On existence of robust combiners for cryptographic hash functions[⋆]

Michal Rjaško

Department of Computer Science, Faculty of Mathematics, Physics and Informatics,
Comenius University, Bratislava
rjasko@dcs.fmph.uniba.sk

Abstract. *A (k,l)-robust combiner for collision resistant hash functions is a construction, which takes l hash functions and combines them so that if at least k of the components are collision resistant, then so is the resulting combination. A black-box (k,l)-robust combiner is robust combiner, which takes its components as black-boxes. A trivial black-box combiner is concatenation of any $(l-k+1)$ of the hash functions. Boneh and Boyen [1] followed by Pietrzak [3] proved, that for collision resistance we cannot do much better that concatenation, i.e. there does not exist black box (k,l)-robust combiner for collision resistance, whose output is significantly shorter that the output of the trivial combiner. In this paper we analyze whether robust combiners for other hash function properties (e.g. preimage resistance and second preimage resistance) exist.*

Key words: *Cryptographic hash function, robust combiner, preimage resistance, second preimage resistance*

1 Introduction

Cryptographic hash functions play important role in the current cryptography. In the last few years, many attacks on popular hash functions believed to be secure (e.g. SHA1, MD5) have been proposed. Within these attacks arises a question, whether we are able to construct a secure hash function. A hash function is a function $H : \{0,1\}^* \to \{0,1\}^v$, which maps messages (strings of 0 and 1) of arbitrary length to strings of fixed length – called images. Practically useful hash functions must guarantee several security properties: they should be collision resistant, what means that it is hard to find two different messages which map to the same image. Other important properties of hash functions are preimage resistance (for given image, it is hard to find its preimage, i.e. a message which maps to that image) and second-preimage resistance (for given message, it is hard to find another message, which maps to the same image). For formal definitions of the properties mentioned above or other notions of hash function security we refer to the works [4], [5].

Natural way how to construct secure hash function is to combine several known (regarded to be secure) hash functions in such a way, that if one of the combined functions appears to be insecure, the combination remains secure. For collision resistance we can achieve this by concatenation. Let $H_1, H_2 : \{0,1\}^* \to \{0,1\}^v$ be some hash functions. We can construct a hash function H, where

$$H(M) = H_1(M)||H_2(M).$$

Note, that if (M, M') is a pair of colliding messages for H, then this pair collides also for H_1 and H_2. Therefore if at least one of the H_1 and H_2 is collision resistant, then H is collision resistant too. However this approach has one important disadvantage – the output length of H is twice as large as the output of underlying hash functions H_1 and H_2, what can lead to problems with practical implementation, mainly on devices with small amount of memory as smart-cards.

Thus the question is, whether one can construct a secure combiner with output shorter than the concatenation. Boneh and Boyen in [1] proved the first negative result in this direction, in particular that there does not exist secure combiner for collision resistant hash functions with output shorter than concatenation, with an assumption, that the combiner queries each hash function exactly once. This result was generalized by Pietrzak [3], where the author proved that the secure combiners for collision resistance with significantly shorter output than concatenation do not exist. The later work consider a (k,l)-robust combiners for collision resistant, which are secure if at least k of the l components are secure.

In this paper we follow the work of Pietrzak [3] and prove the similar results for other important properties of hash functions – second preimage resistance and preimage resistance. We define a (k,l) combiner for preimage resistance and second preimage resistance and for these definitions we prove the impossibility results similar to one from [3].

Organization In the section 2 we start by some useful notation and continue with the formal definitions of (k,l) combiners for collision resistance, preimage resistance and second-preimage resistance. In the section 3 we prove the negative results, namely in the Theorem 1 we prove that secure combiner for preimage resistance with output significantly shorter than concatenation does not exists and in the Theorem 2 we prove the similar result for second-preimage resistance.

⋆ This paper was supported by VEGA grant number 1/0266/09 and by Comenius University grant number UK/365/2009.

2 Preliminaries

In this section we formally define a combiner of l hash functions for three notions of hash function security – collision resistance, preimage resistance and second preimage resistance. The definition of the combiner for collision resistance is from [1], the other definitions (i.e. for preimage and second-preimage resistance) are slight modification of the former one.

