An extended abstract of this work appears in *8th Conference on Security and Cryptography for Networks (SCN 2012)*.

Hash Combiners for Second Pre-Image Resistance, Target Collision Resistance and Pre-Image Resistance have Long Output

Arno Mittelbach

Darmstadt University of Technology, Germany
`http://www.cryptoplexity.de`
`arno.mittelbach@cased.de`

Abstract. A (k, l) hash-function combiner for property P is a construction that, given access to l hash functions, yields a single cryptographic hash function which has property P as long as at least k out of the l hash functions have that property. Hash function combiners are used to hedge against the failure of one or more of the individual components. One example of the application of hash function combiners are the previous versions of the TLS and SSL protocols [10, 8].

The concatenation combiner which simply concatenates the outputs of all hash functions is an example of a robust combiner for collision resistance. However, its output length is, naturally, significantly longer than each individual hash-function output, while the security bounds are not necessarily stronger than that of the strongest input hash-function. In 2006 Boneh and Boyen asked whether a robust black-box combiner for collision resistance can exist that has an output length which is significantly less than that of the concatenation combiner [4]. Regrettably, this question has since been answered in the negative for fully black-box constructions (where hash function and adversary access is being treated as black-box), that is, combiners (in this setting) for collision resistance roughly need at least the length of the concatenation combiner to be robust [4, 5, 16, 17].

In this paper we examine weaker notions of collision resistance, namely: *second pre-image resistance* and *target collision resistance* [20] and *pre-image resistance*. As a generic brute-force attack against any of these would take roughly 2^n queries to an n-bit hash function, in contrast to only $2^{n/2}$ queries it would take to break collision resistance (due to the birthday bound), this might indicate that combiners for weaker notions of collision resistance can exist which have a significantly shorter output than the concatenation combiner (which is, naturally, also robust for these properties). Regrettably, this is not the case.

Keywords. hash functions, combiners, collision resistance, second pre-image resistance, target collision resistance, pre-image resistance

1 Introduction

In theory, hash functions are usually treated as ideal objects, that is, they are assumed to be random oracles or to hold certain properties such as collision resistance (it is difficult to find two messages that hash to the same value) or pre-image resistance (it is difficult, given an image, to find any pre-image). Assuming that these properties hold, this then allows us to prove protocols or constructions to be secure when instantiated with such a function. However, finding functions that provably hold any of the properties usually demanded of good hash functions is a difficult problem. Consequently, in practice, hash functions are heuristics that come only with a very limited number of guarantees. Thus it is not surprising that with time, attacks against practical hash functions are usually found that drastically lower the bounds assumed in theory. Many attacks have been presented for MD5 [25, 23, 21], and also for SHA-1 [13] first attacks have been published [24, 7, 1, 6]. This, in turn, led NIST (National Institute of Standards and Technology) to hold a competition to find a successor to the SHA-1 and SHA-2 families [14].[1]

Combiners.

Hash function combiners can be used to hedge against the failure of one (or more) of the components. A (k, l)-combiner is a construction that given access to l primitives implements the same primitive while guaranteeing that a certain property or multiple properties hold as long as a these are held by k out of the l input primitives. If the combiner ensures this for some property then it is said to be *robust* for that property.

Combiners are usually considered as black-box combiners, that is, the combiner only gets black-box access to its input hash functions. This is due to that i) this allows us the use the combiner with any hash functions; and ii) we are (so far) ignorant as to properly modeling white-box access. Consider, for example, the somewhat "pathological" combiner C^{H_1,H_2} with two input hash functions which on input M returns $H_1(M)$ if H_1 is collision resistant and $H_2(M)$ otherwise. Naturally, this combiner does all we want from a combiner for hash functions, but we have no idea, of how such a combiner could be implemented. In this paper, we limit our investigation to black-box combiners and speak henceforth only of combiner.

For hash functions the classical combiner robust for collision resistance is the concatenation combiner, i.e., $C_{||}(M) = H_1(M)||H_2(M)$ is a robust $(1, 2)$-combiner for collision resistance as naturally any two messages (M, M') that collide under $C_{||}$ also collide under both hash functions (i.e., $H_b(M) = H_b(M')$ for $b = \{1, 2\}$).

However, when simply concatenating the outputs of several hash functions, the output length grows significantly, while the security guarantee of the combined hash function does not necessarily increase. That is, we expect an adversary to find collisions for a hash function with output length n after roughly $2^{n/2}$ queries to the function (due to the generic birthday attack), a bound which can only be met by the concatenation combiner if all input functions were "ideal" hash functions to begin with (for which naturally a combiner would not be needed in the first place). Thus, Boneh and Boyen asked whether robust combiners for collision resistance exist that have a significantly shorter output length than the concatenation combiner [4]. This question has since been answered negatively [4, 5, 16, 17].

Weaker than Collision Resistance.

In practice, "full" collision resistance is not always required, i.e., for many applications a suitable level of security can be achieved with weaker notions such as second pre-image or target collision resistance. Here an adversary has to find a collision for a specific message. Think for example of checksums for programs. If an adversary wants to maliciously change the program then it has to make sure not to change the checksum in the process. Thus, the first part of the collision is fixed. Another example of a weaker property is pre-image resistance where given an image the best strategy for an adversary of finding a corresponding pre-image should be exhaustive search. Think password storage, for an application where pre-image resistance yields sufficient security.

For these variants of collision resistance the concatenation combiner is, naturally, also robust. While the generic birthday attack gives us an estimate of $2^{n/2}$ queries an adversary has to perform to find a (random) collision for an ideal function, an adversary would have to search the entire domain to break any of the

[1]In terms of security guarantees for practical hash functions the SHA-3 competition is also very telling. NIST received 64 entries, many of which were shown to lack the desired properties. Currently the competition is in its final stages and five finalists have been selected [14, 2, 11, 26, 3, 9].

properties second pre-image-, target collision- or pre-image resistance when considering ideal functions. It is thus interesting to study the question of finding combiners with short output that are robust for any of these weaker properties. A promising indication is also that short output combiners exist for the related (but privately-keyed) properties *message authentication codes* and *pseudo-randomness*. Regrettably, we will show that, as for the collision resistance case, such combiners cannot exist.

Impossibility Results.

For our definitions and proofs we closely follow Pietrzak's elegant proof for the collision resistance case [17, 15]. We will define combiners in the style of [4] as a pair of algorithms (C, P) where C implements the combiner logic and P provides a reduction from attacks on the combiner to attacks on the input functions. To prove the non-existence of a black-box combiner C with short output we will design an attack oracle $\mathcal{B}$ (that we will call breaking oracle) which breaks the investigated property (e.g., second pre-image resistance) for the combiner with noticeable probability but which does not help the security reduction P too much in breaking the property for the necessary number of input hash functions (2 in the case of a $(1, 2)$-combiner). The intuition is, that if the combiner compresses too much, than collisions appear on the combiner due to the compression and not due to collisions on the input hash functions. If we can guarantee that the breaking oracle only outputs such collisions, then the reduction P has to find collisions without the aid of the breaking oracle. However, if the combiner was indeed robust then the reduction P must, also with access to this specific breaking oracle, be able to break the property (e.g., find second pre-images) for all input hash functions (in case of a $(1, 2)$-combiner). Such a reduction P, on the other hand, allows us to compress a uniformly chosen random function $R : \{0,1\}^w \to \{0,1\}^v$ to below $2^w v$ bits. As this violates a corollary of Shannon's source coding theorem [22] we can argue that such combiners cannot exist.

The main contribution of our paper is to extend Theorem 1 given by Pietrzak [17] for the properties second pre-image resistance, target collision resistance and pre-image resistance. That is, randomized combiners robust for collision resistance, second pre-image resistance, target collision resistance or pre-image resistance have to have long output. We give an informal version of our main theorem for the case of deterministic combiners for two hash functions:

Theorem 1 (informal) *For some $n, m, v, w \in \mathbb{N}$ assume $C : \{0,1\}^m \to \{0,1\}^n$ is an efficient black-box-combiner for two hash functions of the form $H : \{0,1\}^w \to \{0,1\}^v$ that is robust for collision resistance, second pre-image resistance, target collision resistance or pre-image resistance. Then the combiner's output length n is bounded by:*

$$n \geq 2v - \mathcal{O}(\omega) \tag{1}$$

where ω is logarithmic in the number of hash function queries by the combiner.

Note that the bound roughly corresponds to the concatenation combiner's output length of $2v$. As Canetti et al. showed in [5], it is however possible to chop off a logarithmic number of bits (logarithmic in the number of oracle calls by the combiner) of the concatenation combiner while staying robust. As their combiner achieves the bound given in equation (1) the bound is tight.

Further note that we will prove the result for finite domain hash functions. However, as this is an impossibility result, proving it for a finite domain actually makes the result stronger, as every secure hash function that takes arbitrary length messages also has to be secure when considering the subset of only fixed length input messages.

Related Work.

Combiners for the properties second pre-image resistance and pre-image resistance with short output length have been studied by Rjasko [19] who gives an impossibility result for a special case of deterministic combiners where the reduction can query the breaking oracle only once. As shown in [17] this simplification allows for a much simpler proof (also see Appendix A) than the general case for probabilistic combiners where the reduction can query the adversary (or breaking oracle in our terminology) multiple times.

2 Preliminaries

2.1 Notation

Unless stated otherwise, lower-case letters such as $n \in \mathbb{N}$ represent natural numbers. With 1^n we denote the unary representation of n and with $\langle n \rangle_b$ its binary representation padded with zeros to length b. Upper-case letters in standard typeface like M stand for bit strings. We denote with $\{0,1\}^n$ the set of all bit strings M of length $|M| = n$, while $\{0,1\}^*$ denotes the set of all bit strings. For bit strings $X, Y \in \{0,1\}^*$ we denote with $X||Y$ the concatenation. If $\mathcal{X}$ is a set then by $|\mathcal{X}|$ we denote its cardinality. By $M \leftarrow \mathcal{X}$ we mean that M is chosen uniformly from $\mathcal{X}$, if $\mathcal{X}$ is a distribution then $M \leftarrow \mathcal{X}$ denotes that M is chosen accordingly. The logarithm log is always to base 2.

PTM stands for Polynomial time Turing Machine and PPTM for Probabilistic Polynomial time Turing Machine. Upper-case letters in calligraphy like $\mathcal{A}$ usually denote a PPTM. We will often simply call them adversary or algorithm and we write $\mathcal{A}$(M) if M is initially written on the Turing Machine's input tape (i.e., the algorithm runs with input M). By $X \leftarrow \mathcal{A}$ we denote that X is output by algorithm $\mathcal{A}$. If the Turing Machine has black-box access to one or more oracles $\mathcal{O}_1, ..., \mathcal{O}_z$ (these will usually be hash functions), we denote this by adding the oracles in superscript: $\mathcal{A}^{\mathcal{O}_1,...,\mathcal{O}_z}$.[2] In some instances we will speak of oracle circuits instead of oracle PPTMs to hint that this Turing Machine runs in the non-uniform complexity model.

By $\text{qry}^{\mathcal{O}_i}(\mathcal{A}^{\mathcal{O}_1,...,\mathcal{O}_z}(M))$ we denote the set of all of $\mathcal{A}$'s queries to oracle $\mathcal{O}_i$ when algorithm $\mathcal{A}$ runs on input M.

If X and Z are random variables then $Pr[X = y]$ denotes the probability that X takes on the value of y. By $Pr[X = y|Z]$ we denote the conditional probability of $X = y$ given Z. If **pre** is a predicate or event then $Pr[\textbf{pre}]$ denotes the probability that the predicate is true (the event occurs)

We write $Pr[\text{step}_1; ...; \text{step}_i : \text{condition}]$ which describes a random experiment and which should be read as: the probability that the condition holds after the steps were executed in consecutive order.

If A is an event then by $\neg A$ we denote the complementary event, that is, $Pr[\neg A] = 1 - Pr[A]$. By $\wedge$ (resp. $\vee$) we denote the conjunction (resp. disjunction) of events: if A and B are events then $Pr[A \wedge B]$ is the probability that both events A and B occur and $Pr[A \vee B]$ is the probability that at least one of the events A or B does occur.

2.2 Hash Functions and their Properties

Formally hash functions are defined as a family of functions together with a key generation algorithm HKGen that picks one of the functions to be used. That is, A *hash function* (family) is a pair of efficient algorithms $\mathcal{H} = (\text{HKGen}, H)$ where $\text{HKGen}(1^n)$ is a probabilistic algorithm that takes as input the security parameter 1^n and outputs a key K (note that the security parameter is implicit in K) and $H(K, M) := H_K(M)$ is a deterministic algorithm that takes a key K and message $M \in \{0,1\}^*$ as input and outputs a

[2] In this case the Turing Machine has extra oracle tapes (one per oracle). $\mathcal{A}$ can write query X on oracle tape i and gets the oracle's answer $\mathcal{O}_i$ in the next step written on the tape.

hash value $H_K(M) \in \{0,1\}^n$. We will drop the subscript and write $H(M)$ if it is clear from context which key the function gets.

