2003－ＣＳＥＣ－22　(3)
2003／7／17

Double block length ハッシュ関数の解析

服部　充洋[†]　廣瀬　勝一[†]　吉田　進[†]

[†] 京都大学大学院情報学研究科　〒 606-8501 京都市左京区吉田本町
E-mail: †{hattori,hirose,yoshida}@hanase.kuee.kyoto-u.ac.jp

あらまし　本稿では，double block length ハッシュ関数およびその圧縮関数の安全性について検討する．第一に，Satoh，Haga，Kurosawa による double block length ハッシュ関数の解析結果を検討する．ここでは，彼等の解析結果のうち，鍵の長さが平文あるいは暗号文の長さの 2 倍であるブロック暗号から構成される圧縮関数をもつレート 1 の double block length ハッシュ関数の解析結果について検討する．この検討により，彼等の解析において一部考慮されていない場合のあることを示す．次に，効果的な攻撃法が知られていない double block length ハッシュ関数についてそれらを構成する圧縮関数の安全性を検討する．この検討により，これらの圧縮関数は，高々single block length ハッシュ関数の圧縮関数と同程度の安全性しかもたないことを示す．

キーワード　ハッシュ関数，ブロック暗号，暗号解析，double block length ハッシュ関数．

Analysis of Double Block Length Hash Functions

Mitsuhiro HATTORI[†], Shoichi HIROSE[†], and Susumu YOSHIDA[†]

[†] Graduate School of Informatics, Kyoto University
Yoshida-hommachi, Sakyo-ku, Kyoto, 606-8501 Japan
E-mail: †{hattori,hirose,yoshida}@hanase.kuee.kyoto-u.ac.jp

Abstract　The security of double block length hash functions and their compression functions is analyzed in this paper. First, the analysis of double block length hash functions by Satoh, Haga, and Kurosawa are investigated. The focus of this investigation is their analysis of the double block length hash functions with the rate 1 whose compression functions consist of a block cipher with the key twice longer than the plaintext/ciphertext. It is shown that there exists a case uncovered by their analysis. Second, the compression functions are analyzed with which secure double block length hash functions may be constructed. The analysis shows that these compression functions are at most as secure as the compression functions of single block length hash functions.

Key words　Hash functions, block ciphers, cryptanalysis, double block length hash functions.

1. Introduction

A hash function is a mapping from the set of all binary sequences to the set of binary sequences of some fixed length. It is one of the most important primitives in cryptography [1]. A hash function dedicated to cryptography is called a cryptographic hash function. Cryptographic hash functions are classified into unkeyed hash functions and keyed hash functions. In this paper, the unkeyed hash functions are discussed and they are simply called hash functions.

A hash function usually consists of a compression function. A compression function is the function $f : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}^a$. There are two major methods for construction of a compression function, namely, from scratch and based on a block cipher. The topic of this paper is the latter method. The main motivation of this construction is the minimization of design and implementation effort, which is supported by the expectation that secure hash functions can be constructed from secure block ciphers.

Hash functions based on block ciphers are classified into two categories: single block length hash functions and double block length hash functions. A single block length hash function is a hash function the length of whose output is equal to that of the block cipher. The length of the output of a double block length hash function is twice larger than that of the block cipher. The length of the output of a widely used block cipher is 64 or 128. Thus, single block length hash functions are no longer secure.

The compression functions of double block length hash functions are classified by the number of encryptions and the key length of the block cipher. The double block length hash functions with the compression functions with two encryptions of an (m, m) block cipher were analyzed in [2], [3], where an (m, k) block cipher is the one with the length of the plaintext/ciphertext m and the length of the key k. Satoh, Haga, and Kurosawa [4] analyzed the double block length hash functions with the compression functions with one encryption of an (m, m) or $(m, 2m)$ block cipher. They also made an analysis of the double block length hash functions with the compression functions with two encryptions of an $(m, 2m)$ block cipher. They stated that no effective attacks had not been found for the double block length hash functions with the compression functions, with two encryptions of an $(m, 2m)$ block cipher, satisfying the property exceptional defined by them.

In this paper, first, the analysis of double block length hash functions by Satoh, Haga, and Kurosawa is investigated. The focus of the investigation is their analysis of the double block length hash functions with the rate 1 whose compression functions consist of an $(m, 2m)$ block cipher. This investigation shows that there exists a case uncovered by their analysis. This result implies that there exist double block length hash functions whose compression functions do not satisfy the property exceptional and on which no effective attacks are found.

Second, for the double block length hash functions on which no effective attacks are known, their compression functions are analyzed. It is shown that all of these compression functions are at most as secure as those of single block length hash functions. Thus, even if there may exist secure double block length hash functions, it is impossible to prove it only by relying on the security of their compression functions.

The paper is organized as follows. Some definitions are introduced and mathematical facts are described in Section 2. Block-cipher-based hash functions are defined in Section 3. The analysis by Satoh et.al. is investigated in Section 4. In Section 5, the analysis of compression functions is described. Finally Section 6 concludes this paper with future work.

