

# On Chosen Target Forced Prefix preimage resistance

Michal Rjaško

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

**Abstract.** In this paper we analyze the Chosen Target Forced Prefix (CTFP) preimage resistance security notion for hash functions firstly introduced by Kelsey and Kohno with the Nostradamus attack [4]. We give a formal definition of this property in hash function family settings and work out all the implications and separations between CTFP preimage resistance and other standard notions of hash function security (preimage resistance, collision resistance, etc.). This paper follows the work of Rogaway and Shrimpton [6], where they defined seven basic notions of hash function security and examined all the relationships among these notions. We also define a new property for security of hash function families – always CTFP preimage resistance, which guarantees CTFP security for all the hash functions in the family.

**Key words:** hash function, chosen target forced prefix preimage resistance, provable security

## 1 Introduction

This paper studies the security notion for hash functions called Chosen Target Forced Prefix (CTFP) preimage resistance firstly introduced by Kelsey and Kohno in [4]. The notion relates to the Nostradamus attack on Merkle-Damgård hash functions [4]. A hash function secure in the CTFP sense should be resistant against the Nostradamus attack. We give the formal definition of this notion in hash function family settings and work out all the relationships between CTFP preimage resistance and other security notions for hash function families (i.e. notions of preimage resistance, second-preimage resistance, collision resistance, unforgeability, pseudo-random function and pseudo-random oracle). We also define a new security property for hash function families – always Chosen Target Forced Prefix preimage resistance (aCTFP), which guarantees that a hash function family secure in aCTFP sense does not have a “weak” key, what is not the case of CTFP preimage resistance, which allows insecurity for a small number of keys. Of course we also work out all the implications or separations between aCTFP preimage resistance and CTFP preimage resistance or the other security notions.

We follow the work of Rogaway and Shrimpton [6], where they define seven basic notions of hash function’s security – notions of preimage resistance (Pre,

aPre, ePre), second-preimage resistance (Sec, aSec, eSec) and collision resistance (Coll). They also work out all the relationships among these notions. The letter “a” in the name of the notion (e.g. aPre) represents the word “always”, what means that such security notion guarantees the security for the whole key domain (i.e. all the particular hash functions in the family are secure). The letter “e” represents the word “everywhere”, what means that the hash function family is secure for the whole message space, e.g. for eSec there does not exist a message for which it is easy to find second-preimages. For more complete discussion about these security notions we refer to the work [6].

Besides the seven notions defined in the work Rogaway-Shrimpton [6] we study the relationships between CTFP (aCTFP) preimage resistance and unforgeability (MAC) security notion, pseudo-random function (Prf) or Pseudo-random oracle (Pro). The unforgeability notion is useful when a hash function family is used to construct message authentication codes. It guarantees that an adversary with oracle access to the hash function family can not guess the hash of any message without querying it. The pseudo-random function and pseudo-random oracle notions relates to the terms indistinguishability and indifferntiability. Hash function family secure in Prf sense is indistinguishable from the random oracle. Similarly a hash function family secure in Pro sense is indifferntiable from the random oracle. We note that the term indifferntiability was firstly introduced by Maurer, Renner and Holenstein [5]. The difference between indifferntiability and indistinguishability is in the fact, whether we assume that a hash function family has some public subsystems (e.g. a compression function) or not. We refer to the work [5] or [3] for more details.

*Organization.* We begin by presenting some basic notations and definitions. In the Section 3 we formally define the twelve security notions discussed above (Pre, aPre, ePre, Sec, aSec, eSec, Coll, MAC, Prf, Pro, CTFP, aCTFP). In the section 4 we firstly formally define the implication and separation between two security notions (4.1), then we prove the implications (4.2) and finally we prove the separations (4.3). The summary over all the relationships can be found in the Table 1.

	Pre	aPre	ePre	Sec	aSec	eSec	Coll	MAC	CTFP	aCTFP	Prf	Pro
CTFP	$\not\rightarrow$ 6 $\not\leftarrow$ 5	$\not\rightarrow$ 6 $\not\leftarrow$ 5	$\not\rightarrow$ 6 $\not\leftarrow$ 5	$\not\rightarrow$ 6 $\not\leftarrow$ 5	$\not\rightarrow$ 6 $\not\leftarrow$ 5	$\not\rightarrow$ 6 $\not\leftarrow$ 5	$\not\rightarrow$ 6 $\leftarrow$ 1	$\not\rightarrow$ 6 $\not\leftarrow$ 5	x	$\not\rightarrow$ 4 $\leftarrow$ 3	$\not\rightarrow$ 6 $\not\leftarrow$ 5	$\not\rightarrow$ 6 $\leftarrow$ 2
aCTFP	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\rightarrow$ 3 $\not\leftarrow$ 4	x	$\not\rightarrow$ 8 $\not\leftarrow$ 7	$\not\rightarrow$ 8 $\not\leftarrow$ 7

**Table 1.** Relationships among the definitions. Numbers behind the implication/separation are numbers of theorems, where the proof of the corresponding relation can be found.

## 2 Preliminaries

In the formal definitions of hash function security a hash function family is used instead of a hash function. The hash function family is a hash function parametrized by a key. Its is more universal object than a hash function and it enables us to formally define notions, which are hard to define in the settings when using only hash functions. For example it is hard to define collision resistance when we consider only hash functions, since collisions exist in every hash function (its domains is bigger than its range) and trivial adversary can win against any hash function – it just need to have hardwired a colliding pair (however it can be difficult to find such adversary in practice). In hash function family settings, such adversary would need to have hardwired a colliding pair for every key. Note that one can consider a popular hash function SHA1 to be a member of a hash function family, which key is the initialization vector for the SHA1 algorithm.

**Definition 1 (Hash function family).** *A hash function family is a function  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ , where  $\mathcal{K} = \{0, 1\}^k$ ,  $\mathcal{Y} = \{0, 1\}^y$  for some integers  $k, y > 0$  and  $\mathcal{M} = \{0, 1\}^*$ . Set  $\mathcal{K}$  is called key space, number  $y$  is called hash length of  $H$ .*

We write  $M \stackrel{\$}{\leftarrow} \mathcal{S}$  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  $M_1 M_2$ . Bitwise complement of string  $M$  we write as  $\bar{M}$ . Empty string is denoted by  $\mu$ . Let  $Func(D, R)$  represent the set of all functions  $\rho : D \rightarrow R$  and let  $RF_{D,R}$  be a function chosen randomly from the set  $Func(D, R)$  (i.e.  $RF_{D,R} \stackrel{\$}{\leftarrow} Func(D, R)$ ). We sometimes write  $RF_{d,r}$  when  $D = \{0, 1\}^d$  and  $R = \{0, 1\}^r$ . By  $Prefix_n(M)$  we denote the  $n$ -bit prefix of message  $M$ , similarly by  $Suffix_n(M)$  we denote the  $n$ -bit suffix of  $M$ .