We require different properties from hash functions depending on the application. The properties that we are interested in in this paper are *collision resistance* (CR, it should be difficult to find two messages that hash to the same value), *second pre-image resistance* (SPR, it should be difficult given a message to find a second message that hashes to the same value) its variant *target collision resistance*[3] (TCR) and *pre-image resistance* (OW, given an image it should be difficult to find a corresponding pre-image).

Definition 1 (Collision Resistance) *A* hash function $\mathcal{H} = (HKGen, H)$ *is collision resistant (CR), if the advantage for every efficient adversary* $\mathcal{A}$ *in the following experiment is negligible in* n*:*

$$Adv^{cr}_{\mathcal{A}}(n) = Pr\left[\begin{matrix} K \leftarrow HKGen(1^n); \\ (M, M') \leftarrow \mathcal{A}(K) \end{matrix} \quad : \quad \begin{matrix} M \neq M' \wedge \\ H_K(M) = H_K(M') \end{matrix}\right] \approx 0$$

The probability is over the selection of $K \leftarrow HKGen(1^n)$ *and* $\mathcal{A}$*'s internal coin tosses.*

Second pre-image- and target-collision resistance are closely related to collision resistance. For successfully attacking the collision resistance property of a hash function it is sufficient for an attacker to find any pair of messages (M, M') such that they hash to the same value. The idea behind second pre-image- and target collision resistance is that it should be difficult for any attacker to find a specific collision and not just a random one. By specific we mean that the first part of the collision M is somehow predefined. The difference between the two definitions (second pre-image and target collision resistance) is who is specifying the first part of the collision: when considering second pre-image resistance the first part of the collision is sampled according to some distribution (in the upcoming definition denoted with $\mathcal{M}(1^n)$) while when considering target collision resistance the adversary may choose the first part. The literature does often not clearly distinguish between the two definitions or uses the terms interchangeably. An introduction to several variants of these notions can be found in [20].

Definition 2 (Second Pre-Image Resistance) *We call a hash function* $\mathcal{H} = (HKGen, H)$ second pre-image resistant *(SPR) with respect to distribution* $\mathcal{M}$*, if the advantage for every efficient adversary* $\mathcal{A}$ *in the following experiment is negligible in* n*:*

$$Adv^{spr}_{\mathcal{A}}(n) = Pr\left[\begin{matrix} K \leftarrow HKGen(1^n); M \leftarrow \mathcal{M}(1^n); \\ M' \leftarrow \mathcal{A}(K, M) \end{matrix} \quad : \quad \begin{matrix} M \neq M' \wedge \\ H_K(M) = H_K(M') \end{matrix}\right] \approx 0$$

The probability is over the selection of $K \leftarrow HKGen(1^n)$*, the choice of* $M \leftarrow \mathcal{M}$ *and* $\mathcal{A}$*'s internal coin tosses.*

Definition 3 (Target Collision Resistance) *We call a hash function* $\mathcal{H} = (HKGen, H)$ target collision resistant *(TCR), if the advantage for every efficient adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *in the following experiment is negligible in* n*:*

$$Adv^{tcr}_{\mathcal{A}}(n) = Pr\left[\begin{matrix} (M, st) \leftarrow \mathcal{A}_1(1^n); K \leftarrow HKGen(1^n); \\ M' \leftarrow \mathcal{A}_2(K, M, st) \end{matrix} \quad : \quad \begin{matrix} M \neq M' \wedge \\ H_K(M) = H_K(M') \end{matrix}\right] \approx 0$$

The probability is over the selection of $K \leftarrow HKGen(1^n)$ *and* $\mathcal{A}$*'s internal coin tosses.*

[3]Target collision resistant hash functions are often also referred to as *universal one-way hash functions* [12].

The additional value st in the definition of target collision resistance is meant to allow the adversary to encode some sort of state shared between the first adversary $\mathcal{A}_1$ fixing the first part of the collision and the second adversary $\mathcal{A}_2$ which is tasked with finding a collision for the fixed message.

A notion closely related to second pre-image resistance is that of one-wayness or pre-image resistance. Here an adversary is given an image (but not the pre-image) and is tasked to find any pre-image that hashes to the given image. Formally:

Definition 4 (Pre-Image Resistance) *A hash function $\mathcal{H} = (HKGen, H)$ is called* pre-image resistant *or* one-way *(OW) with respect to distribution $\mathcal{M}$, if the advantage for every efficient adversary $\mathcal{A}$ in the following experiment is negligible in n:*

$$Adv_{\mathcal{A}}^{ow}(n) = Pr\left[\begin{array}{r} K \leftarrow HKGen(1^n); M \leftarrow \mathcal{M}(1^n); \\ M' \leftarrow \mathcal{A}(K, H_K(M)) \end{array} \; : \; H_K(M) = H_K(M')\right] \approx 0$$

The probability is over the selection of $K \leftarrow HKGen(1^n)$, the choice of $M \leftarrow \mathcal{M}$ and $\mathcal{A}$'s internal coin tosses.

2.3 Predicates Capturing Events

In the upcoming proofs and in our definition of a robust combiner we need to formalize the event that an adversary finds second pre-images, target collisions or simply pre-images. For this we examine a random experiment $\mathcal{A}^{H_1,\dots,H_l}$ where an adversary plays against hash functions $H_1, \dots, H_l$. Here predicates **spr** (resp. **tcr** or **ow**) are true if and only if the adversary in the course of the experiment finds a second pre-image (resp., target collision, pre-image). By finding we mean that the adversary actually performs one query to a hash function H_i which yields the second pre-image (or target collision or pre-image) in respect to some target message(reps., target image). The definitions are closely related to Pietrzak's definition for collision resistance [17]. We here present the definition for second pre-image resistance and give the definitions for the collision resistance, target collision resistance and pre-image resistance cases in Appendix B (Definitions 10, 11 and 12).

Definition 5 *The predicate $\boldsymbol{spr}^{H_i(X_i)}(\mathcal{A}^{H_1,\dots,H_l})$ is defined for the random experiment $\mathcal{A}^{H_1,\dots,H_l}$ and target message X_i and holds if $\mathcal{A}$ finds a second pre-image for X_i for function H_i; that is, in the course of the computation of $\mathcal{A}^{H_1,\dots,H_l}$ an oracle call to H_i is made with messages $(X_{\boldsymbol{spr}})$ for which $H_i(X_i) = H_i(X_{\boldsymbol{spr}})$ and $X_i \neq X_{\boldsymbol{spr}}$. Formally:*

$$\boldsymbol{spr}^{H_i(X_i)}(\mathcal{A}^{H_1,\dots,H_l}) \iff$$
$$\exists\, X_{\boldsymbol{spr}} \in qry^{H_i}(\mathcal{A}^{H_1,\dots,H_l}) : X_i \neq X_{\boldsymbol{spr}} \;\wedge H_i(X_i) = H_i(X_{\boldsymbol{spr}})$$

For subset $\mathcal{H} \subseteq \{H_1, ..., H_l\}$ and target messages $X_1, ..., X_l$ we define the predicate $\boldsymbol{spr}^{\mathcal{H}}_{X_1,\dots,X_l}(\mathcal{A}^{H_1,\dots,H_l})$ to hold if second pre-images are found for all hash functions in $\mathcal{H}$ (for corresponding target message X_i):

$$\boldsymbol{spr}^{\mathcal{H}}_{X_1,\dots,X_l}(\mathcal{A}^{H_1,\dots,H_l}) \iff \forall\, H_i \in \mathcal{H} : \boldsymbol{spr}^{H_i(X_i)}(\mathcal{A}^{H_1,\dots,H_l})$$

For $1 \leq n \leq l$ we define the predicate $\boldsymbol{spr}_{n,X_1,\dots,X_l}(\mathcal{A}^{H_1,\dots,H_l})$ to hold if a second pre-image is found for n of the l hash oracles:

$$\boldsymbol{spr}_{n,X_1,\dots,X_l}(\mathcal{A}^{H_1,\dots,H_l}) \iff \exists\, \mathcal{H} \subseteq \{H_1, ..., H_l\}, |\mathcal{H}| \geq n : \boldsymbol{spr}^{\mathcal{H}}_{X_1,\dots,X_l}(\mathcal{A}^{H_1,\dots,H_l})$$

2.4 Randomized Black-Box Combiners for Cryptographic Hash Functions

As usual for impossibility results we want to be as general as possible about the capabilities of the considered combiners. We consider randomized black-box combiners after Pietrzak [17] in the non-uniform setting. This means that the combiners get as additional input their randomness R which can be regarded as some sort of public key. The (k, l) combiner is formalized as a pair of algorithms (C, P) where C is a non-uniform circuit implementing the logic of the combiner and P provides a security reduction. For this C gets access to the l hash functions $H_1, \dots, H_l$ and P gets access to a breaking oracle[4] $\mathcal{B}$ which can produce second pre-images (or target collision, or pre-images respectively) with a certain success probability ρ. If security reduction $P^{\mathcal{B}, H_1, \dots, H_l}$ finds second pre-images (or target collisions, pre-images) for $l - k + 1$ of the input hash functions with noticeable probability (with respect to the breaking oracle's success rate ρ), we call the combiner ρ-robust for second pre-image resistance (or target collision resistance, respectively). Thus a combiner is ρ-robust if and only if such a breaking oracle cannot exist if at least k out of the l input hash functions hold the property.

We here present the definition for randomized combiners for the case of second pre-image resistance. For the collision resistant case see [17] or Definition 13 in the appendix, which also gives the definition for the target collision resistance and pre-image resistance cases.

Definition 6 *A randomized (k, l)-combiner for hash functions $H_1, \dots, H_l$ of the forms $\{0,1\}^w \to \{0,1\}^v$ is a pair of efficient algorithms (C, P) where $C : \mathcal{R} \times \{0,1\}^m \to \{0,1\}^n$ is an oracle circuit and P is an oPPTM[5] providing the security reduction for C.*

Let $0 \leq \rho \leq 1$. Oracle $\mathcal{B}^{\boldsymbol{spr}}$ ρ-breaks $C^{H_1, \dots, H_l}$ (for SPR) if it outputs a second pre-image to input message $M \leftarrow \{0,1\}^m$ for $C^{H_1, \dots, H_l}(R, M)$ with probability ρ (over the choice of message and randomness). If no second pre-image is found it outputs $\perp$.

*The combiner is called **ρ-robust for SPR** if for some non-negligible $0 < \epsilon \leq 1$ and any choice of functions $H_1, \dots, H_l$ and for any breaking oracle $\mathcal{B}^{\boldsymbol{spr}}$ that ρ-breaks $C^{H_1, \dots, H_l}$, the random experiment $P^{\mathcal{B}^{spr}, H_1, \dots, H_l}$ on input $(M_1, \dots, M_l)$ with $M_i \leftarrow \{0,1\}^m$ (for $i = 1, \dots, l$) finds second pre-images for $l - k + 1$ input hash functions with probability:*

$$Pr[\boldsymbol{spr}_{l-k+1, M_1, \dots, M_l}(P^{\mathcal{B}^{spr}, H_1, \dots, H_l}(M_1, \dots, M_l))] \geq \epsilon \tag{2}$$

The combiner is called robust *for SPR if it is efficient and ρ-robust for every non negligible $\rho(v)$.*

Note that the breaking oracle's success probability ρ can be a function of $v \in \mathbb{N}$, i.e., the hash functions' output length. Further note, that for second pre-image resistance we cannot take a fraction of the randomness as success criterion for the breaking oracle (as we have in the TCR case and Pietrzak did for the collision resistance case in [17]). This is due to that a hash function might be designed such that some images have exactly one pre-image and thus a breaking oracle does not stand a chance in finding a second pre-image for such a message/pre-image pair.

Finally, note that we could have fixed the reduction's success probability ϵ to a constant value, due to the fact that we are considering randomized combiners. Given some combiner (C, P) that satisfies equation (2) for some non-negligible ϵ, we can easily construct a new combiner C, P^* that satisfies equation (2) with probability $\epsilon^* > \epsilon$ by simply repeatedly calling P with renewed random coins. We will later fix ϵ to $2/3$ (an arbitrary choice) to simplify notation.

Definition 7 (Efficiency) *Let q_C be the number of oracle queries performed by C and q_P an upper bound for the number of oracle calls made by P, then the combiner (C, P) is called* efficient *if both q_P and q_C are polynomial in v.*

[4] Think of the breaking oracle as the best known adversary against the combiner.

[5] See [17] for how to treat non-uniform reductions.

Remark. [On efficiency definition] Note that it is sufficient to only count successful calls to the breaking oracle, that is, we do not need to count queries to $\mathcal{B}$ where the answer is $\perp$. By not counting unsuccessful calls to the breaking oracle we actually make the impossibility result stronger as even then, as we will see the reduction P will have to make exponentially many calls to the hash function oracle to succeed. This can be done as, unsuccessful even the successful queries to $\mathcal{B}$ will not help P (too much) in its task and unsuccessful queries cannot be used by P to hide hash function calls.

Further note that throughout this paper we assume that q_C, q_P^H and q_P^B are at least one. This can be safely assumed as the contrary would be rather uninteresting: consider for example a combiner which does not use its hash functions. It is trivially non-robust as collisions for the combiner cannot be reduced to collisions on any of the input functions.

