ON CHOSEN TARGET FORCED PREFIX PREIMAGE RESISTANCE

MICHAL RJAŠKO

ABSTRACT. In this paper we analyze the Chosen Target Forced Prefix (CTFP) preimage resistance security notion for hash functions firstly introduced in [Kelsey, J.—Kohno, T.: *Herding hash functions and the Nostradamus attack*, in: Advances in Cryptology—EUROCRYPT '06, 25th Annual Internat. Conf. on the Theory and Appl. of Cryptographic Techniques (S. Vaudenay, ed.), St. Petersburg, Russia, 2006, Lecture Notes in Comput. Sci., Vol. 4004, Springer-Verlag, Berlin, 2006, pp. 183–200]. We give a formal definition of this property in hash function family settings and work out all the implications and separations between the CTFP preimage resistance and other standard notions of hash function security (preimage resistance, collision resistance, etc.). This paper follows the work of [Rogaway, P.—Shrimpton, T.: *Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance*, in: Fast Software Encryption, 11th International Workshop—FSE '04 (B. Roy et al., eds.), Delhi, India, 2004, Lecture Notes in Comput. Sci., Vol. 3017, Springer-Verlag, Berlin, 2004, pp. 371–388], where they define seven basic notions of hash function security and examine 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.

1. Introduction

This paper studies the security notion of cryptographic 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, which attacks Merkle-Damgård hash functions [4]. A hash function secure in the CTFP sense is resistant against the Nostradamus attack. We give a formal definition of this notion in the hash function family settings and work out all the relationships between CTFP preimage resistance and other security notions for hash

2010 Mathematics Subject Classification: 94A60, 60C05, 68U99.

Keywords: hash function, chosen target forced prefix preimage resistance, provable security. This paper was supported by VEGA grant number 1/0266/09 and by Comenius University grant number UK/365/2009.

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—always Chosen Target Forced Prefix preimage resistance (aCTFP), which guarantees security of a hash function family for all keys. This is not the case of CTFP preimage resistance, which allows insecurity for a small number of keys. Similarly to the CTFP case, we also work out all the implications and separations among aCTFP preimage resistance, CTFP preimage resistance and 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”, which 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”, which 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 of Rogaway—Shrimpton we also study the relationships between CTFP (aCTFP) preimage resistance and unforgeability (MAC), pseudo-random function (Prf) and pseudo-random oracle (Pro). The unforgeability 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 cannot guess the hash of any message without querying it. The pseudo-random function and pseudo-random oracle notions relate to the terms indistinguishability and indifferentiability. A hash function family secure in the Prf sense is indistinguishable from the random oracle. Similarly a hash function family secure in the Pro sense is indifferentiable from the random oracle. We note that the term indifferentiability was firstly introduced by Maurer, Renner and Holenstein [5]. The difference between indifferentiability and indistinguishability is in the fact, whether we assume that a hash function family has some publicly available components (e.g., a compression function). In the indistinguishability case, a potential adversary has access to hash function as a monolithic object and does not have any information about components the hash function is built from. On the other hand, indifferentiability framework assumes, that a hash function is built from components, which are accessible to anyone (such component is usually a compression function). In some cases, indifferentiability framework models real world attacks more accurately than indistinguishability. We refer to the works [3], [5] for more details.

This paper is extension of our previous work [7], where we analyzed all the relationships among the ten security notions (Pre, aPre, ePre, Sec, aSec, eSec, Coll, MAC, Prf and Pro). Thus, in this work we expand the table of all relationships with the two additional notions, CTFP and aCTFP.

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 formally define the implication and separation between two security notions (4.1), and then we prove the implications (4.2) and finally we prove the separations (4.3). The summary over all the relationships proved in this work can be found in the Table 1. For completeness, in the Table 2 we present our previous results from [7] together with the results of Rogaway and Shrimpton [6].

