PAPER Special Section on Cryptography and Information Security

Weak Security Notions of Cryptographic Unkeyed Hash Functions and Their Amplifiability

Shoichi HIROSE[†a)], ***Member***

SUMMARY Cryptographic unkeyed hash functions should satisfy preimage resistance, second-preimage resistance and collision resistance. In this article, weak second-preimage resistance and weak collision resistance are defined following the definition of weak one-wayness. Preimage resistance is one-wayness of cryptographic hash functions. The properties of weak collision resistance is discussed in this article. The same kind of results can be obtained for weak second-preimage resistance. Weak collision resistance means that the probability of failing to find a collision is not negligible, while collision resistance means that the success probability is negligible. It is shown that there really exist weakly collision resistant hash functions if collision resistant ones exist. Then, it is shown that weak collision resistance is amplifiable, that is, collision resistant hash functions can be constructed from weakly collision resistant ones. Unfortunately, the method of amplification presented in this article is applicable only to a certain kind of hash functions. However, the method is applicable to hash functions based on discrete logarithms. This implies that collision resistant hash functions can be obtained even if the discrete logarithm problem is much easier than is believed and only weakly intractable, that is, exponentiation modulo a prime is weakly one-way.

key words: *cryptographic hash function, collision resistance, weak collision resistance, second-preimage resistance, weak second-preimage resistance*

1. Introduction

Hash functions are very important primitives in cryptography. Hash functions in cryptography are classified in two types: unkeyed hash functions and keyed hash functions. The former ones are also called manipulation detection codes (MDCs). The latter ones are also called message authentication codes (MACs). Excellent surveys are presented in [6], [8].

Cryptographic unkeyed hash functions should satisfy preimage resistance, second-preimage resistance and collision resistance. Preimage resistance means that, given an output, it is infeasible to obtain an input which produces the output. Thus, preimage resistance is one-wayness for hash functions. Second-preimage resistance means that, given an input, it is infeasible to obtain another input which produces the same output as the given input. Collision resistance means that it is infeasible to obtain two different inputs which produce the same output. In this article, weak second-preimage resistance and weak collision resistance are defined and analyzed.

Actually, the term "weak collision resistance" is also found in previous literature, and is used to represent two different security notions of hash functions. In some literature, second-preimage resistance is called weak collision resistance [6]. In this case, collision resistance is called strong collision resistance. On the other hand, in [2], weak collision resistance is defined to be collision resistance of keyed hash functions, and it is shown to be implied by unforgeability, that is, secure MAC [1]. In both cases, weak collision rsistance is implied by one-wayness. Hence, from Simon's result [9], it is quite different from (strong) collision resistance. In [9], he showed that no provable construction of a collision resistant hash function exists based on a "black box" one-way permutation, which means that the one-way permutation is used as a subroutine, that is, the internal structure is not used.

For one-wayness, on the other hand, "weak" and "strong" represent how difficult it is to find a preimage. A function is called strongly one-way if, for every probabilistic polynomial-time algorithm, the probability that it succeeds in finding a preimage is negligible. A function is called weakly one-way if, for every probabilistic polynomial-time algorithm, the probability that it fails to find a preimage is not negligible. In this framework, there exist weakly one-way functions if there exist strongly one-way functions and weakly one-way functions imply strongly one-way functions. This equivalence is implicit in [10] and the proof is in [5]. Though the proof is not so straightforward, the latter implication is proved by simple alignment as is easily imagined. For example, let f be a function and $g(x_1, x_2) \overset{\text{def}}{=} (f(x_1), f(x_2))$. To find a preimage of a given output of g, it is required to find preimages for both of the two f's of which g is composed.

In this article, weak collision resistance and second-preimage resistance are defined following the definition of weak one-wayness. That is, weak collision resistance means that the probability of failure to find a collision is not negligible. In contrast, strong collision resistance means that the probability of success in finding a collision is negligible. The weak collision resistance is then analyzed using the technique in [5]. The same kind of results are obtained for second-preimage resistance.

First, it is shown that this new definition is not void. It is shown that there exist weakly collision resistant hash functions if there exist collision resistant hash functions. Second, it is shown that weak collision resistance can be amplifiable. However, simple alignment mentioned above

Manuscript received March 22, 2004.
Manuscript revised July 9, 2004.
Final manuscript received August 18, 2004.

