Provably Secure Double-Block-Length Hash Functions in a Black-Box Model

Provably Secure Double-Block-Length Hash Functions in a Black-Box Model

Shoichi Hirose

Graduate School of Informatics, Kyoto University, Kyoto 606-8501 Japan
hirose@i.kyoto-u.ac.jp

Abstract. In CRYPTO'89, Merkle presented three double-block-length hash functions based on DES. They are optimally collision resistant in a black-box model, that is, the time complexity of any collision-finding algorithm for them is $\Omega(2^{\ell/2})$ if DES is a random block cipher, where ℓ is the output length. Their drawback is that their rates are low. In this article, new double-block-length hash functions with higher rates are presented which are also optimally collision resistant in the black-box model. They are composed of block ciphers whose key length is twice larger than their block length.

keywords: double-block-length hash function, black-box model, block cipher

1 Introduction

A cryptographic hash function is a function which maps an input of arbitrary length to an output of fixed length. It is one of the most important primitives in cryptography [14] and should satisfy preimage resistance, second-preimage resistance and collision resistance. Informally, preimage resistance means that, given an output, it is infeasible to obtain an input which produces the output. 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. For simplicity, a cryptographic hash function is called a hash function in this article.

A hash function usually consists of iteration of a compression function with fixed input/output length and is called an iterated hash function. Compression-function constructions are classified into two types: based on block ciphers and from scratch. The topic of this article is the former. It minimizes design and implementation effort with secure block ciphers. Its major drawback is slow processing speed. However, it is compensated by fast block ciphers such as AES. Furthermore, some recent work has pointed out weakness of SHA families [1, 18]. Thus, block-cipher-based hash functions may become more important.

Block-cipher-based hash functions are classified into two categories: single-block-length (SBL) and double-block-length (DBL). A SBL hash function is a hash function whose output length is equal to the block length. The output length of a DBL hash function is twice larger than the block length.

It is well-known that the birthday attack can find a collision of a hash function with time complexity $O(2^{\ell/2})$, where ℓ is the output length of the hash function. The block length of widely used block ciphers is 64 or 128. Thus, SBL hash functions are no longer secure in terms of collision resistance.

For DBL hash functions, many constructions have been presented [4, 7–10, 12, 15]. Among them, three DBL hash functions by Merkle [15] have been shown to be optimally collision resistant in a black-box model: the time complexity of any collision-finding algorithm for them is $\Omega(2^{\ell/2})$, where ℓ is the output length. However, their rates are at most 0.276 and they are not so efficient.

In this article, DBL hash functions are proposed which are more efficient and optimally collision resistant in the black-box model. They can be represented in a simple form. They are of parallel type and their rates are 1/2. They are based on block ciphers whose key length is twice larger than the block length. Thus, they can be constructed with AES or other previous AES candidates, which support 128-bit blocks and 256-bit keys.

The DBL hash functions proposed in this article consist of two different block ciphers to be provably secure. Though it seems their drawback, a genuine tweakable block cipher [13] will help obtain virtually two different block ciphers with different tweaks. Furthermore, it is possible to transform a DBL hash function with different block ciphers to the one with only one block cipher with slightly lower rate by the method used in MDC-2 [4].

Collision resistance as well as preimage resistance of the proposed DBL hash functions is proved in the black-box model. In this model, for the proposed DBL hash functions, second-preimage resistance can be regarded as preimage resistance for the output corresponding to the given input. In the black-box model, a block cipher is assumed to be an invertible keyed random permutation. This is an ideal but still proper assumption in that most of the attacks on block-cipher-based hash functions do not utilize the internal structure of the block ciphers. The technique in [3] is used in the security proofs in this article. It is assumed that two block ciphers are independent in our analysis.

The rest of this article is organized as follows. Section 2 includes notations, definitions and related work. In Section 3, provably secure DBL hash functions with rate 1/2 consisting of two block ciphers are presented. Security proofs are also shown. In Section 4, it is mentioned how to construct provably secure DBL hash functions with one block cipher. A concluding remark is given in Section 5.

2 Preliminaries

2.1 Related Work

Preneel, Govaerts and Vandewalle [16] discussed the security of SBL hash functions against several attacks. They considered SBL hash functions with compression functions represented by $h_i = e(k, x) \oplus z$, where e is an (n, n) block cipher, $k, x, z \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i, v\}$ and v is a constant. They concluded that 12 out of $64 (= 4^3)$ hash functions are secure against the attacks. However, they did not provide any formal proofs.