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

	Pre	aPre	ePre	Sec	aSec	eSec	Coll	MAC	CTFP	aCTFP	Prf	Pro
CTFP	$\nrightarrow 6$ $\nleftarrow 5$	$\nrightarrow 6$ $\nleftarrow 5$	$\nrightarrow 6$ $\nleftarrow 5$	$\nrightarrow 6$ $\nleftarrow 5$	$\nrightarrow 6$ $\nleftarrow 5$	$\nrightarrow 6$ $\nleftarrow 5$	$\nrightarrow 6$ $\leftarrow 1$	$\nrightarrow 6$ $\nleftarrow 5$	$\rightarrow$ $\leftarrow$	$\nrightarrow 4$ $\leftarrow 3$	$\nrightarrow 6$ $\nleftarrow 5$	$\nrightarrow 6$ $\leftarrow 2$
aCTFP	$\nrightarrow 8$ $\nleftarrow 7$	$\nrightarrow 8$ $\nleftarrow 7$	$\nrightarrow 8$ $\nleftarrow 7$	$\nrightarrow 8$ $\nleftarrow 7$	$\nrightarrow 8$ $\nleftarrow 7$	$\nrightarrow 8$ $\nleftarrow 7$	$\nrightarrow 8$ $\nleftarrow 7$	$\nrightarrow 8$ $\nleftarrow 7$	$\rightarrow 3$ $\nleftarrow 4$	$\rightarrow$ $\leftarrow$	$\nrightarrow 8$ $\nleftarrow 7$	$\nrightarrow 8$ $\nleftarrow 7$

2. Preliminaries

Formal definitions of hash function security are usually made in the hash function family settings. The hash function family is a hash function parametrized by a key. It 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 considering only hash functions, since collisions exist in every hash function (as 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 an adversary in practice. In hash function family settings, such an adversary would need to have hardwired a colliding pair for every key.

TABLE 2. Relationships among the definitions from our previous work [7] and from the work of Rogaway, Shrimpton [6]. The relationships among the first seven properties are analyzed in [6], while the others in [7].

	Pre	aPre	ePre	Sec	aSec	eSec	Coll	MAC	Prf	Pro
Pre	$\leftrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$
aPre	$\rightarrow$	$\leftrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$
ePre	$\rightarrow$	$\nrightarrow$	$\leftrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$
Sec	$\rightarrow$	$\nrightarrow$	$\nrightarrow$	$\leftrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$
aSec	$\rightarrow$	$\rightarrow$	$\nrightarrow$	$\rightarrow$	$\leftrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$
eSec	$\rightarrow$	$\nrightarrow$	$\nrightarrow$	$\rightarrow$	$\nrightarrow$	$\leftrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$
Coll	$\rightarrow$	$\nrightarrow$	$\nrightarrow$	$\rightarrow$	$\nrightarrow$	$\rightarrow$	$\leftrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$
Mac	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\leftrightarrow$	$\nrightarrow$	$\nrightarrow$
Prf	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\nrightarrow$	$\rightarrow$	$\leftrightarrow$	$\nrightarrow$
Pro	$\rightarrow$	$\nrightarrow$	$\rightarrow$	$\rightarrow$	$\nrightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\rightarrow$	$\leftrightarrow$

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 $M \xleftarrow{\$} \mathcal{S}$ 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 $\overline{M}$. Empty string is denoted by μ . Let $\text{Func}(D, R)$ represent the set of all functions $\rho: D \rightarrow R$ and let $RF_{D,R}$ be a function chosen randomly from the set $\text{Func}(D, R)$ (i.e., $RF_{D,R} \xleftarrow{\$} \text{Func}(D, R)$). We sometimes write $RF_{d,r}$ when $D = \{0, 1\}^d$ and $R = \{0, 1\}^r$. By $\text{Prefix}_n(M)$ we denote the n -bit prefix of message M , similarly by $\text{Suffix}_n(M)$ 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 $\mathcal{A}$ on some input is the *average* time needed to compute an output (relative to some fixed RAM model) plus the description size of $\mathcal{A}$ (relative to some fixed coding of RAMs).

Note that we append the description size of an adversary into its running time, to avoid the situations, where an adversary has “hardwired” in its code large amount of information, so that it can run very fast. However, the information hardwired into the adversary need to be computed, too. Also note that we consider average running time rather than the worst case running time of an adversary. This is because of cases, when an adversary runs very slowly, but only with negligible probability. It suffices for us, that an adversary is efficient with high probability.

Let $H: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ be a hash function family. We denote by $\text{Time}_{H,n}$ the running time of an algorithm P (i.e., some random access machine) computing H that has the best 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, $\text{Time}_{H,n}$ 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) relate to the terms of indistinguishability and indifferentiability. If a hash function family is secure in the Prf (Pro) sense, then it is indistinguishable (indifferentiable) from a random oracle. The notion Pro was firstly introduced by Coron, Dodis, Malinaud and Puniya [3] and then reused by Bellare and Ristenpart [1], [2]. The term indifferentiability was firstly introduced by Maurer, Renner and Holenstein [5].