[†] The author is with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.
a) E-mail: hirose@i.kyoto-u.ac.jp

does not work for collision resistance and second-preimage resistance. For example, to find another input which produces the same output with the given input (x_1, x_2) of g defined above, it is sufficient to find only one of x'_1 or x'_2 such that $f(x_i) = f(x'_i)$ and $x_i \neq x'_i$ for $i = 1, 2$, because $g(x_1, x_2) = g(x'_1, x_2) = g(x_1, x'_2)$, $(x_1, x_2) \neq (x'_1, x_2)$ and $(x_1, x_2) \neq (x_1, x'_2)$. It is shown in this article that strongly collision resistant hash functions can be constructed from weakly collision resistant hash functions satisfying the following property: $h : X \times X \to Y$, $|X| = |Y|$, and both $h(x, \cdot)$ and $h(\cdot, x)$ are permutations for every $x \in X$. Hash functions based on discrete logarithms [3] satisfy the property. Thus, strongly collision resistant hash functions can be constructed even if the discrete logarithm problem is much easier than is believed and only weakly intractable, that is, exponentiation modulo a prime is weakly one-way. This shows that the amplifiability result in this article also has practical significance.

The rest of this article is organized as follows. In Section 2, weak collision resistance and strong collision resistance are formally defined. In Section 3, the existence of weakly collision resistant hash functions is discussed. The topic of Section 4 is the amplifiability of weak collision resistance. In Section 5, an example of weakly collision resistant hash functions are presented based on discrete logarithms. It is briefly mentioned that the same kind of results can be obtained for weak second-preimage resistance in Section 6. A concluding remark is in Section 7.

2. Preliminaries

Let $\mathbb{N}$ be the set of positive integers. Let H_n be a set of hash functions such that $H_n = \{h_k \,|\, h_k : D \to R,\ k \in K\}$, where $D \subseteq \{0,1\}^{\ell_D(n)}$, $R \subseteq \{0,1\}^{\ell_R(n)}$, $K \subseteq \{0,1\}^{\ell_K(n)}$, $|D| > |R|$, and $\ell_D(n)$, $\ell_R(n)$ and $\ell_K(n)$ are polynomials in n. k is regarded as an index. A pair of inputs $(x, x') \in D \times D$ is called a collision of a hash function h_k if $h_k(x) = h_k(x')$ and $x \neq x'$.

A family of hash functions $\{H_n\}_{n\in\mathbb{N}}$ is called weakly collision resistant (weakly CR) if the probability of failure to find a collision is not negligible for every efficient algorithm.

Definition 1: $\{H_n\}_{n\in\mathbb{N}}$ is called a weakly CR family of hash functions if

- there exists a probabilistic polynomial-time algorithm K_H which, with an input 1^n, outputs $k \in K$,
- there exists a deterministic polynomial-time algorithm M_H which, with inputs $k \in K$ and $x \in D$, outputs $h_k(x)$, and
- there exists some polynomial $p(n)$ such that, for every probabilistic polynomial-time algorithm F_H and every sufficiently large n,

$$\sum_{k\in K} \Pr[\mathsf{K}_H(1^n) = k]\ \Pr[\mathsf{F}_H(k) \text{ fails}] \geq \frac{1}{p(n)},$$

where "$\mathsf{F}_H(k)$ fails" means that it fails to find a collision and the probability is taken over the coin tosses of K_H and F_H. ♦

A family of hash functions $\{H_n\}_{n\in\mathbb{N}}$ is called strongly CR if the probability of success in finding a collision is negligible for every efficient algorithms.

Definition 2: $\{H_n\}_{n\in\mathbb{N}}$ is called a strongly CR family of hash functions if

- there exist a probabilistic polynomial-time algorithm K_H and a deterministic polynomial-time algorithm M_H as stated in Definition 1, and
- for every polynomial $q(n)$, every probabilistic polynomial-time algorithm F_H and every sufficiently large n,

$$\sum_{k\in K} \Pr[\mathsf{K}_H(1^n) = k] \Pr[\mathsf{F}_H(k) \text{ succeeds}] < \frac{1}{q(n)},$$

where "$\mathsf{F}_H(k)$ succeeds" means that it succeeds in finding a collision and the probability is taken over the coin tosses of K_H and F_H. ♦

From the definitions, it is obvious that strong CR implies weak CR. In the followings, a family of hash functions is simply called CR if it is strongly CR or weakly CR.

