2 \cdot 2^{-n}$ *(for sufficiently large n's).*

Proof. Consider the information about the image of a value (iv, m) (not appearing in $\textsc{fval} \cup \textsc{cval}$) available through $\textsc{fval}, \textsc{cval}$. Suppose that this value (iv, m) appears in the course of a collision search —else the claim already follows because the image is completely undetermined— and thus the image belongs to $\textsc{samples} \setminus (\textsc{fval} \cup \textsc{cval})$. This only leaks the information that the image of (iv, m) must be distinct from other images in such a collision search, or else the collision finder would have output (iv, m) as part of the collision. Hence, the information available through $\textsc{fval}, \textsc{cval}$ only exclude the images in $\textsc{samples} \cap (\textsc{fval}_b \cup \textsc{cval}_b)$ —values for the other bit $\overline{b}$ are not relevant— which is a set of size at most $|\textsc{fval}_b \cup \textsc{cval}_b| \leq 3 \cdot 2^{cn}$ (since each of the 2^{cn} calls to $\mathcal{C}_b$ yields at most two entries in $\textsc{cval}_b$). Thus, for large n's there are at least $2^n - 3 \cdot 2^{cn} \geq \frac{1}{2} \cdot 2^n$ candidate images left, each one being equally like. □

The next lemma says that the collision-finders cannot be used to break pre-image resistance, i.e., despite the ability to find collisions via $\mathcal{C}_0, \mathcal{C}_1$, searching for a pre-image to a chosen value is still infeasible. Below we formalize this by executing an adversary $\mathcal{B}$ in mode $\mathsf{challenge}$ first, in which $\mathcal{B}$ explicitly determines an image y for which a pre-image should be found under f_b. To avoid trivial attacks we also presume that no (iv, m) with $f_b(\mathrm{iv}, m) = y$ has been found up to this point. Then, we continue $\mathcal{B}$'s execution in mode find in which $\mathcal{B}$ tries to find a suitable pre-image (iv, m). This assumes that $\mathcal{B}$ cannot try out too many collision-finder replies (i.e., at most 2^{cn} many for some constant $c < \frac{1}{2}$):

Lemma 3 (Chosen Pre-Image Resistance). *For any algorithm $\mathcal{B}$ and any constant $c < \frac{1}{2}$ the following experiment $\mathbf{Exp}^{pre,\mathsf{Comb}}_{\mathcal{B},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ has negligible probability of returning 1:*

Experiment $\mathbf{Exp}^{pre,\mathsf{Comb}}_{\mathcal{B},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$:
initialize $(f_0, IV_0) \leftarrow \mathsf{HKGen}_0(1^n)$, $(f_1, IV_1) \leftarrow \mathsf{HKGen}_1(1^n)$
let $(y, b, state) \leftarrow \mathcal{B}^{f_0,f_1,\mathcal{C}_0,\mathcal{C}_1}(\mathit{challenge}, IV_0, IV_1)$
let $\textsc{val}_b^{ch} = \textsc{fval}_b \cup \textsc{cval}_b$ *at this point*
let $(iv, m) \leftarrow \mathcal{B}^{f_0,f_1,\mathcal{C}_0,\mathcal{C}_1}(\mathit{find}, state)$
return 1 *iff*
  $f_b(iv, m) = y$ *and* $((iv, m), y) \notin \textsc{val}_b^{ch}$, *and*
  $\mathcal{B}$ made at most 2^{cn} calls to collision-finder $\mathcal{C}_b$ (in both phases together), and
  no lucky collisions occured during $\mathcal{B}$'s computation (in both phases together)

The proof is delegated to Appendix A. The proof idea is as follows. For any value appearing in $\textsc{fval}_b \setminus \textsc{cval}_b$ during the find phase the probability of matching y is at most $2 \cdot 2^{-n}$ by the image uncertainty. Furthermore, according to the Birthday Lemma 1 the set $\textsc{fval}_b$ cannot contain more than 2^{dn} elements for some $d > \frac{1}{2}$ (or else a lucky collision is very likely). But then the probability of finding another pre-image among those values is negligible.

The harder part is to show that $\mathcal{B}$ cannot significantly influence the collision finder $\mathcal{C}_b$ to search for a collision with image y (which would then appear in $\textsc{cval}_b$ and could be output by $\mathcal{B}$). Here we use the property of our model saying that the circuit's output $C_b(s)$ for each sample is essentially determined by $\mathcal{B}$ (or, to be precise, by the previous values in $\textsc{fval}$ and $\textsc{cval}$). But then the Image Uncertainty Lemma applies again, and each sample $C_b(s)$ yields y with probability at most $2 \cdot 2^{-n}$. The final step is to note that each collision search most likely requires approximately $2^{\frac{n}{2}}$ or less samples, and $\mathcal{B}$ initiates at most 2^{cn} many searches for $c < \frac{1}{2}$. Hence, with overwhelming probability there is no value with image y in $\textsc{samples}$ in the find phase at all, and thus no such value in $\textsc{cval}_b$. This shows Chosen Pre-Image Resistance.

