Some Plausible Constructions of Double-Block-Length Hash Functions

Shoichi Hirose

Faculty of Engineering, The University of Fukui, Fukui 910-8507 Japan
hirose@fuee.fukui-u.ac.jp

Abstract. In this article, it is discussed how to construct a compression function with $2n$-bit output using a component function with n-bit output. The component function is either a smaller compression function or a block cipher. Some constructions are presented which compose collision-resistant hash functions: Any collision-finding attack on them is at most as efficient as the birthday attack in the random oracle model or in the ideal cipher model. A new security notion is also introduced, which we call indistinguishability in the iteration, with a construction satisfying the notion.

1 Introduction

A cryptographic hash function is a function which maps an input of arbitrary length to an output of fixed length. It satisfies preimage resistance, second-preimage resistance and collision resistance. It is one of the most important primitives in cryptography [19]. 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. This type of hash function is called an iterated hash function. There has been an interest in constructing a compression function from component functions with smaller output length. Many schemes have been presented following the approach [4, 10, 11, 13–15, 17, 20]. It is typical for constructions using block ciphers. For example, suppose that AES is used for construction. The block length of AES is 128 bits, and a hash function with 128-bit output is no longer secure against the birthday attack. Thus, it is desired to construct a compression function with larger output length.

In this article, we study how to construct a compression function with $2n$-bit output using a component function with n-bit output. A hash function with such a compression function is called a double-block-length (DBL) hash function (as opposed to a single-block-length (SBL) hash function, where the compression function has n-bit output). The component function may be either a block cipher or a smaller compression function.

We first discuss constructions using a smaller compression function. We focus on the constructions formalized by Nandi [22]. In his formalization, the compression function is of the form $F(x) = (f(x), f(p(x)))$, where f is a component compression function and p is a permutation such that both p and p^{-1} are easy to compute. We show that any collision-finding attack on a hash function with the compression function F is at most as efficient as the birthday attack if f is a random oracle and p satisfies some properties. Our properties for p are easy to be satisfied; for example, they are satisfied by the permutation $p(x) = x \oplus c$, where $\oplus$ is bit-wise addition and c is a non-zero constant.

Similar results are in fact already obtained by Nandi [21], whose analysis actually applies to a broader range of hash functions than our analysis. However, our results are sharper. We give a significantly better upper bound on the probability of finding a collision as a function of the number of queries made by the adversary.

A new security notion for a compression function is also introduced, which we call indistinguishability in the iteration. It is really weaker than the notion proposed in [5]. However, it may be valuable in practice. Loosely speaking, a compression function $F(x) = (f(x), f(p(x)))$ where f is a random oracle is called indistinguishable in the iteration if F cannot be distinguished from a random oracle in the iterated hash function. We give sufficient conditions on p for F to be indistinguishable in the iteration.

Second, we discuss constructions using a block cipher. A compression function composed of a block cipher is presented and its collision resistance is analyzed. We show that any collision-finding attack on a hash function composed of the compression function is at most as efficient as the birthday attack if the block cipher used is ideal. A block cipher is ideal if it is assumed to be a keyed invertible random permutation. The compression function presented in this article is quite simple but has not been explicitly discussed previously.

In [10], it is shown that a collision-resistant hash function can be easily composed of a compression function using two distinct block ciphers. It is well-known that two distinct block ciphers can be obtained from a block cipher by fixing, for example, one key bit by 0 and 1. However, it is preferable in practice that fixing key bits is avoided. Moreover, fixing one bit may not be sufficient and more bits may be required to be fixed. Our new construction does not involve any fixing of key bits by constants.

The technique in [3] is used in the security proofs in this article. However, the analysis is more complicated than the one in [3] since the relation of two component-compression-function/block-cipher calls in a compression function need to be taken into account.

The rest of this article is organized as follows. Section 2 includes notations, definitions and a brief review of the related works. Section 3 discusses compression functions composed of a smaller compression function, including the results on collision resistance and our new notion of indistinguishability in the iteration. Section 4 exhibits a block-cipher-based compression function whose associated hash function has optimal collision resistance; the proof of collision resistance is given in the appendix. Section 5 gives a concluding remark which mentions a recent collision attack on the scheme in Sect. 4.

2 Preliminaries

2.1 Iterated Hash Function

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 a fixed 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 $1 \le i \le l$ and $h_l = H(m)$. H is called an iterated hash function.

Before being divided into the blocks, unambiguous padding is applied to the input. The length of the padded input is a multiple of ℓ'. In this article, Merkle-Damgård strengthening [6, 20] is assumed for padding. Thus, the last block contains the length of the input.

2.2 Random Oracle Model and Ideal Cipher Model

Random Oracle Model. Let $\boldsymbol{F}_{n',n} = \{f \mid f : \{0,1\}^{n'} \to \{0,1\}^n\}$. In the random oracle model, the function f is assumed to be randomly selected from $\boldsymbol{F}_{n',n}$. The computation of f is simulated by the following oracle.