Black, Rogaway and Shrimpton [3] presented a detailed investigation of provable security of SBL hash functions given in [16] in the black-box model. The most important result shown in their paper is that the time complexity of any collision-finding algorithm against 20 hash functions including the 12 mentioned above is $\Omega(2^{\ell/2})$, where ℓ is the output length.

Knudsen, Lai and Preneel [11] discussed the security of DBL hash functions with rate 1 based on (n, n) block ciphers. Hohl, Lai, Meier and Waldvogel [7] discussed the security of compression functions of DBL hash functions with rate 1/2. On the other hand, the security of DBL hash functions with rate 1 based on $(n, 2n)$ block ciphers was discussed by Satoh, Haga and Kurosawa [17] and by Hattori, Hirose and Yoshida [6].

Many schemes with rate less than 1 were also presented. Merkle [15] presented three DBL hash functions based on DES with rates at most 0.276. They are optimally collision resistant in the black-box model. MDC-2 and MDC-4 [4] are also DBL hash functions based on DES with rates 1/2 and 1/4, respectively. Lai and Massey proposed the tandem/abreast Davies-Meyer [12]. They consist of a $(n, 2n)$ block cipher and their rates are 1/2. It is an open question whether the four schemes are optimally collision resistant or not.

Knudsen and Preneel studied the schemes to construct secure compression functions with longer outputs from secure ones based on error-correcting codes [8–10]. It is also an open question whether optimally collision resistant compression functions are constructed by their schemes.

Recently, Black, Cochran and Shrimpton [2] showed that it is impossible to construct a highly efficient block-cipher-based hash function provably secure in the black-box model. A block-cipher-based hash function is highly efficient if it makes exactly one block-cipher call for each message block and all block-cipher calls use a single key.

2.2 Cryptographic Hash Functions

A cryptographic hash function H is a function which maps an input of arbitrary length to an output of fixed length. H should satisfy the following properties.

Preimage resistance For a given output y, it is intractable to find an input x such that $y = H(x)$.

Second-preimage resistance For a given input x, it is intractable to find an input x' such that $H(x) = H(x')$ and $x \neq x'$.

Collision resistance It is intractable to find a pair of inputs x and x' such that $H(x) = H(x')$ and $x \neq x'$.

A hash function $H : \{0,1\}^* \to \{0,1\}^\ell$ usually consists of a compression function $f : \{0,1\}^\ell \times \{0,1\}^{\ell'} \to \{0,1\}^\ell$ and an initial value $h_0 \in \{0,1\}^\ell$. An input m is divided into the ℓ'-bit blocks $m_1, m_2, \ldots, m_l$. Then,

$$h_i = f(h_{i-1}, m_i)$$

is computed successively for $1 \leq i \leq l$ and $h_l = H(m)$. H is called an iterated hash function.

Unambiguous padding is applied to m if its length is not a multiple of ℓ'. It is outside the scope of this article and is not described here.

2.3 Block Ciphers and a Black-Box Model

A block cipher with the block length n and the key length κ, $e : \{0,1\}^{\kappa} \times \{0,1\}^n \to \{0,1\}^n$, is called an (n, κ) block cipher. An (n, κ) block cipher is an invertible keyed permutation: $e(k, \cdot)$ is a permutation for every $k \in \{0,1\}^{\kappa}$, and it is easy to compute both $e(k, \cdot)$ and $e(k, \cdot)^{-1}$. The set of all (n, κ) block ciphers is denoted by $B(n, \kappa)$.

Most of the attacks on hash functions based on block ciphers do not utilize the internal structure of the block ciphers. Thus, the security of hash functions based on block ciphers is often analyzed in a black-box model, that is, under the assumption that $e(k, \cdot)$ is a random invertible permutation for each k.

In the black-box model, an encryption e and a decryption e^{-1} can be simulated by the following two oracles. An encryption oracle e returns a randomly selected ciphertext for a query which is a pair of a key and a plaintext. A decryption oracle e^{-1} returns a randomly selected plaintext for a query which is a pair of a key and a ciphertext. The oracles e and e^{-1} share a table of triplets of keys, plaintexts and ciphertexts, (k_i, x_i, y_i)'s, which are produced by the queries and the corresponding answers. Referring to the table, they randomly select an answer to a new query under the restriction that $e(k, \cdot)$ is a permutation for every k. They also add the triplet produced by the query and the answer to the table.

Without loss of generality, it is assumed that any adversary with the two oracles e and e^{-1} asks only once on a triplet of a key, a plaintext and a ciphertext obtained by a query and a corresponding answer: Once the adversary obtains (k, x, y) by a query and the answer, he just keeps it and asks neither (k, x) nor (k, y) afterward.