We start with some basic notation. We write $M \overset{\$}{\leftarrow} \mathcal{S}$ for the experiment of choosing random element from the distribution $\mathcal{S}$. If $\mathcal{S}$ is a finite set, then M is chosen uniformly from $\mathcal{S}$. Concatenation of finite strings M_1 and M_2 we denote by $M_1||M_2$ or simply M_1M_2.

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

Let λ be some parameter. In this section, and later, by H_i we denote a function mapping from $\{0,1\}^*$ to $\{0,1\}^v$, where $i = 1 \dots l$ and $v = p(\lambda)$ for some polynomial p. In general, a combiner of l hash functions H_i, $i = 1, \dots, l$ for some notion of hash function security is a pair (C, P), where

– $C : \{0,1\}^m \rightarrow \{0,1\}^n$ does the "combination" ($m = p_m(\lambda)$ and $n = p_n(\lambda)$ for some polynomials p_m, p_n), i.e. it is an oracle turing machine with oracle access to $H_1, \dots, H_l$. It behaves as a standard hash function (i.e. it takes some message on input and outputs a hash of the message – string of fixed length), but during the computation it can query any of its oracles.
– P provides a proof of the security for C. It is an algorithm, which transforms the "ability" of breaking the C (with respect to the particular security property) to the ability of breaking the candidates.

We note that both C and P should be efficient – they run in a time that is polynomial in the security parameter λ (and thus it is polynomial also in m, n or v). The oracle turing machine C is the same for combiners of all security notions. On the other hand, P needs to be modified when going from one security notion to another.

The security of combiner (with respect to some security notion) is determined by the number how many of the candidate hash functions need to be secure in order to guarantee that the resulting combiner is secure. By (k,l)-combiner (C, P) we denote the combiner (C, P), which is secure, if at least k of the l candidate hash functions are secure.

We expect from all candidate hash functions H_i to have the same output length v. We assume this just for simplicity, our results can be easily extended for variable output length of these candidate hash functions.

In this paper we deal only with black box combiners, what means that the combination algorithm (C) has only black box (i.e. oracle) access to the candidate hash functions. It does not know the structure of underlying primitives $H_1, \dots, H_l$.

We start by formal definition of the combiner for collision resistance from [3].

2.1 Combiner for collision resistance

A collision resistant (k,l)-combiner for l hash functions $H_1, \dots, H_l$ is a pair (C, P) of oracle turing machines C and P, where

– $C : \{0,1\}^m \rightarrow \{0,1\}^n$ has oracle access to hash functions $H_1, \dots, H_i$ and performs the "combination" of hash functions. On input it takes a message of fixed length m and outputs an image of fixed length n. The output of C on an input M and with oracle access to $H_1, \dots, H_l$ is denoted as $C^{H_1,\dots,H_l}(M)$.
– P is an oracle machine, which provides a "proof" of security for C. It takes as input a pair of messages (M, M') and outputs two vectors

$$\mathbf{W} = (w_1, \dots, w_l) \text{ and } \mathbf{W}' = (w'_1, \dots, w'_l).$$

The role of the algorithm P is to transform the collision in C to collisions in at least $l - k + 1$ of the underlying hash functions $H_1, \dots, H_l$. If such a P exists, which transforms collisions in C to collisions in H_i for *all* compatible $H_1, \dots, H_l$, then we consider C as a collision resistant (k,l) combiner. Now, we formally define what was discussed above.

We say that P k succeeds on $H_1, \dots, H_l$, M and M' if:

$$\exists J \subseteq \{1, \dots, l\}, |J| \geq l - k + 1 :$$
$$(\forall j \in J) : \ (w_j, w'_j) \text{ is a collision for } H_j$$
$$(\text{i.e. } H_j(w_j) = H_j(w'_j))$$

Let

$$\mathbf{Adv}_P^{\mathrm{Coll}[k]}[(H_1, \dots, H_l), M, M']$$

denote the probability that P k-succeeds.

We say that (C, P) is ε-secure (k, l)-Coll-combiner, if for all $H_1, \ldots, H_l$ and all collisions (M, M') in C we have:

$$\mathbf{Adv}_P^{\mathrm{Coll}[k]}[(H_1, \ldots, H_l), M, M'] \geq 1 - \varepsilon$$

We consider (C, P) to be secure (k, l)-Coll-combiner, if ε is negligible in the security parameter λ.