3 Robust Combiners have Long Output

In this section we will give the formal definition of Theorem 1 together with its proof for the case of second pre-image resistance. In Appendix D we present the necessary adaptations for the cases of target collision resistance and pre-image resistance:

Theorem 2 *For some $n, m, v, w \in \mathbb{N}$ assume (C, P) is an efficient (k,l)-black-box-combiner for hash functions of the form $\{0,1\}^w \to \{0,1\}^v$. Let $C : \mathcal{R} \times \{0,1\}^m \to \{0,1\}^n$ be robust for collision resistance, second pre-image resistance, target collision resistance or pre-image resistance. Then the combiner's output length n is bounded by:*

$$n \geq (l - k + 1)v - \mathcal{O}(\log q_C) \tag{3}$$

where q_C is the number of hash function queries performed by C.

3.1 Additional Definitions

As a gentle introduction to the definitions presented so far we prove a specialized version of Theorem 2 in Appendix A.

For the upcoming proof of our main theorem we need a notion of second pre-images (resp. target collisions and pre-images) that are *safe* for the combiner. "Safe for the combiner" means that the evaluation of the message pair (M, M') on a specific combiner C does not yield a trivial second pre-image for hash function H and target message X.[6] Trivial in the sense that during the evaluation combiner C on messages M and M' queries hash function H on a message $X_{\mathbf{spr}}$ for which $X \neq X_{\mathbf{spr}}$ and $H(X) = H(X_{\mathbf{spr}})$:

Definition 8 (Safe Second Pre-Image) *Let $H : \{0,1\}^w \to \{0,1\}^v$ be a hash function and $C : \{0,1\}^m \to \{0,1\}^n$ some oPPTM. Let $X_{target} \in \{0,1\}^w$. We say that the message $M_{\boldsymbol{spr}} \in \{0,1\}^m$ is a safe second pre-image with respect to message $M \in \{0,1\}^m$ and function H (with respect to C^H and X_{target}) if*

1. *$C^H(M) = C^H(M_{\boldsymbol{spr}})$ (but not necessarily $M \neq M_{\boldsymbol{spr}}$, that is, $(M, M_{\boldsymbol{spr}})$ may be a pseudocollision)*
2. *the evaluation of $C^H(\cdot)$ on inputs M and $M_{\boldsymbol{spr}}$ does not involve a call to $H(X)$ with $X \neq X_{target}$ and $H(X) = H(X_{target})$:*

$$\forall X \in qry^H(C^H(M)) \cup qry^H(C^H(M_{\boldsymbol{spr}})) : X = X_{target} \vee H(X) \neq H(X_{target})$$

We define the predicate $\boldsymbol{safeSpr}_{H,X}^{C^H}(M, M_{\boldsymbol{spr}})$ iff $(M, M_{\boldsymbol{spr}})$ is a safe second pre-image for hash function H and target message X.

[6]Note that for the case of pre-image resistance we do not consider a message pair but only a single message M.

For $k \in \mathbb{N}$ we denote with $\boldsymbol{safeSpr}^{C^{H_1,\dots,H_l}}_{k,X_1,\dots,X_l}(M, M_{\boldsymbol{spr}})$ that $(M, M_{\boldsymbol{spr}})$ is a safe second pre-image for messages $X_1, .., X_l$ for k out of C's l oracles H_i $(i = 1, ..., l)$. Here X_i is the target message for function H_i $(i = 1, ..., l)$.

We give the definitions for safe target-collisions (Definition 14) and safe pre-images (Definition 15) in Appendix B.3.

Compressibility.

The main idea in the upcoming proof is to show that if a robust combiner with short output exists then we can compress a uniformly random function $H : \{0,1\}^w \to \{0,1\}^v$ below $2^w v$ bits. This, however, is not possible due to a result proved by Shannon (which we will present in Proposition 2) and hence such a combiner cannot exist. To express this we need a notion of compressibility:

Definition 9 (Compressibility) *A random variable H can be compressed to s bits, if two functions $\boldsymbol{com}$ and $\boldsymbol{dec}$ (for compression, resp. decompression) exist such that on average the size of $com(\tilde{H})$ (where $\tilde{H}$ is an instantiation of the random variable) is less or equal to s bits and that the probability of $\boldsymbol{dec}(\boldsymbol{com}(\tilde{H}))$ is exactly 1; that is $\boldsymbol{dec}(\cdot)$ is always able to completely restore $\tilde{H}$:*

$$E[|\boldsymbol{com}(\tilde{H})|] \leq s \qquad \text{and} \qquad Pr[\boldsymbol{dec}(\boldsymbol{com}(\tilde{H})) = \tilde{H}] = 1$$

3.2 Proof of Theorem 2

We will now present the proof of our main theorem for the case of second-preimage resistance. A description of what parts need to be changed for the cases of target collision resistance and pre-image resistance is given in Appendix D. Our argument follows Pietrzak's in [17, 15] where he proves the theorem for collision resistance. We are going to prove Theorem 2 indirectly by proving a proposition which informally says that if the output length of a combiner (C, P) is short, then it cannot be efficient, as P will have to make exponentially many queries to its hash oracle for (C, P) to be robust:

Proposition 1 *For some $n, m, w, v, l, k \in \mathbb{N}$ Let $C : \mathcal{R} \times \{0,1\}^m \to \{0,1\}^n$ be an oracle circuit with input domain $m := l \cdot (v + 1)$, with q_C oracle gates for every hash function $H_i : \{0,1\}^w \to \{0,1\}^v$ (for $i = 1, ..., l$) and with output range*

$$n := (l - k + 1) \cdot (v - \log q_C) - t \qquad (t > 0) \tag{4}$$

Let $\rho := (1 - 2^{-t+l+2})/\binom{l}{k}$. If (C, P) is a ρ-robust (for SPR, TCR or OW) (k, l)-combiner where reduction P has success probability at least $2/3$, making $q_P^{\mathcal{B}}$ queries to the breaking oracle and q_P^H queries to its hash functions then

$$v \leq \log q_P^H + 2 + \log l \tag{5}$$

or equivalently

$$2^v \leq q_P^H \cdot 4 \cdot l \tag{6}$$

Before we prove the proposition, let us show how it implies Theorem 2:

Proof (of Theorem 2). Proposition 1 states that if a (k, l) combiner (C, P) is ρ-robust with $\rho = (1 - 2^{-t+l+2})/\binom{l}{k}$ and the combiner has domain $m := l \cdot (v + 1)$ and range $n := (l - k + 1) \cdot (v - \log q_C) - t$ for some $t > 0$, then it needs to make exponentially many calls to its oracles as $q_P^H \geq 2^{v-2-l}$. That is, P is not efficient. As every robust combiner is also ρ-robust for $\rho = (1 - 2^{-t+l+2})/\binom{l}{k}$ this proves Theorem 2 for the special case where the combiner is shrinking by $m - n = l \cdot (v+1) - (l-k+1) \cdot (v - \log q_C) + t$ as the combiner is either not efficient or not robust. However, as a combiner for arbitrary length hash functions necessarily has to work for fixed length functions as well the result presented here directly applies for arbitrary and infinite domain combiners. $\square$

3.3 An Outline

To prove Proposition 1 we have to find oracles $H_1, \ldots, H_l$ and $\mathcal{B}$ such that the breaking oracle $\mathcal{B}$ finds second pre-images for combiner $C^{H_1,\ldots,H_l} : \{0,1\}^m \to \{0,1\}^n$ with range $n < (l-k+1)v - \mathcal{O}(\log q_C)$ while k out of the l hash functions stay second pre-image resistant in relation to $P^{\mathcal{B},H_1,\ldots,H_l}$; that is, $P^{\mathcal{B},H_1,\ldots,H_l}$ is not able to find second pre-images for more than $l-k+1$ hash functions even with access to the powerful breaking oracle $\mathcal{B}$.

For this we are going to use hash functions chosen uniformly at random from the space of all functions of the form $\{0,1\}^w \to \{0,1\}^v$. For the breaking oracle $\mathcal{B}$ we are going to carefully design a function which outputs only safe second pre-images (cf. Definition 8) with respect to the target messages $X_1, \ldots, X_l \in \{0,1\}^w$ (denoted by $\mathcal{B}^{X_1,\ldots,X_l}$) given to the security reduction P; that is, P cannot simply use the breaking oracle to get all the necessary second pre-images. We use the following idea: we take another random function $\phi : \{0,1\}^* \to \{0,1\}^m$ to completely define the breaking oracle. On receiving input (R, M) the oracle starts counting upwards and tries every natural number $i = 1, 2, ...$ concatenated to the randomness as a possible collision:

$$C^{H_1,\ldots,H_l}(R, \phi(R||\langle i \rangle)) \stackrel{?}{=} C^{H_1,\ldots,H_l}(R, M)$$

If a collision is found (note that this may be a pseudocollision, i.e., $M = \phi(R||\langle i \rangle)$) we need to make sure that the found second pre-image does not yield trivial second pre-images for $l-k+1$ hash functions. The problem is that, a priori, we do not know the target messages an adversary using the breaking oracle is interested in. For now we will simply think of the breaking oracle as being initialized with the correct message $X_1, \ldots, X_l \in \{0,1\}^w$. That is, the breaking oracle $\mathcal{B}^{X_1,\ldots,X_l}$ will ensure that no trivial second pre-images for messages $X_1, \ldots, X_l$ are found. For this it checks whether the hash function calls during the evaluation of $C^{H_1,\ldots,H_l}(R, M)$ and $C^{H_1,\ldots,H_l}(R, \phi(R||\langle i \rangle))$ (i.e., the input the to breaking oracle (R, M) and the found (pseudo)-collision by the breaking oracle) contains $l-k+1$ trivial second pre-images for message $X_1, \ldots, X_l$ and hash functions $H_1, \ldots, H_l$. If that is not the case the breaking oracle outputs $\phi(R||\langle i \rangle)$, otherwise it exits and outputs $\perp$. In other words, we make sure that the collision found by the breaking oracle is a safe second pre-image (cf. Definition 8) for messages $X_1, \ldots, X_l$.

After proving that such a breaking oracle initialized to message $X_1, \ldots, X_l \in \{0,1\}^w$ does indeed break the security of any (k,l)-combiner with high probability it remains to prove that the reduction $P^{\mathcal{B},H_1,\ldots,H_l}$ with access to our breaking oracle and the (uniformly random) hash functions makes a poor job in finding second pre-images. For this we will use a corollary from Claude E. Shannon's source coding theorem (see any good introduction to information theory or Shannon's original [22]):

Proposition 2 *A uniformly random function $H : \{0,1\}^w \to \{0,1\}^v$ (with prefix-free domain[7]) cannot be compressed to less than $2^w v$ bits.*

What we will show is that if $P^{\mathcal{B},H_1,\ldots,H_l}$ is able to find enough (i.e., more than $l-k$) second pre-images for $H_1, \ldots, H_l$ with noticeable probability, then we are able to compress $H_1, \ldots, H_l$ to less than $l2^w v$ bits. That is, we are going to design a custom compression (**com**) and decompression (**dec**) algorithm that given $H_1, \ldots, H_l$, P and $\mathcal{B}$ uses P to compress the hash functions $H_1, \ldots, H_l$. As these are uniformly random, this forms a contradiction to Proposition 2 and hence such a P cannot exist. What is left to show

[7]Note that all functions $\{0,1\}^w \to \{0,1\}^v$ have a prefix-free domain and range, as all elements in either set have the same length.

is that in our case **com** and **dec** can initialize the breaking oracle with the target messages given to P before P is given access to the breaking oracle (remember that our breaking oracle $\mathcal{B}^{X_1,\dots,X_l}$ outputs only second pre-images that are safe for the specific messages $X_1,\dots,X_l$). We will see that this is in fact the case and that thus $\mathcal{B}$ will not be of too much use for P.

3.4 The Proof

The Oracles.

To prove Proposition 1 we begin by defining the $l+1$ oracles $H_1,...,H_l$ and $\mathcal{B}$. The hash functions H_i ($i = 1,...,l$) are each sampled uniformly at random from the set of all functions of the form $\{0,1\}^w \to \{0,1\}^v$. The breaking oracle $\mathcal{B}$ will be defined by a function $\phi : \{0,1\}^* \to \{0,1\}^m$ which is also sampled uniformly at random from all functions of the form $\{0,1\}^* \to \{0,1\}^m$. Function ϕ defines for a message $M \in \{0,1\}^m$ and randomness $R \in \mathcal{R}$ a pseudo-second pre-image $M_{\mathbf{spr}}$ for $C^{H_1,\dots,H_l}(R,M)$ as $M_{\mathbf{spr}} := \phi(R||\langle i\rangle)$, where i is the smallest integer such that $C^{H_1,\dots,H_l}(R,M) = C^{H_1,\dots,H_l}(R,M_{\mathbf{spr}})$.