2.4 DBL Hash Functions

DBL hash functions with two block-cipher calls in their compression functions are discussed in the article. Let f be a compression function such that

$$(h_i, g_i) = f(h_{i-1}, g_{i-1}, m_i),$$

where $h_i, g_i, m_i \in \{0,1\}^n$ and n is the block length. f consists of f_U and f_L such that

$$\begin{cases} h_i = f_U(h_{i-1}, g_{i-1}, m_i) \\ g_i = f_L(h_{i-1}, g_{i-1}, m_i). \end{cases}$$

h_i is not fed into f_L and this kind of compression function is called the parallel type. This type of compression function is considered in this article.

Each of f_U and f_L is composed of a block cipher as follows:

$$\begin{cases} h_i = e_U(k_U, x_U) \oplus z_U \\ g_i = e_L(k_L, x_L) \oplus z_L, \end{cases}$$

where k_U, x_U, z_U and k_L, x_L, z_L are uniquely defined by h_{i-1}, g_{i-1}, m_i.

The rate r of an iterated hash function of block-cipher-based f is defined by

$$r = \frac{|m_i|}{(\#\text{ of block-cipher calls in } f) \times n} .$$

It is a measure of the efficiency of block-cipher-based hash functions.

The major difference should be noticed between the DBL hash functions previously proposed and ones proposed in the article. e_U and e_L are identical for the former, but are different for the latter.

2.5 Definitions of Security

As has been discussed in this section, the security of DBL hash functions is analyzed in the black-box model. Insecurity is quantified by success probability of an optimal resource-bounded adversary. In the black-box model, the resource is the number of the queries to encryption and decryption oracles.

For a set S, $z \leftarrow_{\mathrm{R}} S$ represents random sampling from S under the uniform distribution. For a probabilistic algorithm $\mathcal{M}$, $z \leftarrow_{\mathrm{R}} \mathcal{M}(x)$ means that z is an output of $\mathcal{M}$ with an input x and the output distribution is based on the random choices of $\mathcal{M}$ and the input distribution.

Collision Resistance. The following experiment FindColHF$(\mathcal{A}, H)$ is introduced to define the collision resistance of a DBL hash function H with two block ciphers e_U and e_L. The adversary $\mathcal{A}$ is a collision-finding algorithm of H with oracles e_U, e_U^{-1} and e_L, e_L^{-1}. Let $e_P^{\pm 1}$ represent a pair of oracles e_P and e_P^{-1} for $P \in \{U, L\}$.

```
FindColHF($\mathcal{A}, H$)
  $e_U \leftarrow_{\mathrm{R}} B(n,\kappa)$; $e_L \leftarrow_{\mathrm{R}} B(n,\kappa)$;
  $(m, m') \leftarrow_{\mathrm{R}} \mathcal{A}^{e_U^{\pm 1}, e_L^{\pm 1}}$;
  if $m \neq m' \wedge H(m) = H(m')$ return 1; else return 0;
```

FindColHF$(\mathcal{A}, H)$ returns 1 iff $\mathcal{A}$ finds a collision. Let $\mathbf{Adv}_H^{\mathrm{coll}}(\mathcal{A})$ be the probability that FindColHF$(\mathcal{A}, H)$ returns 1. The probability is taken over the uniform distribution on $B(n, \kappa)$ and coin tosses of $\mathcal{A}$.

Definition 1 (Collision resistance of a hash function). *For $q \geq 1$, let*

$$\mathbf{Adv}_H^{\mathrm{coll}}(q) = \max_{\mathcal{A}} \left\{ \mathbf{Adv}_H^{\mathrm{coll}}(\mathcal{A}) \right\} ,$$

where $\mathcal{A}$ makes at most q queries to each of $e_U^{\pm 1}$ and $e_L^{\pm 1}$. ♢

The following experiment FindColCF($\mathcal{A}, f, h_0$) is introduced to define the collision resistance of a compression function f with two block ciphers e_U and e_L. h_0 is an initial value of an iterated hash function of f.

```
FindColCF($\mathcal{A}, f, h_0$)
    $e_U \leftarrow_{\mathrm{R}} B(n,\kappa)$; $e_L \leftarrow_{\mathrm{R}} B(n,\kappa)$;
    $((h,m),(h',m')) \leftarrow_{\mathrm{R}} \mathcal{A}^{e_U^{\pm 1}, e_L^{\pm 1}}$;
    if $((h,m) \neq (h',m') \wedge f(h,m) = f(h',m')) \vee f(h,m) = h_0$ return 1;
    else return 0;
```

FindColCF($\mathcal{A}, f, h_0$) returns 1 iff $\mathcal{A}$ finds a collision of f or a preimage of h_0. Let $\mathbf{Adv}_f^{\mathrm{comp}}(\mathcal{A})$ be the probability that FindColCF($\mathcal{A}, f, h_0$) returns 1.