2. Preliminaries

$\mathbb{N}$ denotes the set of natural numbers. $\oplus$ denotes the bitwise exclusive OR. $a \| b$ denotes the concatenation of $a \in \{0,1\}^i$ and $b \in \{0,1\}^j$, where $a \| b \in \{0,1\}^{i+j}$.

2.1 Block Ciphers

A block cipher is a keyed function which maps an m-bit plaintext block to an m-bit ciphertext block. Let $\kappa, m \in \mathbb{N}$. An (m, k) block cipher is a mapping $E : \{0,1\}^\kappa \times \{0,1\}^m \rightarrow \{0,1\}^m$. For each $k \in \{0,1\}^\kappa$, the function $E_k(\cdot) = E(k, \cdot)$ is a one-to-one mapping from $\{0,1\}^m$ to $\{0,1\}^m$. $\{0,1\}^\kappa$ and $\{0,1\}^m$ in the domain $\{0,1\}^\kappa \times \{0,1\}^m$ and $\{0,1\}^m$ in the range are called the key space, the plaintext space, and the ciphertext space, respectively. m is called the block length and k is called the key length.

2.2 Hash Functions

2.2.1 Iterated Hash Functions

A hash function is a mapping from the set of all binary sequences to the set of binary sequences of some fixed length. A hash function is denoted by $h : \{0,1\}^* \rightarrow \{0,1\}^a$, where $\{0,1\}^* = \bigcup_{i \geq 0} \{0,1\}^i$.

A hash function $h : \{0,1\}^* \rightarrow \{0,1\}^a$ usually consists of a compression function $f : \{0,1\}^a \times \{0,1\}^b \rightarrow \{0,1\}^a$ and an initial value $IV \in \{0,1\}^a$. h is computed by the iterated application of f to the given input. Thus, h is called an iterated hash function. The output of the hash function h for an input $M \in \{0,1\}^*$, $h(M)$, is calculated as follows. M is called a message.

(Step 1) The message M is divided into the blocks of the equal length b. If the length of M is not a multiple of b, M is padded using an unambiguous padding rule. Let $M_1, M_2, \ldots, M_n$ be the blocks from the (padded) message M, where $M_i \in \{0,1\}^b$ for $i = 1, 2, \ldots, n$.

(Step 2) $H_i = f(H_{i-1}, M_i)$ is calculated for $i = 1, 2, \ldots, n$, where $H_i \in \{0,1\}^a$ and $H_0 = IV$. H_n is the output of h for the message M, that is $H_n = h(M)$. If the initial value should be specified, the equation is described as $H_n = h(H_0, M)$.

2.3 Properties Required for Hash Functions

For a hash function h, there exist many pairs $(M, \hat{M})$ such that $h(M) = h(\hat{M})$ and $M \neq \hat{M}$. For cryptographic use, the hash function h must satisfy the following properties.

preimage resistance Given a hash value H, it is computationally infeasible to find a message M such that $h(M) = H$.

second preimage resistance Given a message M, it is computationally infeasible to find a message $\hat{M}$ such that $h(M) = h(\hat{M})$ and $M \neq \hat{M}$.

collision resistance It is computationally infeasible to find a pair of messages, M and $\hat{M}$, such that $h(M) = h(\hat{M})$ and $M \neq \hat{M}$.

The relationships among the properties are [1]:

- If a hash function satisfies the second preimage resistance, then it also satisfies the preimage resistance, and
- If a hash function satisfies the collision resistance, then it also satisfies the second preimage resistance.

Therefore, it is the easiest to satisfy preimage resistance, and it is the most difficult to satisfy collision resistance.

2.4 Attacks on Hash Functions

The following attacks [3] are against the properties listed in Section 2.3.

the preimage attack Given an initial value H_0 and a hash value H, find a message M such that $H = h(H_0, M)$.

the second preimage attack Given an initial value H_0 and a message M, find a message $\hat{M}$ such that $h(H_0, M) = h(H_0, \hat{M})$ and $M \neq \hat{M}$.

the free-start preimage attack Given a hash value H, find an initial value H_0 and a message M such that $h(H_0, M) = H$.

the free-start second preimage attack Given an initial value H_0 and a message M, find an initial value $\hat{H}_0$ and a message $\hat{M}$ such that $h(H_0, M) = h(\hat{H}_0, \hat{M})$ and $(H_0, M) \neq (\hat{H}_0, \hat{M})$.

the collision attack Given an initial value H_0, find two messages M, $\hat{M}$ such that $h(H_0, M) = h(H_0, \hat{M})$ and $M \neq \hat{M}$.

the semi-free-start collision attack Find an initial value H_0 and two messages M, $\hat{M}$ such that $h(H_0, M) = h(H_0, \hat{M})$ and $M \neq \hat{M}$.

the free-start collision attack Find two initial values H_0, $\hat{H}_0$ and two messages M, $\hat{M}$ such that $h(H_0, M) = h(\hat{H}_0, \hat{M})$ and $(H_0, M) \neq (\hat{H}_0, \hat{M})$.