3. Existence of a Weakly Collision Resistant Family of Hash Functions

In this section, it is shown that there exists a weakly CR family of hash functions if there exists a CR family of hash functions.

Let $H_n = \{h_k \,|\, h_k : D \to R,\ k \in K\}$. For a family of hash functions $\{H_n\}_{n\in\mathbb{N}}$, a family of hash functions $\{F_n\}_{n\in\mathbb{N}}$ is defined as follows.

- $F_n = H_n \cup \{f\}$, where $f : D \to R$, f is polynomial-time computable, and its collision is easy to find.
- K_F is an algorithm for sampling an index. With an input 1^n, it proceeds as follows.
  1. It selects $u_1 \in \{0,1\}^{\lceil \log n \rceil}$ at random.
  2. It runs $\mathsf{K}_H(1^n)$.
  3. It outputs $u = (u_1, u_2)$, where u_2 is the output of $\mathsf{K}_H(1^n)$ in the previous step.
- Let $F_n = \{f_u \,|\, f_u : D \to R, u \in \{0,1\}^{\lceil \log n \rceil} \times K\}$. Then,

$$f_{(u_1,u_2)} = \begin{cases} h_{u_2} & \text{if } u_1 = 0^{\lceil \log n \rceil} \\ f & \text{otherwise.} \end{cases}$$

It is obvious that $\{F_n\}_{n\in\mathbb{N}}$ is not strongly CR. There exists a probabilistic polynomial-time algorithm such that the probability of its success in finding a collision is at least $1 - 1/n$.

Theorem 1: If $\{H_n\}_{n\in\mathbb{N}}$ is CR, then $\{F_n\}_{n\in\mathbb{N}}$ is weakly CR. ♦

(Proof) Suppose that $\{F_n\}_{n\in\mathbb{N}}$ is not weakly CR. For simplicity, let $m = \lceil \log n \rceil$ and $L = \{0,1\}^m$. Then, for every polynomial $p(n)$, there exists a probabilistic polynomial-time algorithm A_F such that,

$$\sum_{u \in L \times K} \Pr[\mathsf{K}_F(1^n) = u] \Pr[\mathsf{A}_F(u) \text{ succeeds}] \geq 1 - \frac{1}{p(n)}$$

for infinitely many n's. "$\mathsf{A}_F(u)$ succeeds" means that $\mathsf{A}_F(u)$ succeeds in finding a collision, that is, $\mathsf{A}_F(u) = (x, x')$, $f_u(x) = f_u(x')$ and $x \neq x'$.

$$\begin{aligned}
&\sum_{u \in L \times K} \Pr[\mathsf{K}_F(1^n) = u] \Pr[\mathsf{A}_F(u) \text{ succeeds}] \\
&= \sum_{\substack{u \in L \times K \\ u_1 = 0^m}} \Pr[\mathsf{K}_F(1^n) = u] \Pr[\mathsf{A}_F(u) \text{ suc.}] \\
&\quad + \sum_{\substack{u \in L \times K \\ u_1 \neq 0^m}} \Pr[\mathsf{K}_F(1^n) = u] \Pr[\mathsf{A}_F(u) \text{ suc.}] \\
&\leq \sum_{u_2 \in K} \Pr[\mathsf{K}_F(1^n) = (0^m, u_2)] \Pr[\mathsf{A}_F(0^m, u_2) \text{ suc.}] \\
&\quad + \sum_{\substack{u \in L \times K \\ u_1 \neq 0^m}} \Pr[\mathsf{K}_F(1^n) = u] \\
&= \sum_{u_2 \in K} \Pr[\mathsf{K}_F(1^n) = (0^m, u_2)] \Pr[\mathsf{A}_F(0^m, u_2) \text{ suc.}] \\
&\quad + \left(1 - \frac{1}{2^m}\right).
\end{aligned}$$

Thus,

$$\begin{aligned}
&\sum_{u_2 \in K} \Pr[\mathsf{K}_F(1^n) = (0^m, u_2)] \Pr[\mathsf{A}_F(0^m, u_2) \text{ suc.}] \\
&\quad \geq \frac{1}{2^m} - \frac{1}{p(n)}.
\end{aligned}$$