Definition 2 (Collision resistance of a compression function). *For $q \geq 1$, let*

$$\mathbf{Adv}_f^{\mathrm{comp}}(q) = \max_{\mathcal{A}} \left\{ \mathbf{Adv}_f^{\mathrm{comp}}(\mathcal{A}) \right\},$$

where $\mathcal{A}$ asks at most q queries to each of $e_U^{\pm 1}$ and $e_L^{\pm 1}$. $\diamondsuit$

Preimage Resistance. The following experiment FindPreImg($\mathcal{A}, G$) is introduced to define the preimage resistance of G with two block ciphers e_U and e_L. G is a hash function or a compression function.

```
FindPreImg($\mathcal{A}, G$)
    $e_U \leftarrow_{\mathrm{R}} B(n,\kappa)$; $e_L \leftarrow_{\mathrm{R}} B(n,\kappa)$; $y \leftarrow_{\mathrm{R}} \{0,1\}^{\ell}$;
    $x \leftarrow_{\mathrm{R}} \mathcal{A}(y)^{e_U^{\pm 1}, e_L^{\pm 1}}$;
    if $G(x) = y$ return 1; else return 0;
```

FindPreImg($\mathcal{A}, G$) returns 1 iff $\mathcal{A}$ finds a preimage of G for an output y chosen randomly. Let $\mathbf{Adv}_G^{\mathrm{img}}(\mathcal{A})$ be the probability that FindPreImg($\mathcal{A}, G$) returns 1.

Definition 3 (Preimage resistance). *For $q \geq 1$, let*

$$\mathbf{Adv}_G^{\mathrm{img}}(q) = \max_{\mathcal{A}} \left\{ \mathbf{Adv}_G^{\mathrm{img}}(\mathcal{A}) \right\},$$

where $\mathcal{A}$ makes at most q queries to each of $e_U^{\pm 1}$ and $e_L^{\pm 1}$. $\diamondsuit$

Generally speaking, second-preimage resistance is stronger security requirement than preimage resistance. A preimage may have some information of another preimage which produces the same output. However, in the black-box model, for the hash functions or the compression functions considered in the subsequent sections, a preimage has no information useful to find another preimage. Thus, only preimage resistance is discussed in this article.

3 Provably Secure DBL Hash Functions with Two Block Ciphers

In this section, the security of DBL hash functions with compression functions shown in Fig. 1 is analyzed. Let f be a compression function such that $(h_i, g_i) =$

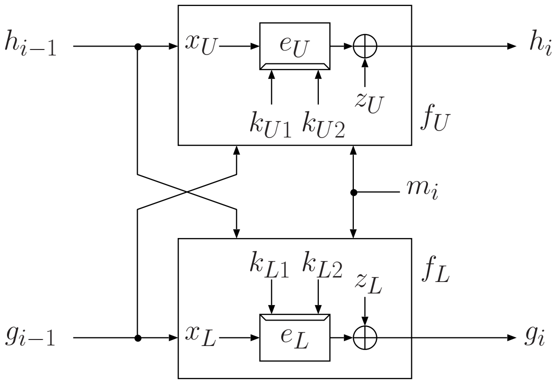

Fig. 1. A Diagram of Compression Functions with Two Block Ciphers and with Rate 1/2

$f(h_{i-1}, g_{i-1}, m_i)$ and

$$\begin{cases} h_i = f_U(h_{i-1}, g_{i-1}, m_i) \\ g_i = f_L(h_{i-1}, g_{i-1}, m_i). \end{cases}$$

f_U and f_L consist of $(n, 2n)$ block ciphers e_U and e_L, respectively, and are represented as follows:

$$\begin{cases} h_i = e_U(k_{U1} \| k_{U2}, x_U) \oplus z_U \\ g_i = e_L(k_{L1} \| k_{L2}, x_L) \oplus z_L, \end{cases}$$

where '$\|$' is the concatenation and $k_{U1}, k_{U2}, x_U, z_U,\ k_{L1}, k_{L2}, x_L, z_L \in \{0,1\}^n$ are represented by linear combinations of $h_{i-1}, g_{i-1}, m_i \in \{0,1\}^n$. Namely,

$$\begin{pmatrix} k_{U1} \\ k_{U2} \\ x_U \\ z_U \end{pmatrix} = U \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_i \end{pmatrix}, \quad \begin{pmatrix} k_{L1} \\ k_{L2} \\ x_L \\ z_L \end{pmatrix} = L \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_i \end{pmatrix}$$

and both U and L are 4×3 $\{0,1\}$-matrices.