The following two propositions [5] are often used to estimate the amount of computation of the attacks.

[Proposition 1] Suppose that a sample of size r is drawn from a set of N elements with replacement. If $r, N \to \infty$, then the probability that a given element is drawn converges to

$$1 - \exp\left(-\frac{r}{N}\right). \tag{1}$$

[Proposition 2] (Birthday Paradox) Suppose that a sample of size r is drawn from a set of N elements with replacement. If $r, N \to \infty$ and r is $O(\sqrt{N})$, then the probability that there is at least one coincidence is converges to

$$1 - \exp\left(-\frac{r^2}{2N}\right). \tag{2}$$

3. Hash Functions Based on Block Ciphers

3.1 Compression Function Construction

There are two major methods for constructing compression functions: construction based on block ciphers and construction from scratch. The topic of this paper is the former construction.

3.2 Single Block Length Hash Functions and Double Block Length Hash Functions

Let $h : \{0,1\}^* \to \{0,1\}^a$ be an iterated hash function and $E : \{0,1\}^\kappa \times \{0,1\}^m \to \{0,1\}^m$ be a block cipher used in the compression function of h. If $a = m$, then h is called a single block length hash function. If $a = 2m$, then h is called a double block length hash function.

Let σ be the number of the encryptions of the block cipher used in the compression function. Let $b = |M_i|$. Then, the rate is defined as $b/(\sigma \cdot m)$ and is used as a speed index.

3.3 Hash Functions Considered in This Paper

We consider double block length hash functions with the rate 1, whose compression functions are composed of an $(m, 2m)$ block cipher.

Let $M_i = (M_i^1, M_i^2) \in \{0,1\}^{2m}$ be a message block, where $M_i^1, M_i^2 \in \{0,1\}^m$. The compression function $H_i = f(H_{i-1}, M_i)$ is defined by the two functions f^1, f^2 such as

$$\left\{\begin{array}{lcl} H_i^1 & = & f^1(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2) \\ H_i^2 & = & f^2(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2), \end{array}\right. \tag{3}$$

where $H_j = (H_j^1, H_j^2)$ and $H_j^1, H_j^2 \in \{0,1\}^m$, for $j = i-1, i$. Each of f^1 and f^2 contains one encryption of the $(m, 2m)$ block cipher E. H_i^1 and H_i^2 are represented by

$$\left\{\begin{array}{lcl} H_i^1 & = & E_{A\|B}(C) \oplus D \\ H_i^2 & = & E_{W\|X}(Y) \oplus Z, \end{array}\right. \tag{4}$$

where $A, B, C, D, W, X, Y, Z \in \{0,1\}^m$. A, B, C, D, W, X, Y and Z are represented by linear combinations of $H_{i-1}^1, H_{i-1}^2, M_i^1$ and M_i^2 as follows:

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = L_1 \begin{pmatrix} H_{i-1}^1 \\ H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} W \\ X \\ Y \\ Z \end{pmatrix} = L_2 \begin{pmatrix} H_{i-1}^1 \\ H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{pmatrix}, \tag{6}$$

where L_1 and L_2 are 4×4 binary matrices.

Let $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ denote row vectors of L_1 and let $\boldsymbol{w}$, $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ denote row vectors of L_2.

3.4 The Black-Box Model

The black-box model [6] is used in our analysis. In this model, a block cipher is assumed to be random, that is, $E_k : \{0,1\}^m \to \{0,1\}^m$ is a random permutation for each $k \in \{0,1\}^\kappa$. The oracle E^{-1}, on input (k, y), returns x such that $E_k(x) = y$.

4. A Comment on the Analysis by Satoh, Haga and Kurosawa

Satoh, Haga, and Kurosawa [4] have analyzed the security of the double block length hash functions defined in Section 3.3. In this section, their analysis is investigated. It is shown that there exists a case uncovered by their analysis.

4.1 Analysis by Satoh et.al.

Satoh et.al. presented the following claim (Theorem 16 in their paper [4]).

[Claim 1] For the double block length hash functions of the rate 1, whose round function has the form of (4), suppose that at least one of L_1 and L_2 is not *exceptional*. Then, there exist second preimage and preimage attacks with about 4×2^m complexity. Furthermore, there exists a collision attack with about $3 \times 2^{m/2}$ complexity.

Throughout this paper, as in the above claim, the complexity of an attack is the required number of encryptions and decryptions of the block cipher.

The notion of *exceptional* is defined as follows.

[Definition 1] Let L be a 4×4 binary matrix. Let L_r be the 4×2 submatrix of L, where L_r consists of the right half elements of L. Let L_r^3 be the 3×2 submatrix of L_r such that the third row of L_r is deleted. Let L_r^4 be the 3×2 submatrix of L_r such that the fourth row of L_r is deleted. L is called *exceptional* if $\mathrm{Rank}(L) = 4$ and $\mathrm{Rank}(L_r^3) = \mathrm{Rank}(L_r^4) = 2$.