For the final conclusions about our model, we prove that, given a collision (iv, m), (iv', m') produced by a collision finder $\mathcal{C}_b$, generating another pre-image also mapping to $f_b(\mathrm{iv}, m) = f_b(\mathrm{iv}', m')$, is infeasible. The proof is in two steps, first showing that one cannot use the f_b-boxes to find such an additional value, and the second lemma shows that this remains true if one tries to use the collision finder (if one does not call the collision finder more than a polynomial number of times). We remark that this aspect refers to collisions *for the compression functions* only; given a collision generated by the finders one can of course extend this to further collisions *for the iterated hash function* by appending message blocks:

Lemma 4 (f-Replication Resistance). *Assume adversary $\mathcal{A}$ in $\mathbf{Exp}^{amp,\mathsf{Comb}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ makes at most 2^{cn} calls to each collision-finder $\mathcal{C}_0, \mathcal{C}_1$ and that each set* $\text{FVAL}_0, \text{FVAL}_1$ *contains at most 2^{dn} elements for constants c, d with $c + d < 1$. Then the probability that there exist values $((iv, m), y) \in \text{CVAL}_b$ and $((iv', m'), y) \in \text{FVAL}_b \setminus \text{CVAL}_b$ for $b \in \{0, 1\}$, is negligible.*

Proof. Fix a bit b. Since $\mathcal{A}$ makes at most 2^{cn} calls to $\mathcal{C}_b$ and each reply returns two elements, the set CVAL_b is of size at most $2 \cdot 2^{cn}$. Consider any value $((\text{iv}, m), y) \in \text{CVAL}_b$ and any value $((\text{iv}', m'), y') \in \text{FVAL}_b \setminus \text{CVAL}_b$. Then, because $((\text{iv}', m'), y') \notin \text{CVAL}_b$, we must have $y' \neq y$ or $(\text{iv}, m) \neq (\text{iv}', m')$. In the first case we have no match, in the second case a match can occur with probability at most $2 \cdot 2^{-n}$ by the image uncertainty (considering the point in the execution where the the second of the two values appears for the first time).

Now sum over all $2 \cdot 2^{cn} \cdot 2^{dn} = 2 \cdot 2^{(c+d)n}$ combinations, such that the probability of finding any match is at most $4 \cdot 2^{(c+d-1)n}$. Since $c + d < 1$ this is negligible, and stays negligible if we sum over both choices for b. □

Note that the fact above indicates that, after having generated collisions through the finder, finding other matching function values through the f-boxes is infeasible. This holds at any point in the execution, i.e., $\mathcal{A}$ may not even successfully produce a collision but rather stop prematurely. Next, we use this fact (together with pre-image resistance) to prove replication resistance with respect to the collision finders:

Lemma 5 ($\mathcal{C}$-Replication Resistance). *Assume adversary $\mathcal{A}$ in $\mathbf{Exp}^{amp,\mathsf{Comb}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ makes at most poly(n) calls to each collision-finder $\mathcal{C}_0, \mathcal{C}_1$ and that* $\text{FVAL}_0, \text{FVAL}_1$ *each contain at most 2^{dn} elements for a constant $d < 1$. Then the probability that there exist values $((iv, m), y)$, $((iv', m'), y)$, $((iv^*, m^*), y) \in \text{CVAL}_b$ for $b \in \{0, 1\}$ with pairwise distinct $(iv, m), (iv', m'), (iv^*, m^*)$, is negligible.*

The proof is in Appendix B. The basic idea is that, at some point in the execution, there must be at most two of the three values in CVAL_b and then another call adds the third value with the same image. But then this contradicts the chosen pre-image resistance, because the right call to the collision finder among the polynomially many ones can be guessed with probability $1/\text{poly}(n)$. We note that the full argument needs to take care of the case that the third value appears in FVAL_b before.

5 A Security-Amplifying Combiner

Our (input-restricted) security-amplifying combiner takes messages $M = m_0 || \ldots || m_{t-1}$ of exactly t blocks with $t \leq en$ for some constant $e < \frac{1}{4}$ and applies each of the two hash functions H_0, H_1 to the message $m_0 || \ldots || m_t$ and outputs the concatenation:

Theorem 1. *Let $\mathcal{H}_0, \mathcal{H}_1$ be idealized Merkle-Damgård hash functions. Let $e < \frac{1}{4}$ be a constant and assume that $t \le en$. Then the combiner*

$$\mathsf{Comb}^{H_0 H_1}_{amp,t}(M) = H_0(m_0||\ldots||m_{t-1}) \mid\mid H_1(m_0||\ldots||m_{t-1})$$

of $\mathcal{H}_0$ and $\mathcal{H}_1$ is $\alpha(n)$-security-amplifying for $\alpha(n) = poly(n)$ if the adversary in experiment $\mathbf{Exp}^{amp,\mathsf{Comb}_{amp,t}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ makes at most $\alpha(n) = poly(n)$ calls to each collision finder.

We also remark that our combiner is obviously a (classically) secure combiner in the non-idealized setting. The theorem shows that we get the improved security-amplification guarantee against attacks in the idealized world.