3.1 Collision Resistance

In this subsection, a sufficient and simple condition of U and L is presented for an iterated hash function of f to be collision resistant.

The collision resistance of compression functions is focused on in the remaining part. It has been shown in [5, 15] that an iterated hash function is collision resistant if its compression function is. The following lemma states the fact in the black-box model.

Lemma 1. *[3] Let H be an iterated hash function of f. Then, for $q \geq 1$, $\mathbf{Adv}_H^{\mathrm{coll}}(q) \leq \mathbf{Adv}_f^{\mathrm{comp}}(q)$.* $\Diamond$

First, a notation and a simple lemma are given for later use. For $1 \leq r \leq 4$, let $U(r)$ and $L(r)$ denote 3×3 $\{0,1\}$-matrices obtained by deleting the r-th row of U and L, respectively.

Lemma 2. *If both $U(3)$ and $U(4)$ are non-singular, then*

$$z_U \in \{x_U, x_U \oplus k_{U1}, x_U \oplus k_{U2}, x_U \oplus k_{U1} \oplus k_{U2}\}.$$

$\Diamond$

Proof. Since $U(4)$ is non-singular, z_U can be represented by a linear combination of x_U, k_{U1}, k_{U2}. On the other hand, since $U(3)$ is non-singular, z_U cannot be represented by any linear combinations of k_{U1}, k_{U2}. $\Box$

A sufficient condition is given for a compression function to be collision resistant in the following lemma.

Lemma 3. *Suppose that all of $U(3), U(4), L(3), L(4)$ are non-singular. Then, for every $1 \leq q \leq 2^{n-1} + 1$,*

$$\mathbf{Adv}_f^{\mathrm{comp}}(q) \leq q(q+1)/2^{2n-1}.$$

$\Diamond$

Proof. Let $\mathcal{A}$ be a collision-finding algorithm of f with oracles $e_U^{\pm 1}$ and $e_L^{\pm 1}$. $\mathcal{A}$ asks q queries to each of $e_U^{\pm 1}$ and $e_L^{\pm 1}$.

Since both $U(4)$ and $L(4)$ are non-singular and

$$\begin{pmatrix} k_{U1} \\ k_{U2} \\ x_U \end{pmatrix} = U(4) \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_i \end{pmatrix}, \quad \begin{pmatrix} k_{L1} \\ k_{L2} \\ x_L \end{pmatrix} = L(4) \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_i \end{pmatrix},$$

the correspondence between (k_{U1}, k_{U2}, x_U) and (k_{L1}, k_{L2}, x_L) is 1-to-1. Thus, once a pair of an input and an output of e_U, $(k_{U1}, k_{U2}, x_U, y_U)$, is fixed by $\mathcal{A}$'s query to e_U or e_U^{-1} and its reply, an input to e_L, (k_{L1}, k_{L2}, x_L), is uniquely determined. Similarly, $\mathcal{A}$'s query to e_L or e_L^{-1} and its reply also uniquely determine an input to e_U.

On the other hand, it is necessary to ask a query to each of $e_U^{\pm 1}$ and $e_L^{\pm 1}$ in order to obtain a pair of an input and an output of f. The fact mentioned above implies that the correspondence between a pair of a query and a reply of $e_U^{\pm 1}$ and that of $e_L^{\pm 1}$ is 1-to-1. Hence, without loss of generality, it is assumed that $\mathcal{A}$ asks a query to an oracle and the corresponding query to the other oracle at a time.

Since $h_i = e_U(k_{U1} \| k_{U2}, x_U) \oplus z_U = y_U \oplus z_U$ and

$$z_U \in \{x_U, x_U \oplus k_{U1}, x_U \oplus k_{U2}, x_U \oplus k_{U1} \oplus k_{U2}\}$$

from Lemma 2, h_i depends both on x_U and on y_U and one of x_U and y_U is determined randomly by a reply of the oracle. Thus, h_i is randomly determined by the oracle. g_i is also randomly determined by the other oracle.

It is assumed that $z_U = x_U$ and $z_L = x_L$ in the rest of the proof. The proof is similar for the other cases.