**Definition 2 (Adversary).** *An adversary is a random access machine (RAM) with any number of inputs (i.e. it can access  $i$ th bit of input  $j$  in unit time) that can toss a coin in unit time (i.e. it can choose a sample from the set  $\{0, 1\}$  in a unit time). Running time of an adversary  $A$  on some input is the average time needed to compute an output (relative to some fixed RAM model) plus the description size of  $A$  (relative to some fixed coding of RAMs).*

Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family. We denote by  $Time_{H,n}$  the running time of an algorithm  $P$  (i.e. some random access machine) computing  $H$  that has the best worst case running time over all inputs  $(K, M); K \in \mathcal{K}; M \in \mathcal{M}; |M| = n$ , that is, any other algorithm  $P'$  computing  $H$  has the worst case running time over all the inputs  $(K, M); K \in \mathcal{K}; M \in \mathcal{M}; |M| = n$  greater or equal to  $P$ 's. Informally speaking,  $Time_{H,n}$  is the time needed to compute  $H_K$  on an input of length  $n$ .

A function  $f : \mathbb{N} \rightarrow \mathbb{R}^+$  is negligible, if it descends faster than any polynomial powered to  $-1$ . The formal definition is following.

**Definition 3 (Negligible function).** A function  $f : \mathbb{N} \rightarrow \mathbb{R}^+$  is negligible, if for every constant  $c > 0$ , there exists an integer  $N_0 \in \mathbb{N}$ , such that for all integers  $n > N_0$  it holds

$$f(n) < \frac{1}{n^c}.$$

The term negligible we mostly use when considering an advantage of an adversary attacking a hash function family  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ . We consider some advantage as negligible when it is a function of  $k$  or  $y$  and this function is negligible.

### 3 Definitions of the security notions

*Standard notions.* Below we give the definitions of notions for hash function security. The first seven notions are those from Rogaway-Shrimpton [6] — notions of collision resistance, second-preimage resistance, preimage resistance and their always and everywhere versions. The pseudo-random function (Prf) and pseudo-random oracle (Pro) notions relates to the terms of indistinguishability and indifferentiability. If a hash function family is secure in Prf (Pro) sense, then it is indistinguishable (indifferentiable) from a random oracle. The Pro notion was firstly introduced by Coron, Dodis, Malinaud and Puniya [3] and then reused by Bellare and Ristenpart [2], [1]. The term indifferentiability was firstly introduced by Maurer, Renner and Holenstein [5]. Finally, the unforgeability notion (MAC) is the notion, which should a hash function family preserve if it is used to create message authentication codes.

We note, that the parameter  $[\lambda]$  is used in the following definitions to avoid random selection from an infinite set  $\mathcal{M}$  and also to bound the length of randomly selected messages. Also note, that from the definition of hash function family (Definition 1) we know, that  $\{0, 1\}^\lambda \subseteq \mathcal{M}$  for every positive integer  $\lambda$ , where  $\mathcal{M}$  is the message space of the hash function family.

Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family and let  $\lambda$  be a positive integer. Let  $A$  be an adversary. Then we define the following advantage measures:

$$\begin{aligned} \mathbf{Adv}_H^{\text{Pre}[\lambda]}(A) &= \Pr \left[ K \xleftarrow{\$} \mathcal{K}; M \xleftarrow{\$} \{0, 1\}^\lambda; Y \leftarrow H_K(M); \right. \\ &\quad \left. M' \leftarrow A(K, Y) : H_K(M') = Y \right] \\ \mathbf{Adv}_H^{\text{ePre}}(A) &= \max_{Y \in \mathcal{Y}} \left( \Pr \left[ K \xleftarrow{\$} \mathcal{K}; M \leftarrow A(K) : H_K(M) = Y \right] \right) \\ \mathbf{Adv}_H^{\text{aPre}[\lambda]}(A) &= \max_{K \in \mathcal{K}} \left( \Pr \left[ M \xleftarrow{\$} \{0, 1\}^\lambda; Y \leftarrow H_K(M); \right. \right. \\ &\quad \left. \left. M' \leftarrow A(Y) : H_K(M') = Y \right] \right) \\ \mathbf{Adv}_H^{\text{Sec}[\lambda]}(A) &= \Pr \left[ K \xleftarrow{\$} \mathcal{K}; M \xleftarrow{\$} \{0, 1\}^\lambda; M' \leftarrow A(K, M) : \right. \\ &\quad \left. (M \neq M') \wedge (H_K(M) = H_K(M')) \right] \end{aligned}$$

$$\begin{aligned}
\mathbf{Adv}_H^{\text{eSec}[\lambda]}(A) &= \max_{M \in \{0,1\}^\lambda} \left( \Pr \left[ K \xleftarrow{\$} \mathcal{K}; M' \leftarrow A(K) : \right. \right. \\
&\quad \left. \left. (M \neq M') \wedge (H_K(M) = H_K(M')) \right] \right) \\
\mathbf{Adv}_H^{\text{aSec}[\lambda]}(A) &= \max_{K \in \mathcal{K}} \left( \Pr \left[ M \xleftarrow{\$} \{0,1\}^\lambda; M' \leftarrow A(M) : \right. \right. \\
&\quad \left. \left. (M \neq M') \wedge (H_K(M) = H_K(M')) \right] \right) \\
\mathbf{Adv}_H^{\text{Coll}}(A) &= \Pr \left[ K \xleftarrow{\$} \mathcal{K}; (M, M') \leftarrow A(K) : \right. \\
&\quad \left. (M \neq M') \wedge (H_K(M) = H_K(M')) \right] \\
\mathbf{Adv}_H^{\text{Prf}}(A) &= \left| \Pr \left[ K \xleftarrow{\$} \mathcal{K}; 1 \leftarrow A^{H_K(\cdot)} \right] - \Pr \left[ \mathcal{F} \xleftarrow{\$} \text{Func}(\mathcal{M}, \mathcal{Y}); 1 \leftarrow A^{\mathcal{F}} \right] \right| \\
\mathbf{Adv}_H^{\text{MAC}}(A) &= \Pr \left[ K \xleftarrow{\$} \mathcal{K}; (M, Y) \leftarrow A^{H_K} : H_K(M) = Y \wedge M \text{ not queried} \right]
\end{aligned}$$

We say that  $H$  is  $(t, L, \varepsilon)$ -xxx for  $\text{xxx} \in \{\text{Pre}, \text{aPre}, \text{Sec}, \text{eSec}, \text{aSec}\}$  if any adversary  $A$  running in time at most  $t$  and outputting messages of length less than or equal to  $L$  has advantage  $\mathbf{Adv}_H^{\text{xxx}[\lambda]}(A) \leq \varepsilon$  for all  $\lambda$ . We say that  $H$  is  $(t, L, \varepsilon)$ -yyy for  $\text{yyy} \in \{\text{ePre}, \text{Coll}\}$ , if any adversary  $A$  running in time at most  $t$  and outputting messages of length less than or equal to  $L$  has advantage  $\mathbf{Adv}_H^{\text{yyy}}(A) \leq \varepsilon$ . We say that  $H$  is  $(t, q, L, \varepsilon)$ -zzz for  $\text{zzz} \in \{\text{MAC}, \text{Prf}\}$ , if any adversary  $A$  running in time at most  $t$ , making at most  $q$  queries to its oracle each of length at most  $L$  has advantage  $\mathbf{Adv}_H^{\text{zzz}}(A) \leq \varepsilon$ .