Our goal is to make sure that the breaking oracle outputs only safe second pre-images (with respect to some target messages $X_1,\dots,X_l$) where by safe we mean that the oracle's output second pre-images should be safe for k out of the l hash functions (i.e., $\mathbf{safeSpr}^{C^{H_1,\dots H_l}}_{k,X_1,\dots,X_l}(M,M_{\mathbf{spr}})$, compare Definition 8). Furthermore, each second pre-image output by $\mathcal{B}$ has to be safe for exactly the same k hash functions.

Let us assume that $\mathcal{B}^{X_1,\dots,X_l}$ was initialized with target messages $X_1,...,X_l$. As abbreviation we will simply write $\mathcal{B}$ when we mean a breaking oracle that was initialized with fixed but random target messages. We now introduce sets S_i (for $i = 1,...,l$) that comprise all safe message, randomness pairs for hash function H_i. If $M_{\mathbf{spr}}$ is defined through ϕ as described above then we can define sets S_i as:

$$S_i := \{(R,M) \in \mathcal{R} \times \{0,1\}^m : \mathbf{safeSpr}^{C_{H_1,\dots,H_l}}_{H_i,X_i}(M,M_{\mathbf{spr}})\} \quad \text{for } i \in \{1,...,l\}$$

We now define the intersection of k out of the l sets S_i such as to maximize the number of elements in the intersection. Let therefor

$$\Gamma_{\max} := \underset{\substack{\Gamma\subseteq\{1,\dots,l\} \\ |\Gamma|=k}}{\operatorname{argmax}} \left| \bigcap_{i\in\Gamma} S_i \right| \quad .$$

With $\Gamma_{\max}$ we can define the maximal intersection

$$\mathcal{R}_\Gamma := \bigcap_{i\in\Gamma_{\max}} S_i$$

which allows us to formalize the breaking oracle $\mathcal{B}$ as:

$$\mathcal{B}^{X_1,\dots,X_l}(R,M) := \begin{cases} M_{spr} & \text{if } (R,M) \in \mathcal{R}_\Gamma \\ \perp & \text{otherwise} \end{cases}$$

Thus, on input (R,M) our breaking oracle $\mathcal{B}^{X_1,\dots,X_l}$ will output a second pre-image $M_{\mathbf{spr}}$ such that $C^{H_1,\dots,H_l}(R,M) = C^{H_1,\dots,H_l}(R,M_{\mathbf{spr}})$ only if it is safe for at least k of the l hash functions in regard to the target messages $X_1,...,X_l$. Also note that two second pre-images for inputs (R,M) and (R',M') will be safe for (at least) the same k hash functions.

The breaking oracle $\mathcal{B}$ is ρ-robust for all (k,l)-combiners.

We will now show that our breaking oracle ρ-breaks every (k,l)-combiner with some noticeable ρ. Note that, by definition, the breaking oracle ρ-breaks $C^{H_1,\dots,H_l}$ with

$$\rho = \frac{|\mathcal{R}_\Gamma|}{|\mathcal{R}| \cdot 2^m}$$

This we will prove using the following lemma stating informally that sampled second pre-images will be safe for at least k out of l hash functions for (k,l)-combiners most of the time.[8] (The proof for Lemma 1 is given in Appendix C.1.).

Lemma 1 *Let $C : \{0,1\}^m \to \{0,1\}^n$ be any oracle machine with q_C oracle gates per hash function $H_i : \{0,1\}^w \to \{0,1\}^v$ (with $i = 1,..,l$). Let $X_1,\dots,X_l \in \{0,1\}^w$ be target messages for functions $H_1,\dots,H_l$. For messages $M, M_{\boldsymbol{spr}}$ sampled as $M \leftarrow \{0,1\}^m$ and $M_{spr} \leftarrow (C^{H_1,\dots,H_l})^{-1}(M)$ we have:*

$$Pr[\boldsymbol{safeSpr}_{k,X_1,\dots,X_l}^{C^{H_1,\dots,H_l}(\cdot)}(M, M_{spr})] \geq 1 - q_C^{l-k+1} \cdot \binom{l}{l-k+1} \cdot 2^{n-(v+1)(l-k+1)}$$

From Lemma 1 we will now deduce a lower bound for ρ. Let $I_{R,M} := 1$ if (R,M) is safe for at least k of the l hash functions H_i and $I_{R,M} = 0$ otherwise. Formally:

$$I_{R,M} := \begin{cases} 1 & \text{if } \mathbf{safeSpr}_{k,X_1,\dots,X_l}^{C^{H_1,\dots,H_l}(R,\cdot)}(M, M_{\mathbf{spr}}) \\ 0 & \text{otherwise} \end{cases}$$

Lemma 1 gives us a lower bound on the probability for event $I_{R,M} = 1$ as our breaking oracle samples random second pre-images. Using an upper bound of 2^l on the binomial coefficient we have that

$$Pr[I_{R,M} = 1] \geq 1 - q_C^{l-k+1} \cdot 2^{n-(v+1)(l-k+1)+l}$$

Now with $n := (l-k+1)\cdot(v - \log q_C) - t$ (see equation 4) we get

$$\begin{aligned} Pr[I_{R,M} = 1] &\geq 1 - q_C^{l-k+1} \cdot 2^{n-(v+1)(l-k+1)+l} \\ &\geq 1 - q_C^{l-k+1} \cdot 2^{(l-k+1)\cdot(v-\log q_C)-t-(v+1)(l-k+1)+l} \\ &= 1 - q_C^{l-k+1} \cdot 2^{\log q_C^{-(l-k+1)}} \cdot 2^{-(l-k+1)} \cdot 2^{-t+l} \\ &\geq 1 - 2^{-t+l} \end{aligned}$$

Remember that, by definition, the breaking oracle $\mathcal{B}$ has an inherent set $\mathcal{R}_\Gamma$ and only outputs second pre-images if the input (R,M) is an element of $\mathcal{R}_\Gamma$. With Γ we have fixed the set of k hash functions H_i ($i \in \Gamma_{\max}$) such that $\mathcal{R}_\Gamma$ is maximized. The indicator $I_{R,M}$ equals 1, if we have a safe second pre-image for k hash functions but not necessarily for the k hash functions specified by $\Gamma_{\max}$. We can however safely assume that ρ is greater than the probability of a safe second pre-image ($Pr[I_{R,M} = 1]$) divided by the possibilities of choosing k out of l elements (as $\Gamma_{\max}$) was designed to maximize this probability). Thus, with the expectation value for event $I_{R,M}$ (note that $E[I_{R,M}] = Pr[I_{R,M} = 1]$) we can lowerbound $\rho \geq E[I_{R,M}]/\binom{l}{k}$.

[8]Note that this lemma is essentially the only place where we need that the combiner's output length is significantly less than that of the concatenation combiner, $n = (l-k+1)(v - \log(q_C) - t$ (see equation(4)).

Setting $\tilde{e} := 1 - E[I_{R,M}]$ and applying the reverse Markov inequality, we have that:

$$\begin{aligned}\Pr[E[I_{R,M}] < 1 - \gamma 2^{-t+l}] &= Pr[\tilde{e} \geq \gamma 2^{-t+l}] \\ &\leq \frac{1 - E[I_{R,M}]}{1 - 1 + \gamma 2^{-t+l}} \\ &\leq \frac{1 - 1 + 2^{-t+l}}{1 - 1 + \gamma 2^{-t+l}} = \frac{1}{\gamma}\end{aligned}$$

Setting $\gamma := 4$ yields: $Pr[E[I_{R,M}] < 1 - 2^{-t+l+2}] \leq \frac{1}{4}$. Using the estimate $\rho \geq E[I_{R,M}]/\binom{l}{k}$ we thus have:

$$\begin{aligned}Pr\left[\rho < (1 - 2^{-t+l+2})/\binom{l}{k}\right] &\leq Pr\left[E[I_{R,M}]/\binom{l}{k} < (1 - 2^{-t+l+2})/\binom{l}{k}\right] \\ &\leq \frac{1}{4}\end{aligned} \quad (7)$$

We proved that with probability of at least 0.75, our breaking oracle ρ-breaks a (k, l)-combiner with $\rho = (1 - 2^{-t+l+2})/\binom{l}{k}$.

3.5 $\mathcal{B}$ does not help P

We will now present the lemma that allows us to prove Proposition 1. Informally this lemma states that if the range of the combiner is as in the proposition and the reduction $P^{\mathcal{B},H_1,...,H_l}$ finds second pre-images with some probability then we can use this to compress the combined function table of hash functions $H_1, ..., H_l$ below $l2^w v$ bits. This can then be used to prove Proposition 1 as the H_i $(i = 1, ..., l)$ were chosen uniformly at random and therefore, by Proposition 2, cannot be compressed below $l2^w v$ bits.

Lemma 2 *Let (C,P) be as in Proposition 1 with*

$$v > \log q_P^H + \log l + 2 \quad (8)$$

Also let for any set of target messages $X_1, ..., X_l \in \{0, 1\}^w$

$$Pr[\boldsymbol{spr}_{l-k+1,X_1,...,X_l}(P^{\mathcal{B}^{X_1,...,X_l},H_1,...,H_l}(X_1, ..., X_l))] \geq 0.5 \quad (9)$$

Then $H_1, ..., H_l$ can be compressed below $l2^w v$ bits.

Let us show how we can now prove our main proposition, Proposition 1:

Proof (of Proposition 1). Let Δ denote the event that $\mathcal{B}$ ρ-breaks C^H with $\rho \geq (1 - 2^{-t+3})/\binom{l}{k}$ (note that $Pr[\Delta] \geq 0.75$). With equation (7) and the bound on P from the definition of robust second-pre-image combiners (see Proposition 1 and the discussion following Definition 6), we can derive the following bound for any collection X^l of input messages:

$$\begin{aligned}&\forall X^l \in \overbrace{\{0,1\}^w \times ... \times \{0,1\}^w}^{l \text{ times}} : \\ &Pr[\mathbf{spr}_{l-k+1,X^l}(P^{\mathcal{B},H_1,...,H_l}(X^l))] = Pr[\Delta] \cdot Pr[\mathbf{spr}_{l-k+1,X^l}(P^{\mathcal{B},H_1,...,H_l}(X^l))|\Delta] \\ &\qquad \geq \frac{3}{4} \cdot \frac{2}{3} = 0.5\end{aligned}$$

Assume the $H_1, ..., H_l$ are uniformly random, then by Lemma 2 the combined function table for $H_1, ..., H_l$ can be compressed below $l2^w v$ bits. This contradicts Proposition 2 and hence equation (23) must be wrong. Thus:

$$v \leq \log q_P^H + 2 + \log l$$

which concludes the proof. □

Compressing $H_1, ..., H_l$ with Second Pre-Images by $P^{\mathcal{B},H_1,...,H_l}$

We will here only present a proof sketch for Lemma 2 (the full proof is given in Appendix C.2).

Let us first take a closer look at the second pre-images sampled by the breaking oracle. By definition $\mathcal{B}(R, M)$ will only output a (pseudo) second pre-image if it is a safe one. Let's assume that $P^{\mathcal{B},H_1,...,H_l}$ was given the target messages $X_1, ..., X_l$ and that the breaking oracle $\mathcal{B}^{X_1,...,X_l}$ was initialized with exactly these. For now think of P as being a fair player who honestly is telling $\mathcal{B}$ its target values (we will later see why this can be assumed).

The breaking oracle $\mathcal{B}^{X_1,...,X_l}$ now only outputs safe second pre-images that are safe for k of the l hash functions. For the reduction $P^{\mathcal{B}^{X^l},H^l}$ (for notational simplicity we write $P^{\mathcal{B}^{X^l},H^l}$ instead of $P^{\mathcal{B}^{X_1,...,X_l},H_1,...,H_l}$) to be successful it has for its input $(X_1, ..., X_l)$ to find second pre-images for $l - k + 1$ of the l hash functions where X_i is the first part of the collision for hash function H_i $(i = 1, ..., l)$. This, however, directly implies that if $P^{\mathcal{B}^{X^l},H^l}(X_1, ..., X_l)$ succeeds and outputs $l-k+1$ second-pre-images then at least for one of the $l - k + 1$ hash functions the second pre-image was not generated by the breaking oracle directly.