Let A_H be an algorithm which, with an input $k \in K$, runs $\mathsf{A}_F(0^m, k)$ and outputs its output. Then,

$$\begin{aligned}
&\sum_{k \in K} \Pr[\mathsf{K}_H(1^n) = k] \Pr[\mathsf{A}_H(k) \text{ suc.}] \\
&= 2^m \sum_{u_2 \in K} \Pr[\mathsf{K}_F(1^n) = (0^m, u_2)] \Pr[\mathsf{A}_F(0^m, u_2) \text{ suc.}] \\
&\geq 1 - \frac{2^m}{p(n)} > 1 - \frac{2n}{p(n)},
\end{aligned}$$

which implies that $\{H_n\}_{n \in \mathbb{N}}$ is not weakly CR. ■

4. Amplifiability of Weak Collision Resistance

In this section, it is shown that weak CR can be amplifiable, that is, a strongly CR family of hash functions can be constructed from a weakly CR family of hash functions. Unfortunately, the proposed method of construction is applicable only to families of hash functions with an additional property. However, in the next section, it is mentioned that there really exists a (weakly) CR family of hash functions with the property.

Theorem 2: A strongly CR family of hash functions is able to be constructed from any weakly CR family of hash functions

$$H_n = \{h_k \,|\, h_k : X \times X \to Y,\ k \in K\}$$

such that $|X| = |Y|$ and both $h_k(x, \cdot)$ and $h_k(\cdot, x)$ are permutations for every k and x. ♦

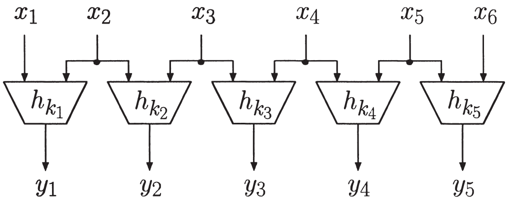

Fig. 1 A hash function $g_v \in G_{n,5}$.

(Proof) Let

$$G_{n,d} = \{g_v \,|\, g_v : X^{d+1} \to Y^d,\ v \in K^d\}$$

such that

$$\begin{aligned}
&g_v(x_1, x_2, \ldots, x_{d+1}) \\
&\quad = (h_{k_1}(x_1, x_2), h_{k_2}(x_2, x_3), \ldots, h_{k_d}(x_d, x_{d+1}))
\end{aligned}$$

for $k_j \in K$ for $j = 1, \ldots, d$ and $v = (k_1, k_2, \ldots, k_d)$. Fig. 1 shows an example for $d = 5$.

It is obvious that there exists a probabilistic polynomial-time algorithm M_G which, with inputs $v = (k_1, \ldots, k_d) \in K^d$ and $x = (x_1, \ldots, x_{d+1}) \in X^{d+1}$, outputs $g_v(x)$. M_G simply runs $\mathsf{M}_H(k_i, (x_i, x_{i+1}))$ for $i = 1, 2, \ldots, d$.

Let K_G be an algorithm which, with an input 1^n, runs $\mathsf{K}_H(1^n)$ d times independently. K_G is a probabilistic polynomial-time algorithm for sampling an index of the hash functions in $G_{n,d}$ if d is a polynomial in n.

Lemma 1: For every $g_v \in G_{n,d}$, if

- $g_v(x_1, x_2, \ldots, x_{d+1}) = g_v(x'_1, x'_2, \ldots, x'_{d+1})$ and
- $(x_1, x_2, \ldots, x_{d+1}) \neq (x'_1, x'_2, \ldots, x'_{d+1})$,

then $x_j \neq x'_j$ for $j = 1, 2, \ldots, d + 1$. ◊

(Proof) This lemma is obvious from the assumption on H_n that both $h_k(x, \cdot)$ and $h_k(\cdot, x)$ are permutations for every k and x. □

In the remaining part, it is shown that $\{G_{n,d}\}_{n \in \mathbb{N}}$ is strongly CR with respect to the algorithm K_G for every weakly CR family $\{H_n\}_{n \in \mathbb{N}}$, where d is determined based on the weakness of CR of $\{H_n\}_{n \in \mathbb{N}}$.

Let $p(n)$ be the polynomial such that, for every probabilistic polynomial-time algorithm F_H and every sufficiently large n,

$$\sum_{k \in K} \Pr[\mathsf{K}_H(1^n) = k]\ \Pr[\mathsf{F}_H(k) \text{ fails}] \geq \frac{1}{p(n)}.$$

Let $d = n\,p(n)$ and d is denoted by $d(n)$ to make it explicit that it depends on n.