The pseudo-random oracle security notion requires a hash function family  $H$  to be build from some small ideal compression function  $f : \{0,1\}^{y+d} \rightarrow \mathcal{Y}; d > 0$  and an algorithm computing  $H$  has oracle access to  $f$  (we say that  $H$  extends the domain of  $f$ , i.e. the algorithm  $H$  is a domain extension transform). Therefore when comparing Pro with other notions, all the adversaries need to have an oracle access to  $f$ , since otherwise they would not be able to compute a hash value of  $H^f$  for an arbitrary key (for example it would make impossible for an adversary attacking in Prf sense to perform a brute force key finding attack).

Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family. Let  $A$  be an adversary,  $f = RF_{y+d,y}$  for some integer  $d > 0$  ( $f$  represents an ideal compression function) and let  $\mathcal{S}$  be a simulator (the simulator  $\mathcal{S}$  is an algorithm (i.e. a RAM), which simulates  $f$  to make distinguishing more difficult, for more details we refer to [3]). Then we define the following advantage measure:

$$\begin{aligned}
\mathbf{Adv}_{H,f,\mathcal{S}}^{\text{Pro}}(A) &= \left| \Pr \left[ K \xleftarrow{\$} \mathcal{K}; 1 \leftarrow A^{H_K^f(\cdot), f(\cdot)}(K) \right] - \right. \\
&\quad \left. \Pr \left[ K \xleftarrow{\$} \mathcal{K}; \mathcal{F} \xleftarrow{\$} \text{Func}(\mathcal{M}, \mathcal{Y}); 1 \leftarrow A^{\mathcal{F}(\cdot), \mathcal{S}^{\mathcal{F}}(K, \cdot)}(K) \right] \right|
\end{aligned}$$

We say that  $H$  is  $(t_A, t_S, q_1, q_2, L, \varepsilon)$ -Pro if for any adversary  $A$  running in time at most  $t_A$  and making at most  $q_1$  ( $q_2$ ) queries to its first (second) oracle each of length less than or equal to  $L$ , there exists a simulator  $\mathcal{S}$  running in time  $t_S$  such that the advantage  $\mathbf{Adv}_{H,f,\mathcal{S}}^{\text{Pro}}(A) \leq \varepsilon$ .

*CTFP preimage resistance.* In 2006, Kelsey and Kohno [4] proposed a new type of attack on Merkle-Damgård hash functions called the herding attack (or Nostradamus Attack). They introduced a new security property that a hash function should have – Chosen Target Forced Prefix (CTFP) preimage resistance and showed that Merkle-Damgård hash functions do not guarantee the same security as a random oracle does with respect to this property.

What the Chosen Target Forced Prefix preimage resistance security notion ensures can be illustrated on the following example from [4]:

One day in early 2006, the following ad appears in a news:

I, Nostradamus, hereby provide the MD5 hash  $Y$  of many important predictions about the future, including the closing prices of all stocks in the S&P500 as of the last business day of 2006.

Few weeks after the last business day of 2006, Nostradamus publishes a message  $M$  containing in its first block precise closing prices of the S&P500 stocks. The message then continues with many uncertain predictions which haven't come true yet.

The MD5 hash  $Y$  that Nostradamus firstly provides represents the *chosen target* part from the name of the CTFP security notion and the precise closing prices of the S&P500 stocks represent the *forced prefix*. The question is whether Nostradamus could cheat about his predictive capabilities.

Now we formally define the CTFP preimage resistance security notion adapted to hash function family settings.

Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family,  $\lambda$  be a positive integer and let  $A$  be an adversary. Then we define the following advantage measure:

$$\mathbf{Adv}_H^{\text{CTFP}[\lambda]}(A) = \Pr \left[ K \xleftarrow{\$} \mathcal{K}; (Y, S) \leftarrow A(K); P \xleftarrow{\$} \{0, 1\}^\lambda; \right. \\ \left. M \leftarrow A(P, S) : H_K(P||M) = Y \right]$$

We say that  $H$  is  $(t, L, \varepsilon)$ -CTFP if any adversary  $A$  running in time at most  $t$  and outputting messages of length less than or equal to  $L$  has advantage  $\mathbf{Adv}_H^{\text{CTFP}[\lambda]}(A) \leq \varepsilon$  for all  $\lambda$ .

The variable  $S$  in the definition is adversary's state. It is a string of an arbitrary length, where  $A$  can store some information (i.e. its state) for the second stage. The image  $Y$  which  $A$  chooses in the first stage corresponds to *chosen target* from the name of the security notion (i.e. the hash, which Nostradamus provides). Similarly,  $P$  corresponds to the *forced prefix*, that is the precise closing prices of the S&P500 stocks from the example above.

One can see that if we maximize the advantage above over all prefixes  $P$ , i.e. to define everywhere version of CTFP, then a trivial adversary returning  $H_K(P_0||M)$  in the first step and  $M$  in the second step would prevail. On the

other hand we can maximize the advantage over all keys  $\mathcal{K}$  to get an always chosen target forced prefix preimage resistance security notion.

Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family,  $\lambda$  be a positive integer and let  $A$  be an adversary. Then we define the following advantage measure:

$$\mathbf{Adv}_H^{\text{aCTFP}[\lambda]}(A) = \max_{K \in \mathcal{K}} \left( \Pr \left[ (Y, S) \leftarrow A; P \xleftarrow{\$} \{0, 1\}^\lambda; \right. \right. \\ \left. \left. M \leftarrow A(P, S) : H_K(P||M) = Y \right] \right)$$

We say that  $H$  is  $(t, L, \varepsilon)$ -aCTFP if any adversary  $A$  running in time at most  $t$  and outputting messages of length less than or equal to  $L$  has advantage  $\mathbf{Adv}_H^{\text{aCTFP}[\lambda]}(A) \leq \varepsilon$  for all  $\lambda$ .