For the proof idea it is instructive to investigate why the straightforward application of the attack of Joux for the case of at most $t \le \frac{n}{4}$ message blocks fails. In this case one would again build a multi-collision set for either hash function of size at most $2^t \le 2^{\frac{n}{4}}$. But this time the probability that any of the $2^{2t} < 2^{\frac{n}{2}}$ pairs in such a multi-collision set also collides under the other hash function, should be approximately $2^{\frac{n}{2}} \cdot 2^{-n} = 2^{-\frac{n}{2}}$. Most likely, even approximately $2^{\frac{n}{2}}$ multi-collsion sets should therefore not help to find a collision under both hash functions. Our proof follows these lines of reasoning, i.e., bounding the size of multi-collision sets and the probability that message pairs in such a multi-collision set also collide under the other hash function. We stress, however, that a full proof in our model still needs to deal with more general adversaries, possibly taking advantage of the collision finders through "clever" queries.

To process messages of arbitrary length without losing the security-amplification property we apply a hash-tree construction [10] to our combiner. Since the construction is somewhat standard we merely give an example for $t = 2$ in Figure 3. For a similar and more formal treatment see for instance [2]. In general the input restriction t of the hash combiner gives us an t-ary tree, processing k message blocks $m_0 \ldots m_{k-1}$.

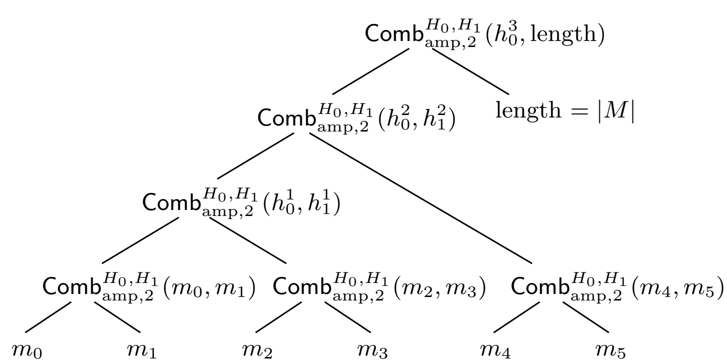


Fig. 3. Example of a hash tree construction for our combiner ($t = 2, k = 6$)

If two messages $M \neq M'$ lead to a collision in the root of the hash tree, it can be either the result of a non-trivial collision in the final application of the combiner for different message lengths $|M| \neq |M'|$ (in which case we get a non-trivial collision for the basic combiner), or else the tree structures must be identical. In the latter case the collision can always be traced back to a collision for an earlier application of the combiner. Hence, in both cases the reason for the tree collision is at least one collision for the basic combiner.

As for the efficiency, for a full t-ary tree (with $k = t^r$, the number of message blocks, being a power of t) we apply our basic combiner $\frac{k-1}{t-1} + 1$ times. Each time we need $2t$ applications of the compression functions, making our solution about $\frac{t}{t-1}$ times slower than the classical combiner with $2k$ applications (but with the advantage of security amplification for our combiner).

6 Proof of Security Amplification

Before giving the proof we first show a technical conclusions stating that the adversary against our (input-restricted) combiner essentially cannot win if the function values of the output are undetermined (the proof of this first lemma follows from the image uncertainty and appears in Appendix C):

Lemma 6 (Output Knowledge). *Assume that $\mathcal{A}$ in experiment $\mathbf{Exp}^{amp,\mathsf{Comb}_{amp,t}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ makes at most 2^{cn} calls to each collision-finder $\mathcal{C}_0, \mathcal{C}_1$ for some constant $c < 1$. Assume that $\mathcal{A}$ eventually outputs $M = m_0||\ldots||m_{t-1} \neq M' = m'_0||\ldots||m'_{t-1}$ such that*

$$\begin{aligned} iv_{b,0} = iv'_{b,0} = IV_b, \quad & iv_{b,i+1} = f_b(iv_{b,i}, m_i), \\ & iv'_{b,i+1} = f_b(iv'_{b,i}, m'_i) \quad \text{for } b \in \{0,1\} \text{ and } i \in \{0,1,\ldots,t-1\} \end{aligned}$$

Suppose further that $((iv_{b,i}, m_i), iv_{b,i+1})$ or $((iv'_{b,i}, m'_i), iv'_{b,i+1}))$ does not belong to $\textsc{fval}_b \cup \textsc{cval}_b$ for some $b \in \{0,1\}$ and some $i \in \{0,1,\ldots,t-1\}$. Then the probability that the experiment returns 1 is negligible.

The following lemma proves that, for t message blocks there can only be 2^t multi-collisions, as long as each collision finder is only called a polynomial number of times:

Lemma 7 (Multi-Collisions). *Assume attacker $\mathcal{A}$ in experiment $\mathbf{Exp}^{amp,\mathsf{Comb}_{amp,t}}_{\mathcal{A},\mathcal{H}_0,\mathcal{H}_1,\mathcal{C}_0,\mathcal{C}_1}(n)$ makes at most poly(n) calls to each collision-finder $\mathcal{C}_0, \mathcal{C}_1$ and that the experiment returns 1. Then, the probability that for some $b \in \{0,1\}$ and some $iv_{b,t}$, the set*

$$\textsc{multi}_b(iv_{b,t}) := \left\{ M = m_0||\ldots||m_{t-1} \;:\; \begin{array}{l} iv_{b,i+1} = f_b(iv_{b,i}, m_i) \in \textsc{fval}_b \cup \textsc{cval}_b \\ \text{for } i = 0,1,\ldots,t-1, \text{ where } iv_{b,0} = IV_b \end{array} \right\}$$

contains more than 2^t elements, is negligible.