Let s denote the index of the hash function H_s $(s \in \{1, ..., l\})$ for which the second pre-image output by $P^{\mathcal{B}^{X^l},H^l}$ was not trivially created via the breaking oracle $\mathcal{B}^{X^l}$. Let $X^s_{\mathbf{spr}}$ be the second pre-image for hash function H_s for message X_s, i.e., $H_s(X_s) = H_s(X_{\mathbf{spr}})$ and $X_s \neq X_{\mathbf{spr}}$. What we now know is that the oracle query $H_s(X^s_{\mathbf{spr}})$ which resulted in this second pre-image is made by P directly and not by the breaking oracle $\mathcal{B}^{X^l}$. Hence, $X^s_{\mathbf{spr}}$ is not present in any of the queries to H_s resulting from calls to the the breaking oracle. Let us by

$$\mathrm{qry}^{H_i}_{\mathcal{B}^{X^l}(R,M)} = \mathrm{qry}^{H_i}(C^{H^l}(R, M)) \cup \mathrm{qry}^{H_i}(C^{H^l}(R, M^{\mathcal{B}}_{spr})) \tag{10}$$

denote all the queries to hash function H_i that occur in the evaluation of a $\mathcal{B}^{X^l}$ query on input (R, M), i.e. the queries resulting from evaluating the combiner $C^{H^l}(\cdot)$ with input (R, M) (the request to the breaking oracle) and $(R, M^{\mathcal{B}}_{spr})$ (the oracle's answer). Then we can rephrase the above statement about the second pre-image $X^s_{\mathbf{spr}}$ found by $P^{\mathcal{B}^{X^l},H^l}$ for hash function H_s:

$$X^s_{\mathbf{spr}} \notin \mathrm{qry}^{H_s}_{\mathcal{B}^{X^l}(M_B)} \quad \forall\ M_B \in \mathrm{qry}^{\mathcal{B}}(P^{\mathcal{B}^{X^l},H^l}(M)) \tag{11}$$

This could again be rephrased as: $P^{\mathcal{B}^{X^l},H^l}(M_1, ..., M_l)$ cannot find trivial second-pre-images for all $l-k+1$ hash functions; $P^{\mathcal{B}^{X^l},H^l}$ cannot simply let $\mathcal{B}^{X^l}$ do all the work.

Now how does this help in compressing the function table of $H_1, \ldots, H_l$? The idea is to design a compression algorithm **com** together with a corresponding decompression algorithm **dec**. Both algorithms **com** and **dec** share a common, combined target message $X_\tau = (X_1, ..., X_l)$ with $(X_i \in \{0, 1\}^w$ for $i = 1, ..., l)$. The algorithms make use of the security reduction P and provide P with the input X_τ and the oracles that P expects: a breaking oracle $\mathcal{B}$ and hash functions $H_1, \ldots, H_l$.

If P succeeds in generating a second-preimage $X_{\mathbf{spr}}$, the compression algorithm **com** reduces the combined function table of $H_1, \ldots, H_l$ by X_τ. It further removes all the calls from P and $\mathcal{B}$ to any of the hash functions H_i (note that as **com** provided P and $\mathcal{B}$ with the hash functions it can easily track all those queries). The compression algorithm **com** then prepends the reduced function table with the hash values that P and $\mathcal{B}$ request during the execution of P. The new function table now contains one hash value less (the one from the second pre-image for one of the target messages in X_τ) than the original function table and we have thus compressed the function table by v bits. For decompression we can simply again simulate P (with the same random coins) and if we can identify the query with the second pre-image we can reconstruct the function table.

The difficulty will be to identify the call $X^s_{\mathbf{spr}}$ from P to hash function H^s which yields the second pre-image as it may not be P's last call. As we have said, **com** and **dec** can track which query goes to which hash function. Thus if we store the index of the call to H_s which yielded the second pre-image relative to the number of calls made by P (we know that this is bound by q_P, compare Definition 7).

Finally we have to make sure that the breaking oracle $\mathcal{B}$ given to P is initialized with the correct messages $(X_1, \ldots, X_l) = X_\tau$. Again, as **com** and **dec** provide the breaking oracle to P they can simply initialize the breaking oracle before "handing it over". $\Diamond$

With this we have completed the proof of Theorem 2 for the case of second pre-images. See Appendix D for a description of the changes needed for target collision resistance and pre-image resistance.

4 Conclusion and Outlook

We have given a strong indication that combiners with short output robust for second pre-image resistance, target collision resistance or pre-image resistance do not exist. By this, we have extended Pietrzak's Theorem where he gave the result for the case of plain collision resistance [17]. Note that our work, as well as Pietrzak's only applies to fully black-box reductions in the terminology of Reingold et al. [18]. One possibility of bypassing such an impossibility result, is to consider white-box access to the hash functions (resp. the breaking oracle). A different approach would be to consider combiners only for a specific class of hash functions (e.g., efficiently implementable functions) instead of combiners that need to be robust for any choice of functions. Ideally, this class of hash functions should contain functions used in practice, such as the SHA familiy.

Acknowledgments

I thank the anonymous reviewers for valuable comments. I would like to especially thank Christina Brzuska and Marc Fischlin for insightful discussions. This work was supported by CASED (`www.cased.de`).

References

[1] Aoki, K., Sasaki, Y.: Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1. In: Halevi, S. (ed.) Advances in Cryptology – CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 70–89. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 16–20, 2009) (Cited on page 2.)

[2] Aumasson, J.P., Henzen, L., Meier, W., Phan, R.C.W.: SHA-3 proposal BLAKE. Submission to NIST (Round 3) (2010), `http://131002.net/blake/blake.pdf` (Cited on page 2.)

[3] Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: The keccak SHA-3 submission. Submission to NIST (Round 3) (2011), `http://keccak.noekeon.org/Keccak-submission-3.pdf` (Cited on page 2.)

[4] Boneh, D., Boyen, X.: On the impossibility of efficiently combining collision resistant hash functions. In: Dwork, C. (ed.) Advances in Cryptology – CRYPTO 2006. Lecture Notes in Computer Science, vol. 4117, pp. 570–583. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 20–24, 2006) (Cited on pages 1, 2, and 3.)

[5] Canetti, R., Rivest, R.L., Sudan, M., Trevisan, L., Vadhan, S.P., Wee, H.: Amplifying collision resistance: A complexity-theoretic treatment. In: Menezes, A. (ed.) Advances in Cryptology – CRYPTO 2007. Lecture Notes in Computer Science, vol. 4622, pp. 264–283. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 19–23, 2007) (Cited on pages 1, 2, and 3.)

[6] Cannière, C.D., Rechberger, C.: Preimages for reduced SHA-0 and SHA-1. In: Wagner, D. (ed.) Advances in Cryptology – CRYPTO 2008. Lecture Notes in Computer Science, vol. 5157, pp. 179–202. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 17–21, 2008) (Cited on page 2.)

[7] De Cannière, C., Rechberger, C.: Finding SHA-1 characteristics: General results and applications. In: Lai, X., Chen, K. (eds.) Advances in Cryptology – ASIACRYPT 2006. Lecture Notes in Computer Science, vol. 4284, pp. 1–20. Springer, Berlin, Germany, Shanghai, China (Dec 3–7, 2006) (Cited on page 2.)

[8] Dierks, T., Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (Proposed Standard) (Aug 2008), `http://www.ietf.org/rfc/rfc5246.txt`, updated by RFCs 5746, 5878, 6176 (Cited on page 1.)

[9] Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., Walker, J.: The skein hash function family. Submission to NIST (Round 3) (2010), `http://www.skein-hash.info/sites/default/files/skein1.3.pdf` (Cited on page 2.)

[10] Freier, A., Karlton, P., Kocher, P.: The Secure Sockets Layer (SSL) Protocol Version 3.0. RFC 6101 (Historic) (Aug 2011), `http://www.ietf.org/rfc/rfc6101.txt` (Cited on page 1.)

[11] Gauravaram, P., Knudsen, L.R., Matusiewicz, K., Mendel, F., Rechberger, C., Schlffer, M., Thomsen, S.S.: Grstl – a SHA-3 candidate. Submission to NIST (Round 3) (2011), `http://www.groestl.info/Groestl.pdf` (Cited on page 2.)

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

[13] NIST: FIPS PUB 180-3 Secure Hash Standard (SHS), `http://csrc.nist.gov/publications/fips/fips180-3/fips180-3_final.pdf` (Cited on page 2.)

[14] NIST: NIST SHA-3 Competition, `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html` (Cited on page 2.)

[15] Pietrzak, K.: Compression from collisions, or why CRHF combiners have a long output (full version) (Cited on pages 3 and 9.)

[16] Pietrzak, K.: Non-trivial black-box combiners for collision-resistant hash-functions don't exist. In: Naor, M. (ed.) Advances in Cryptology – EUROCRYPT 2007. Lecture Notes in Computer Science, vol. 4515, pp. 23–33. Springer, Berlin, Germany, Barcelona, Spain (May 20–24, 2007) (Cited on pages 1 and 2.)

[17] Pietrzak, K.: Compression from collisions, or why CRHF combiners have a long output. In: Wagner, D. (ed.) Advances in Cryptology – CRYPTO 2008. Lecture Notes in Computer Science, vol. 5157, pp. 413–432. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 17–21, 2008) (Cited on pages 1, 2, 3, 4, 6, 7, 9, 15, 18, and 21.)

[18] Reingold, O., Trevisan, L., Vadhan, S.P.: Notions of reducibility between cryptographic primitives. In: Naor, M. (ed.) TCC 2004: 1st Theory of Cryptography Conference. Lecture Notes in Computer Science, vol. 2951, pp. 1–20. Springer, Berlin, Germany, Cambridge, MA, USA (Feb 19–21, 2004) (Cited on page 15.)

[19] Rjaško, M.: On existence of robust combiners for cryptographic hash functions. In: ITAT. pp. 71–76 (2009) (Cited on pages 4 and 18.)

[20] Rogaway, P., Shrimpton, T.: Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance. In: Roy, B.K., Meier, W. (eds.) Fast Software Encryption – FSE 2004. Lecture Notes in Computer Science, vol. 3017, pp. 371–388. Springer, Berlin, Germany, New Delhi, India (Feb 5–7, 2004) (Cited on pages 1 and 5.)

[21] Sasaki, Y., Aoki, K.: Finding preimages in full MD5 faster than exhaustive search. In: Joux, A. (ed.) Advances in Cryptology – EUROCRYPT 2009. Lecture Notes in Computer Science, vol. 5479, pp. 134–152. Springer, Berlin, Germany, Cologne, Germany (Apr 26–30, 2009) (Cited on page 2.)

[22] Shannon, C., Petigara, N., Seshasai, S.: A mathematical theory of. Communication, Bell System Technical Journal 27, 379–423 (1948) (Cited on pages 3 and 10.)

[23] Stevens, M., Sotirov, A., Appelbaum, J., Lenstra, A.K., Molnar, D., Osvik, D.A., de Weger, B.: Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In: Halevi, S. (ed.) Advances in Cryptology – CRYPTO 2009. Lecture Notes in Computer Science, vol. 5677, pp. 55–69. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 16–20, 2009) (Cited on page 2.)

[24] Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full SHA-1. In: Shoup, V. (ed.) Advances in Cryptology – CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 17–36. Springer, Berlin, Germany, Santa Barbara, CA, USA (Aug 14–18, 2005) (Cited on page 2.)

[25] Wang, X., Yu, H.: How to break MD5 and other hash functions. In: Cramer, R. (ed.) Advances in Cryptology – EUROCRYPT 2005. Lecture Notes in Computer Science, vol. 3494, pp. 19–35. Springer, Berlin, Germany, Aarhus, Denmark (May 22–26, 2005) (Cited on page 2.)

[26] Wu, H.: The hash function JH. Submission to NIST (round 3) (2011), `http://www3.ntu.edu.sg/home/wuhj/research/jh/jh_round3.pdf` (Cited on page 2.)

A Proving a Specialized Case

In this section we prove that a certain class of combiners have to have long output if they are robust for second pre-image resistance, target collision resistance or pre-image resistance. This result (which has also been studied in [19]) is a special case of Theorem 2 but allows for a much simpler proof, which will allow us to introduce some of the notation presented in Section 2. What is different from the general case is that we want the security reduction P to generate a second pre-image (target collision or pre-image, resp.) given only a single second pre-image (target collision or pre-image, resp.) for the combiner. We model this, by allowing P to make only a single call to the breaking oracle. We will present this for the $(1, 2)$-deterministic-combiner case which is the form of combiner that is usually used in practice. As the proofs are almost identical for second pre-image resistance and target collision resistance we will present the sections that differ side by side (second pre-image resistance on the left and target collision resistance on the right). We omit the prove for pre-image resistance as it is (almost) identical to the one for second pre-image resistance: simply exchange target messages for target images. We closely follow the reasoning for the collision resistance case (see Proposition 1 in [17]).

Proposition 3 *For some $n, m, v, w \in \mathbb{N}$ with $m > n$, assume that (C, P) with $C : \{0,1\}^m \to \{0,1\}^n$ is a robust $(1, 2)$-black-box-combiner for SPR, TCR or OW, for hash functions of the form $\{0,1\}^w \to \{0,1\}^v$, with the additional constraint, that P* is querying the breaking oracle only once.

Let ϵ denote the success probability of P, i.e., for target messages $X_1, X_2 \leftarrow \{0,1\}^w$ and a breaking oracle $\mathcal{B}$ which on input $M \in \{0,1\}^m$ outputs a second pre-image M' such that (M, M') collide under C^{H_1,H_2}:

$$Pr[\boldsymbol{spr}^{H_1,H_2}_{X_1,X_2}(P^{\mathcal{B},H_1,H_2}(X_1, X_2))] \geq \epsilon$$

Let ϵ denote the success probability of P, i.e., for a breaking oracle $\mathcal{B}$ that breaks the two round TCR protocol for $C^{H_1,\ldots,H_l}$ (i.e., $\mathcal{B}$ on input some randomness outputs a target message and after getting access to $C^{H_1,\ldots,H_l}$ outputs a second message which is different from the target message but shares the same hash value):

$$Pr[\boldsymbol{tcr}^{H_1,H_2}(P^{\mathcal{B},H_1,H_2})] \geq \epsilon$$

Then the output length n of the combiner C satisfies:

$$n \geq 2v - 2 \log q_P + \log \epsilon - 1 \tag{12}$$

The idea in the upcoming proof is to choose two hash functions randomly from the space of all functions and then to derive two bounds on the probability that an adversary can find second pre-images (target collision or pre-images, resp.): an upper bound using the union bound and a lower bound using the reduction P guaranteed by robust combiners. By then combining the two bounds we can derive the bound on the combiner's output length given in equation (12).