Suppose that $\{G_{n,d(n)}\}_{n \in \mathbb{N}}$ is not strongly CR, that is, there exists a probabilistic polynomial-time algorithm A_G and a polynomial q such that

$$\sum_{v\in K^{d(n)}} \Pr[\mathsf{K}_G(1^n) = v]\ \Pr[\mathsf{A}_G(v) \text{ succeeds}] \geq \frac{1}{q(n)} \qquad (1)$$

for infinitely many n's.

Let C_H be a probabilistic polynomial-time algorithm which, with an input $k \in K$, runs the following procedure B_H $b(n) = 2\,n\,d(n)\,q(n)$ times.

```
$\mathsf{B}_H(k)$ /* $k \in K$ */
{
  for ($j = 1$ to $d(n)$) {
    $(k_1, k_2, \ldots, k_{d(n)}) \leftarrow \mathsf{K}_G(1^n)$;
    $(x, x') \leftarrow \mathsf{A}_G(v)$;
      /* $v = (k_1, \ldots, k_{j-1}, k, k_{j+1}, \ldots, k_{d(n)})$
*/
    if ($g_v(x) = g_v(x')$ and $x \neq x'$) {
      output $((x_j, x_{j+1}), (x'_j, x'_{j+1}))$;
      halt;
    }
  }
}
```

From Lemma 1, $\mathsf{B}_H(k)$ succeeds in finding a collision of h_k if $\mathsf{A}_G(v)$ succeeds in finding a collision of g_v. Let $B = \{k \,|\, k \in K,\ \Pr[\mathsf{B}_H(k) \text{ succeeds}] > n/b(n)\}$.

Lemma 2: $\Pr[\mathsf{C}_H(k) \text{ succeeds}] > 1 - \dfrac{1}{2^n}$ for every $k \in B$. ◊

(Proof) Since $\mathsf{C}_H(k)$ runs $\mathsf{B}_H(k)$ $b(n)$ times,

$$\Pr[\mathsf{C}_H(k) \text{ fails}] < \left(1 - \frac{n}{b(n)}\right)^{b(n)} < \frac{1}{2^n}. \qquad \square$$

Lemma 3: $\Pr[\mathsf{K}_H(1^n) \in B] > 1 - \dfrac{1}{2\,p(n)}$ for infinitely many n's. ◊

(Proof) Suppose that $\Pr[\mathsf{K}_H(1^n) \in B] \leq 1 - \dfrac{1}{2\,p(n)}$ for every sufficiently large n. It is shown that there is a contradiction between this assumption and the inequality (1).

$$\begin{aligned}
&\sum_{v\in K^{d(n)}} \Pr[\mathsf{K}_G(1^n) = v] \Pr[\mathsf{A}_G(v) \text{ succeeds}] \\
&\quad = \sum_{v\in K^{d(n)}\setminus B^{d(n)}} \Pr[\mathsf{K}_G(1^n) = v] \Pr[\mathsf{A}_G(v) \text{ suc.}] \\
&\qquad + \sum_{v\in B^{d(n)}} \Pr[\mathsf{K}_G(1^n) = v] \Pr[\mathsf{A}_G(v) \text{ suc.}].
\end{aligned}$$

Let

$$\sigma_1(n) \overset{\text{def}}{=} \sum_{v\in K^{d(n)}\setminus B^{d(n)}} \Pr[\mathsf{K}_G(1^n) = v] \Pr[\mathsf{A}_G(v) \text{ suc.}],$$

$$\sigma_2(n) \overset{\text{def}}{=} \sum_{v\in B^{d(n)}} \Pr[\mathsf{K}_G(1^n) = v] \Pr[\mathsf{A}_G(v) \text{ suc.}].$$

Let $\mathsf{K}_H^{(l)}(1^n)$ represent the l-th run of $\mathsf{K}_H(1^n)$ of $\mathsf{K}_G(1^n)$ for $1 \leq l \leq d(n)$.