For example, a secure $(1, 2)$-combiner for collision resistance can look like follows:

– $C^{H_1,H_2}(M) = H_1(M)||H_2(M)$
– $P^{H_1,H_2}(M, M') = (M, M), (M, M')$

The turing machine C just passes its input to the candidates and returns the concatenation of their output. Note that collision in C implies collision in both H_1 and H_2. In more detail, if (M, M') is a collision for such a C, then (M, M') is also a collision in both H_1 and H_2. Thus P has easy work – it copies its input to the output. It is easy to see, that for all H_1, H_2 and all collisions (M, M') in C is

$$\mathbf{Adv}_P^{\mathrm{Coll}[1]}[(H_1, H_2), M, M'] = 1.$$

Therefore (C, P) is 0-secure $(1, 2)$-Coll-combiner.

2.2 Combiner for preimage resistance

The combiner for preimage resistance is similar to one for collision resistance. Only difference is in the algorithm P, which provides "a proof of the security". Algorithm P is given on its input challenge images $y_1, \ldots, y_l$ of $H_1, \ldots, H_l$, for which it has to find preimages. It plays a game, in which it chooses an image of C for which it gets a preimage.

A preimage resistant combiner for l hash functions $H_1, \ldots, H_l$ is a pair (C, P) of oracle turing machines C and P, where

– $C : \{0,1\}^m \to \{0,1\}^n$ is the same as in the collision resistant combiner.
– P is an oracle turing machine, which provides a "proof" of security for C. It plays the following game. Let $f : \{0,1\}^n \to \{0,1\}^m \cup \{\perp\}$ be a function, such that for all $Y \in \{0,1\}^n$ is $f(Y) = M$, for which $C^{H_1,\ldots,H_l}(M) = Y$. If such a M does not exists, then $f(Y) = \perp$.

Pre-Comb^f game:
1. Messages $w_1, \ldots, w_l \in \{0,1\}^m$ are chosen at random, then images $y_1 = H_1(w_1), \ldots, y_l = H_l(w_l)$ are computed and given to P on its input.
2. P with oracle access to $H_1, \ldots, H_l$ outputs some image $Y \in \{0,1\}^n$.
3. A message $M = f(Y)$, is given to P (note that $C^{H_1,\ldots,H_l}(M) = Y$), if $f(Y) = \perp$, then the game ends and P fails.
4. P continues (still can query $H_1, \ldots, H_l$) and outputs a vector $\mathbf{W} = (w'_1, \ldots, w'_l)$.

We note, that the function f, which parametrizes the Pre-Comb game can be understood as a deterministic "device", which finds preimages in C (it need not to be efficient). An algorithm P from a secure preimage resistant combiner should win the Pre-Comb game for all possible functions f (or at least for non-negligible part of such functions, however in this paper we consider combiners to be secure if they win for all possible fs).

We say that P k succeeds on $H_1, \ldots, H_l$, $y_1, \ldots, y_l$ and f if:

$$\exists J \subseteq \{1, \ldots, l\}, |J| \geq l - k + 1 :$$

$$(\forall j \in J) : \ w'_j \text{ is preimage of } y_j \text{ on } H_j$$

$$(\text{i.e. } H_j(w'_j) = y_j)$$

Let

$$\mathbf{Adv}_P^{\mathrm{Pre}[k]}[(H_1, \ldots, H_l), (y_1, \ldots, y_l), f]$$

denote the probability that P k-succeeds.

Finally, (C, P) is ε-secure (k, l)-Pre-combiner, if for all $H_1, \ldots, H_l$, all images $y_1, \ldots, y_l$ and all possible f we have:

$$\mathbf{Adv}_P^{\mathrm{Pre}[k]}[(H_1, \ldots, H_l), (y_1, \ldots, y_l), f] \geq 1 - \varepsilon$$

We say (C, P) is secure (k, l)-Coll-combiner, if ε is negligible in the security parameter λ.

For example consider the following $(1, 2)$-combiner for preimage resistance:

– C^{H_1,H_2} $(M_1||M_2) = H_1(M_1) \;||\; H_2(M_2)$ – C gets a message on its input, divides it into two parts of roughly equal length and passes these two parts to H_1 and H_2.
– P – given images y_1, y_2 on input, P computes the image $Y = y_1||y_2$ and returns it in the second step of the Pre-Comb game. In turn P receives a message M, where $C^{H_1,H_2}(M) = Y$. Finally P divides the message M into two parts w_1 and w_2 (exactly as C divides its input) and returns (w_1, w_2).