Finally, a hash function family should be unforgeable (MAC) 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 λ , 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 λ be a positive integer. Let $\mathcal{A}$ be an adversary. Then we define the following advantage measures:

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

We say that H is (t, L, ε) -xxx for $\text{xxx} \in \{\text{Pre}, \text{aPre}, \text{Sec}, \text{eSec}, \text{aSec}\}$ if any adversary $\mathcal{A}$ running in time at most t and outputting messages of length less than

or equal to L has advantage

$$\mathbf{Adv}_H^{\text{xxx}[\lambda]}(\mathcal{A}) \leq \varepsilon \quad \text{for all } \lambda.$$

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

$$\mathbf{Adv}_H^{\text{zzz}}(\mathcal{A}) \leq \varepsilon.$$

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 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 be impossible for an adversary attacking in the 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 $\mathcal{A}$ 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:

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

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

$$\mathbf{Adv}_{H,f,\mathcal{S}}^{\text{Pro}}(\mathcal{A}) \leq \varepsilon.$$

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 advertisement appears in a news:

I, Nostradamus, hereby provides 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 have not 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, λ be a positive integer and let $\mathcal{A}$ be an adversary. Then we define the following advantage measure:

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

We say that H is (t, L, ε) -CTFP if any adversary $\mathcal{A}$ running in time at most t and outputting messages of length less than or equal to L has advantage

$$\mathbf{Adv}_H^{\text{CTFP}[\lambda]}(\mathcal{A}) \leq \varepsilon \quad \text{for all } \lambda.$$

The variable S in the definition is adversary's state. It is a string of an arbitrary length, where $\mathcal{A}$ can store some information (i.e., its state) for the second stage. The string S is the only way how to pass some information to the second stage (i.e., $\mathcal{A}$'s only input in the second stage are P and S , it is not allowed to read any additional information from external files etc.). The image Y which $\mathcal{A}$ 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, λ be a positive integer and let $\mathcal{A}$ be an adversary. Then we define the following advantage measure:

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

We say that H is (t, L, ε) -aCTFP if any adversary $\mathcal{A}$ running in time at most t and outputting messages of length less than or equal to L has advantage

$$\mathbf{Adv}_H^{\text{aCTFP}[\lambda]}(\mathcal{A}) \leq \varepsilon \quad \text{for all } \lambda.$$

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 given that randomly selected values. These “two stage” definitions of the security notions are more demonstrative and they are more suitable for our proofs.

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

$$\mathbf{Adv}_H^{\text{ePre}}(\mathcal{A}) = \Pr \left[(Y, S) \leftarrow \mathcal{A}; K \xleftarrow{\$} \mathcal{K}; M' \leftarrow \mathcal{A}(K, S) : H_K(M') = Y \right],$$

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

$$\mathbf{Adv}_H^{\text{eSec}}(\mathcal{A}) = \Pr \left[(M, S) \leftarrow \mathcal{A}; K \xleftarrow{\$} \mathcal{K}; M' \leftarrow \mathcal{A}(K, S) : \right. \\ \left. (M \neq M') \wedge (H_K(M) = H_K(M')) \right],$$

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

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. Implications and separations

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 a xxx sense, then so it is in a yyy sense. Similarly, a security notion xxx non-implies security notion yyy, if there exists a hash function family H secure in a xxx sense, but insecure in a 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 a xxx sense, but insecure in a yyy sense. For briefer presentation, let $Atks$ temporarily denote the set {Pre, aPre, ePre, Sec, aSec, eSec, Coll, CTFP, aCTFP, MAC, Prf, Pro}.