$$\begin{aligned}
&\sigma_1(n) \\
&= \sum_{v\in K^{d(n)}\setminus B^{d(n)}} \left(\prod_{l=1}^{d(n)} \Pr[\mathsf{K}_H^{(l)}(1^n) = k_l]\right) \Pr[\mathsf{A}_G(v) \text{ suc.}] \\
&\leq \sum_{j=1}^{d(n)} \sum_{\substack{k_i\in K \\ 1\leq i\leq d(n), i\neq j \\ k_j\in K\setminus B}} \left(\prod_{l=1}^{d(n)} \Pr[\mathsf{K}_H^{(l)}(1^n) = k_l]\right) \Pr\,[\mathsf{A}_G(v) \text{ suc.}] \\
&\overset{\text{def}}{=} \sum_{j=1}^{d(n)} \sigma'_1(n, j).
\end{aligned}$$

$$\begin{aligned}
&\sigma'_1(n, j) \\
&= \sum_{k_j\in K\setminus B} \Pr[\mathsf{K}_H^{(j)}(1^n) = k_j] \times \\
&\qquad \sum_{\substack{k_i\in K \\ 1\leq i\leq d(n), i\neq j}} \left(\prod_{\substack{l=1 \\ l\neq j}}^{d(n)} \Pr[\mathsf{K}_H^{(l)}(1^n) = k_l]\right) \Pr[\mathsf{A}_G(v) \text{ suc.}] \\
&\leq \max_{k_j\in K\setminus B} \sum_{\substack{k_i\in K \\ 1\leq i\leq d(n) \\ i\neq j}} \left(\prod_{\substack{l=1 \\ l\neq j}}^{d(n)} \Pr[\mathsf{K}_H^{(l)}(1^n) = k_l]\right) \Pr[\mathsf{A}_G(v) \text{ suc.}] \\
&\leq \max_{k_j\in K\setminus B} \Pr[\mathsf{B}_H(k_j) \text{ suc.}] \\
&\leq \frac{n}{b(n)}.
\end{aligned}$$

Thus, $\sigma_1(n) \leq \dfrac{n\,d(n)}{b(n)}$.

On the other hand, from the assumption that $\Pr[\mathsf{K}_H(1^n) \in B] \leq 1 - \dfrac{1}{2\,p(n)}$,

$$\begin{aligned}
\sigma_2(n) &\leq \sum_{v\in B^{d(n)}} \Pr[\mathsf{K}_G(1^n) = v] \\
&\leq \left(1 - \frac{1}{2p(n)}\right)^{d(n)} < \frac{1}{2^{n/2}} < \frac{n\,d(n)}{b(n)}
\end{aligned}$$

for every sufficiently large n.

Hence,

$$\begin{aligned}
&\sum_{v\in K^{d(n)}} \Pr[\mathsf{K}_G(1^n) = v] \Pr[\mathsf{A}_G(v) \text{ succeeds}] \\
&\quad < \frac{2n\,d(n)}{b(n)} < \frac{1}{q(n)},
\end{aligned}$$

which causes a contradiction. □

From Lemmas 2 and 3, for infinitely many n's,

$$\begin{aligned}
&\sum_{k\in K} \Pr[\mathsf{K}_H(1^n) = k] \Pr[\mathsf{C}_H(k) \text{ succeeds}] \\
&\quad \geq \sum_{k\in B} \Pr[\mathsf{K}_H(1^n) = k] \Pr[\mathsf{C}_H(k) \text{ succeeds}] \\
&\quad > \left(1 - \frac{1}{2^n}\right) \sum_{k\in B} \Pr[\mathsf{K}_H(1^n) = k] \\
&\quad > \left(1 - \frac{1}{2^n}\right)\left(1 - \frac{1}{2\,p(n)}\right)
\end{aligned}$$

$$> 1 - \frac{1}{p(n)},$$

which contradicts the assumption that every probabilistic polynomial-time algorithm fails to find a collision with probability at least $1/p(n)$ for every sufficiently large n. ■

From Theorem 2, a strongly CR family of hash functions with fixed input length is obtained. A strongly CR family of hash functions with variable input length can be obtained with cascade construction owing to Merkle [7] and Damgård [4].

5. A Weakly CR Family of Hash Functions with the Property in Theorem 2

A family of hash functions based on discrete logarithms [3] is introduced first. Precisely speaking, the definition of index-sampling algorithm is different from the one in Definitions 1 and 2.

For a positive integer a, let $\mathbb{Z}_a = \{0, 1, 2, \ldots, a-1\}$ and $\mathbb{Z}_a^* = \{z \,|\, z \in \mathbb{Z}_a \wedge \gcd(z, a) = 1\}$.