The oracle f first receives an input x_i as a query. Then, it returns a randomly selected output y_i if the query has never been asked before. It keeps a table of pairs of queries and replies, and it returns the same reply to the same query.

Ideal Cipher Model. A block cipher with the block length n and the key length κ is called an (n, κ) block cipher. Let $e : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$ be an (n, κ) block cipher. Then, $e(k, \cdot)$ is a permutation for every $k \in \{0,1\}^\kappa$, and it is easy to compute both $e(k, \cdot)$ and $e(k, \cdot)^{-1}$.

Let $\boldsymbol{B}_{n,\kappa}$ be the set of all (n, κ) block ciphers. In the ideal cipher model, e is assumed to be randomly selected from $\boldsymbol{B}_{n,\kappa}$. The encryption e and the decryption e^{-1} are simulated by the following two oracles.

The encryption oracle e first receives a pair of a key and a plaintext as a query. Then, it returns a randomly selected ciphertext. On the other hand, the decryption oracle e^{-1} first receives a pair of a key and a ciphertext as a query. Then, it returns a randomly selected plaintext. 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 replies. Referring to the table, they select a reply 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 reply to the table.

2.3 DBL Hash Function

An iterated hash function whose compression function is composed of a block cipher is called a single-block-length (SBL) hash function if its output length is equal to the block length of the block cipher. It is called a double-block-length (DBL) hash function if its output length is twice larger than the block length.

Let F be a compression function composed of a block cipher. For an iterated hash function composed of F, the rate r defined below is often used as a measure of efficiency:

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

In this article, we also call an iterated hash function a DBL hash function if its compression function F is composed of a smaller compression function f and its output length is twice larger than the output length of f.

2.4 Related Work

Knudsen and Preneel studied the schemes to construct secure compression functions with longer outputs from secure ones based on error-correcting codes [13–15]. It is an open question whether optimally collision-resistant compression functions are constructed by their schemes. A hash/compression function is optimally collision-resistant if any attack to find its collision is at most as efficient as the birthday attack.

Our work is largely motivated by the recent works by Lucks [18] and Nandi [22]. Nandi generalized the results by Lucks and by Hirose [10]. He discussed how to construct DBL hash functions and presented optimally collision-resistant ones. However, their security analysis is not so sharp as ours, which is mentioned in Sect. 1.

Coron, Dodis, Malinaud and Puniya [5] discussed how to construct a random oracle with arbitrary input length given a random oracle with fixed input length.

As is reviewed in the following, there are many papers on hash functions composed of block ciphers.

Preneel, Govaerts and Vandewalle [25] discussed the security of SBL hash functions against several generic 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, $k, x, z \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i, c\}$ and c 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 [25] in the ideal cipher model. The most important result in their paper is that 20 hash functions including the 12 mentioned above is optimally collision-resistant.

Knudsen, Lai and Preneel [16] discussed the insecurity of DBL hash functions with the rate 1 composed of (n, n) block ciphers. Hohl, Lai, Meier and Waldvogel [11] discussed the security of compression functions of DBL hash functions with the rate 1/2. On the other hand, the security of DBL hash functions with the rate 1 composed of $(n, 2n)$ block ciphers was discussed by Satoh, Haga and Kurosawa [26] and by Hattori, Hirose and Yoshida [8]. These works presented no construction for DBL hash functions with optimal collision resistance.

Many schemes with the rates less than 1 were also presented. Merkle [20] presented three DBL hash functions composed of DES with the rates at most 0.276. They are optimally collision-resistant in the ideal cipher model. MDC-2 and MDC-4 [4] are also DBL hash functions composed of DES with the rates 1/2 and 1/4, respectively. Lai and Massey proposed the tandem/abreast Davies-Meyer [17]. They consist of an $(n, 2n)$ block cipher and their rates are 1/2. It is an open question whether the four schemes are optimally collision-resistant or not.

Hirose [10] presented a large class of DBL hash functions with the rate 1/2, which are composed of $(n, 2n)$ block ciphers. They were shown to be optimally collision-resistant in the ideal cipher model. However, his construction requires two independent block ciphers, which makes the results less attractive.

Nandi, Lee, Sakurai and Lee [23] also proposed an interesting construction with the rate 2/3. However, they are not optimally collision-resistant. Knudsen and Muller [12] presented some attacks against it and illustrated its weaknesses, none of which contradicts the security proof in [23].

Black, Cochran and Shrimpton [2] showed that it is impossible to construct a highly efficient block-cipher-based hash function provably secure in the ideal cipher 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.

Gauravaram, Millan and May proposed a new approach based on iterated halving to design a hash function with a block cipher [7].

3 DBL Hash Function in the Random Oracle Model

3.1 Compression Function

In this section, we consider the DBL hash functions with compression functions given in the following definition.