Since $C^{H_1,H_2}(w_1||w_2) = H_1(w_1)||H_2(w_2)$, we can see that $H_1(w_1) = y_1$ and $H_2(w_2) = y_2$. Thus (C, P) is 0-secure $(1, 2)$-Pre-combiner.

2.3 Combiner for second-preimage resistance

Here we define a combiner for second-preimage resistance. Again, only difference from combiners for preimage or collision resistance is in the algorithm P.

A second-preimage resistant combiner for l hash functions $H_1, \ldots, H_l$ is a pair (C, P) of oracle turing machines C and P, where

– $C : \{0,1\}^m \to \{0,1\}^n$ is the same as in the case of preimage resistant or collision resistant combiner.
– P is an oracle turing machine, which provides a "proof" of security for C. It plays the following game. Let $f : \{0,1\}^m \to \{0,1\}^m \cup \{\perp\}$ be a function, such that for all $M \in \{0,1\}^m$ is $f(M) = M'$, for $M' \neq M$ and $C^{H_1,\ldots,H_l}(M) = C^{H_1,\ldots,H_l}(M')$. If such a M' does not exist, then $f(M) = \perp$.

Sec-Comb^f game:
1. Message $w \in \{0,1\}^m$ is chosen at random and given to P on its input.
2. P with oracle access to $H_1, \ldots, H_l$ outputs some message $M \in \{0,1\}^m$.
3. P is given a message $M' = f(M)$ ($M' \neq M$ and $C^{H_1,\ldots,H_l}(M) = C^{H_1,\ldots,H_l}(M')$).
4. P continues (still can query $H_1, \ldots, H_l$) and outputs a vector $(w'_1, \ldots, w'_l)$.

We say that P k succeeds on $H_1, \ldots, H_l$, w and f if:

$$\exists J \subseteq \{1, \ldots, l\}, |J| \geq l - k + 1 :$$
$$(\forall j \in J) : \ w'_j \text{ is second-preimage of } w \text{ on } H_j$$
$$(\text{i.e. } H_j(w'_j) = H_j(w) \text{ and } w'_j \neq w_j)$$

Now, let

$$\mathbf{Adv}_P^{\mathrm{Sec}[k]}[(H_1, \ldots, H_l), w, f]$$

denote the probability that P k-succeeds.

Finally, (C, P) is ε-secure (k, l)-Sec-combiner, if for all $H_1, \ldots, H_l$, all messages w and all possible f we have:

$$\mathbf{Adv}_P^{\mathrm{Sec}[k]}[(H_1, \ldots, H_l), w, f] \geq 1 - \varepsilon$$

And we say (C, P) is secure (k, l)-Coll-combiner, if ε is negligible in the security parameter λ.

Consider the following example of secure $(1, 2)$ combiner for second-preimage resistance:

– $C^{H_1,H_2}(M) = H_1(M)||H_2(M)$
– P – on input w in the second step of the Sec-Comb game P returns a message $M = w$. In turn P receives a message $M' \neq M$, where $C^{H_1,H_2}(M') = C^{H_1,H_2}(M)$. Finally P returns a vector (M', M').

It is easy to see,

$$\begin{aligned} C^{H_1,H_2}(M') &= H_1(M')||H_2(M') \\ &= H_1(M)||H_2(M) \\ &= C^{H_1,H_2}(M), \end{aligned}$$

thus $H_1(M') = H_1(M)$, $H_2(M') = H_2(M)$.

3 Impossibility proofs

Boneh and Boyen [1] followed by Pietrzak [3] showed, that there does not exist a collision resistant (k, l) combiner with short output. In this section we prove the similar results for preimage resistance and second-preimage resistance.

The impossibility result for preimage resistant combiners is given in the Theorem 1, and in the Theorem 2 is given the impossibility result for second-preimage resistance.