4.2 A Comment

Let N_2 be the 2×2 submatrix of L_r, where N_2 consists of the upper half elements of L_r. Satoh et.al. presented their proof of Claim 1 for two cases: (i) $\mathrm{Rank}(L) = 3$ and $\mathrm{Rank}(N_2) = 2$ and (ii) $\mathrm{Rank}(L) = 4$. The first case is investigated in the remaining part. Their proof proceeds as follows.

Since $\mathrm{Rank}(N_2) = 2$, one can find (by elementary row operations) $\alpha, \beta = 0, 1$ such that

$$L' = \begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \\ \boldsymbol{c} \\ \boldsymbol{d} \oplus \alpha\boldsymbol{a} \oplus \beta\boldsymbol{b} \end{pmatrix} = \begin{pmatrix} N_1 & N_2 \\ N_3' & N_4' \end{pmatrix}, \qquad (7)$$

where

$$N_4' = \begin{pmatrix} * & * \\ 0 & 0 \end{pmatrix}. \qquad (8)$$

Let

$$\begin{pmatrix} A \\ B \\ C \\ D' \end{pmatrix} = L' \begin{pmatrix} H_{n-1}^1 \\ H_{n-1}^2 \\ M_n^1 \\ M_n^2 \end{pmatrix}. \qquad (9)$$

Then, $D' = 0$, H_{n-1}^1, H_{n-1}^2 or $H_{n-1}^1 \oplus H_{n-1}^2$.

Subsequently, they stated in their proofs that $\boldsymbol{c} = \lambda_1\boldsymbol{a} \oplus \lambda_2\boldsymbol{b}$ when $D' \neq 0$. However, in general, there may be a case that $\boldsymbol{c} = \lambda_1\boldsymbol{a} \oplus \lambda_2\boldsymbol{b} \oplus \boldsymbol{d}$ even if $D' \neq 0$. Furthermore, in this case, their attack for the case that $\mathrm{Rank}(L) = 3$, $\mathrm{Rank}(N_2) = 2$, and $D' \neq 0$ cannot be applied.

In their attack, the adversary chooses random triples (A, B, C) such that $C = \lambda_1 A \oplus \lambda_2 B$ and computes $D = E_{A\|B}(C) \oplus H_n^1$. Then the adversary computes $D' = D \oplus \alpha A \oplus \beta B$. However, if $\boldsymbol{c} = \lambda_1\boldsymbol{a} \oplus \lambda_2\boldsymbol{b} \oplus \boldsymbol{d}$, C is calculated by A, B, and D. Therefore, the adversary cannot compute D by $E_{A\|B}(C) \oplus H_n^1$.

5. Collision-Resistance of Compression Functions

From the results by Satoh, Haga, and Kurosawa and the discussion in the last section, no effective attacks are found in the double block length hash functions defined in Section 3.3 with compression functions satisfying

(i) the property exceptional, or

(ii) $\mathrm{Rank}(L_1) = \mathrm{Rank}(L_2) = 3$, $\mathrm{Rank}(N_{1,2}) = \mathrm{Rank}(N_{2,2}) = 2$, and

$\boldsymbol{c} \oplus \boldsymbol{d} = \lambda_1\boldsymbol{a} \oplus \lambda_2\boldsymbol{b}$ for some $\lambda_1, \lambda_2 \in \{0, 1\}$

$\boldsymbol{y} \oplus \boldsymbol{z} = \lambda_3\boldsymbol{w} \oplus \lambda_4\boldsymbol{x}$ for some $\lambda_3, \lambda_4 \in \{0, 1\}$,

where $N_{i,2}$ is the upper right 2×2 submatrix of L_i for $i = 1, 2$.

In this section, some effective attacks are presented on compression functions satisfying $\mathrm{Rank}(L_1) \geqq 3$ and $\mathrm{Rank}(L_2) \geqq 3$. This implies that it is impossible to prove the security of double block length hash functions, on which no effective attacks are found, mentioned above, only by relying on the security of their compression functions.

[Theorem 1] Let f be any compression function represented by the equation (4) such that $\mathrm{Rank}(L_1) \geqq 3$ and $\mathrm{Rank}(L_2) \geqq 3$. Then, there exist a free-start second preimage attack and a free-start collision attack on f with complexities about 2×2^m and $2 \times 2^{m/2}$, respectively.

This theorem is proved for the following two cases:

1. $\mathrm{Rank}(L_1) = 4$ or $\mathrm{Rank}(L_2) = 4$,
2. $\mathrm{Rank}(L_1) = \mathrm{Rank}(L_2) = 3$.

The following lemma is for the former case.