*Equivalent two stage adversaries.* We note that for the definitions of advantages where we maximize over some quantity (keys or messages), there exists equivalent definition where the adversary performs the attack in two stages – in the first step it chooses the specific value (key or message), then the random choice is made by the environment and in the second phase the adversary continues with the attack with given that randomly selected values. These “two stage” definitions of the security notions are more demonstrative and they are more suitable for our proofs.

$$\begin{aligned} \mathbf{Adv}_H^{\text{aPre}[\lambda]}(A) &= \Pr \left[ (K, S) \leftarrow A; M \xleftarrow{\$} \{0, 1\}^\lambda; Y \leftarrow H_K(M); M' \leftarrow A(Y, S) : \right. \\ &\quad \left. H_K(M') = H_K(M) \right] \\ \mathbf{Adv}_H^{\text{ePre}}(A) &= \Pr \left[ (Y, S) \leftarrow A; K \xleftarrow{\$} \mathcal{K}; M' \leftarrow A(K, S) : H_K(M') = Y \right] \\ \mathbf{Adv}_H^{\text{aSec}[\lambda]}(A) &= \Pr \left[ (K, S) \leftarrow A; M \xleftarrow{\$} \{0, 1\}^\lambda; M' \leftarrow A(M, S) : \right. \\ &\quad \left. (M \neq M') \wedge (H_K(M) = H_K(M')) \right] \\ \mathbf{Adv}_H^{\text{eSec}}(A) &= \Pr \left[ (M, S) \leftarrow A; K \xleftarrow{\$} \mathcal{K}; M' \leftarrow A(K, S) : \right. \\ &\quad \left. (M \neq M') \wedge (H_K(M) = H_K(M')) \right] \\ \mathbf{Adv}_H^{\text{aCTFP}[\lambda]}(A) &= \Pr \left[ (Y, K, S) \leftarrow A; P \xleftarrow{\$} \{0, 1\}^\lambda; M \leftarrow A(P, S) : H_K(P||M) = Y \right] \end{aligned}$$

For the proof of equivalence between “one-stage” and “two-stage” definitions, we refer to the work of Rogaway-Shrimpton [6]. The proof is quite easy and straightforward.

## 4 Relationships

### 4.1 Implication and Separation

In this section we provide definitions for implication and separation between the security notions defined above. Intuitively, a security notion  $xxx$  implies a security notion  $yyy$ , when for all hash function families  $H$  holds, that if  $H$  is secure in  $xxx$  sense, then so it is in  $yyy$  sense. Similarly, a security notion  $xxx$  non-implies security notion  $yyy$ , if there exists a hash function family  $H$  secure in  $xxx$  sense, but insecure in  $yyy$  sense. However, in most situations it is very hard to find an unconditionally  $xxx$  secure hash function family, thus when proving the separation we rather assume the existence of some  $xxx$  secure hash function family  $H$ , from which we construct a hash function family  $H'$  also secure in  $xxx$  sense, but insecure in  $yyy$  sense. For briefer presentation, let  $Atks$  temporarily denote the set  $\{\text{Pre}, \text{aPre}, \text{ePre}, \text{Sec}, \text{aSec}, \text{eSec}, \text{Coll}, \text{CTFP}, \text{aCTFP}, \text{MAC}, \text{Prf}, \text{Pro}\}$ .

We consider a hash function family  $H$  to be secure in some sense (Prf, MAC, Sec,...), if any polynomial adversary has negligible advantage (with respect to  $k$  and  $y$ ) against  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  in that sense. In the Pro case, we consider a hash function family  $H$  to be secure in Pro sense when for any polynomial adversary  $A$  there exists a simulator  $\mathcal{S}$  running in a polynomial time such that the advantage  $\text{Adv}_{H,\mathcal{S},f}^{\text{Pro}}(A)$  is negligible for  $f = RF_{y+d,y}$ . Polynomial adversary runs in a time that is a polynomial of  $k$ ,  $y$  and  $l$ , where  $l$  is the length of its input (if it has some).

The formal definition of implication between security notions arises straightly from the intuition above. We note that in the following definition, and later,  $[\cdot]$  is a placeholder which is either  $[\lambda]$  (for Pre, aPre, Sec, aSec, eSec, CTFP, aCTFP) or empty (for ePre, Coll, Prf, Pro). We also write  $\text{Adv}_{H,\cdot,\cdot}^{\text{xxx}[\cdot]}$ , which is either  $\text{Adv}_{H,\mathcal{S},f}^{\text{xxx}[\cdot]}$  (when  $xxx$  is Pro), or  $\text{Adv}_{H,f}^{\text{xxx}[\cdot]}$  (when  $xxx$  is something else than Pro, but we are comparing it to Pro (e.g.  $yyy$  is Pro)), or  $\text{Adv}_H^{\text{xxx}[\cdot]}$  (when both security notions  $xxx$  and  $yyy$  are different from Pro).

**Definition 4 ( $xxx \rightarrow yyy$ ).** Let  $\mathcal{K} = \{0,1\}^k$ ,  $\mathcal{M} = \{0,1\}^*$  and  $\mathcal{Y} = \{0,1\}^y$  for some fixed  $k$  and  $y$ , let  $\lambda$  be some fixed positive integer,  $f = RF_{y+d,y}$  and suppose, that  $xxx, yyy \in Atks$ . We say that the definition of security notion  $xxx$  implies security notion  $yyy$  (shortly  $xxx \rightarrow yyy$ ), if for any hash function family  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  and any adversary  $A$  running in polynomial time  $t$ , with non-negligible advantage (with respect to  $k$ ,  $y$  or  $\lambda$ ) in  $yyy$  sense (for all polynomial simulators  $\mathcal{S}$  if  $yyy$  is Pro), there exists an adversary  $A'$  such that  $A'$  runs in polynomial time  $t'$  and has non-negligible advantage in  $xxx$  sense (for all polynomial simulators  $\mathcal{S}$  if  $xxx$  is Pro).

Similarly we can formally define a separation between two security notions.

**Definition 5** ( $\text{xxx} \not\rightarrow \text{yyy}$ ). Let  $\mathcal{K} = \{0, 1\}^k$ ,  $\mathcal{M} = \{0, 1\}^*$  and  $\mathcal{Y} = \{0, 1\}^y$  for some fixed  $k$  and  $y$ , let  $\lambda$  be some fixed positive integer,  $f = RF_{y+d,y}$  and suppose that  $\text{xxx}, \text{yyy} \in \text{Atks}$ . We say that the definition of security notion  $\text{xxx}$  non-implies security notion  $\text{yyy}$  (shortly  $\text{xxx} \not\rightarrow \text{yyy}$ ), if for any hash function family  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  there exists a hash function family  $H' : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ , such that if  $\text{Adv}_{H', \cdot, \cdot}^{\text{xxx}[\cdot]}(t)$  is non-negligible (for all polynomial simulators  $\mathcal{S}$  if  $\text{xxx}$  is Pro), then so is  $\text{Adv}_{H, \cdot, \cdot}^{\text{xxx}[\cdot]}(t')$  (for all polynomial simulators  $\mathcal{S}$  if  $\text{xxx}$  is Pro), and  $\text{Adv}_{H', \cdot, \cdot}^{\text{yyy}[\cdot]}(t')$  is non-negligible too (for all polynomial simulators  $\mathcal{S}$  if  $\text{yyy}$  is Pro), where  $t$  and  $t'$  are some polynomial running times.