Proof. Let $H_1, H_2 : \{0,1\}^w \to \{0,1\}^v$ be two functions chosen uniformly at random and independently from the space of all functions of the form $\{0,1\}^w \to \{0,1\}^v$. Let $\mathcal{A}_1$ be an oPPTM that makes at most q_A calls to each of its oracles. As the hash functions are random functions we can examine the random experiment, that $\mathcal{A}_1$ given a pre-image $X \leftarrow \{0,1\}^w$ finds one (or possibly more) second pre-image(s) within its q_A oracle queries to H_1 for hash function H_1. This experiment has a binomial distribution $B(q_A, 2^{-v})$ and we can hence upper-bound $\mathcal{A}_1$'s success probability of finding a second-pre-image for H_1 with:

$$Pr[\mathbf{spr}^{H_1(X)}(\mathcal{A}_1^{H_1,H_2}(X))] = 1 - B(0|q_A, 2^{-v}) = 1 - (1 - 2^{-v})^{q_A} \leq \frac{q_A}{2^v} \tag{13}$$

As H_1 and H_2 were chosen independently, the probability that adversary $\mathcal{A}_1$ finds a second pre-image for target messages X_1 and X_2 for both its hash functions is simply the square of what we had in equation (13):

$$Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(\mathcal{A}_1^{H_1,H_2}(X_1,X_2))] \leq (\frac{q_A}{2^v})^2 \tag{14}$$

Note that the probability is the same if adversary $\mathcal{A}_1$ was allowed to choose messages X_1, X_2 before getting access to the hash functions (TCR case).

Let us now consider a robust combiner (C, P) as given in the proposition. We will compute a lower bound for the success probability of a second adversary $\mathcal{A}_2$. Adversary $\mathcal{A}_2$ simply calls reduction P to do the work and tries to simulate the breaking oracle $\mathcal{B}$ that is expected by reduction:

Adversary $\mathcal{A}_2$ simply passes its input messages X_1 and X_2 to algorithm $P^{\mathcal{B},H_1,H_2}$. It relays P's queries to hash functions H_1 and H_2 to its oracles and answers P's single query M_b to the breaking oracle $\mathcal{B}$ with message $M'_b \leftarrow \{0,1\}^*$ sampled uniformly at random.

Adversary $\mathcal{A}_2$ calls P. Before P can call the hash function oracles it has to commit to messages X_1 and X_2. On receiving the messages $\mathcal{A}_2$ commits to them as well. Now P (and $\mathcal{A}_2$) may query the hash functions and $\mathcal{A}_2$ simply relays P's queries to its respective oracle. At some point (this may come before or after P committed to its messages) P makes its query to $\mathcal{B}$ and expects a commitment message. $\mathcal{A}_2$ answers that query with a random message $M_b \leftarrow \{0,1\}^m$. Now P expects $\mathcal{B}$ to come up with a target collision for M_b and $\mathcal{A}_2$ fulfills this by generating another random message M'_b.

Reduction P will output second pre-images (or target collisions, respectively) for H_1 and H_2 with probability ϵ conditioned on the event that M'_b is a correct second pre-image (target collision) for M_b (i.e., $C^{H_1,H_2}(M_b) = C^{H_1,H_2}(M'_b)$ and $M_b \neq M'_b$). We can thus give a lower bound on $\mathcal{A}_2$'s success probability. To shorten the notation we define the event Φ to hold if $\mathcal{A}_2$ answers the oracle query to $\mathcal{B}$ "correctly", i.e:

$$\Phi \iff C^{H_1,H_2}(M_b) = C^{H_1,H_2}(M'_b) \wedge M_b \neq M'_b$$

As $Pr[M_b = M'_b] = 2^{-m}$ and $Pr[C^{H_1,H_2}(M_b) = C^{H_1,H_2}(M'_b)] \geq 2^{-n}$ we can lowerbound the probability of event Φ by the probability of having a random collision minus the probability of sampling the same message (using $m > n$ in the last step):[9]

$$Pr[\Phi] \geq 2^{-n} - 2^{-m} \geq 2^{-n-1}$$

Hence, we can derive a lower bound on $\mathcal{A}_2$'s success probability as it is equal to that of $P^{\mathcal{B}^*,H_1,H_2}$ where B^* represents the simulated breaking oracle.[10]

$$\begin{aligned} Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(\mathcal{A}^{H_1,H_2}(X_1,X_2))] =& Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(P^{\mathcal{B}^*,H_1,H_2}(X_1,X_2))] \\ =& Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(P^{\mathcal{B}^*,H_1,H_2}(X_1,X_2))|\Phi] \cdot Pr[\Phi] + \\ & Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(P^{\mathcal{B}^*,H_1,H_2}(X_1,X_2))|\neg\Phi] \cdot Pr[\neg\Phi] \end{aligned}$$

Let $\mathcal{B}$ be a breaking oracle, succeeding in finding second pre-images with probability 1, then by dropping the second summand, and observing that

$$Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(P^{\mathcal{B}^*,H_1,H_2}(X_1,X_2))|\Phi] = Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(P^{\mathcal{B},H_1,H_2}(X_1,X_2))]$$

[9] Note that for pre-image resistance the restriction of not sampling the "same" pre-image is not required, as there is no original pre-image. Hence we have that $Pr[\Phi] \geq 2{-}n$.

[10] We will prove the following steps for SPR. For TCR simply exchange the predicate.

(notice the changing breaking oracle: $\mathcal{B}$ instead of $\mathcal{B}^*$), we arrive at the following lower bound for $\mathcal{A}_2$'s success probability:

$$\begin{aligned} Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(\mathcal{A}^{H_1,H_2}(X_1,X_2))] &\geq Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(P^{\mathcal{B},H_1,H_2}(X_1,X_2))] \cdot Pr[\Phi] \\ &\geq \epsilon \cdot (2^{-n} - 2^{-m}) \geq \epsilon \cdot 2^{-n-1} \end{aligned} \tag{15}$$

As this bound is valid for any choice of functions H_1 and H_2 it is also valid for uniformly random functions. We can thus combine the bounds computed for $\mathcal{A}_1$ in equation (14) and for $\mathcal{A}_2$ in equation (15) to arrive at an inequality:

$$\epsilon \cdot 2^{-n-1} \leq Pr[\mathbf{spr}^{H_1,H_2}_{X_1,X_2}(\mathcal{A}^{H_1,H_2}(X_1,X_2))] \leq (\frac{q_A}{2^v})^2$$

Solving it for n yields (note that $q_A = q_P$):

$$n \geq 2v - 2\log(q_A) + \log(\epsilon) - 1$$

$\square$

Note that, for the practically relevant case where reduction P is efficient and ϵ not negligible, there exists an upper bound on $\log q_P$ and a lower bound on $\log \epsilon$ (with respect to v). Thus, the derived bound can be rewritten as $n \geq 2v - \mathcal{O}(\log v)$ which matches the bound specified in equation (3) for Theorem 2 for the case $l = 2$ and $k = 1$.

B Additional Definitions for Collision Resistance, Target Collision Resistance and Pre-Image Resistance

In this section we give missing definitions from Sections 2 and 3 for the cases collision resistance, target collision resistance and pre-image resistance.

B.1 Predicate Noting Found Collisions, Target Collisions or Pre-Images

Here we present the corresponding definitions to Definition 5 for collision resistance, target collision resistance, and pre-image resistance. We define predicates denoting that in the course of the execution of a certain experiment, a collision, a target collision or a pre-image was found.

Definition 10 *The predicate* $\boldsymbol{col}^{H_i}(\mathcal{A}^{H_1,...,H_l})$ *(with* $1 \leq i \leq l$*) is defined for the random experiment* $\mathcal{A}^{H_1,...,H_l}$ *and holds if* $\mathcal{A}$ *finds a collision for* H_i*; that is, in the course of the computation of* $\mathcal{A}^{H_1,...,H_l}$ *two oracle calls to* H_i *are made with messages* (M, M') *for which* $H_i(M) = H_i(M')$ *and* $M \neq M'$*. Formally:*

$$\boldsymbol{col}^{H_i}(\mathcal{A}^{H_1,...,H_l}) \iff \exists\, M, M' \in qry^{H_i}(\mathcal{A}^{H_1,...,H_l}) : M \neq M' \wedge H_i(M) = H_i(M')$$

For $\mathcal{H} \subseteq \{H_1, ..., H_l\}$ *we define the predicate* $\boldsymbol{col}^{\mathcal{H}}(\mathcal{A}^{H_1,...,H_l})$ *to hold if collisions are found for all hash functions in* $\mathcal{H}$*:*

$$\boldsymbol{col}^{\mathcal{H}}(\mathcal{A}^{H_1,...,H_l}) \iff \forall\, H \in \mathcal{H} : \boldsymbol{col}^{H}(\mathcal{A}^{H_1,...,H_l})$$

and for $1 \leq n \leq l$ *we define the predicate* $\boldsymbol{col}_n(\mathcal{A}^{H_1,...,H_l})$ *to hold if a collision is found for* n *of the* l *hash oracles:*

$$\boldsymbol{col}_n(\mathcal{A}^{H_1,...,H_l}) \iff \exists\, \mathcal{H} \subseteq \{H_1, ..., H_l\}, |\mathcal{H}| \geq n : \boldsymbol{col}^{\mathcal{H}}(\mathcal{A}^{H_1,...,H_l})$$

For target collision resistance we define the predicate family $\mathbf{tcr}(\mathcal{A}^{H_1,\ldots,H_l})$ analogously to the one for second pre-image resistance in Definition 5. They hold if algorithm $\mathcal{A}$ breaks the two round TCR-protocol as defined in Definition 3 for the specified hash functions.

Definition 11 *The predicate $\boldsymbol{tcr}^{H_i}(\mathcal{A}^{H_1,\ldots,H_l})$ (with $1 \leq i \leq l$) is defined for the random experiment $\mathcal{A}^{H_1,\ldots,H_l}$ and holds if $\mathcal{A}$ finds a target collision for H_i; that is, in the course of the computation of $\mathcal{A}^{H_1,\ldots,H_l}$, adversary $\mathcal{A}$ commits to target message M before getting access to the hash function's key and then performs an oracle call to H_i with messages M' for which $H_i(M) = H_i(M')$ and $M \neq M'$. Formally:*

$$\begin{aligned}\boldsymbol{tcr}^{H_i}(\mathcal{A}^{H_1,\ldots,H_l}) \iff& \\ &\exists\, M, M' : \mathcal{A} \text{ commits to } M \wedge M' \in qry^{H_i}(\mathcal{A}^{H_1,\ldots,H_l}) \wedge \\ &\qquad M \neq M' \wedge H_i(M) = H_i(M')\end{aligned}$$

For $\mathcal{H} \subseteq \{H_1, \ldots, H_l\}$ we define the predicate $\boldsymbol{tcr}^{\mathcal{H}}_{X_1,\ldots,X_l}(\mathcal{A}^{H_1,\ldots,H_l})$ to hold if collisions are found for all hash functions in $\mathcal{H}$:

$$\boldsymbol{tcr}^{\mathcal{H}}(\mathcal{A}^{H_1,\ldots,H_l}) \iff \forall\, H \in \mathcal{H} : \boldsymbol{tcr}^{H}(\mathcal{A}^{H_1,\ldots,H_l})$$

and for $1 \leq n \leq l$ we define the predicate $\boldsymbol{tcr}_n(\mathcal{A}^{H_1,\ldots,H_l})$ to hold if a collision is found for n of the l hash oracles:

$$\boldsymbol{tcr}_n(\mathcal{A}^{H_1,\ldots,H_l}) \iff \exists\, \mathcal{H} \subseteq \{H_1, \ldots, H_l\}, |\mathcal{H}| \geq n : \boldsymbol{tcr}^{\mathcal{H}}(\mathcal{A}^{H_1,\ldots,H_l})$$

Finally, the definition for pre-image resistance, which is also closely related to the definition for the second pre-image resistant case (Definition 5). Note, that we speak of target images I instead of target messages X.