The lemma holds because if there was a multi-collision set with more than 2^t elements, then there must be distinct values $(\mathrm{iv}_{b,i}, m_i)$, $(\mathrm{iv}'_{b,i}, m'_i)$ and $(\mathrm{iv}^*_{b,i}, m^*_i)$ mapping to the same image under f_b. According to the previous lemma we can assume that all of them belong to $\textsc{fval}_b \cup \textsc{cval}_b$, but then they would either be a lucky collision (two or three values in $\textsc{fval}_b$), refute f-replication resistance (one value in $\textsc{fval}_b$) or contradict $\mathcal{C}$-replication resistance (no value in $\textsc{fval}_b$).

With these two lemmas we can now prove that our combiner is security-amplifying. The full proof appears in Appendix C. For an outline consider the multi-collision sets defined in the previous lemma. Lemma 6 implies that, in order to win, the adversary must know the images of the final output $M \neq M'$. Hence, each of the two messages must appear in some multi-collision set, and to constitute a collision under hash function H_b, they must appear in the same multi-collision set $\text{MULTI}_b(y_b)$ for some y_b. Moreover, since the messages must collide under both hash functions simultaneously they must belong to an intersection $\text{MULTI}_0(y_0) \cap \text{MULTI}_1(y_1)$ for some y_0, y_1.

Lemma 7 now says that each multi-collision set has at most 2^t elements. Thus, there are at most $2^{2t} \leq 2^{2en}$ such pairs in each multi-collision set. Furthermore, we can bound the number of multi-collision sets by the number of elements in $\text{FVAL}_b \cup \text{CVAL}_b$, and therefore by $3 \cdot 2^{dn}$ for a constant $d > \frac{1}{2}$ with $d + 2e < 1$ (here we use the fact that $e < \frac{1}{4}$). We therefore have at most $3 \cdot 2^{(d+2e)n}$ possible pairs $M \neq M'$. The proof then shows that, by the image uncertainty, any of the pairs M, M' in some multi-collision set $\text{MULTI}_b(y_b)$ also collides under the other hash function $H_{\overline{b}}$, with probability at most $6 \cdot 2^{(d+2e-1)n}$ which is negligible. Put differently, with overwhelming probability the intersections of mulit-collision sets for both hash functions are empty and the adversary cannot find appropriate messages M, M'.

Acknowledgments

We thank the anonymous reviewers for valuable comments. Both authors are supported by the Emmy Noether Program Fi 940/2-1 of the German Research Foundation (DFG).

References

1. Dan Boneh and Xavier Boyen. *On the Impossibility of Efficiently Combining Collision Resistant Hash Functions.* Advances in Cryptology — Crypto 2006, Volume 4117 of Lecture Notes in Computer Science, pages 570–583. Springer-Verlag, 2006.
2. Mihir Bellare and Phillip Rogaway. *Collision-Resistant Hashing: Towards Making UOWHFs Practical.* Advances in Cryptology — Crypto'97, Volume 1294 of Lecture Notes in Computer Science, pages 470–484. Springer-Verlag, 1997.
3. Ran Canetti, Ron Rivest, Madhu Sudan, Luca Trevisan, Salil Vadhan, and Hoeteck Wee. *Amplifying Collision Resistance: A Complexity-Theoretic Treatment.* Advances in Cryptology — Crypto 2007, Lecture Notes in Computer Science. Springer-Verlag, 2007. *These proceedings.*
4. Ivan Damgård. *A Design Principle for Hash Functions.* Advances in Cryptology — Crypto'89, Volume 435 of Lecture Notes in Computer Science, pages 416–427. Springer-Verlag, 1990.
5. Oded Goldreich, Russell Impagliazzo, Leonid Levin, Ramarathnam Venkatesan, and David Zuckerman. *Security Preserving Amplification of Hardness.* Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)'90, pages 318–326. IEEE Computer Society Press, 1990.
6. Amir Herzberg. *On Tolerant Cryptographic Constructions.* 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.
7. Jonathan Hoch and Adi Shamir. *Breaking the ICE — Finding Multicollisions in Iterated Concatenated and Expanded (ICE) Hash Functions.* Fast Software Encryption (FSE) 2006, Volume 4047 of Lecture Notes in Computer Science. Springer-Verlag, 2006.
8. Antoine Joux. *Multicollisions in Iterated Hash Functions.* Advances in Cryptology — Crypto 2004, Volume 3152 of Lecture Notes in Computer Science. Springer-Verlag, 2004.
9. Henry Lin, Luca Trevisan, and Hoeteck Wee. *On Hardness Amplification of One-Way Functions.* Theory of Cryptography Conference (TCC) 2005, Volume 3378 of Lecture Notes in Computer Science, pages 34–49. Springer-Verlag, 2005.
10. Ralph Merkle. *One Way Hash Functions and DES.* Advances in Cryptology — Crypto'89, Volume 435 of Lecture Notes in Computer Science, pages 428–446. Springer-Verlag, 1990.