We start by notation used in the rest of this paper. Let (C, P) be some combiner and let:

– $\mathbf{W}_i(M)$ be the set of oracle queries to H_i made while evaluating $C^{H_1,\ldots,H_l}$ on the message M
– $\mathbf{V}_i(M) = \{H_i(M); M \in \mathbf{W}_i(M)\}$ be the set of corresponding answers.
– $w_{i,j}(M)$ be the j-th query to H_i made while evaluating $C^{H_1,\ldots,H_l}(M)$ and $v_{i,j}(M)$ be the corresponding answer.

To simplify the presentation we will assume that P can output only messages that it has queried during the game. Note that we can assume this without loss of generality.

Theorem 1. *Let (C, P) be a (k, l)-combiner for preimage resistance, where C can make at most q_C oracle queries. Suppose, that*

$$n < (v - \lg(q_C))(l - k + 1) - l$$

Then there exist $y_1, \ldots, y_l$, $H_1, \ldots, H_l$ and f, such that

$$\mathbf{Adv}_P^{Pre[k]}[(H_1, \ldots, H_l), (y_1, \ldots, y_l), f] \textit{ is negl. in } \lambda$$

Proof. Let $H_1, \ldots, H_l : \{0,1\}^* \to \{0,1\}^v$ be all uniformly random hash functions. Let q_P be the total number of queries that P makes in the Pre-Comb game and let $y_1, \ldots, y_l$ be the challenge images that P gets in the first step of the game. From the fact, that P runs in a polynomial time we have that $q_P = p(\lambda)$ for some polynomial p. Therefore the probability that P queries any H_i with some message w, such that $H_i(w) = y_i$, is negligible in λ. To see this only a simple idea is needed. All of the H_is are random thus the probability that P gets output y_i for some $i = 1, \ldots, l$ in one query is $l/2^v$. The probability that P queries y_i for some i in q_P queries is therefore

$$q_P \frac{l}{2^v} = p(\lambda)\frac{l}{2^{p'(\lambda)}},$$

what is negligible in λ.

Thus P's only chance to win is in the message M it gets as a preimage for the image Y chosen in the second step of the Pre-Comb game. We show, that if $n < (v - \lg(q_C))(l-k+1) - l$, then there exist images $y_1, \ldots, y_l$, and a function f such that for all images Y which P can output in the second step, evaluating of $C(f(Y))$ does not present a preimage for at least k of the $y_1, \ldots, y_l$. This means, that for such $H_1, \ldots, H_l$, $y_1, \ldots, y_l$ and f is

$$\mathbf{Adv}_P^{\mathrm{Pre}[k]}[(H_1, \ldots, H_l), (y_1, \ldots, y_l), f]$$

negligible in λ, what we want to prove.

Let $H_1, \ldots, H_l$ be as defined above (independent random hash functions) and let $Y \in \{0,1\}^n$ be some image of $C^{H_1,\ldots,H_l}$. Consider the following random experiment. First, images $y_1, \ldots, y_l \in \{0,1\}^v$ are chosen at random and then a message $M \in \{0,1\}^m$ is randomly chosen. Now consider the following events:

$$\begin{aligned} \mathcal{E}_1 &\iff C^{H_1,\ldots,H_l}(M) = Y \\ \mathcal{E}_2 &\iff \exists J \subseteq \{1, \ldots, l\}, |J| > l - k : \\ &\qquad \forall j \in J : y_j \in \mathbf{V}_j(M) \end{aligned}$$

Note that if $\Pr[\mathcal{E}_1] > \Pr[\mathcal{E}_2]$. then $\Pr[\mathcal{E}_1 \wedge \neg\mathcal{E}_2] > 0$, what means, that there exist images $y_1, \ldots, y_l \in \{0,1\}^v$ such that for each $Y \in \{0,1\}^n$ there exists a message $M \in \{0,1\}^m$ for which $C^{H_1,\ldots,H_l}(M) = Y$ and the evaluation of $C^{H_1,\ldots,H_l}(M)$ does not present preimages for $y_1, \ldots, y_l$. In other words, it means that there exist images $y_1, \ldots, y_l$ and a function f[1] for which the theorem holds.