Definition 12 *The predicate $\boldsymbol{ow}^{H_i(I_i)}(\mathcal{A}^{H_1,\ldots,H_l})$ is defined for the random experiment $\mathcal{A}^{H_1,\ldots,H_l}$ and target image I_i and holds if $\mathcal{A}$ finds any pre-image for I_i for function H_i; that is, in the course of the computation of $\mathcal{A}^{H_1,\ldots,H_l}$ an oracle call to H_i is made with messages $(X_{\boldsymbol{ow}})$ for which $H_i(X_{\boldsymbol{ow}}) = I_i$. Formally:*

$$\boldsymbol{ow}^{H_i(I_i)}(\mathcal{A}^{H_1,\ldots,H_l}) \iff \exists\, X_{\boldsymbol{ow}} \in qry^{H_i}(\mathcal{A}^{H_1,\ldots,H_l}) : H_i(X_{\boldsymbol{ow}}) = I_i$$

For subset $\mathcal{H} \subseteq \{H_1, \ldots, H_l\}$ and target images $I_1, \ldots, I_l$ we define the predicate $\boldsymbol{ow}^{\mathcal{H}}_{I_1,\ldots,I_l}(\mathcal{A}^{H_1,\ldots,H_l})$ to hold if pre-images are found for all hash functions in $\mathcal{H}$ (for corresponding target images I_i):

$$\boldsymbol{ow}^{\mathcal{H}}_{I_1,\ldots,I_l}(\mathcal{A}^{H_1,\ldots,H_l}) \iff \forall\, H_i \in \mathcal{H} : \boldsymbol{ow}^{H_i(I_i)}(\mathcal{A}^{H_1,\ldots,H_l})$$

For $1 \leq n \leq l$ we define the predicate $\boldsymbol{ow}_{n,I_1,\ldots,I_l}(\mathcal{A}^{H_1,\ldots,H_l})$ to hold if a pre-image is found for n of the l hash oracles:

$$\boldsymbol{ow}_{n,I_1,\ldots,I_l}(\mathcal{A}^{H_1,\ldots,H_l}) \iff \exists\, \mathcal{H} \subseteq \{H_1, \ldots, H_l\}, |\mathcal{H}| \geq n : \boldsymbol{ow}^{\mathcal{H}}_{I_1,\ldots,I_l}(\mathcal{A}^{H_1,\ldots,H_l})$$

B.2 Randomized Combiner for Collision Resistance, Target Collision Resistance, and Pre-Image Resistance

Here we give the definition of robust combiners for collision resistance (also see [17]), target collision resistance, and pre-image resistance (cf. Definition 6).

Definition 13 *Let the situation be as in Definition 6.*

Oracle $\mathcal{B}^{\boldsymbol{col}}$ ρ-breaks $C^{H_1,\dots,H_l}$ (for CR) if it outputs a collision for $C^{H_1,\dots,H_l}(R,\cdot)$ for a ρ-fraction of the possible choices of randomness $R \in \mathcal{R}$ and $\perp$ otherwise.

Oracle $\mathcal{B}^{\boldsymbol{tcr}}$ ρ-breaks $C^{H_1,\dots,H_l}$ (for TCR) if it successfully completes the TCR protocol for $C^{H_1,\dots,H_l}(R,\cdot)$ for a ρ-fraction of the possible choices of randomness $R \in \mathcal{R}$. If it is not able to find a target collision for the previously specified message it outputs $\perp$ in the second step.

Oracle $\mathcal{B}^{\boldsymbol{ow}}$ ρ-breaks $C^{H_1,\dots,H_l}$ (for OW) if it outputs a pre-image to input image $C^{H_1,\dots,H_l}(R,M)$ (with $M \leftarrow \{0,1\}^m$) for a ρ-fraction of the randomness $R \in \mathcal{R}$. If no pre-image is found it outputs $\perp$.

A randomized (k,l) combiner for hash functions $H_1,\dots,H_l$ is called

ρ-robust for CR *if for some non-negligible $0 < \epsilon \leq 1$ and any choice of functions $H_1,...,H_l$ and for any breaking oracle $\mathcal{B}^{\boldsymbol{col}}$ that ρ-breaks $C^{H_1,\dots,H_l}$, the random experiment $P^{\mathcal{B}^{\boldsymbol{col}},H_1,\dots,H_l}$ finds collisions for $l-k+1$ input hash functions with probability:*

$$Pr[\boldsymbol{col}_{l-k+1}(P^{\mathcal{B}^{\boldsymbol{col}},H_1,\dots,H_l})] \geq \epsilon \tag{16}$$

ρ-robust for TCR *if for some non-negligible $0 < \epsilon \leq 1$ and any choice of functions $H_1,...,H_l$ and for any breaking oracle $\mathcal{B}^{\boldsymbol{tcr}}$ that ρ-breaks $C^{H_1,\dots,H_l}$, the random experiment $P^{\mathcal{B}^{\boldsymbol{tcr}},H_1,\dots,H_l}$ breaks the TCR-protocol for $l-k+1$ input hash functions with probability:*

$$Pr[\boldsymbol{tcr}_{l-k+1}(P^{\mathcal{B}^{\boldsymbol{tcr}},H_1,\dots,H_l})] \geq \epsilon \tag{17}$$

ρ-robust for OW *if for some non-negligible $0 < \epsilon \leq 1$ and any choice of functions $H_1,...,H_l$ and for any breaking oracle $\mathcal{B}^{\boldsymbol{ow}}$ that ρ-breaks $C^{H_1,\dots,H_l}$, the random experiment $P^{\mathcal{B}^{\boldsymbol{ow}},H_1,\dots,H_l}$ on input $(C(M_1),...,C(M_l))$ with $M_i \leftarrow \{0,1\}^m$ (for $i = 1,...,l$) finds pre-images for $l-k+1$ input hash functions with probability:*

$$Pr[\boldsymbol{ow}_{l-k+1}(P^{\mathcal{B}^{\boldsymbol{ow}},H_1,\dots,H_l})] \geq \epsilon \tag{18}$$

The combiner is called robust *for CR (TCR, OW) if it is efficient and ρ-robust for every non negligible $\rho(v)$.*

Note that for pre-image resistance we can define the success probability over the choice of randomness (in contrast of the definition for second pre-image resistance) as we sample the images not directly from the codomain but from the domain and then use the hash function to map them to an image (cf. Definition 4).

B.3 Safe Target Collisions and Pre-Images

Here we give the definition of safe target collisions and pre-images corresponding to Definition 8 (safe second pre-images) on page 8.

The notion for safe target collisions is exactly the same as for second pre-images. As the only difference between the two schemes is who chooses the first part of the collision (cf. Definition 3) we call a message $M_{\mathbf{tcr}}$ safe with respect to message M (this time chosen by the adversary) and oracles C^H and H if it does not yield a trivial target collision for some target message $X \in \{0,1\}^w$ and oracle H.

Definition 14 (Safe Target Collisions) *The predicate $\boldsymbol{safeTcr}^{C^H}_{H,X}(M, M_{\boldsymbol{tcr}})$ holds if and only if $(M, M_{\boldsymbol{tcr}})$ is a safe second pre-image for hash function H and target message X.*

A safe pre-image denotes a pre-image for some image on the combiner which does not yield trivial pre-images on the (input) hash functions.

Definition 15 (Safe Pre-Image) *Let $H : \{0,1\}^w \to \{0,1\}^v$ be a hash function and $C : \{0,1\}^m \to \{0,1\}^n$ some oPPTM. Let $I_{H\text{-}target} \in \{0,1\}^v \cap H(\{0,1\}^w)$ be a target image for hash function H, $I_{C\text{-}target} \in \{0,1\}^n \cap C(\{0,1\}^m)$ a target image for combiner C, and $M_{\boldsymbol{ow}} \in \{0,1\}^m$. We say that $M_{\boldsymbol{ow}}$ is a safe pre-image for H (with respect to C^H , $I_{H\text{-}target}$ and $I_{C\text{-}target} \in$) if*

1. *$C^H(M_{\boldsymbol{ow}}) = I_{C\text{-}target}$*
2. *the evaluation of $C^H(\cdot)$ on input $M_{\boldsymbol{ow}}$ does not involve a call to $H(X)$ with $H(X) = I_{H\text{-}target}$):*

$$\forall X \in qry^H(C^H(M)) \cup qry^H(C^H(M_{\boldsymbol{spr}})) : X = X_{target} \vee H(X) \neq H(X_{target})$$

We define the predicate $\boldsymbol{safeOw}^{C^H}_{H,I}(M, M_{\boldsymbol{spr}})$ iff $(M_{\boldsymbol{ow}})$ is a safe pre-image for hash function H and target image I.

For $k \in \mathbb{N}$ we denote with $\boldsymbol{safeOw}^{C^{H_1,\dots,H_l}}_{k,I_1,\dots,I_l}(M_{\boldsymbol{ow}})$ that $(M_{\boldsymbol{ow}})$ is a safe pre-image for target images $I_1,..,I_l$ for k out of C's l oracles H_i $(i = 1,...,l)$. Here I_i is the target image for function H_i $(i = 1,...,l)$.

C Missing Proofs

C.1 Proof of Lemma 1

Before we prove Lemma 1 we need some additional notation: For hash functions $H_1, \dots, H_l$ we may simply write H^l. We use the same abbreviation for messages and write X^l for the messages $X_1, \dots, X_l$. With $\mathbf{spr}_{k,X^l}(C^{H^l}(\cdot), Y, Y')$ we denote the event that k second pre-images for target messages $X_1, \dots, X_l$ are found during the evaluation of $C^{H^l}(Y)$ and $C^{H^l}(Y')$ (compare Definition 5).

Let us now repeat the Lemma and present the proof:

Lemma 1. *Let $C : \{0,1\}^m \to \{0,1\}^n$ be any oracle machine with q_C oracle gates per hash function $H_i : \{0,1\}^w \to \{0,1\}^v$ (with $i = 1,..,l$). Let $X_1,...,X_l \in \{0,1\}^w$ be target messages for functions $H_1,...,H_l$. For messages $M, M_{\boldsymbol{spr}}$ sampled as $M \leftarrow \{0,1\}^m$ and $M_{spr} \leftarrow (C^{H_1,\dots,H_l})^{-1}(M)$ we have:*

$$Pr[\boldsymbol{safeSpr}^{C^{H_1,\dots,H_l}(\cdot)}_{k,X_1,\dots,X_l}(M, M_{spr})] \geq 1 - q_C^{l-k+1} \cdot \binom{l}{l-k+1} \cdot 2^{n-(v+1)(l-k+1)} \tag{19}$$

Proof (of Lemma 1). Let $Y \leftarrow \{0,1\}^m$ and $Y' \leftarrow \{0,1\}^m$ be random variables uniformly distributed over $\{0,1\}^m$. Then, the distribution of $M, M_{\mathbf{spr}}$ is equivalent to the distribution of Y, Y' conditioned on $C^{H_1,\dots,H_l}(Y) = C^{H_1,\dots,H_l}(Y')$. Hence we have:

$$Pr[\mathbf{safeSpr}^{C^{H^l}}_{k,X^l}(M, M_{\mathbf{spr}})] = Pr[\mathbf{safeSpr}^{C^{H^l}}_{k,X^l}(Y, Y')|C^{H^l}(Y) = C^{H^l}(Y')]$$

This probability is (using the definition of the predicate **safeSpr**) simply that of the predicate's second condition (there are no trivial second pre-images for $l-k+1$ out of the l target messages $X_1,...,X_l$ on hash functions $H_1,...,H_l$ during the evaluation of $C^{H^l}(\cdot)$ with inputs Y and Y') again conditioned on $C^{H^l}(Y) = C^{H^l}(Y')$.

$$\begin{aligned}
&Pr[\mathbf{safeSpr}^{C^{H^l}}_{k,X^l}(Y, Y')|C^{H^l}(Y) = C^{H^l}(Y')] \\
&= Pr[C^{H^l}(Y) = C^{H^l}(Y') \wedge \neg\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot), Y, Y')|C^{H^l}(Y) = C^{H^l}(Y')] \\
&= Pr[\neg\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot), Y, Y')|C^{H^l}(Y) = C^{H^l}(Y')]
\end{aligned}$$

This can be restated as:

$$\begin{aligned}
&Pr[\neg\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')|C^{H^l}(Y)=C^{H^l}(Y')] \\
&= \frac{Pr[\neg\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')\wedge C^{H^l}(Y)=C^{H^l}(Y')]}{Pr[C^{H^l}(Y)=C^{H^l}(Y')]} \\
&\geq \frac{Pr[C^{H^l}(Y)=C^{H^l}(Y')]-(1-Pr[\neg\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')])}{Pr[C^{H^l}(Y)=C^{H^l}(Y')]} \\
&= \frac{Pr[C^{H^l}(Y)=C^{H^l}(Y')]-Pr[\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')]}{Pr[C^{H^l}(Y)=C^{H^l}(Y')]} \\
&= 1-\frac{Pr[\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')]}{Pr[C^{H^l}(Y)=C^{H^l}(Y')]} \geq \epsilon \in [0,1] \qquad (20)
\end{aligned}$$

Here we used $Pr[A \wedge B] = Pr[A] - Pr[\neg B] + Pr[\neg(A \vee B)]$ in step 2. Now for $0 \leq \epsilon \leq 1$, equation (20) holds if and only if:

$$1-\epsilon \geq \frac{Pr[\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')]}{Pr[C^{H^l}(Y)=C^{H^l}(Y')]} \qquad (\iff)$$

$$(1-\epsilon)\cdot Pr[C^{H^l}(Y)=C^{H^l}(Y')] \geq Pr[\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')] \qquad (21)$$

As $Pr[C^{H^l}(Y)=C^{H^l}(Y')] \geq 2^{-n}$ and from Proposition 3 we know that:

$$Pr[\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(Y))] \leq \binom{l}{l-k+1}\left(\frac{q_C}{2^v}\right)^{l-k+1}$$

As we are now evaluating $C^{H^l}(\cdot)$ not only for message Y but also for message Y' we can bound the probability on the right hand side of equation (21) with:

$$Pr[\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')] \leq \binom{l}{l-k+1}\left(\frac{q_C}{2^{v-1}}\right)^{l-k+1}$$

Now we know that the following two inequalities hold:

$$(1-\epsilon)\cdot Pr[C^{H^l}(Y)=C^{H^l}(Y')] \geq (1-\epsilon)2^{-n}$$

$$\binom{l}{l-k+1}\left(\frac{q_C}{2^{v-1}}\right)^{l-k+1} \geq Pr[\mathbf{spr}_{l-k+1,X^l}(C^{H^l}(\cdot),Y,Y')]$$

Thus if

$$(1-\epsilon)2^{-n} \geq \binom{l}{l-k+1}\left(\frac{q_C}{2^{v-1}}\right)^{l-k+1}$$

and equivalently

$$\epsilon \leq 1-q_C^{l-k+1}\cdot\binom{l}{l-k+1}\cdot 2^{n-(v+1)(l-k+1)} \qquad (22)$$

then equation (21) holds.