11. M. Nandi and D. Stinson. *Multicollision Attacks on a Class of Hash Functions.* Number 2004/330 in Cryptology eprint archive. `eprint.iacr.org`, 2004.
12. Krzysztof Pietrzak. *Non-Trivial Black-Box Combiners for Collision-Resistant Hash-Functions don't Exist.* Advances in Cryptology — Eurocrypt 2007, Lecture Notes in Computer Science. Springer-Verlag, 2007.
13. Andrew Yao. *Theory and Applications of Trapdoor Functions.* Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS) 1982. IEEE Computer Society Press, 1982.
14. Hongbo Yu and Xiaoyun Wang. *MultiCollision Attack on the Compression Functions of MD4 and 3-Pass HAVAL.* Number 2007/085 in Cryptology eprint archive. `eprint.iacr.org`, 2007.

A Proof of Chosen Pre-Image Resistance (Lemma 3)

In this section we prove Lemma 3, showing that no adversary $\mathcal{B}$ can determine an image y and later find another pre-image to this value.

Proof. Let d be a constant with $\frac{1}{2} < d < 1$. Assume that $\textsc{fval}_b$ contains more than 2^{dn} elements at the end. Then Lemma 1 implies that such executions can only contribute with negligible probability to $\mathcal{B}$'s success. From now on we can therefore condition on this bound 2^{dn} of number on elements in $\textsc{fval}_b$.

By the image uncertainty we conclude that the probability that any $((\text{iv}, m), f_b(\text{iv}, m)) \in \textsc{fval}_b \setminus \textsc{cval}_b$ in $\mathcal{B}$'s find phase yields y, is at most $2 \cdot 2^{-n}$. Here we use the fact that any function evaluation adding to $\textsc{fval}_b \setminus \textsc{cval}_b$ is either via a direct call to the f_b-box, or via an f_b-gate evaluation in the computation of a circuit $C(s)$, carried out through one of the collision finders. In any case, the input to the function only depends on the values in $\textsc{fval}$ and $\textsc{cval}$ before the corresponding query; for f_b-box queries this is clear and for circuit computations it follows as the circuit is chosen by $\mathcal{B}$ and all previous function evaluations immediately appear in $\textsc{fval}_b$. Therefore, the uncertainty bound applies. Summing over all at most 2^{dn} many values in $\textsc{fval}_b$ shows that the probability of hitting y is bounded from above by $2 \cdot 2^{(d-1)n}$ and is thus negligible. In the sequel we therefore presume that no $((\text{iv}, m), y) \in \textsc{fval}_b \setminus \textsc{cval}_b$ appears (unless it has been in $\textsc{val}_b^{\mathsf{ch}}$ before, in which case $\mathcal{B}$ cannot use it anymore for a successful run).

We next investigate the effect of collision finder calls on $\textsc{cval}_b$, addressing the question if $\mathcal{B}$ can force the collision finder to bias collisions towards y in some way. Recall that the collision finder makes at most 2^{cn} many runs for $c < \frac{1}{2}$. Let $e = \frac{3}{4} - \frac{c}{2} > \frac{1}{2}$. Then we can assume that each run probes at most 2^{en} new elements previously not in $\textsc{samples}$. This is so since, for a single run, the probability of finding no collisions after 2^{en} many trials for fresh values, is double-exponentially small (see Lemma 1 and note that this remains true for a slightly larger probability of $2 \cdot 2^{-n}$). The probability that any of the 2^{cn} calls would require more fresh samples, is therefore still negligible. From now on we thus presume that each call adds at most 2^{en} new entries to $\textsc{samples}$.

Consider the j-th call C_b to the collision finder $\mathcal{C}_b$ in the find stage. Let $\textsc{cval}_{b,j}^{\mathsf{before}}$ be the set $\textsc{cval}_b$ before this call, such that $\textsc{cval}_{b,1}^{\mathsf{before}}$ denotes the set $\textsc{cval}_b$ at the beginning of the find phase. Note that $\textsc{cval}_{b,j}^{\mathsf{before}}$ does not change during the collision search, but only when the finder returns the collision. Suppose further that $\textsc{cval}_{b,j}^{\mathsf{before}}$ does not contain any element $((\text{iv}, m), y)$ which is not already in $\textsc{val}_b^{\mathsf{ch}}$. This is obviously true for $\textsc{cval}_{b,1}^{\mathsf{before}}$.

A crucial aspect in our consideration is that all circuit values $C_b(s)$ during the collision search are fully determined given $\textsc{fval}_b$ (containing the pairs of the entire execution but whose images are distinct from y by assumption) as well as $\textsc{cval}_{b,j}^{\mathsf{before}}$. Hence, the uncertainty bound applies again, and the probability that a specific sample $C_b(s)$ gives a new pair $(C_b(s), y) \notin \text{CVAL}^{\mathsf{before}}_{b,j} \cup \text{VAL}^{\mathsf{ch}}_b$, is at most $2 \cdot 2^{-n}$ (noting that any entry $(C_b(s), f_b(C_b(s))) \in (\text{FVAL}_b \cup \text{CVAL}^{\mathsf{before}}_{b,j}) \setminus \text{VAL}^{\mathsf{ch}}_b$ has an image different from y by assumption). Since there are at most 2^{en} new samples, only with probability at most $2 \cdot 2^{(e-1)n}$ some new sample $C_b(s)$ in $\mathcal{C}_b$'s search yields y. It follows that, except with probability $2 \cdot 2^{(e-1)n}$, the set $\text{CVAL}^{\mathsf{before}}_{b,j+1}$ including the new collisions will not contain a suitable entry.