We consider a hash function family H to be secure in some sense (Prf, MAC, Sec, ...) if any polynomial time 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 the Pro sense when for any polynomial time adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$ running in a polynomial time such that the advantage $\mathbf{Adv}_{H,S,f}^{\text{Pro}}(\mathcal{A})$ is negligible for $f = RF_{y+d,y}$. Polynomial time 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 $\mathbf{Adv}_{H,\cdot}^{\text{xxx}[\cdot]}$, which is either $\mathbf{Adv}_{H,S,f}^{\text{xxx}[\cdot]}$ (when xxx is Pro), or $\mathbf{Adv}_{H,f}^{\text{xxx}[\cdot]}$ (when xxx is something else than Pro, but we are comparing it to Pro (e.g., yyy is Pro)), or $\mathbf{Adv}_H^{\text{xxx}[\cdot]}$ (when both security notions xxx and yyy are different from Pro).

DEFINITION 4 (xxx $\rightarrow$ yyy). Let

$$\mathcal{K} = \{0, 1\}^k, \mathcal{M} = \{0, 1\}^* \quad \text{and} \quad \mathcal{Y} = \{0, 1\}^y$$

for some fixed k and y , let λ 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 (denoted by xxx $\rightarrow$ yyy), if for any hash function family $H: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ and any adversary $\mathcal{A}$ running in polynomial time t with non-negligible advantage (with respect to k , y or λ) in the yyy sense (for all polynomial simulators $\mathcal{S}$ if yyy is Pro), there exists an adversary $\mathcal{A}'$ such that $\mathcal{A}'$ runs in polynomial time t' and has non-negligible advantage in the 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

$$\mathcal{K} = \{0, 1\}^k, \mathcal{M} = \{0, 1\}^* \quad \text{and} \quad \mathcal{Y} = \{0, 1\}^y$$

for some fixed k and y , let λ 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 xxx *non-implies* security notion yyy (denoted by $\text{xxx} \not\rightarrow \text{yyy}$), 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 $\text{Adv}_{H',\cdot}^{\text{xxx}[\cdot]}(t)$ is non-negligible (for all polynomial simulators $\mathcal{S}$ if xxx is Pro), then so is $\text{Adv}_{H,\cdot}^{\text{xxx}[\cdot]}(t')$ (for all polynomial simulators $\mathcal{S}$ if xxx is Pro), and $\text{Adv}_{H,\cdot}^{\text{yyy}[\cdot]}(t')$ is non-negligible, too (for all polynomial simulators $\mathcal{S}$ if yyy is Pro), where t and t' are some polynomial running times.

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

Thus our definitions of implication and separation above are less strict than ones in [6]. One can easily see that the Rogaway-Shrimpton's definitions imply our definitions above, that is, if xxx implies yyy with respect to the definition of Rogaway and Shrimpton, then xxx implies yyy also with respect to our definition. Our definition of implication provides more freedom when reducing the ability of breaking a hash function family H in the yyy sense (represented by the adversary $\mathcal{A}$), to the ability of breaking H in the xxx sense (this ability is represented by the adversary $\mathcal{B}$). Similarly, if xxx non-implies yyy with respect to the definition of Rogaway and Shrimpton, then the separation holds with respect to our definition, too.

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 $\mathcal{A}$ an adversary running in a polynomial time and attacking H in the CTFP sense with non-negligible advantage ε . Consider the following adversary $\mathcal{B}$ performing attack in the Coll sense against H :

Adversary $\mathcal{B}(K)$

- (1) $(Y, S) \leftarrow \mathcal{A}(K)$,
- (2) **let** $P_1 \xleftarrow{\$} \{0, 1\}^\lambda$,
- (3) $M_1 \leftarrow \mathcal{A}(P_1, S)$,
- (4) **let** $P_2 \xleftarrow{\$} (\{0, 1\}^\lambda - \{P_1\})$,
- (5) $M_2 \leftarrow \mathcal{A}(P_2, S)$,
- (6) **return** $(P_1 \| M_1, P_2 \| M_2)$.

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

$$\varepsilon \left(\varepsilon - \frac{1}{2^\lambda} \right) = \varepsilon^2 - \frac{\varepsilon}{2^\lambda},$$

which is non-negligible, if ε is non-negligible.

Note that all information the adversary $\mathcal{A}$ can pass from the first to the second phase is stored in the variable S (which is returned by $\mathcal{A}$ in its first phase and given as input in its second phase). The adversary $\mathcal{B}$ simulates the first phase of $\mathcal{A}$ only once, but in the second simulation (line 5 of the adversary $\mathcal{B}$) $\mathcal{A}$ cannot notice, that its second phase is going to be run for the second time, since $\mathcal{A}$ has the same inputs. Hence in the second simulation $\mathcal{A}$ must succeed with the same probability as in the first simulation (line 3 in $\mathcal{B}$'s code), there is only a little difference that the prefix P_1 chosen in the first simulation cannot be chosen again in the second simulation. $\square$

THEOREM 2. $\text{Pro} \rightarrow \text{CTFP}$.