We note that the definitions of implication and separation in Rogaway and Shrimpton [6] are different from the definitions above. We needed to modify the definitions in [6] since they do not apply in the following settings. Consider that we have an adversary  $A$  attacking in CTFP sense, from which we construct an adversary  $B$  attacking in Coll sense. The adversary  $B$  simulates the second stage of the adversary  $A$  twice and succeeds when  $A$  wins in both simulations. Thus  $B$ 's advantage in Coll sense is square of  $A$ 's advantage in CTFP sense. Intuitively, one should then consider, that security notion Coll implies security notion CTFP, what really holds with respect to our definition above. On the other, the definition from [6] restricts the running time of  $B$  to be only constantly greater than  $A$ 's running time (what is not the our case), moreover the advantage of  $B$  in Coll sense is restricted (with respect to the definition in [6]) to be only constantly smaller than  $A$ 's advantage in CTFP sense.

Thus our definitions of implication and separation above are more general than ones in [6]. One can easily see that the Rogaway-Shrimpton's definitions implies our definitions above, that is, if  $\text{xxx}$  implies  $\text{yyy}$  with respect to the definition of Rogaway and Shrimpton, then  $\text{xxx}$  implies  $\text{yyy}$  also with respect to our definition.

## 4.2 Implications

Here we investigate which of the security notions imply CTFP or aCTFP notions and vice-versa. We begin by proving that collision resistance implies chosen target forced prefix preimage resistance. All the implications and separations are summarized in the Table 1.

**Theorem 1.**  $\text{Coll} \rightarrow \text{CTFP}$

*Proof.* Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family and  $A$  an adversary running in polynomial time and attacking  $H$  in CTFP sense with non-negligible advantage  $\varepsilon$ . Consider the following adversary  $B$  performing attack in Coll sense against  $H$ :

<b>Adversary</b> $B(K)$	
1	$(Y, S) \leftarrow A(K)$
2	<b>let</b> $P_1 \xleftarrow{\$} \{0, 1\}^\lambda$
3	$M_1 \leftarrow A(P_1, S)$
4	<b>let</b> $P_2 \xleftarrow{\$} (\{0, 1\}^\lambda - \{P_1\})$
5	$M_2 \leftarrow A(P_2, S)$
6	<b>return</b> $(P_1    M_1, P_2    M_2)$

The running time of  $B$  is only polynomially slower than the running time of  $A$ . The advantage of  $B$  against  $H$  is given by the probability, that  $A$  succeeds in the both simulations, that is it returns suffixes  $M_1$  and  $M_2$  that with prefixes  $P_1$  and  $P_2$  make hash  $Y$  (i.e.  $H(P_1 || M_1) = H(P_2 || M_2) = Y$ ). View of the adversary  $A$  in the first simulation on line 3 is the same as in standard CTFP attack, thus in this case  $A$  wins with probability  $\varepsilon$ . However in the second simulation, one can note that the prefix  $P_1$  can not be chosen. Consider that the prefix  $P_1$  is the prefix for which the adversary  $A$  wins with probability 1 (what is actually the worst case). Then the probability that  $A$  wins in the second simulation (line 5) is at least  $\varepsilon - 1/2^\lambda$ . Thus  $B$ 's advantage in Coll sense is  $\varepsilon(\varepsilon - 1/2^\lambda) = \varepsilon^2 - \varepsilon 2^{-\lambda}$ , what is non-negligible, if  $\varepsilon$  is non-negligible.

**Theorem 2.**  $Pro \rightarrow CTFP$

*Proof.* Let  $f = RF_{y+d,y}$  be a random function,  $H^f : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be an arbitrary hash function family and let  $\lambda$  be a positive integer. Consider an adversary  $A$  running in polynomial time  $t$  and attacking  $H$  in CTFP sense with non-negligible advantage  $\mathbf{Adv}_{H,f}^{CTFP[\lambda]}(A) = \varepsilon$ . We construct the following adversary  $B$  attacking  $H$  in Pro sense:

<b>Adversary</b> $B^{f_1, f_2}(K)$	
1	$(Y, S) \leftarrow A^{f_2}(K)$
2	<b>let</b> $P \xleftarrow{\$} \{0, 1\}^\lambda$
3	$M \leftarrow A^{f_2}(P, S)$
4	<b>if</b> $f_1(P    M) = Y$ <b>then return</b> 1
5	<b>otherwise return</b> 0

It is clear that there exists a polynomial  $p(k, y, \lambda)$  such that the running time of  $B$  is  $p(k, y, \lambda) \cdot t$ . When  $B$ 's oracles are  $H$  and  $f$ , then it returns 1 with a probability equal to the probability that  $A$  wins in the simulation on line 3, i.e.  $B$ 's chance to win is  $\varepsilon$ . Now consider the case, when  $B$ 's oracles are a randomly chosen function  $\mathcal{F}$  and some polynomial simulator  $\mathcal{S}$ . Total number of queries made by the simulator  $\mathcal{S}$  to its oracle  $\mathcal{F}$  can be at most  $t_{\mathcal{S}} \cdot t$ , where  $t_{\mathcal{S}}$  is running time of the simulator  $\mathcal{S}$ . Because  $\mathcal{F}$  is random, the probability of finding  $M$  such that  $\mathcal{F}(P || M) = Y$  is at most  $(t_{\mathcal{S}} \cdot t) / |\mathcal{Y}|$ . Since both  $\mathcal{S}$  and  $A$  run in the polynomial time, then  $(t_{\mathcal{S}} \cdot t) / |\mathcal{Y}|$  is negligible, what means that  $A^{\mathcal{S}^{\mathcal{F}}}$  does not make enough queries to find the valid message  $M$  such that  $\mathcal{F}(P || M) = Y$  with non-negligible probability. Therefore  $B$  has non-negligible advantage against  $H$  in Pro sense, what completes the proof.

**Theorem 3.**  $aCTFP \rightarrow CTFP$

*Proof.* Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family,  $\lambda$  be a positive integer and let  $A$  be an adversary running in polynomial time  $t$  and performing attack against  $H$  with non-negligible advantage  $\mathbf{Adv}_H^{CTFP[\lambda]}(A) = \varepsilon$ . Notice, that there must exist a key  $K_0 \in \mathcal{K}$  which if chosen by the environment, then  $A$ 's chance to win is at least  $\varepsilon$  (otherwise  $A$ 's advantage would be smaller than  $\varepsilon$ ). Now consider the following adversary  $B$  performing attack in aCTFP sense:

**Adversary  $B$**   
 [1<sup>st</sup> stage]  
 $(Y, S) \leftarrow A(K_0)$   
**return**  $(Y, K_0, S)$   
 [2<sup>nd</sup> stage with input  $(P, S)$ ]  
 $M \leftarrow A(P, S)$   
**return**  $M$

Clearly  $B$  is only polynomially slower than  $A$ . From the assumption that  $K_0$  is the key, where  $A$ 's chance to win against  $H$  in CTFP sense is at least  $\varepsilon$  we have, that  $B$ 's chance to win against  $H$  in aCTFP sense is at least  $\varepsilon$  too, what means that  $B$ 's advantage in aCTFP sense is non-negligible.