Finally, sum over all at most 2^{cn} many calls to $\mathcal{C}_b$ to derive that CVAL_b does not contain a new entry $((\text{iv}, m), y) \in \text{CVAL}_b \setminus \text{VAL}^{\mathsf{ch}}_b$, except with probability $2 \cdot 2^{(c+e-1)n}$ for $c + e = \frac{3}{4} + \frac{c}{2} < 1$ which is negligible. Since the same holds for $\text{FVAL}_b \setminus \text{CVAL}_b$ the overall probability of finding a suitable pre-image (iv_0, m), including possibly the final output which is not a member in $\text{FVAL}_b \cup \text{CVAL}_b$, is negligible. □

B Proof of $\mathcal{C}$-Replication Resistance (Lemma 5)

In this section we prove that no adversary can find three values in CVAL mapping to the same image.

Proof. We discuss that if $\mathcal{A}$ could find three (or more) of those values then this would contradict either f-replication resistance or chosen pre-image resistance. Consider adversary $\mathcal{B}$ against the chosen pre-image resistance which basically runs a black-box simulation of $\mathcal{A}$. In the $\mathsf{challenge}$-phase, $\mathcal{B}$ initially makes a guess for a specific call j adversary $\mathcal{A}$ makes to one of the collision finders. Then $\mathcal{B}$ runs $\mathcal{A}$ up to the point where $\mathcal{A}$ receives the answer $((\text{iv}, m), y)$, $((\widehat{\text{iv}}, \widehat{m}), y)$ of $\mathcal{C}_b$ for this j-th call. Then $\mathcal{B}$ outputs y, b (and all internal information of $\mathcal{A}$ as state) and concludes this stage. In the find-phase $\mathcal{B}$ continues $\mathcal{A}$'s simulation and waits to see a value $((\text{iv}^*, m^*), y)$ in the execution, and then outputs (iv^*, m^*) and stops.

We next analyze $\mathcal{B}$'s success probability. Since each call to the collision-finders adds at most two new values to CVAL_b, there must be a point in $\mathcal{A}$'s execution where there is $(\text{iv}, m) \in \text{CVAL}_b$ (and possibly $(\text{iv}', m') \in \text{CVAL}_b$) and only the next call to $\mathcal{C}_b$ adds the value (iv^*, m^*) to CVAL_b, i.e., so far $(\text{iv}^*, m^*) \notin \text{CVAL}_b$. Suppose that the conditional probability (given such a value (iv^*, m^*) with the same image really appears in the execution) that this value belongs to FVAL_b after the corresponding call to $\mathcal{C}_b$, was noticeable. Then this would clearly contradict the f-replication resistance (bounding the polynomial number of calls by 2^{cn} for the constant $c = \frac{1}{2} - \frac{d}{2}$ with $c + d < 1$). We may therefore assume that $(\text{iv}^*, m^*) \notin \text{VAL}^{\mathsf{ch}}_b = \text{CVAL}_b \cup \text{FVAL}_b$ at this point. But then $\mathcal{B}$ guesses the right call j with probability $1/\text{poly}(n)$, and thus predicts a function value with noticeable probability. This, however, contradicts the chosen pre-image resistance. □

C Proof of Security Amplification (Theorem 1)

In this section we provide the proofs of the claims in Section 6 and of the theorem. First we prove that an adversary must essentially know the function values of the output (Lemma 6):

Proof (of Lemma 6). Suppose $\mathcal{A}$ outputs such values M, M' and succeeds with noticeable probability. Assume for simplicity that $((\text{iv}_{b,i}, m_i), \text{iv}_{b,i+1}) \notin \text{FVAL}_b \cup \text{CVAL}_b$; the case $((\text{iv}'_{b,i}, m'_i), \text{iv}'_{b,i+1})$ is treated analogously. Let i be maximal and fix the bit b.

By Lemma 1 we can assume $|\text{FVAL}_b| \leq 2^{dn}$ for $d = \max\{\frac{3}{4}, c\}$, except with negligible probability. Hence, from now on we can condition on $|\text{FVAL}_b \cup \text{CVAL}_b| \leq 3 \cdot 2^{dn}$. For a success the messages M and M' must collide under H_b. If $i = t-1$ then $f_b(\text{iv}_{b,i}, m_i) = \text{iv}_{b,i+1}$ is the output of the hash function, and since this value does not appear in $\text{FVAL}_b \cup \text{CVAL}_b$, the probability of matching $\text{iv}'_{b,i+1}$ is bounded from above by $2 \cdot 2^{-n}$ by the image uncertainty.