For every $1 \leq j \leq q$, let C_j be the event such that

$$\begin{aligned}&(x_{Uj} \oplus y_{Uj} = h_0 \wedge x_{Lj} \oplus y_{Lj} = g_0) \vee \\ &\exists j' < j\,(x_{Uj} \oplus y_{Uj} = x_{Uj'} \oplus y_{Uj'} \wedge x_{Lj} \oplus y_{Lj} = x_{Lj'} \oplus y_{Lj'}),\end{aligned}$$

where x_{Uj}, y_{Uj} and x_{Lj}, y_{Lj} correspond to the pairs of the j-th query and its reply of $e_U^{\pm 1}$ and $e_L^{\pm 1}$, respectively. Then,

$$\Pr[\mathsf{C}_j] \leq \frac{j}{(2^n - (j-1))^2}.$$

Thus, if $q \leq 2^{n-1} + 1$, then

$$\begin{aligned}\mathbf{Adv}_f^{\mathrm{comp}}(\mathcal{A}) &\leq \Pr[\mathsf{C}_1 \vee \cdots \vee \mathsf{C}_q] \leq \sum_{j=1}^{q} \Pr[\mathsf{C}_j] \\ &\leq \sum_{j=1}^{q} \frac{j}{(2^n - (j-1))^2} \leq \sum_{j=1}^{q} \frac{j}{(2^n - 2^{n-1})^2} \\ &= \frac{q(q+1)}{2^{2n-1}}.\end{aligned}$$

□

The following theorem is led immediately from Lemmas 1 and 3.

Theorem 1. *Let H be an iterated hash function of f. Suppose that all of $U(3), U(4), L(3), L(4)$ are non-singular for f. Then,*

$$\mathbf{Adv}_H^{\mathrm{coll}}(q) \leq q(q+1)/2^{2n-1}$$

for every $1 \leq q \leq 2^{n-1} + 1$. ◊

From this theorem, any constant probability of success in finding a collision implies that $q = \Omega(2^n)$.

There are many compression functions satisfying the condition given in Theorem 1. The number of U's such that $U(3)$ and $U(4)$ are non-singular is 672. Thus, the number of compression functions satisfying the condition in Theorem 1 is $672^2 = 451584$.

3.2 Preimage Resistance

Preimage resistance of iterated hash functions presented in the previous subsection is discussed here.

The following lemma shows the relationship between preimage resistance of an iterated hash function and that of its compression function. This lemma is also implicit in [19].

Lemma 4. *[3] Let H be an iterated hash function of f. Then, for $q \geq 1$, $\mathbf{Adv}_H^{\mathrm{img}}(q) \leq \mathbf{Adv}_f^{\mathrm{img}}(q)$.* $\Diamond$

The preimage resistance of compression functions given in the previous subsection is presented in the following lemma.

Lemma 5. *Suppose that all of $U(3), U(4), L(3), L(4)$ are non-singular. Then, for every $g \geq 1$,*

$$\mathbf{Adv}_f^{\mathrm{img}}(q) \leq q/(2^n - q)^2.$$

$\Diamond$

Proof. Let $\mathcal{A}$ be a preimage-finding algorithm of f with oracles $e_U^{\pm 1}$ and $e_L^{\pm 1}$. $\mathcal{A}$ asks q queries to each of $e_U^{\pm 1}$ and $e_L^{\pm 1}$. Let w be the input of $\mathcal{A}$ and $w = (w_U, w_L)$, where $w_U, w_L \in \{0,1\}^n$.

It is necessary to ask a query to each of $e_U^{\pm 1}$ and $e_L^{\pm 1}$ in order to obtain a pair of an input and an output of f. As in the proof of Lemma 3, the correspondence between a pair of a query and a reply of $e_U^{\pm 1}$ and that of $e_L^{\pm 1}$ is 1-to-1. Hence, without loss of generality, it is assumed that $\mathcal{A}$ asks a query to an oracle and the corresponding query to the other oracle at a time.

Since $h_i = y_U \oplus z_U$ and

$$z_U \in \{x_U, x_U \oplus k_{U1}, x_U \oplus k_{U2}, x_U \oplus k_{U1} \oplus k_{U2}\}$$

from Lemma 2, h_i depends both on x_U and on y_U and one of x_U and y_U is determined randomly by a reply of the oracle. Thus, h_i is randomly determined by the oracle. g_i is also randomly determined by the other oracle.

It is assumed that $z_U = x_U$ and $z_L = x_L$ in the rest of the proof. The proof is similar for the other cases.