[Lemma 1] Suppose that $\mathrm{Rank}(L_1) \geqq 3$ and $\mathrm{Rank}(L_2) \geqq 3$. If $\mathrm{Rank}(L_1) = 4$ or $\mathrm{Rank}(L_2) = 4$, then there exist a free-start (second) preimage attack and a free-start collision attack with complexities about 2×2^m and $2 \times 2^{m/2}$, respectively.

(Proof) Without loss of generality, it is assumed that $\mathrm{Rank}(L_1) = 4$.

Since $\mathrm{Rank}(L_1) = 4$, from (5),

$$\begin{pmatrix} H_{i-1}^1 \\ H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{pmatrix} = L_1^{-1} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix}. \qquad (10)$$

For the (second) preimage attack, the adversary *Adv* proceeds as follows.

the free-start (second) preimage attack

(Step 0) (This step is only for the free-start second preimage attack.) *Adv* computes the output (H_i^1, H_i^2) from the given input $(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2)$.

(Step 1) *Adv* chooses 2^m random triples $(\tilde{A}, \tilde{B}, \tilde{C})$ and computes $\tilde{D} = E_{\tilde{A}\|\tilde{B}}(\tilde{C}) \oplus H_i^1$. Since the block cipher is assumed to be random, $\tilde{D}$ is also random.

(Step 2) For each 4-tuples $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$, *Adv* computes $(\tilde{H}_{i-1}^1, \tilde{H}_{i-1}^2, \tilde{M}_i^1, \tilde{M}_i^2)$ with (10). Since $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$ is random, $(\tilde{H}_{i-1}^1, \tilde{H}_{i-1}^2, \tilde{M}_i^1, \tilde{M}_i^2)$ is also random.

(Step 3) For each $(\tilde{H}_{i-1}^1, \tilde{H}_{i-1}^2, \tilde{M}_i^1, \tilde{M}_i^2)$, *Adv* computes $(\tilde{W}, \tilde{X}, \tilde{Y}, \tilde{Z})$ with (6) and computes $\tilde{H}_i^2 = E_{\tilde{W}\|\tilde{X}}(\tilde{Y}) \oplus \tilde{Z}$.

Since $\mathrm{Rank}(L_2) \geqq 3$, $(\boldsymbol{w}, \boldsymbol{x}) \neq (\boldsymbol{0}, \boldsymbol{0})$. Therefore, at least one of $\tilde{W}$ and $\tilde{X}$ is expressed by a linear combination of $\tilde{H}_{i-1}^1$, $\tilde{H}_{i-1}^2$, $\tilde{M}_i^1$ and $\tilde{M}_i^2$. Since $\tilde{H}_{i-1}^1$, $\tilde{H}_{i-1}^2$, $\tilde{M}_i^1$ and $\tilde{M}_i^2$ are random, $E_{\tilde{W}\|\tilde{X}}(\tilde{Y})$ is random, and $\tilde{H}_i^2$ is also random. Thus, according to Proposition 1, *Adv* can find $\tilde{H}_i^2$ such that $H_i^2 = \tilde{H}_i^2$ with probability about 0.63. The total complexity is about 2×2^m.

For the free-start collision attack, *Adv* proceeds as follows.

the free-start collision attack

Adv chooses arbitrary H_i^1. Then it chooses $2^{m/2}$ random triples $(\tilde{A}, \tilde{B}, \tilde{C})$ and computes $\tilde{H}_i^2$ in the same way as in the steps 1–3 above. According to Proposition 2, *Adv* can find a collision of f^2 with probability about 0.39. The total complexity is about $2 \times 2^{m/2}$. □

[Lemma 2] Suppose that $\mathrm{Rank}(L_1) = \mathrm{Rank}(L_2) = 3$. Then, there exist a free-start (second) preimage attack and a free-start collision attack with complexities abouot 2×2^m and $2 \times 2^{m/2}$, respectively.

This lemma is lead from the following two lemmas.

[Lemma 3] Suppose that $\mathrm{Rank}(L_1) = \mathrm{Rank}(L_2) = 3$. If $\boldsymbol{c} \oplus \boldsymbol{d}$ is not represented by any linear combination of $\boldsymbol{a}$ and $\boldsymbol{b}$, or $\boldsymbol{y} \oplus \boldsymbol{z}$ is not represented by any linear combination of $\boldsymbol{w}$ and $\boldsymbol{x}$, then there exist a free-start (second) preimage attack and a free-start collision attack with complexities about 2×2^m and $2 \times 2^{m/2}$, respectively.

(Proof)

Without loss of generality, $\boldsymbol{c} \oplus \boldsymbol{d}$ is not assumed to be represented by any linear combination of $\boldsymbol{a}$ and $\boldsymbol{b}$.