We proved that for $0 \leq \epsilon \leq 1$

$$Pr[\mathbf{safeSpr}_{k,X_1,...,X_l}^{C^{H_1,...,H_l}(\cdot)}(M, M_{spr})] \geq \epsilon$$

holds, if

$$\epsilon \leq 1 - q_C^{l-k+1} \cdot \binom{l}{l-k+1} \cdot 2^{n-(v+1)(l-k+1)}$$

This directly implies the proposition and we can conclude our proof. □

C.2 Proof of Lemma 2

Lemma 2. *Let (C,P) be as in Proposition 1 with*

$$v > \log q_P^H + \log l + 2 \tag{23}$$

Also let for any set of target messages $X_1, ..., X_l \in \{0,1\}^w$

$$Pr[\boldsymbol{spr}_{l-k+1,X_1,...,X_l}(P^{\mathcal{B}^{X_1,...,X_l},H_1,...,H_l}(X_1,...,X_l))] \geq 0.5 \quad . \tag{24}$$

Then $H_1, ..., H_l$ can be compressed below $l2^w v$ bits.

For an outline of the proof see page 14 in Section 3.5.

Proof (of Lemma 2). Let X_τ be the combined target message, defined as:

$$X_\tau = (X_1, ..., X_l) \in \overbrace{\{0,1\}^w \times ... \times \{0,1\}^w}^{l \text{ times}}$$

Let the oracle $\mathcal{B}^{X_1,...,X_l}$ be initialized with the target messages.

Let us start by defining the compression function **com**. For compressing $H_1, ..., H_l$ we run $P^{\mathcal{B}^{X^l},H^l}(X_\tau)$ and distinguish between two cases. If P does not produce the necessary second pre-images then **com** simply outputs each of the entire function tables prepended with a 0 to distinguish this case. If by $\tilde{H}_i$ we denote the function table of hash function H_i then we can define this case as:

$$\mathbf{com}(\tilde{H}_1||...||\tilde{H}_l) = 0||\tilde{H}_1||...||\tilde{H}_l \quad \text{if} \quad \neg\mathbf{spr}_{l-k+1,X^l}(P^{\mathcal{B}^{X^l},H^l}(X_1,...,X_l))$$

If, on the other hand, P succeeds in finding all necessary $l-k+1$ second pre-images, then we know that for some hash function H_s for which P succeeded in finding a second pre-image $X^s_{\mathbf{spr}}$ the H_s query that resulted in the second pre-image did not come from the breaking oracle but from P directly.

As **com** can distinguish between the calls to the hash oracles from P and $\mathcal{B}^{X^l}$, **com** can identify H_s. The compression algorithm **com** will output a 1 appended with the index of hash function H_s appended with the index (relative to the maximum number of calls to H_s by P) of the second pre-image $X^s_{\mathbf{spr}}$ appended with a compressed version of the function table for H_s appended with the full function tables for all other hash functions. For this case, let $A_1, A_2, ..., A_\sigma$ denote all the H_s queries done by $P^{\mathcal{B}^{X^l},H^l}$ plus the H queries made by $\mathcal{B}$ (in the order they occur, up to query $X^s_{\mathbf{spr}}$, the one that finds the second pre-image, and without repetitions; thus $A_\sigma = X^s_{spr}$). Each H query by P increases the sequence $A_1, A_2, ...$ by at most one, while a query to $\mathcal{B}$ by P can increase the sequence by arbitrarily many values. Let $\tilde{H}_s^-$ denote the reduced function table of H_s with the rows $A_1, ..., A_\sigma$ and X_s (the s^{th} component of X_τ) removed. We know that query A_σ is requested by P. Let this be the i^{th} query to H_s by P. Also let $\tilde{H}^{l-s}$ be the function tables of hash functions H_i with $i \in \{1, ..., l\} \backslash s$. Algorithm **com** is now defined as:

$$\mathbf{com}(\tilde{H}_1||...|\tilde{H}_l) =$$
$$\begin{cases} 0||\tilde{H}_1||...||\tilde{H}_l & \text{if } \neg\mathbf{spr}_{l-k+1,X^l}(P^{\mathcal{B}^{X^l},H^l}(X_1,...,X_l)) \\ 1||\langle s\rangle_{\log_l}||\langle i\rangle_{\log_{q_P}}||H_s(A_1)||...||H_s(A_\sigma)||\tilde{H}_s^-||\tilde{H}^{l-s} & \text{otherwise} \end{cases}$$

Before we describe the decompression algorithm **dec**, let us examine **com**'s compression rate. If

$$\neg\mathbf{spr}_{l-k+1,X^l}(P^{\mathcal{B}^{X^l},H^l}(X_1,...,X_l))$$

then $|\mathbf{com}(H)| = l2^wv + 1$. Since $Pr[\mathbf{spr}_{l-k+1,X^l}(P^{\mathcal{B}^{X^l},H^l}(X_1,...,X_l))]$ is bounded from below by 0.5, this happens with probability at most

$$p := Pr[\neg\mathbf{spr}_{l-k+1,X^l}(P^{\mathcal{B}^{X^l},H^l}(X_1,...,X_l))] \leq 0.5$$

In the other case (when the necessary second pre-images are generated) the compressed function table has length $1 + (2^w - 1)v + \log q_P^H + (l-1)2^wv + \log l$. Thus we can give the expected overall length as:

$$\begin{aligned} E[|\mathbf{com}(H)|] &\leq p(l2^wv + 1) + \\ &\qquad (1-p)(1 + (2^w-1)v + \log q_P^H + (l-1)2^wv + \log l) \\ &= 1 + l2^wv - (1-p)(v - \log q_P^H - \log l) \end{aligned} \tag{25}$$

We will now define the decompression algorithm **dec**. On input T, $\mathbf{dec}(T)$ parses T as $b||T'$ and if $b = 0$ outputs T', which in this case is exactly the concatenation of all function tables $\tilde{H}_1||...||\tilde{H}_l$. If $b = 1$, **dec** goes on parsing T' as

$$\langle s\rangle_{\log l}||\langle i\rangle_{\log q_P}||H_s(A_1)||...||H_s(A_\sigma)||\tilde{H}_s^-||\tilde{H}^{l-s}$$

It now simulates $P^{\mathcal{B}^{X^l},H^l}(X_1,...,X_l)$ up to the point where P makes its i^{th} query $X^s_{\mathbf{spr}} = A_\sigma$ to oracle H_s. We know that $H_s(X_s) = H_s(X^s_{\mathbf{spr}}) = H_s(A_\sigma)$ so we now have all the missing values (and indices) in $\tilde{H}_s^-$ needed to reconstruct the full function table H_s together with all the other function tables.

com does compress H on average

The final step will now be to show that the achieved compression rate is in fact less than $l2^wv$ bits. The upper bound on the expected compression rate from equation (25) is less than $l2^wv$ if and only if:

$$\begin{aligned} 1 + l2^wv - (1-p)(v - \log q_P^H - \log l) &< l2^wv \qquad (\iff) \\ (1-p)(v - \log q_P^H - \log l) &> 1 \end{aligned}$$

(and thus if)

$$0.5(v - \log q_P^H - \log l) > 1 \tag{26}$$

Since the requirement given by equation (23) is $v > \log q_P^H + \log l + 2$, the last inequality (26) holds. This completes the proof. □

D Main Result for Target Collision Resistance and Pre-Image Resistance

D.1 Adaptation for Target Collision Resistance

Let us describe what is necessary to adapt the proof (of Theorem 2) to the case of target collision resistance. For this let us quickly recapitulate what we have done for second preimages. The proof idea was to take uniformly random functions $H_1, \ldots, H_l$ as hash functions and define a breaking oracle which outputs only safe second preimages; that is, the second pre-images found by $\mathcal{B}$ do not help the reduction $P^{\mathcal{B},H_1,\ldots,H_l}$ too much in finding all necessary second pre-images for the hash functions. We have then shown that the defined breaking oracle ρ-breaks (for SPR) every (k, l)-combiner with a certain probability and finally that: if $P^{\mathcal{B},\cdot H_1,\ldots,H_l}$ is still able to find all $l - k + 1$ second pre-images then we can use those to compress the combined function table of $H_1, \ldots, H_l$ below $l2^w v$ bits; but due to proposition 2 this is not possible and we have hence found a contradiction.

To adapt the argument to work for target collision resistance there is only little we have to change: namely the breaking oracle and the final compression based proof where the oracle is used. In Definition 14 we give the definition for a safe target collision. In the SPR case the first part of the collision is specified from the outside (according to some distribution). We have captured this notion by giving the breaking oracle an input message for which it should come up with a second pre-image. In the case of target collision resistance the adversary may specify the first part of the collision. What changes is that now the breaking oracle is called in two steps. In the first step the breaking oracle is called with some randomness $R \in \mathcal{R}$ and it outputs a target message for which it will in the second step tries to find a collision. If you recall the definition of the breaking oracle for the SPR case, then the oracle was completely defined by some random function $\phi : \{0,1\}^* \to \{0,1\}^m$. We define the TCR oracle in exactly the same way: For the first step (choosing the first part of the collision) the oracle returns $\phi(R)$. The second part (finding a target collision for $\phi(R)$, i.e. a message M such that $C^H(\phi(R)) = C^M$) the oracle does exactly the same as the breaking oracle for SPR. As in the SPR case the breaking oracle has to be initialized with target message $X \in \{0,1\}^w$ (from the hash functions domain) for which it should make sure that no trivial target collision occurs.

The next part of the argument, that the defined breaking oracle breaks every (k, l)-combiner with high probability, does not need to be adapted. The second and final change is in the proof of Lemma 2 where the reduction is used to compress the function table of hash functions $H_1, \ldots, H_l$. Here we gave both the compression and decompression algorithm access to a shared message $X_\tau \in \{0,1\}^w$. For the TCR case we can omit this message as the reduction $P^{\mathcal{B},H_1,\ldots,H_l}$ has to output the first part of the collision and this will be the same for **com** and **dec** if they call P with the same random coins, which they have to do in any case. Note that it is sufficient for **com** and **dec** to initialize the breaking oracle $\mathcal{B}$ with the target messages after it is generated by P as $\mathcal{B}$ needs the target messages for its second round only.

D.2 Adaptation for Pre-Image Resistance

For the case of pre-image resistance we have to adapt slightly more. First notice that the breaking oracle is given randomness and a message in the SPR case, whereas now the breaking oracle gets randomness and an image. As before the breaking oracle $\mathcal{B}$ outputs only safe pre-images. In a next step we have to adapt Lemma 1. This is also straight forward. The main differences are, that we now consider a target image I_C for the combiner and only a single message $M_{\mathbf{ow}} \leftarrow (C^{H_1,\ldots,H_l})^{-1}(I_C)$. In the proof we then sample only a single $Y \leftarrow \{0,1\}^m$. Exchanging $\mathbf{safeSpr}_{k,X_1,\ldots,X_l}^{C^{H_1,\ldots,H_l}(\cdot)}(M, M_{\mathbf{spr}})$ for $\mathbf{safeOw}_{k,I_1,\ldots,I_l}^{C^{H_1,\ldots,H_l}(\cdot)}(M_{\mathbf{ow}})$ (where the I_i are the target images for the various hash functions) and $C^{H^l}(Y) = C^{H^l}(Y')$ for $C^{H^l}(Y) = I_C$ yields the required result.

The next part of the argument, that the defined breaking oracle breaks every (k, l)-combiner with high probability, does not need to be adapted. The final change then is to give algorithms **com** and **dec** not a target images but also target messages which induces the images (hash value of message). We now need that if P finds pre-images which are different from the target messages. As there are, on average, 2^{w-v} pre-images for each image (as the functions were chosen randomly) this can be assumed without loosing too much probability.