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 λ be a positive integer. Consider an adversary $\mathcal{A}$ running in a polynomial time t and attacking H in the CTFP sense

with non-negligible advantage $\mathbf{Adv}_{H,f}^{\text{CTFP}[\lambda]}(\mathcal{A}) = \varepsilon$. We construct the following adversary $\mathcal{B}$ attacking H in the Pro sense:

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

It is clear that there exists a polynomial $p(k, y, \lambda)$ such that the running time of $\mathcal{B}$ is $p(k, y, \lambda) \cdot t$. When $\mathcal{B}$'s oracles are H and f , then it returns 1 with the probability equal to the probability that $\mathcal{A}$ wins in the simulation on line 3, i.e., $\mathcal{B}$'s chance to win is ε . Now consider the case, when $\mathcal{B}$'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 $q_S \cdot q$, where q_S is the maximum number of queries made by $\mathcal{S}$ to $\mathcal{F}$ and q is the maximum number of queries made by $\mathcal{A}$ to its oracle (in this case, its oracle is $\mathcal{S}^{\mathcal{F}}$; note that $q \leq t$). Because $\mathcal{F}$ is random, the probability of finding M such that $\mathcal{F}(P\|M) = Y$ is at most $(q_S \cdot q)/|\mathcal{Y}|$. Since both $\mathcal{S}$ and $\mathcal{A}$ run in the polynomial time, $q_S \cdot q$ is also polynomial and thus $(q_S \cdot q)/|\mathcal{Y}|$ is negligible. Therefore $\mathcal{B}$'s advantage against H is

$$\Pr[\mathcal{B}^{H^f, f} \rightarrow 1] - \Pr[\mathcal{B}^{\mathcal{F}, \mathcal{S}^{\mathcal{F}}} \rightarrow 1] = \varepsilon - \frac{(q_S \cdot q)}{|\mathcal{Y}|},$$

which is non-negligible if ε is non-negligible. $\square$

THEOREM 3. $a\text{CTFP} \rightarrow \text{CTFP}$.

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

Adversary $\mathcal{B}$
 [1st stage]
 $(Y, S) \leftarrow \mathcal{A}(K_0)$,
return (Y, K_0, S) ;
 [2nd stage with input (P, S)]
 $M \leftarrow \mathcal{A}(P, S)$,
return M .

Clearly, $\mathcal{B}$ is only polynomially slower than $\mathcal{A}$. From the assumption that K_0 is the key, where $\mathcal{A}$'s chance to win against H in the CTFP sense is at least ε ,

we have that $\mathcal{B}$'s chance to win against H in the aCTFP sense is at least ε , too. This means that $\mathcal{B}$'s advantage in the aCTFP sense is non-negligible. $\square$

4.3. Separations

In this section we investigate the separations between the CTFP, aCTFP and the other notions. We provide all the constructions used in proofs of the separations in the Figure 1.

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

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

THEOREM 4. $CTFP \not\rightarrow aCTFP$.

Proof. Let $H: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ be a hash function family, λ 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 $\mathcal{A}$ be an adversary attacking $H^{(5)}$ in the CTFP sense with non-negligible advantage. We investigate $\mathcal{A}$'s advantage against H in the CTFP sense. If a key $K \neq K_0$ is chosen by the environment, then $\mathcal{A}$'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 $\mathcal{A}$ can win against H with probability 0. However the probability that the key K_0 is chosen is $1/|\mathcal{K}|$. Thus

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

Hence $\mathcal{A}$'s advantage against H is non-negligible, too.