Since $\mathrm{Rank}(L_1) = 3$, the following three cases should be considered:

(a) $\boldsymbol{a}$ and $\boldsymbol{b}$ are linearly dependent,

(b) $\boldsymbol{a}$ and $\boldsymbol{b}$ are linearly independent and $\boldsymbol{c} = \lambda_1 \boldsymbol{a} \oplus \lambda_2 \boldsymbol{b}$ for some $\lambda_1, \lambda_2 \in \{0, 1\}$,

(c) $\boldsymbol{a}$ and $\boldsymbol{b}$ are linearly independent and $\boldsymbol{d} = \lambda_1 \boldsymbol{a} \oplus \lambda_2 \boldsymbol{b}$ for some $\lambda_1, \lambda_2 \in \{0, 1\}$.

In the case (a), either $\boldsymbol{a}$ or $\boldsymbol{b}$ is $\boldsymbol{0}$, or $\boldsymbol{a} = \boldsymbol{b}$. Thus the key $A\|B$ is $A\|0$, $0\|B$, or $A\|A$. Without loss of generality, the key is assumed $A\|A$. Then, from the equation (5),

$$\begin{pmatrix} A \\ C \\ D \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{h}^1 & \boldsymbol{h}^2 & \boldsymbol{m}^1 & \boldsymbol{m}^2 \end{pmatrix}}_{L'} \begin{pmatrix} H_{i-1}^1 \\ H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{pmatrix}, \qquad (11)$$

where L' is the 3×4 submatrix of L_1 and $\boldsymbol{h}^1$, $\boldsymbol{h}^2$, $\boldsymbol{m}^1$, and $\boldsymbol{m}^2$ are the column vectors of L'. Since $\mathrm{Rank}(L') = 3$, one column vector of L' is expressed by a linear combination of the other column vectors. Without loss of generality, it is assumed that $\boldsymbol{h}^1$ is expressed by a linear combination of the other vectors. Then,

$$\begin{pmatrix} A \\ C \\ D \end{pmatrix} = L'' \begin{pmatrix} H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{pmatrix} \oplus \boldsymbol{h}^1 H_{i-1}^1, \qquad (12)$$

where $L'' = \begin{pmatrix} \boldsymbol{h}^2 & \boldsymbol{m}^1 & \boldsymbol{m}^2 \end{pmatrix}$. Since $\mathrm{Rank}(L'') = 3$,

$$\begin{pmatrix} H_{i-1}^2 \\ M_i^1 \\ M_i^2 \end{pmatrix} = L''^{-1} \left(\begin{pmatrix} A \\ C \\ D \end{pmatrix} \oplus \boldsymbol{h}^1 H_{i-1}^1 \right). \qquad (13)$$

the free-start (second) preimage attack

(Step 0) (This step is only for the free-start second preimage attack.) *Adv* computes the output (H_i^1, H_i^2) for the given input $(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2)$.

(Step 1) The adversary *Adv* chooses 2^m random 2-tuples $(\tilde{A}, \tilde{C})$ and computes $\tilde{D} = E_{\tilde{A}\|\tilde{A}}(\tilde{C}) \oplus H_i^1$.

(Step 2) For each $(\tilde{A}, \tilde{C}, \tilde{D})$, *Adv* chooses a random $\tilde{H}_{i-1}^1$ and computes $(\tilde{H}_{i-1}^2, \tilde{M}_i^1 \tilde{M}_i^2)$ from (13).

(Step 3) For each $(\tilde{H}_{i-1}^1, \tilde{H}_{i-1}^2, \tilde{M}_i^1, \tilde{M}_i^2)$, *Adv* computes $(\tilde{W}, \tilde{X}, \tilde{Y}, \tilde{Z})$ with (6) and computes $\tilde{H}_i^2 = E_{\tilde{W}\|\tilde{X}}(\tilde{Y}) \oplus \tilde{Z}$. Since $\mathrm{Rank}(L_2) = 3$, $(\boldsymbol{w}, \boldsymbol{x}) \neq (\boldsymbol{0}, \boldsymbol{0})$. Therefore, at least one of $\tilde{W}$ and $\tilde{X}$ is expressed by a linear combination of $\tilde{H}_{i-1}^1$, $\tilde{H}_{i-1}^2$, $\tilde{M}_i^1$ and $\tilde{M}_i^2$. Since $\tilde{H}_{i-1}^1$, $\tilde{H}_{i-1}^2$, $\tilde{M}_i^1$ and $\tilde{M}_i^2$ are random, $E_{\tilde{W}\|\tilde{X}}(\tilde{Y})$ is random, and $\tilde{H}_i^2$ is also random. Thus, according to Proposition 1, *Adv* can find $\tilde{H}_i^2$ such that $H_i^2 = \tilde{H}_i^2$ with probability about 0.63. The total complexity is about 2×2^m.

the free-start collision attack

Adv chooses arbitrary H_i^1. *Adv* chooses $2^{m/2}$ random 2-tuples $(\tilde{A}, \tilde{C})$ and computes $\tilde{D} = E_{\tilde{A}\|\tilde{A}}(\tilde{C}) \oplus H_i^1$. After that, it computes $\tilde{H}_i^2$ in the same way as in the steps 2–3 above. According to Proposition 2, *Adv* can find a collision of f^2 with probability about 0.39. The total complexity is about $2 \times 2^{m/2}$.