### 4.3 Separations

In this section we investigate the separations between CTFP, aCTFP and the other notions. We provide all the constructions used in proofs of the separations in the Figure 1. Let's begin by proving that aCTFP security notion does not imply CTFP.

**Theorem 4.**  $CTFP \not\rightarrow aCTFP$

*Proof.* Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family,  $\lambda$  be a positive integer and consider the following construction from the Figure 1:

$$H_K^{(5)}(M) = \begin{cases} \text{Suffix}_y(M) & \text{if } K = K_0 \\ H_K(M) & \text{otherwise} \end{cases}$$

Let  $A$  be an adversary attacking  $H^{(5)}$  in CTFP sense with non-negligible advantage. We investigate  $A$ 's advantage against  $H$  in CTFP sense. If a key  $K \neq K_0$  is chosen by the environment, then  $A$ 's chance to win against  $H$  is the same as against  $H^{(5)}$ . On the other hand, if the key  $K = K_0$  is chosen, then in the worst case  $A$  can win against  $H$  with probability 0. However the probability that the key  $K_0$  is chosen is  $1/|\mathcal{K}|$ . Thus

$$\mathbf{Adv}_H^{CTFP[\lambda]}(A) \geq \mathbf{Adv}_{H^{(5)}}^{CTFP[\lambda]}(A) - \frac{1}{|\mathcal{K}|}.$$

Thus  $A$ 's advantage against  $H$  non-negligible too.

$$\begin{aligned}
H_K^{(1)}(M) &= \begin{cases} K[1 \dots \min\{k, y\}] || 0^{\max\{y-k, 0\}} & \text{if } \text{Suffix}_k(M) = K \\ H_K(M) & \text{otherwise} \end{cases} \\
H_K^{(2)}(M) &= \begin{cases} M & \text{if } |M| = y \\ H_K(M) & \text{otherwise} \end{cases} \\
H_K^{(3)}(M) &= H_K(M[1 \dots |M| - 1] || 0) \\
H_K^{(4)}(M) &= \begin{cases} 0^y & \text{if } M = 0 \\ H_K(M) & \text{otherwise} \end{cases} \\
H_K^{(5)}(M) &= \begin{cases} \text{Suffix}_y(M) & \text{if } K = K_0 \\ H_K(M) & \text{otherwise} \end{cases}
\end{aligned}$$

**Fig. 1.** Constructions of hash function families used in proofs of separations. The constructions  $H^{(3)}$  and  $H^{(4)}$  are from [6].

Now consider the following adversary  $B$  attacking  $H^{(5)}$  in aCTFP sense:

**Adversary  $B$**   
 $[1^{st} \text{ stage}]$   
**return**  $(0^y, K_0, K_0)$   
 $[2^{nd} \text{ stage with input } (P, S)]$   
**return**  $0^y$

From the definition of  $H^{(5)}$  we can see, that  $B$ 's advantage in aCTFP sense against  $H^{(5)}$  is 1. For completeness we note that  $B$  runs in a polynomial time.

**Theorem 5.**

- (1)  $Pre, aPre \not\rightarrow CTFP$
- (2)  $Sec, aSec \not\rightarrow CTFP$
- (3)  $ePre, eSec \not\rightarrow CTFP$
- (4)  $MAC \not\rightarrow CTFP$
- (5)  $Prf \not\rightarrow CTFP$

*Proof.* Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  be a hash function family,  $\lambda$  a positive integer and consider the construction  $H^{(1)}$  from the Figure 1:

$$H_K^{(1)}(M) = \begin{cases} K[1 \dots \min\{k, y\}] || 0^{\max\{y-k, 0\}} & \text{if } \text{Suffix}_k(M) = K \\ H_K(M) & \text{otherwise} \end{cases}$$

First, we show that there exists an adversary performing attack in CTFP sense with non-negligible advantage against  $H^{(1)}$ . For that purpose consider the following adversary  $A$ :

**Adversary  $A$**   
 [1<sup>st</sup> stage with input  $K$ ]  
   **return**  $(K[1 \dots \min\{y, k\}] || 0^{\max\{y-k, 0\}}, K)$   
 [2<sup>nd</sup> stage with input  $(P \in \{0, 1\}^\lambda, S)$ ]  
   **let**  $K := S$   
   **return**  $K$

Clearly  $A$  runs in a polynomial time of  $k$ ,  $y$  and the length of its input. From the definition of  $H^{(1)}$  we can see, that  $\mathbf{Adv}_{H^{(1)}}^{\text{CTFP}[\lambda]} = 1$ , what completes the first part of the proof.

Now we need to show that for any adversary  $B$  attacking  $H^{(1)}$  in xxx sense, there exists an adversary attacking  $H$  in xxx sense, where  $\text{xxx} \in \{\text{Pre}, \text{aPre}, \text{ePre}, \text{Sec}, \text{aSec}, \text{eSec}, \text{MAC}, \text{Prf}\}$ . Let  $B_{\text{xxx}}$  be an adversary attacking  $H^{(1)}$  in xxx sense with non-negligible advantage  $\mathbf{Adv}_{H^{(1)}}^{\text{xxx}[\cdot]}(B_{\text{xxx}}) = \varepsilon$ . We investigate  $B_{\text{xxx}}$ 's advantage against  $H$ .

In the case of preimage resistance and second preimage resistance and their always versions (i.e. the part (1) and (2) of the Theorem), consider the message  $M$  chosen randomly by the environment and let  $K$  be a key either chosen by the environment (Pre, Sec) or by the adversary in the first stage (aPre, aSec). If a suffix of  $M$  is different from  $K$ , then  $B_{\text{xxx}}$ 's chance to win against  $H$  is the same as against  $H^{(1)}$  (since in this case  $H^{(1)}$  looks exactly like  $H$ ). On the other hand if  $K$  is the suffix of  $M$ , then  $B_{\text{xxx}}$  can win against  $H$  in the worst case with probability 0. Thus the following holds:

$$\mathbf{Adv}_H^{\text{xxx}[\lambda]}(B_{\text{xxx}}) \geq \mathbf{Adv}_{H^{(1)}}^{\text{xxx}[\lambda]}(B_{\text{xxx}}) - \Pr[\text{message } M \text{ with suffix } K \text{ is chosen}].$$

If  $|M| \geq k$ , then the probability that  $K$  is the suffix of  $M$  is  $1/|K|$ . If  $|M| > k$  then  $M$  can not have the suffix  $K$ . Therefore

$$\Pr[\text{message } M \text{ with suffix } K \text{ is chosen}] \leq \frac{1}{|K|},$$

what is negligible. Thus  $B_{\text{xxx}}$  has non-negligible advantage against  $H$ , what completes the proof for the parts (1) and (2) of the Theorem.