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

Adversary $\mathcal{B}$
 $[1^{st} \text{ stage}]$
 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 $\mathcal{B}$'s advantage in the aCTFP sense against $H^{(5)}$ is 1. For completeness we note that $\mathcal{B}$ runs in a polynomial time. $\square$

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, λ 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 the CTFP sense with non-negligible advantage against $H^{(1)}$. For that purpose consider the following adversary $\mathcal{A}$:

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

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

Now we need to show that for any adversary $\mathcal{B}$ attacking $H^{(1)}$ in a xxx sense, there exists an adversary attacking H in the xxx sense, where $\text{xxx} \in \{Pre, aPre, ePre, Sec, aSec, eSec, MAC, Prf\}$. Let $\mathcal{B}_{\text{xxx}}$ be an adversary attacking $H^{(1)}$ in the xxx sense with a non-negligible advantage $\mathbf{Adv}_{H^{(1)}}^{\text{xxx}[\cdot]}(\mathcal{B}_{\text{xxx}}) = \varepsilon$. We investigate $\mathcal{B}_{\text{xxx}}$'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 $\mathcal{B}_{\text{xxx}}$'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 $\mathcal{B}_{\text{xxx}}$ can win against H in the worst case with probability 0. Thus the following holds:

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

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

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

which is negligible. Thus $\mathcal{B}_{xxx}$ has non-negligible advantage against H and that completes the proof for the parts (1) and (2) of Theorem 5.

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 choses an image Y . After the adversary makes the selection (i.e., the 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 $\mathcal{B}_{\text{eSec}}$'s chance to win against H is the same as against $H^{(1)}$. On the other hand, the probability that K is the suffix of M is negligible (at most $1/|K|$), thus we have the similar situation as we had above, in particular:

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

And thus the $\mathcal{B}_{\text{eSec}}$ 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 $\mathcal{B}_{\text{ePre}}$ 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 $\mathcal{B}_{\text{ePre}}$ 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 $\mathcal{B}_{\text{ePre}}$'s advantage against H is non-negligible.

The idea behind the proof of the parts (4) and (5) of this theorem is, that an adversary attacking in the MAC or Prf sense does not have access to the key K chosen randomly by the environment. Thus $\mathcal{B}_{\text{xxx}}$ 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 $\mathcal{B}_{\text{xxx}}$ can make to its oracle. The following holds:

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

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

THEOREM 6.

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

Proof. Let $H: \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ and λ 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 $\mathcal{A}$ be an adversary running in a polynomial time and attacking $H^{(2)}$ in the CTFP sense with non-negligible advantage. Consider an advantage of $\mathcal{A}$ when attacking H . Let Y be an image that $\mathcal{A}$ choses in the first stage. If the prefix P chosen randomly by the environment is not prefix of Y , then $\mathcal{A}$'s chance to win against H is the same as against $H^{(2)}$. On the other hand, if P is prefix of Y , then in the worst case $\mathcal{A}$ wins against H with the probability 0. Thus

$$\begin{aligned} \mathbf{Adv}_H^{\text{CTFP}[\lambda]}(\mathcal{A}) &\geq \mathbf{Adv}_{H^{(2)}}^{\text{CTFP}[\lambda]}(\mathcal{A}) \\ &\quad - \Pr[\text{forced prefix } P \text{ is prefix of chosen image } Y]. \end{aligned}$$

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 exists. Therefore

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

which is negligible and thus $\mathcal{A}$ has non-negligible advantage against H .

For the second part of the proof, the construction $H^{(2)}$ is clearly not Pre, aPre, ePre secure. The adversaries attacking in the Pre and aPre sense simply copy their input (image Y) to the output ($H^{(2)}(Y) = Y$). The adversary attacking in

the ePre sense in the first stage choses 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 $\mathcal{A}$ be an adversary attacking $H^{(3)}$ in the CTFP sense with non-negligible advantage and running in a polynomial time. Consider the following adversary $\mathcal{B}$ performing attack in the CTFP sense against H :

Adversary $\mathcal{B}$
 [1st stage with input K]
 $Y \leftarrow \mathcal{A}(K),$
return $(Y, K);$
 [2nd stage with input (P, S)]
 $M \leftarrow \mathcal{A}(P, S),$
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 an adversary $\mathcal{B}$ runs in a time that is only polynomially slower than the running time of $\mathcal{A}$. From the definition of $H^{(3)}$ we can see, that if $\mathcal{A}$ wins against $H^{(3)}$, then $\mathcal{B}$ wins against H . Thus $\mathcal{B}$ has non-negligible advantage against H in the CTFP sense.

The construction $H^{(3)}$ is clearly not secure in the Sec, aSec, eSec or Coll sense, 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 inverted.