In the case (b), the adversary *Adv* proceeds as follows.

the free-start (second) preimage attack

(Step 0) (This step is only for the free-start second preimage attack.) *Adv* computes the output (H_i^1, H_i^2) for the given input $(H_{i-1}^1, H_{i-1}^2, M_i^1, M_i^2)$.

(Step 1) *Adv* chooses 2^m random triples $(\tilde{A},\tilde{B},\tilde{C})$ such that $\tilde{C}=\lambda_1\tilde{A}\oplus\lambda_1\tilde{B}$ and computes $\tilde{D}=E_{\tilde{A}\|\tilde{B}}(\tilde{C})\oplus H_i^1$.
(Step 2) For each $(\tilde{A},\tilde{B},\tilde{C},\tilde{D})$, *Adv* chooses a random 4-tuple $(\tilde{H}_{i-1}^1,\tilde{H}_{i-1}^2,\tilde{M}_i^1,\tilde{M}_i^2)$ which satisfies (5).
(Step 3) For each $(\tilde{H}_{i-1}^1,\tilde{H}_{i-1}^2,\tilde{M}_i^1,\tilde{M}_i^2)$, *Adv* computes $(\tilde{W},\tilde{X},\tilde{Y},\tilde{Z})$ with (6) and computes $\tilde{H}_i^2=E_{\tilde{W}\|\tilde{X}}(\tilde{Y})\oplus\tilde{Z}$. Since $\mathrm{Rank}(L_2)=3$, $(\boldsymbol{w},\boldsymbol{x})\neq(\boldsymbol{0},\boldsymbol{0})$. Therefore, at least one of $\tilde{W}$ and $\tilde{X}$ is expressed by a linear combination of $\tilde{H}_{i-1}^1$, $\tilde{H}_{i-1}^2$, $\tilde{M}_i^1$ and $\tilde{M}_i^2$. Since $\tilde{H}_{i-1}^1$, $\tilde{H}_{i-1}^2$, $\tilde{M}_i^1$ and $\tilde{M}_i^2$ are random, $E_{\tilde{W}\|\tilde{X}}(\tilde{Y})$ is random, and $\tilde{H}_i^2$ is also random. Thus, according to Proposition 1, *Adv* can find $\tilde{H}_i^2$ such that $H_i^2=\tilde{H}_i^2$ with probability about 0.63. The total complexity is about 2×2^m.

the free-start collision attack

Adv chooses arbitrary H_i^1. Then it chooses $2^{m/2}$ random triples $(\tilde{A},\tilde{B},\tilde{C})$ such that $\tilde{C}=\lambda_1\tilde{A}\oplus\lambda_2\tilde{B}$ and computes $\tilde{H}_i^2$ in the same way as in the steps (1)–(3). According to Proposition 2, *Adv* can find a collision of f^2 with probability about 0.39. The total complexity is about $2\times 2^{m/2}$.

The attacks in the case (c) are almost similar to those in the case (b). □

The next lemma is also for the case that $\mathrm{Rank}(L_1)=\mathrm{Rank}(L_2)=3$. In one part of this case, no effective free-start preimage attack is found.

[Lemma 4] Suppose that $\mathrm{Rank}(L_1)=\mathrm{Rank}(L_2)=3$. If $\boldsymbol{c}\oplus\boldsymbol{d}=\lambda_1\boldsymbol{a}\oplus\lambda_2\boldsymbol{b}$ and $\boldsymbol{y}\oplus\boldsymbol{z}=\lambda_3\boldsymbol{w}\oplus\lambda_4\boldsymbol{x}$ for some $\lambda_1,\lambda_2,\lambda_3,\lambda_4\in\{0,1\}$, then there exist a free-start second preimage attack and a free-start collision attack with complexities about 2^m and $2^{m/2}$, respectively.

(Proof)

Since $\mathrm{Rank}(L_1)=3$, the number of the 4-tuples $(H_{i-1}^1,H_{i-1}^2,M_i^1,M_i^2)$ which correspond to the same (A,B,C,D) is 2^m. Let $\mathcal{V}$ be the set of 4-tuples $(H_{i-1}^1,H_{i-1}^2,M_i^1,M_i^2)$ corresponding to the same (A,B,C,D). The following three cases are considered.
Case (I) If all of $\boldsymbol{w}$, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ are represented by linear combinations of $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$, then (W,X,Y,Z) is constant for every $(H_{i-1}^1,H_{i-1}^2,M_i^1,M_i^2)\in\mathcal{V}$.
Case (II) If at least one of $\boldsymbol{w}$ and $\boldsymbol{x}$ is not represented by any linear combination of $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$, then $W\|X$ takes 2^m different values for the elements in $\mathcal{V}$.
Case (III) If both $\boldsymbol{w}$ and $\boldsymbol{x}$ are represented by linear combinations of $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$, and $\boldsymbol{y}$ is not represented by any linear combination of $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$, then $\boldsymbol{z}$ is not represented by any linear combination of $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$ since $\boldsymbol{y}\oplus\boldsymbol{z}=\lambda_3\boldsymbol{w}\oplus\lambda_4\boldsymbol{x}$ for some $\lambda_3,\lambda_4\in\{0,1\}$. In this case, both Y and Z take 2^m different values for the elements in $\mathcal{V}$.