S_H is a probabilistic polynomial-time algorithm which, with an input 1^n, produces p and α, where p is an n-bit prime such that $q = (p-1)/2$ is a prime and α is an element of order q in $\mathbb{Z}_p^*$. p and α are shared by all the hash functions corresponding to the parameter n. K_H is a probabilistic polynomial-time algorithm which, with inputs p and α, produces an element β of order q in $\mathbb{Z}_p^*$ at random. β is regarded as an index.

Let $G_p = \{\alpha^s \,|\, s \in \mathbb{Z}_q\}$. The family of hash functions $\{H_n^{(p,\alpha)}\}_{n\in\mathbb{N}}$ is defined by

$$H_n^{(p,\alpha)} = \left\{ h_\beta \,\middle|\, \begin{array}{l} h_\beta : \mathbb{Z}_q \times \mathbb{Z}_q \to G_p,\ \beta \in G_p, \\ h_\beta(x_1, x_2) = \alpha^{x_1}\beta^{x_2} \bmod p \end{array} \right\}.$$

Let us discuss the CR of $\{H_n^{(p,\alpha)}\}_{n\in\mathbb{N}}$. First, a discrete logarithm problem (DLP) is defined.

Definition 3 (DLP): For given p, α, β, compute $\log_\alpha \beta \bmod p$, where p and α is the output of $\mathsf{S}_H(1^n)$, β is the output of $\mathsf{K}_H(p, \alpha)$. ♦

The following lemma states that $\{H_n^{(p,\alpha)}\}_{n\in\mathbb{N}}$ is weakly CR even if the DLP turns out to be much easier than is believed. The proof is easy and omitted.

Lemma 4: Suppose that the DLP is weakly intractable, that is, the probability of failure to solve the DLP with respect to (p, α) produced by $\mathsf{S}_H(1^n)$ is not negligible: There exists some polynomial $\xi(n)$ such that, for every probabilistic polynomial-time algorithm J and every sufficiently large n,

$$\sum_{\beta \in G_p} \Pr[\mathsf{K}_H(p, \alpha) = \beta] \Pr[\mathsf{J}(p, \alpha, \beta) \text{ fails}] \geq \frac{1}{\xi(n)}.$$

Then, $\{H_n^{(p,\alpha)}\}_{n\in\mathbb{N}}$ is weakly CR. ♦

$\{H_n^{(p,\alpha)}\}_{n\in\mathbb{N}}$ is not strongly CR if the probability of success in solving the DLP is not negligible.

It is easy to see that both $h_\beta(x, \cdot)$ and $h_\beta(\cdot, x)$ are permutations for every $x \in \mathbb{Z}_q$ and $\beta \in G_p$. Thus, the following theorem is immediately lead from Lemma 4 and Theorem 2.

Theorem 3: A strongly CR family of hash functions is constructed if the DLP is weakly intractable. ♦

6. Second-Preimage Resistance

In this section, it is mentioned that similar results can be obtained for second-preimage resistance.

Let $h : D \to R$ be a hash function such that, for every $n \in \mathbb{N}$ and $x \in D \cap \{0,1\}^{\ell_D(n)}$, $h(x) \in R \cap \{0,1\}^{\ell_R(n)}$ and $\ell_R(n) < \ell_D(n)$. Both $\ell_D(n)$ and $\ell_R(n)$ are polynomials in n. Let $D_n = D \cap \{0,1\}^{\ell_D(n)}$ and $R_n = R \cap \{0,1\}^{\ell_R(n)}$.

A hash function h is called weakly second-preimage resistant (2nd-PR) if, for a given $x \in D$, the probability of failure to find another input x' such that $h(x) = h(x')$ is not negligible for every efficient algorithm.

Definition 4: A hash function h is called weakly 2nd-PR if

- there exists a probabilistic polynomial-time algorithm D_h which, with an input 1^n, outputs $x \in D_n$,
- there exists a deterministic polynomial-time algorithm M_h which, with an input $x \in D_n$, outputs $h(x)$, and
- there exists some polynomial $p(n)$ such that, for every probabilistic polynomial-time algorithm F_h and every sufficiently large n,

$$\sum_{x \in D_n} \Pr[\mathsf{D}_h(1^n) = x] \Pr[\mathsf{F}_h(x) \text{ fails}] \geq \frac{1}{p(n)},$$

where "$\mathsf{F}_h(x)$ fails" means that it fails to find another preimage in D_n and the probability is taken over the coin tosses of D_h and F_h. ♦

A hash function h is called strongly 2nd-PR if the probability of success in finding another preimage is negligible for every efficient algorithm.