We know that for particular Y and randomly chosen M is

$$\Pr[C^{H_1,\ldots,H_l}(M) = Y] \geq 2^{-n}.$$

Thus

$$\Pr[\mathcal{E}_1] \geq 2^{-n}.$$

To find $\Pr[\mathcal{E}_2]$, let q_i be the number of queries to H_i made by C (note that $\sum_{i=1}^{l} q_i = q_C$). The $\Pr[\mathcal{E}_2]$ can be upper bounded by the probability that the best oracle algorithm $A^{H_1,\ldots,H_l}$, which is allowed to query H_i at most q_i times finds a preimage for at least $l-k+1$ of y_i (A can evaluate $C^{H_1,\ldots,H_l}$ and then A's success probability is equal to $\Pr[\mathcal{E}_2]$). Since H_i are all independent random functions, the best A can do is to query each H_i with q_i distinct inputs. Now we follow the same steps as Pietrzak [3] did in the proof of the similar theorem for collision resistant combiner.

$$\begin{aligned} \Pr[\mathcal{E}_2] &\leq \Pr[A^{H_1,\ldots,H_l} \text{ finds } l-k+1 \text{ preimages}] \\ &\leq \sum_{\substack{J \subseteq \{1,\ldots,l\} \\ |J| = l-k+1}} \Pr[\forall i \in J : A \text{ finds preimage for } y_i] \\ &\leq \sum_{\substack{J \subseteq \{1,\ldots,l\} \\ |J| = l-k+1}} \prod_{i \in J} \frac{q_i}{2^v} \\ &< \sum_{\substack{J \subseteq \{1,\ldots,l\} \\ |J| = l-k+1}} \frac{q_C^{l-k+1}}{2^{v(l-k+1)}} \\ &\leq \binom{l-k+1}{l} \frac{q_C^{l-k+1}}{2^{v(l-k+1)}} < \frac{2^l q_C^{l-k+1}}{2^{v(l-k+1)}} \end{aligned}$$

When we put everything together:

$$\lg(\Pr[\mathcal{E}_1]) \geq \lg(2^{-n}) = -n$$

and

$$\begin{aligned} \lg(\Pr[\mathcal{E}_2]) &< \lg\left(\frac{2^l q_C^{l-k+1}}{2^{v(l-k+1)}}\right) \\ &= (-(v - \lg(q_C))(l-k+1) + l). \end{aligned}$$

Thus if $n < (v - \lg(q_C))(l-k+1) - l$ we have $\Pr[\mathcal{E}_1] > \Pr[\mathcal{E}_2]$, what we wanted to prove.

$\square$

The condition $n < (v - \lg(q_C))(l-k+1) - l$ gives the lower bound how short can be the output of a secure (k,l)-Pre-combiner. It looks rather unnatural, however if C is allowed to query each H_i exactly once and we consider only $(1,l)$-combiners (what means, that a combiner is secure if at least one of the candidates is secure) then the condition can be rewritten to $n < (v-1)l$.

We note, that this lower bound need not to be optimal – maybe there exists a higher lower bound. We leave such an analysis for future work. Similar analysis for collision resistant combiners can be found in the work [2].

Theorem 2. *Let (C, P) be a (k,l)-combiner for 2nd-preimage resistance, where C makes at most q_C oracle queries. Suppose, that*

$$n < (v - \lg(q_C))(l-k+1) - l + 1$$

Then there exist w, $H_1, \ldots, H_l$ and f, for which

$$\mathbf{Adv}_P^{Sec[k]}[(H_1, \ldots, H_l), w, f] \text{ is negligible in } \lambda$$

[1] for each Y we set $f(Y)$ to the corresponding message M

Proof. The proof is very similar to one in the Theorem 1, therefore we provide just a sketch of the proof. Let $H_1, \dots, H_l$ be all independent random hash functions. We claim, that the probability where P queries a second-preimage of w for at least one of the H_i is negligible. This is due the fact, that P runs in a polynomial time and therefore it can make at most polynomial number of queries, what is not enough for winning against random functions (for more formal discussion see the similar part in the proof of the Theorem 1).

Therefore we only need to prove that if $n < (v - \lg(q_C))(l-k+1) - l + 1$, then there exist a message w and a function f, where for all messages M that P can output in the second step of the Sec-Comb game evaluation of $C(f(M))$ does not present a second preimage of w for at least k of the $H_1, \dots, H_l$.