The proof of the part (3) is very similar to the one above. In the case of ePre and eSec, a message  $M$  is not chosen randomly by the environment, but in the eSec case it is chosen by the adversary in the first stage. In the ePre case, the adversary chooses an image  $Y$ . After the adversary makes the selection (i.e. first stage ends), a key  $K$  is chosen by the environment. Now consider only the eSec case. If the key  $K$  is not suffix of  $M$ , then  $B_{\text{eSec}}$ 's chance to win against  $H$  is the same as against  $H^{(1)}$ . On the other hand, the probability that  $K$  is suffix of  $M$  is negligible (at most  $1/|K|$ ), thus we have the similar situation as we had above, in particular:

$$\begin{aligned} \mathbf{Adv}_H^{\text{eSec}[\lambda]}(B_{\text{eSec}}) &\geq \mathbf{Adv}_{H^{(1)}}^{\text{eSec}[\lambda]}(B_{\text{eSec}}) - \Pr[K \text{ is suffix of } M] \\ &\geq \mathbf{Adv}_{H^{(1)}}^{\text{eSec}[\lambda]}(B_{\text{eSec}}) - \frac{1}{|K|}. \end{aligned}$$

And thus the  $B_{\text{eSec}}$  advantage against  $H$  is non-negligible. For the ePre case suppose that an image  $Y$  is chosen by the adversary. We know that if the key  $K$  is chosen by the environment, such that  $Y = K[1 \dots \min\{k, y\}]||0^{\max\{y-k, 0\}}$ , then in the worst case  $B_{\text{ePre}}$  wins against  $H$  with probability 0, but when  $Y \neq K[1 \dots \min\{k, y\}]||0^{\max\{y-k, 0\}}$ , then the chance of  $B_{\text{ePre}}$  to win against  $H$  is the same as against  $H^{(1)}$ . However, the probability that the key  $K$  is chosen, where  $Y = K[1 \dots \min\{k, y\}]||0^{\max\{y-k, 0\}}$  is negligible ( $1/\min\{|\mathcal{K}|, |\mathcal{Y}|\}$ ), thus the  $B_{\text{ePre}}$ 's advantage against  $H$  is non-negligible.

The idea behind the proof of the parts (4) and (5) of the Theorem is, that an adversary attacking in MAC or Prf sense does not have access to the key  $K$  chosen randomly by the environment. Thus  $B_{\text{xxx}}$  for  $\text{xxx} \in \{\text{MAC}, \text{Prf}\}$  can notice some difference when attacking  $H$  from the case when attacking  $H^{(1)}$  only when a message  $M$  with suffix  $K$  is queried by the adversary. Let  $q$  be the maximum number of queries  $B_{\text{xxx}}$  can make to its oracle. The following holds:

$$\mathbf{Adv}_H^{\text{xxx}}(B_{\text{xxx}}) \geq \mathbf{Adv}_{H^{(1)}}^{\text{xxx}}(B_{\text{xxx}}) - \frac{q}{|\mathcal{K}|}.$$

And since  $B_{\text{xxx}}$  runs in a polynomial time, then  $\frac{q}{|\mathcal{K}|}$  is negligible. Thus the proof is complete.

**Theorem 6.**

- (1)  $\text{CTFP} \not\vdash \text{Pre}, \text{aPre}, \text{ePre}$
- (2)  $\text{CTFP} \not\vdash \text{Sec}, \text{aSec}, \text{eSec}$
- (3)  $\text{CTFP} \not\vdash \text{Coll}$
- (4)  $\text{CTFP} \not\vdash \text{MAC}$
- (5)  $\text{CTFP} \not\vdash \text{Prf}$
- (6)  $\text{CTFP} \not\vdash \text{Pro}$

*Proof.* Let  $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$  and  $\lambda$  be a positive integer.

- (1) Consider a construction  $H^{(2)}$  from the Figure 1.

$$H_K^{(2)}(M) = \begin{cases} M & \text{if } |M| = y \\ H_K(M) & \text{otherwise} \end{cases}$$

Let  $A$  be an adversary running in polynomial time and attacking  $H^{(6)}$  in CTFP sense with non-negligible advantage. Consider advantage of  $A$  when attacking  $H$ . Let  $Y$  be an image that  $A$  chooses in the first stage. If the prefix  $P$  chosen randomly by the environment is not prefix of  $Y$ , then  $A$ 's chance to win against  $H$  is the same as against  $H^{(6)}$ . On the other hand, if  $P$  is prefix of  $Y$ , then in the worst case  $A$  wins against  $H$  win probability 0. Thus

$$\begin{aligned} \mathbf{Adv}_H^{\text{CTFP}[\lambda]}(A) &\geq \mathbf{Adv}_{H^{(6)}}^{\text{CTFP}[\lambda]}(A) - \\ &\quad - \Pr[\text{forced prefix } P \text{ is prefix of chosen image } Y] \end{aligned}$$

The forced prefix  $P$  is uniformly selected from the set  $\{0, 1\}^\lambda$ , thus if  $\lambda \leq y$  then the probability of choosing  $P$  that is prefix of  $Y$  is  $1/2^\lambda$ , if  $\lambda > y$ , then such  $P$  does not exist. Therefore

$$\Pr[\text{forced prefix } P \text{ is prefix of chosen image } Y] \leq \frac{1}{2^\lambda},$$

what is negligible and thus  $A$  has non-negligible advantage against  $H$ .

For the second part of the proof, the construction  $H^{(6)}$  is clearly not Pre, aPre, ePre secure. The adversaries attacking in Pre and aPre sense simply copies its input (image  $Y$ ) to its output ( $H^{(6)}(Y) = Y$ ). The adversary attacking in ePre sense in the first stage chooses an arbitrary image  $Y$  and in the second stage it returns the same.