Definition 5: A hash function h is called strongly 2nd-PR if

- there exist a probabilistic polynomial-time algorithm D_h and a deterministic polynomial-time algorithm M_h as stated in Definition 4, and
- for every polynomial $q(n)$, every probabilistic polynomial-time algorithm F_h and every sufficiently large n,

$$\sum_{x \in D_n} \Pr[\mathsf{D}_h(1^n) = x] \Pr[\mathsf{F}_h(x) \text{ succeeds}] < \frac{1}{q(n)},$$

where "$\mathsf{F}_h(x)$ succeeds" means that it succeeds in finding another preimage in D_n and the probability is taken over the coin tosses of D_h and F_h. ♦

The proofs of the following theorems are omitted because they are similar to the ones for CR.

Theorem 4: If there exists a strongly 2nd-PR hash function, then there exists a weakly 2nd-PR hash function which is not strongly 2nd-PR. ♦

Theorem 5: A strongly 2nd-PR hash function is able to be constructed from any weakly 2nd-PR hash function

$$h : \bigcup_{n \in \mathbb{N}} X_n \times X_n \to \bigcup_{n \in \mathbb{N}} Y_n$$

such that, for every $n \in \mathbb{N}$,

- for every $x_1, x_2 \in X_n$, $h(x_1, x_2) \in Y_n$,
- $|X_n| = |Y_n|$ and $h(x, \cdot)$ and $h(\cdot, x)$ are permutations for every $x \in X_n$. ♦

7. Conclusion

In this article, for cryptographic unkeyed hash functions, a definition of weak CR has been presented. Then, it has been shown that there really exists a weakly CR family of hash functions if there exists a CR family of hash functions. A method has also been presented to construct a strongly CR family of hash functions from a weakly CR one. With this method, a strongly CR family of hash functions is obtained if the discrete logarithm problem is only weakly intractable. A definition of weak 2nd-PR has also been presented, and it has been mentioned that similar results can be obtained for weak 2nd-PR to the ones for CR.

Acknowledgements

The author would like to thank one of the anonymous reviewers for his/her valuable comments which greatly improve the contents of Section 5.

References

[1] J.H. An and M. Bellare, “Constructing VIL-MACs from FIL-MACs: Message authentication under weakened assumptions,” CRYPTO’99 Proceedings, Lecture Notes in Computer Science 1666, pp.252–269, 1999.

[2] M. Bellare, R. Canetti, and H. Krawczyk, “Keying hash functions for message authentication,” CRYPTO’96 Proceedings, Lecture Notes in Computer Science 1109, pp.1–15, 1996.

[3] D. Chaum, E. van Heijst, and B. Pfitzmann, “Cryptographically strong undeniable signatures, unconditionally secure for the signer,” CRYPTO’91 Proceedings, Lecture Notes in Computer Science 576, pp.470–484, 1992.

[4] I. Damgård, “A design principle for hash functions,” CRYPTO’89 Proceedings, Lecture Notes in Computer Science 435, pp.416–427, 1990.

[5] O. Goldreich, Foundations of Cryptography: Basic Tools, Cambridge University Press, 2001.

[6] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[7] R.C. Merkle, “One way hash functions and DES,” CRYPTO’89 Proceedings, Lecture Notes in Computer Science 435, pp.428–446, 1990.

[8] B. Preneel, “The state of cryptographic hash functions,” in Lectures on Data Security, Lecture Notes in Computer Science 1561, pp.158–182, 1998.

[9] D.R. Simon, “Finding collisions on a one-way street: Can secure hash functions be based on general assumptions?,” EUROCRYPT’98 Proceedings, Lecture Notes in Computer Science 1403, pp.334–345, 1998.

[10] A.C.-C. Yao, “Theory and applications of trapdoor functions,” Proc. 23rd IEEE Symposium on Foundations of Computer Science, pp.80–91, 1982.

Shoichi Hirose received the B.E., M.E. and D.E. degrees in information science from Kyoto University, Kyoto, Japan, in 1988, 1990 and 1995, respectively. From 1990 to 1997, he was a research associate at Faculty of Engineering, Kyoto University. From 1998, he is a lecturer at the Graduate School of Informatics, Kyoto University. His current interests include cryptography, information security and computational complexity. He received Young Engineer Award from IEICE in 1997. He is a member of IACR (International Association for Cryptologic Research), ACM, IEEE and IPSJ.