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

The hash function family $H^{(4)}$ is clearly not MAC, Prf or Pro secure. Adversary attacking $H^{(4)}$ in the MAC sense returns $(0, 0^y)$ and wins with probability 1. Adversaries attacking in the Prf and Pro sense check 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 the Prf and Pro sense with non-negligible advantage, since the probability that for a random function $\mathcal{F}$ holds $\mathcal{F}(0) = 0^y$ is negligible. $\square$

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 $Pre \rightarrow CTFP$. This leads to 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 $\mathcal{A}$ be an polynomial adversary performing attack in the Coll sense against $H^{(5)}$ with non-negligible advantage. Consider $\mathcal{A}$'s advantage against H . If a key K chosen by the environment is different from K_0 , then $\mathcal{A}$'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 $\mathcal{A}$ wins with the probability 0. Therefore

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

Hence, $\mathcal{A}$ 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 the aCTFP sense with non-negligible advantage. $\square$

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) $aCTFP \not\rightarrow Pro$.

Remark. 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.

5. Conclusion

In this paper we analyzed the relationships between the notions of Chosen Target Forced Prefix preimage resistance 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 the CTFP preimage resistance is independent from the most of the other standard security notions, except collision resistance and pseudo-random oracle, which imply CTFP. The aCTFP security notion implies CTFP and is independent from all of the other notions.

REFERENCES

- [1] BELLARE, M.—RISTENPART, T.: *Hash functions in the dedicated-key setting: Design choices and MPP transforms*, in: Automata, Languages and Programming, 34th Internat. Colloquium—ICALP '07 (L. Arge et al., eds.), Wroclaw, Poland, 2007, Lecture Notes in Comput. Sci., Vol. 4596, Springer-Verlag, Berlin, 2007, pp. 399–410.
- [2] BELLARE, M.—RISTENPART, T.: *Multi-property-preserving hash domain extension and the EMD transform*, in: Advances in Cryptology—ASIACRYPT '06, 12th Internat. Conf. on the Theory and Application of Cryptology and Information Security (X. Lai et al., eds.), Shanghai, China, 2006, Lecture Notes in Comput. Sci., Vol., 4284, Springer-Verlag, Berlin, 2006, pp. 299–314.
- [3] CORON, J. S.—DODIS, Y.—MALINAUD, C.—PUNIYA, P.: *Merkle-Damgård revisited: How to construct a hash function*, in: Advances in Cryptology—CRYPTO '05, 25th Annual Internat. Cryptology Conf. (V. Shoup, ed.), Santa Barbara, CA, USA, 2005, Lecture Notes in Comput. Sci., Vol. 3621, Springer-Verlag, Berlin, 2005, pp. 430–448.
- [4] KELSEY, J.—KOHNO, T.: *Herding hash functions and the Nostradamus attack*, in: Advances in Cryptology—EUROCRYPT '06, 25th Annual Internat. Conf. on the Theory and Appl. of Cryptographic Techniques (S. Vaudenay, ed.), St. Petersburg, Russia, 2006, Lecture Notes in Comput. Sci., Vol. 4004, Springer-Verlag, Berlin, 2006, pp. 183–200.
- [5] MAURER, U.—RENNER, R.—HOLENSTEIN, C.: *Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology*, in: Theory of Cryptography, 1st Theory of Cryptography Conf.—TCC '04 (M. Naor, ed.), Cambridge, MA, USA, 2004, Lecture Notes in Comput. Sci., Vol. 2951, Springer-Verlag, Berlin, 2004, pp. 21–39.

ON CHOSEN TARGET FORCED PREFIX PREIMAGE RESISTANCE

- [6] ROGAWAY, P.—SHRIMPTON, T.: *Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance*, in: Fast Software Encryption, 11th International Workshop—FSE '04 (B. Roy et al., eds.), Delhi, India, 2004, Lecture Notes in Comput. Sci., Vol. 3017, Springer-Verlag, Berlin, 2004, pp. 371–388.
- [7] RJAŠKO, M.: *Properties of cryptographic hash functions*, in: Mikulášska Kryptobesídka, 2008, Prague, Trusted Network Solution, Prague, 2008, pp. 53–62.

Received September 12, 2009

*Department of Computer Science
Faculty of Mathematics,
Physics and Informatics
Comenius University
SK-842-48 Bratislava
SLOVAKIA
E-mail: rjasko@dcs.fmph.uniba.sk*