the free-start second preimage attack

(Step 1) *Adv* computes the output (H_i^1,H_i^2) for the given input $(H_{i-1}^1,H_{i-1}^2,M_i^1,M_i^2)$.
(Step 2) *Adv* repeatedly chooses a random 4-tuple $(\tilde{H}_{i-1}^1,\tilde{H}_{i-1}^2,\tilde{M}_i^1,\tilde{M}_i^2)$ corresponding to (A,B,C,D) such that

$$\begin{pmatrix}A\\B\\C\\D\end{pmatrix}=L_1\begin{pmatrix}H_{i-1}^1\\H_{i-1}^2\\M_i^1\\M_i^2\end{pmatrix},\tag{14}$$

and computes $(\tilde{W},\tilde{X},\tilde{Y},\tilde{Z})$ by (6) and $\tilde{H}_i^2=E_{\tilde{W}\|\tilde{X}}(\tilde{Y})\oplus\tilde{Z}$ until $H_i^2=\tilde{H}_i^2$.

This succeeds

- with probability about 1 and with complexity 3 for Case (I),
- with probability about 0.63 and with complexity about 2^m for Case (II) and Case (III).

the collision attack

(Step 1) *Adv* chooses arbitrary $(H_{i-1}^1,H_{i-1}^2,M_i^1,M_i^2)$ and computes H_i^1.
(Step 2) *Adv* repeatedly chooses a random 4-tuple $(\tilde{H}_{i-1}^1,\tilde{H}_{i-1}^2,\tilde{M}_i^1,\tilde{M}_i^2)$ corresponding to (A,B,C,D) such that

$$\begin{pmatrix}A\\B\\C\\D\end{pmatrix}=L_1\begin{pmatrix}H_{i-1}^1\\H_{i-1}^2\\M_i^1\\M_i^2\end{pmatrix},\tag{15}$$

and computes $(\tilde{W},\tilde{X},\tilde{Y},\tilde{Z})$ by (6) and $\tilde{H}_i^2=E_{\tilde{W}\|\tilde{X}}(\tilde{Y})\oplus\tilde{Z}$ until a collision of f^2 is found.

This succeeds

- with probability about 1 and with complexity 3 for Case (I),
- with probability about 0.39 and with complexity about $2^{m/2}$ for Case (II) and Case (III). □

In Lemma 4, if at least one of $\boldsymbol{w}$, $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ is linearly independent of $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, $\boldsymbol{d}$, then there exists a free-start preimage attack with probability 0.63^2 and with complexity 2×2^m. This attack is obtained from the free-start second preimage attack in the proof of Lemma 4 by replacing Step 0 with the following Step 0',
(Step 0') *Adv* repeatedly chooses $(H_{i-1}^1,H_{i-1}^2,M_i^1,M_i^2)$ at random until the output corresponding to it is equal to the first half of the given output (H_i^1,H_i^2).
This step succeeds with probability about 0.63 and with complexity about 2^m.

6. Conclusion

The security of double block length hash functions and their compression functions have been analyzed. First, the analysis by Satoh, Haga, and Kurosawa on double block length hash functions have been investigated and it has been shown that there is a case uncovered by their analysis. Then, some effective attacks have been presented on the compression functions which may produce secure double block length hash functions.

Future work includes the analysis of the double block length hash functions whose security remains unclear.

References

[1] A. Menezes, P. van Oorschot, and S. Vanstone, "*Handbook of Applied Cryptography*," CRC Press, 1996.

[2] L. Knudsen and X. Lai, "New attacks on all double block length hash functions of hash rate 1, including the paralled-DM," *EUROCRYPT'94*, *Lecture Notes in Computer Science*, vol. 950, pp.410–418, 1995.

[3] L. Knudsen, X. Lai, and B. Preneel, "Attacks on fast double block length hash functions," *Journal of Cryptology*, vol. 11, no. 1, pp.59–72, 1998.

[4] T. Satoh, M. Haga, and K. Kurosawa, "Towards secure and fast hash functions," *IEICE Transactions of Fundamentals*, vol. 82-A, no. 1, pp.55–62, 1999.

[5] M. Girault, R. Cohen, and M. Campana, "A generalized birthday attack," *EUROCRYPT'88*, *Lecture Notes in Computer Science*, vol. 330, pp.129–156, 1988

[6] J. Black, P. Rogaway, and T. Shrimpton, "Black-box analysis of the block-cipher-based hash-function constructions from PGV," *CRYPTO'02*, *Lecture Notes in Computer Science*, vol. 2442, pp.320–335, 2002.