(2), (3) Consider the construction  $H^{(3)}$  from the Figure 1.

$$H_K^{(3)}(M) = H_K(M[1 \dots |M| - 1]||0)$$

Let  $A$  be an adversary attacking  $H^{(4)}$  in CTFP sense with non-negligible advantage and running in polynomial time. Consider the following adversary  $B$  performing attack in CTFP sense against  $H$ :

**Adversary  $B$**   
 [1<sup>st</sup> stage with input  $K$ ]  
 $Y \leftarrow A(K)$   
**return**  $(Y, K)$   
 [2<sup>nd</sup> stage with input  $(P, S)$ ]  
 $M \leftarrow A(P, S)$   
**let**  $K := S$   
**if**  $H_K(P||M) = Y$  **then return**  $M$   
**else let**  $b := M[|M|]$ ; **return**  $M[1 \dots |M| - 1]||\bar{b}$

Such adversary  $B$  runs in a time that is only polynomially slower than the running time of  $A$ . From the definition of  $H^{(3)}$  we can see, that if  $A$  wins against  $H^{(3)}$ , then  $B$  wins against  $H$ . Thus  $B$  has non-negligible advantage against  $H$  in CTFP sense.

The construction  $H^{(3)}$  is clearly not secure in Sec, aSec, eSec, Coll senses, since for every message  $M$  holds, that  $H_K^{(3)}(M) = H_K^{(3)}(M')$ , where  $M'$  is equal to  $M$  but with the last bit negated.

(4), (5), (6) Here we use the construction  $H^{(4)}$  from the Figure 1.

$$H_K^{(4)}(M) = \begin{cases} 0^y & \text{if } M = 0 \\ H_K(M) & \text{otherwise} \end{cases}$$

Let  $A$  be a polynomial adversary attacking  $H^{(4)}$  in CTFP sense with non-negligible advantage. Consider  $A$ 's advantage against  $H$ . The hash function family  $H^{(4)}$  differs from  $H$  only in the message  $M = 0$  for every key. Thus

$A$  attacking  $H$  can have different probability of success from the case when attacking  $H^{(4)}$  only when the forced prefix chosen randomly by the environment is prefix of the message  $M = 0$ . However the probability that randomly chosen prefix  $P$  is prefix of the message  $M = 0$  is negligible. Thus  $A$  has non-negligible advantage against  $H$  too.

The hash function family  $H^{(4)}$  is clearly not MAC, Prf, Pro secure, since an adversary attacking  $H^{(4)}$  in MAC sense returns  $(0, 0^y)$  and wins with probability 1. Adversaries attacking in Prf and Pro senses checks by querying their (first) oracle  $f$ , whether  $f(0) = 0^y$  and if so, they return 1, otherwise 0. Such adversaries win against  $H^{(4)}$  in Prf and Pro senses with non-negligible advantage, since the probability that for a random function  $\mathcal{F}$  holds  $\mathcal{F}(0) = 0^y$  is negligible.

**Theorem 7.**

- (1)  $Pre, aPre \not\rightarrow aCTFP$
- (2)  $Sec, aSec \not\rightarrow aCTFP$
- (3)  $ePre, eSec \not\rightarrow aCTFP$
- (4)  $MAC \not\rightarrow aCTFP$
- (5)  $Prf \not\rightarrow aCTFP$
- (6)  $Coll \not\rightarrow aCTFP$

*Proof.* The separations (1), (2), (3), (4) and (5) hold, because the same separations hold for CTFP (Theorem 5) on the right side and because aCTFP implies CTFP (Theorem 3). For example, suppose that  $Pre \rightarrow aCTFP$ . However  $aCTFP \rightarrow CTFP$ , thus  $Pre \rightarrow CTFP$ , what is the contradiction with the Theorem 3. Therefore  $Pre \not\rightarrow aCTFP$ . Similarly we can prove the other separations.

For the separation (6) we use the construction  $H^{(5)}$

$$H_K^{(5)}(M) = \begin{cases} \text{Suffix}_y(M) & \text{if } K = K_0 \\ H_K(M) & \text{otherwise} \end{cases}$$

Let  $A$  be an polynomial adversary performing attack in Coll sense against  $H^{(5)}$  with non-negligible advantage. Consider  $A$ 's advantage against  $H$ . If a key  $K$  chosen by the environment is different from  $K_0$ , then  $A$ 's chance to win against  $H$  is the same as against  $H^{(5)}$ . However when the key  $K_0$  is chosen, then in the worst case  $A$  wins with the probability 0. Therefore

$$\mathbf{Adv}_H^{Coll}(A) \geq \mathbf{Adv}_{H^{(5)}}^{Coll}(A) - \frac{1}{|\mathcal{K}|},$$

what means, that  $A$  has non-negligible advantage against  $H$  too. The second part of the proof can be found in the proof of the Theorem 4 where we proved, that there exists an adversary attacking  $H^{(5)}$  in aCTFP sense with non-negligible advantage.

**Theorem 8.**

- (1)  $aCTFP \not\rightarrow Pre, aPre, ePre$

- (2)  $aCTFP \not\rightarrow Sec, aSec, eSec$
- (3)  $aCTFP \not\rightarrow Coll$
- (4)  $aCTFP \not\rightarrow MAC$
- (5)  $aCTFP \not\rightarrow Prf$
- (6)  $aCTFP \not\rightarrow Pro$

*Proof.* We do not provide the exact proof for this Theorem, since it is very similar to one of the Theorem 6. The security of the constructions  $H^{(2)}$ ,  $H^{(3)}$  and  $H^{(4)}$  does not depend on the selection of the key, thus these constructions are also aCTFP secure (if  $H$  is aCTFP secure). Thus the separations proved in the Theorem 6 hold also with the aCTFP notion on their left hand side. xxx

## 5 Conclusion

In this paper we analyzed the relationships between Chosen Target Forced Prefix preimage resistance notions and the other standard notions of hash function's security. We formally defined CTFP preimage resistance in the hash function family settings and also provided a new security property for hash function families – always Chosen Target Forced Prefix preimage resistance. The summary of the relationships can be found in the Table 1.

We showed that CTFP preimage resistance is independent from the most of the other standard security notions, except collision resistance and pseudo-random oracle, which implies CTFP. The aCTFP security notion implies CTFP and is independent from all of the other notions.

## References

1. M. Bellare and T. Ristenpart. Hash Functions in the Dedicated-Key Setting: Design Choices and MPP Transforms. In *International Colloquium on Automata, Languages, and Programming, LNCS vol. 4596*, pages 399–410. Springer, 2006.
2. M. Bellare and T. Ristenpart. Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In *Advances in Cryptology - ASIACRYPT 2006, LNCS vol. 4284*, pages 299–314. Springer, 2006.
3. J.S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-Damgård Revisited: How to Construct a Hash Function. In *Advances in Cryptology - CRYPTO 2005, LNCS vol. 3621*, pages 430–448. Springer, 2005.
4. J. Kelsey and T. Kohno. Herding Hash Functions and the Nostradamus Attack. In *Advances in Cryptology - EUROCRYPT 2006, LNCS vol. 4004*, pages 183–200. Springer, 2006.
5. U. Maurer, R. Renner, and C. Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In *Theory of Cryptography, LNCS vol. 2951*, pages 21–39. Springer, 2004.
6. P. Rogaway and T. Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In *Fast Software Encryption, LNCS vol. 3017*, pages 371–388. Springer, 2004.