For every $1 \leq j \leq q$, let I_j be the event such that

$$x_{Uj} \oplus y_{Uj} = w_U \wedge x_{Lj} \oplus y_{Lj} = w_L$$

where x_{Uj}, y_{Uj} and x_{Lj}, y_{Lj} correspond to the pairs of the j-th query and its reply of $e_U^{\pm 1}$ and $e_L^{\pm 1}$, respectively. Then,

$$\Pr[\mathsf{I}_j] \leq \frac{1}{(2^n - (j-1))^2}.$$

Thus,

$$\begin{aligned}\mathbf{Adv}_f^{\mathrm{img}}(\mathcal{A}) &\leq \Pr[\mathsf{I}_1 \vee \cdots \vee \mathsf{I}_q] \leq \sum_{j=1}^{q} \Pr[\mathsf{I}_j] \leq \sum_{j=1}^{q} \frac{1}{(2^n - (j-1))^2} \\ &\leq \frac{q}{(2^n - q)^2}.\end{aligned}$$

□

The following theorem is led immediately from Lemmas 4 and 5.

Theorem 2. *Let H be an iterated hash function of f. Suppose that all of $U(3), U(4)$, $L(3), L(4)$ are non-singular for f. Then, for every $q \geq 1$,*

$$\mathbf{Adv}_H^{\mathrm{img}}(q) \leq \frac{q}{(2^n - q)^2}.$$

◊

Theorem 2 implies nothing about the preimage resistance for $q \geq 2^n - 2^{n/2} + 1$. It states, however, that the success probability is (asymptotically) negligible as long as $q = c\,2^n$ for any positive constant $c < 1$:

$$\mathbf{Adv}_H^{\mathrm{img}}(c\,2^n) \leq \frac{c}{(1-c)^2} \frac{1}{2^n}.$$

For example, if $c = 1/2$, then $\mathbf{Adv}_H^{\mathrm{img}}(2^{n-1}) \leq 1/2^{n-1}$.

4 Provably Secure DBL Hash Functions with One Block Cipher

Let e be an (n, κ) block cipher and $n + 2 \leq \kappa$. In this section, the security of DBL hash functions with compression functions shown in Fig. 2 is analyzed. The left-side function is focused on. Let us call it f.

The compression function f is represented as follows:

$$\begin{cases} h_i = e(g_{i-1} \| m_i \| v_U, h_{i-1}) \oplus h_{i-1} \\ g_i = e(h_{i-1} \| m_i \| v_L, g_{i-1}) \oplus g_{i-1}, \end{cases}$$

where $m_i \in \{0,1\}^\ell$ for some $1 \leq \ell < \kappa - n$, and v_U and v_L are constants in $\{0,1\}^{\kappa-n-\ell}$ such that $v_U \neq v_L$.

Since $v_U \neq v_L$, in the black-box model, e with v_U and e with v_L can be regarded as two independent random block ciphers. Furthermore, there exists 1-to-1 correspondence between a pair of an input and an output of e with v_U and that of e with v_L.

From these observations, it is clear that the following lemma can be proved in the similar way as Lemma 3.

Lemma 6. *For the compression function f, if $1 \leq q \leq 2^{n-1} + 1$, then*

$$\mathbf{Adv}_f^{\mathrm{comp}}(q) \leq q(q+1)/2^{2n-1}.$$

◊

The following theorem states the collision resistance of an iterated hash function of f. This is immediately lead from Lemmas 1 and 6.

Theorem 3. *Let H be an iterated hash function of f. Then,*

$$\mathbf{Adv}_H^{\mathrm{coll}}(q) \leq q(q+1)/2^{2n-1}$$

for every $1 \leq q \leq 2^{n-1}+1$. ◊

For preimage resistance, similarly, the following theorem is obtained.

Theorem 4. *Let H be an iterated hash function of f. Then, for $q \geq 1$,*

$$\mathbf{Adv}_H^{\mathrm{img}}(q) \leq \frac{q}{(2^n - q)^2}.$$

◊

In the black-box model, it is sufficient that $v_U, v_L \in \{0,1\}$ and $v_U \neq v_L$. However, in practice, v_U, v_L should be longer in order to avoid weak keys and to increase independence. Suppose that ℓ_{con} be the length of v_U or v_L and $\kappa = 2\,n$. Then, the rate of H is $(1-\ell_{\mathrm{con}}/n)/2$. For example, the rate is 7/16 if $\ell_{\mathrm{con}} = n/8$.

The idea that two block ciphers are obtained from one block cipher by fixing a part of the key with different constants is found in the design of MDC-2 [4]. However, the security proof as shown above does not seem to be applied to MDC-2.