If $i < t-1$ then there must exist an entry $((\text{iv}_{b,i+1}, m_{i+1}), \text{iv}_{b,i+2}) \in \text{FVAL}_b \cup \text{CVAL}_b$ (because i is chosen to be maximal). However, the probability that the value $f_b(\text{iv}_{b,i}, m_i)$ appears as a prefix in any of the $3 \cdot 2^{dn}$ values in $\text{FVAL}_b \cup \text{CVAL}_b$, is at most $6 \cdot 2^{(d-1)n}$ and thus negligible. On the other hand, if the prefix $f_b(\text{iv}_{b,i}, m_i)$ does not appear in $\text{FVAL}_b \cup \text{CVAL}_b$, then this contradicts the maximal choice of i. Doubling the probability for both choices of b concludes the proof. □

We next prove Lemma 7, bounding the number of messages in a multi-collision set by 2^t:

Proof (of Lemma 7). Assume that the experiment returns 1 (such that, except with negligible probability, $\text{FVAL}_0, \text{FVAL}_1$ are of size at most 2^{dn} each, for some constant $d < 1$). If some set $\text{MULTI}_b(\text{iv}_{b,t})$ contains more than 2^t elements then there must be an index i such that there are (at least) three distinct values $(\text{iv}_{b,i}, m_i)$, $(\text{iv}'_{b,i}, m'_i)$ and $(\text{iv}^*_{b,i}, m^*_i)$ mapping to the same image under f_b. If two or more of those values belong to $\text{FVAL}_b \setminus \text{CVAL}_b$ then this constitutes a lucky collision and refutes the fact that the experiment returns 1. If one of the values lies in $\text{FVAL}_b \setminus \text{CVAL}_b$, whereas the other two values belong to CVAL_b, then this contradicts the f-replication resistance and this can only happen with negligible probability. Finally, the case that all three values belong to CVAL_b can only happen with negligible probability, too, under the $\mathcal{C}$-replication resistance. □

Finally, we give the full proof that our combiner is security amplyifing:

Proof (of Theorem 1). According to our definition a combiner is called security-amplifying if for any algorithm $\mathcal{A}$ making at most $\alpha(n) \cdot (T_0(n) + T_1(n))$ steps the probability of finding a collision is negligible (for some $\alpha(n) > 1$). Hence we will show that, with overwhelming probability, no collisions for $\mathsf{Comb}_{\text{amp},t}$ (with $t < en$ for constant $e < \frac{1}{4}$) can be computed for any $\alpha(n) = \text{poly}(n)$ when calling each collision finders at most $\alpha(n) = \text{poly}(n)$ many times.

Let $d = \frac{3}{4} - e$ such that the constant d is at larger than $\frac{1}{2}$ and $d + 2e < 1$. Then we can assume that $\text{FVAL}_0, \text{FVAL}_1$ in $\mathcal{A}$'s attack each contain at most 2^{dn} elements, otherwise the probability of winning would be negligible. Also assume that the number of collision finder calls is bounded by $2 \cdot \text{poly}(n) \leq 2^{dn}$ (for sufficiently large n's). Hence, in the following, we can assume that $\text{FVAL}_b \cup \text{CVAL}_b$ contains at most $3 \cdot 2^{dn}$ many elements for $b \in \{0,1\}$.

For any $b \in \{0,1\}$ and any $\text{iv}_{b,t}$ we again consider all sets of multi-collisions,

$$\text{MULTI}_b(\text{iv}_{b,t}) = \left\{ M = m_0 || \ldots || m_{t-1} \ : \ \begin{array}{l} \text{iv}_{b,i+1} = f_b(\text{iv}_{b,i}, m_i) \in \text{FVAL}_b \cup \text{CVAL}_b \\ \text{for } i = 0, 1, \ldots, t-1, \text{ where } \text{iv}_{b,0} = \text{IV}_b \end{array} \right\}$$

but this time we divide them into different stages (depending on the calls to the collision finders). We denote by $\text{MULTI}^{\mathsf{before}}_{b,j}(y)$ the set of multi-collisions *before* the j-th call to one of the two collision finders. The transition to the next phase therefore adds all messages with respect to the new function values from the collision finder's reply as well as all subsequent function evaluations through the f-boxes. Clearly, $\text{MULTI}^{\mathsf{before}}_{b,j}(y) \subseteq \text{MULTI}^{\mathsf{before}}_{b,j+1}(y)$ for all j

and $\text{MULTI}^{\mathsf{before}}_{b,2\cdot\text{poly}(n)+1}(y)$ —which we denote by $\text{MULTI}^{\mathsf{end}}_{b}(y)$— contains all multi-collisions for y under H_b at the end of the execution.

By Lemma 6 adversary $\mathcal{A}$ must "know" all function values in the final output, i.e., they must belong to $\text{FVAL}_b \cup \text{CVAL}_b$ for some $b \in \{0,1\}$. Hence, both messages of the collision $M \neq M'$ for H_b output by $\mathcal{A}$ must also appear in the same set $\text{MULTI}^{\mathsf{end}}_{b}(y_b)$ for some y_b. This basically reduces the task of showing that $\mathcal{A}$ fails, to the proof that no $M \neq M'$ and y_0, y_1 with $M, M' \in \text{MULTI}^{\mathsf{end}}_{0}(y_0) \cap \text{MULTI}^{\mathsf{end}}_{1}(y_1)$ exist (except with some very small probability or if one of the success requirements such as the absence of lucky collisions is violated).

We will show that, given that no success requirements are violated, with overwhelming probability the intersection of multi-collision sets for $b = 0, 1$ will be empty in the course of the execution. This is done by a careful inductive argument, where we use the invariant that for no y_b the set $\text{MULTI}^{\mathsf{before}}_{b,j}(y_b)$ contains $M \neq M'$ such that they collide under $H_{\overline{b}}$. This is clearly true for $\text{MULTI}^{\mathsf{before}}_{b,1}(y_b)$ because up to the point where the first collision finder is called, only f-queries have been made, and each set $\text{MULTI}^{\mathsf{before}}_{b,1}(y_b)$ can contain only one element (or a lucky collision already occurs).