Thus let $H_1, \dots, H_l$ be as defined above and let $M \in \{0,1\}^m$ be some message. Consider the random experiment, where $w \in \{0,1\}^m$ and $M' \in \{0,1\}^m$ are chosen uniformly at random. We define the following events $\mathcal{E}_1$ and $\mathcal{E}_2$:

$$\begin{aligned}\mathcal{E}_1 &\iff M' \neq M \wedge C^{H_1,\dots,H_l}(M) = C^{H_1,\dots,H_l}(M') \\ \mathcal{E}_2 &\iff \exists J \subseteq \{1,\dots,l\}, |J| > l-k : (\forall j \in J)(\exists i) : \\ &\qquad v_{j,i}(M') = H_j(w) \wedge w_{j,i}(M') \neq w\end{aligned}$$

In other words, $\mathcal{E}_1$ is the event when M' is the second-preimage of M in the sense of $C^{H_1,\dots,H_l}$ and $\mathcal{E}_2$ means that the evaluation of $C^{H_1,\dots,H_l}(M')$ does presents at least $l-k+1$ second-preimages for w. Again, if we prove that $\Pr[\mathcal{E}_1] > \Pr[\mathcal{E}_2]$, then $\Pr[\mathcal{E}_1 \wedge \neg\mathcal{E}_2] > 0$, what means that for each M the above $w_1, \dots, w_l$ and M' exist and therefore there exists a function f (we set $f(M) := M'$) for which the theorem holds (together with $w_1, \dots, w_l$).

Now, since $m > n$

$$\Pr[\mathcal{E}_1] = 2^{-n} - 2^{-m} \geq 2^{-n-1}$$

Now, let q_i be the maximum number of queries to H_i made by C. We can upper bound $\Pr[\mathcal{E}_2]$ by the probability that the best oracle algorithm A which can query each H_i at most q_i times finds second-preimage of w for at least $l-k+1$ of the H_is. However $H_1, \dots, H_l$ are all independent random functions, thus the best A can do is to query each H_i on q_i distinct inputs different from w. If we follow the steps as in the corresponding part of the proof of the Theorem 1 we get

$$\Pr[\mathcal{E}_2] < \frac{2^l q_C^{l-k+1}}{2^{v(l-k+1)}}$$

When we put everything together:

$$\lg(\Pr[\mathcal{E}_1]) \geq \lg(2^{-n-1}) = -n - 1$$

and

$$\begin{aligned}\lg(\Pr[\mathcal{E}_2]) &< \lg\left(\frac{2^l q_C^{l-k+1}}{2^{v(l-k+1)}}\right) \\ &= -(v - \lg(q_C))(l-k+1) + l.\end{aligned}$$

Thus if $n < (v - \lg(q_C))(l-k+1) - l + 1$ we have $\Pr[\mathcal{E}_1] > \Pr[\mathcal{E}_2]$, what we wanted to prove.

$\square$

4 Conclusion

We proved that combiners for preimage resistance and second-preimage resistance with output (significantly) shorter than concatenation do not exist. Our results are similar to ones for collision resistance from [1] and [3]. In particular, we showed that one can not create a secure (k,l)-combiner (for some mentioned security notion) of l *arbitrary* hash functions, with output shorter than concatenation of $l-k+1$ candidates.

These results are, however, too strict in a point, that we want from a combiner to work for *all* l-tuples of (compatible) hash functions and that the algorithm P providing the proof of combiner's security must succeed with non-negligible probability on all possible inputs. Such a condition is not very practically relevant – one can think of creating a combiner, which combines only hash functions from some subset of all hash functions (e.g. set of functions computable in polynomial time). Another way how to weaken the restrictions on combiners is to allow the algorithm P to fail on negligible part of inputs. We leave the analysis of such "weaker" combiners for a future work.

References

1. D. Boneh and X. Boyen: *On the impossibility of efficiently combining collision resistant hash functions.* LNCS, 4117, 2006.
2. R. Canetti, R. Rivest, M. Sudan, L. Trevisan, S. Vadhan and H. Wee: *Amplifying collision resistance: a complexity-theoretic treatment.* LNCS, 4622, 2007.
3. K. Pietrzak: *Non-trivial black-box combiners for collision-resistant hash-functions don't exist.* LNCS, 4515, 2007.
4. P. Rogaway and T. Shrimpton: *Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance.* In Fast Software Encryption, LNCS, Springer, 3017, 2004, 371–388.
5. M. Rjaško: *Properties of cryptographic hash functions.* Mikulášska Kryptobesídka, 2008.