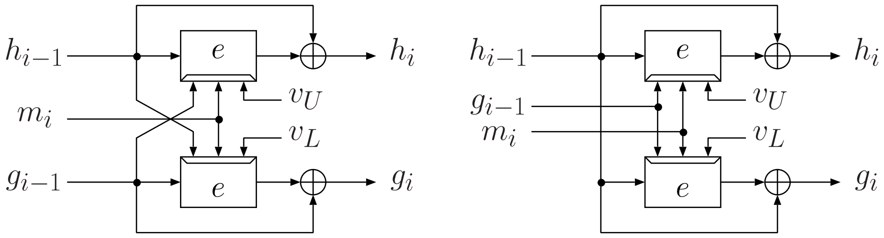

Fig. 2. Compression Functions with One Block Cipher

5 Conclusion

In this article, DBL hash functions provably secure in the black-box model have been presented. They are based on $(n, 2n)$ block ciphers and can be represented in a simple form. Future work is to explore more efficient DBL hash functions optimally collision resistant.

References

1. E. Biham and R. Chen. Near-collisions of SHA-0. Cryptology ePrint Archive, Report 2004/146, 2004. http://eprint.iacr.org/.

2. J. Black, M. Cochran, and T. Shrimpton. On the impossibility of highly efficient blockcipher-based hash functions. Cryptology ePrint Archive, Report 2004/062, 2004. http://eprint.iacr.org/.
3. J. Black, P. Rogaway, and T. Shrimpton. Black-box analysis of the block-cipher-based hash-function constructions from PGV. In *CRYPTO 2002 Proceedings*, pages 320–335, 2002. Lecture Notes in Computer Science 2442.
4. B. O. Brachtl, D. Coppersmith, M. M. Hyden, S. M. Matyas Jr., C. H. W. Meyer, J. Oseas, S. Pilpel, and M. Schilling. Data authentication using modification detection codes based on a public one-way encryption function, mar 1990. U. S. Patent # 4,908,861.
5. I. Damgård. A design principle for hash functions. In *CRYPTO'89 Proceedings*, pages 416–427, 1990. Lecture Notes in Computer Science 435.
6. M. Hattori, S. Hirose, and S. Yoshida. Analysis of double block length hash functions. In *9th IMA International Conference on Cryptography and Coding*, pages 290–302, 2003. Lecture Notes in Computer Science 2898.
7. W. Hohl, X. Lai, T. Meier, and C. Waldvogel. Security of iterated hash functions based on block ciphers. In *CRYPTO'93 Proceedings*, pages 379–390, 1994. Lecture Notes in Computer Science 773.
8. L. Knudsen and B. Preneel. Hash functions based on block ciphers and quaternary codes. In *ASIACRYPT'96 Proceedings*, pages 77–90, 1996. Lecture Notes in Computer Science 1163.
9. L. Knudsen and B. Preneel. Fast and secure hashing based on codes. In *CRYPTO'97 Proceedings*, pages 485–498, 1997. Lecture Notes in Computer Science 1294.
10. L. Knudsen and B. Preneel. Construction of secure and fast hash functions using nonbinary error-correcting codes. *IEEE Transactions on Information Theory*, 48(9):2524–2539, 2002.
11. L. R. Knudsen, X. Lai, and B. Preneel. Attacks on fast double block length hash functions. *Journal of Cryptology*, 11(1):59–72, 1998.
12. X. Lai and J. L. Massey. Hash function based on block ciphers. In *EUROCRYPT'92 Proceedings*, pages 55–70, 1993. Lecture Notes in Computer Science 658.
13. M. Liskov, R. L. Rivest, and D. Wagner. Tweakable block ciphers. In *CRYPTO 2002 Proceedings*, pages 31–46, 2002. Lecture Notes in Computer Science 2442.
14. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
15. R. C. Merkle. One way hash functions and DES. In *CRYPTO'89 Proceedings*, pages 428–446, 1990. Lecture Notes in Computer Science 435.
16. B. Preneel, R. Govaerts, and J. Vandewalle. Hash functions based on block ciphers: A synthetic approach. In *CRYPTO'93 Proceedings*, pages 368–378, 1994. Lecture Notes in Computer Science 773.
17. T. Satoh, M. Haga, and K. Kurosawa. Towards secure and fast hash functions. *IEICE Transactions on Fundamentals*, E82-A(1):55–62, 1999.
18. X. Wang, D. Feng, X. Lai, and H. Yu. Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD. Cryptology ePrint Archive, Report 2004/199, 2004. http://eprint.iacr.org/.
19. R. S. Winternitz. A secure one-way hash function built from DES. In *IEEE Symposium on Security and Privacy*, pages 88–90, 1984.