We also use that, according to the Multi-Collision Lemma 7, each set $\text{MULTI}^{\mathsf{before}}_{b,j}(y_b)$ can contain at most 2^t elements (with overwhelming probability). Additionally, we always have at most $3 \cdot 2^{dn}$ non-empty multi-collision sets, because there can only be an element in a such set if there is at least one value from $\text{FVAL}_b \cup \text{CVAL}_b$. Hence, at any point there are at most $2^{2t} \cdot 3 \cdot 2^{dn} \leq 3 \cdot 2^{(d+2e)n}$ many collision pairs (M, M') appearing together in one of the multi-collision sets, for the constant $d + 2e < 1$.

Now suppose we make the j-th call to one of the collision finders, $\mathcal{C}_b$. After this call (and all subsequent f-function evaluations) take any pair $M \neq M'$ belonging to the same set $\text{MULTI}^{\mathsf{before}}_{b,j+1}(y_b)$ for some y_b. The next step is to note that, most likely, this pair M, M' cannot belong to some $\text{MULTI}^{\mathsf{before}}_{\overline{b},j+1}(y_{\overline{b}})$. Note that if M and M' lie in multi-collision sets $\text{MULTI}^{\mathsf{before}}_{\overline{b},j+1}(y_{\overline{b}})$ and $\text{MULTI}^{\mathsf{before}}_{\overline{b},j+1}(y'_{\overline{b}})$ for $y_{\overline{b}} \neq y'_{\overline{b}}$ then they clearly do not collide under $H_{\overline{b}}$ as those sets must be disjoint.

Assume, towards contradiction, that M, M' appear in a single multi-collision set for $\overline{b}$. We already know that M, M' cannot belong to some $\text{MULTI}^{\mathsf{before}}_{\overline{b},j}(y_{\overline{b}})$ of the previous stage, because none of these pairs constitutes a collision under $H_{\overline{b}}$, except with negligible probability. Hence, at least one of the two messages (say, M) must have been added to $\text{MULTI}^{\mathsf{before}}_{\overline{b},j+1}(y_{\overline{b}})$ because of an $f_{\overline{b}}$-function evaluation of $\mathcal{C}_b$ or via a direct evaluation of $f_{\overline{b}}$, taking into account that $\text{CVAL}_{\overline{b}}$ does not change between the two points in time.

Suppose that M is added to some set $\text{MULTI}^{\mathsf{before}}_{\overline{b},j+1}(y_{\overline{b}})$ via a new $f_{\overline{b}}$-value (which has not been in $\text{CVAL}_{\overline{b}}$), and assume that either M' is added only now or has already been in this set before the call. Consider the maximal i for which a new function value is added (when one would process the blocks m_i of message M through the iterated hash function). If the final value $\text{iv}_{\overline{b},t} = f_{\overline{b}}(\text{iv}_{\overline{b},t-1}, m_{t-1})$ is added ($i = t-1$) then, if for M' processing the final message block $\text{iv}_{\overline{b},t} = f_{\overline{b}}(\text{iv}'_{\overline{b},t-1}, m'_{t-1})$ has been in $\text{FVAL}_{\overline{b}}$ before or is added to $\text{FVAL}_{\overline{b}}$ now, we would have a lucky collision. So $\text{iv}_{\overline{b},t} = f_{\overline{b}}(\text{iv}'_{\overline{b},t-1}, m'_{t-1})$ must have been in $\text{CVAL}_{\overline{b}}$ before. But then this would contradict the f-replication resistance. For any other $i < t-1$ we note that, if $f_{\overline{b}}(\text{iv}_{\overline{b},j}, m_i)$ has not been determined before by $\mathcal{A}$, the probability that it matches any prefix of the at most $3 \cdot 2^{dn}$ previous values in $\text{FVAL}_{\overline{b}} \cup \text{CVAL}_{\overline{b}}$, is negligible (namely, at most $6 \cdot 2^{(d-1)n}$ by the image uncertainty). But this would contradict the maximal choice of i.

In conclusion, for any of the pairs M, M' there must still be an $f_{\overline{b}}$-value not in $\textsc{fval}_{\overline{b}} \cup \textsc{cval}_{\overline{b}}$ at this point, and the probability that the pair M, M' collides under $H_{\overline{b}}$ at all, is thus at most $2 \cdot 2^{-n}$. Therefore, the probability that any of the at most $3 \cdot 2^{(d+2e)n}$ pairs M, M' for $d + 2e < 1$ constitutes a collision under $H_{\overline{b}}$, is negligible. The same argument applies now vice versa, no pair M, M' from a set $\textsc{multi}^{\mathsf{before}}_{\overline{b},j+1}(y_{\overline{b}})$ yields a collision under H_b, except for some negligible error. This gives us the invariant.

The argument can now be set forth to the at most $2 \cdot \mathrm{poly}(n) + 1$ many phases, showing that the final multi-collision sets for $b = 0, 1$ never intersect in more than one element. This proves the theorem. $\square$