Definition 1. *Let $F : \{0,1\}^{2n} \times \{0,1\}^b \to \{0,1\}^{2n}$ be a compression function such that $(g_i, h_i) = F(g_{i-1}, h_{i-1}, m_i)$, where $g_i, h_i \in \{0,1\}^n$ and $m_i \in \{0,1\}^b$. F consists of $f : \{0,1\}^{2n} \times \{0,1\}^b \to \{0,1\}^n$ and a permutation $p : \{0,1\}^{2n+b} \to \{0,1\}^{2n+b}$ as follows:*

$$\begin{cases} g_i = F_U(g_{i-1}, h_{i-1}, m_i) = f(g_{i-1}, h_{i-1}, m_i) \\ h_i = F_L(g_{i-1}, h_{i-1}, m_i) = f(p(g_{i-1}, h_{i-1}, m_i)) \end{cases} .$$

p satisfies the following properties:

- *It is easy to compute both p and p^{-1},*
- *$p(p(\cdot))$ is an identity permutation,*
- *p has no fixed points, that is, $p(g_{i-1}, h_{i-1}, m_i) \neq (g_{i-1}, h_{i-1}, m_i)$ for any (g_{i-1}, h_{i-1}, m_i).*

3.2 Collision Resistance

We will analyze the collision resistance of DBL hash functions composed of F under the assumption that f is a random oracle.

Two queries to the oracle f are required to compute the output of F for an input. For this compression function, a query to f for F_U or F_L uniquely determines the query to f for the other since p is a permutation. Moreover, for every $w \in \{0,1\}^{2n+b}$, $f(w)$ and $f(p(w))$ are only used to compute $F(w)$ and $F(p(w))$, and $w \neq p(w)$ from the properties for p in Definition 1. Thus, it is reasonable to assume that a pair of queries w and $p(w)$ to f are asked at a time.

Definition 2. *A pair of distinct inputs w, w' to F are called a matching pair if $w' = p(w)$. Otherwise, they are called a non-matching pair.*

Notice that $w' = p(w)$ iff $w = p(w')$ since $p(p(\cdot))$ is an identity permutation.

Definition. Insecurity is quantified by success probability of an optimal resource-bounded adversary. The resource is the number of the queries to f in the random oracle model.

For a set S, let $z \leftarrow_{\mathrm{R}} S$ represent random sampling from S under the uniform distribution. For a probabilistic algorithm $\mathcal{M}$, let $z \leftarrow_{\mathrm{R}} \mathcal{M}$ mean that z is an output of $\mathcal{M}$ and its distribution is based on the random choices of $\mathcal{M}$.

Let H be a DBL hash function composed of a compression function F in Definition 1. The following experiment $\mathtt{FindColHF}(\mathcal{A}, H)$ is introduced to quantify the collision resistance of H. The adversary $\mathcal{A}$ with the oracle f is a collision-finding algorithm of H.

$\mathtt{FindColHF}(\mathcal{A}, H)$
$\quad f \leftarrow_{\mathrm{R}} \boldsymbol{F}_{2n+b,n}$;
$\quad (m, m') \leftarrow_{\mathrm{R}} \mathcal{A}^f$;
$\quad$ **if** $m \neq m' \wedge H(m) = H(m')$ **return** 1; **else return** 0;

$\mathtt{FindColHF}(\mathcal{A}, H)$ returns 1 iff $\mathcal{A}$ finds a collision. Let $\mathbf{Adv}_H^{\mathrm{coll}}(\mathcal{A})$ be the probability that $\mathtt{FindColHF}(\mathcal{A}, H)$ returns 1. The probability is taken over the uniform distribution on $\boldsymbol{F}_{2n+b,n}$ and random choices of $\mathcal{A}$.

Definition 3. *For $q \geq 1$, let*

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

where $\mathcal{A}$ makes at most q pairs of queries to f in total.

Without loss of generality, it is assumed that $\mathcal{A}$ does not ask the same query twice. $\mathcal{A}$ can keep pairs of queries and their corresponding answers by himself.

Analysis. The following theorem shows the collision resistance of a hash function composed of F in Definition 1.

Theorem 1. *Let H be a hash function composed of a compression function F specified in Definition 1. Then, for every $1 \leq q \leq 2^n$,*

$$\mathbf{Adv}_H^{\mathrm{coll}}(q) \leq \left(\frac{q}{2^n}\right)^2 + \frac{q}{2^n} \ .$$

Proof. Let $\mathcal{A}$ be a collision-finding algorithm of H with the oracle f. $\mathcal{A}$ asks q pairs of queries to f in total. Suppose that $\mathcal{A}$ finds a colliding pair m, m' of H. Then, it is easy to find a colliding pair of inputs for F without any additional queries to the oracle. Moreover, a pair of inputs to F are either matching or non-matching, so are the colliding pair of inputs for F.

For $2 \leq j \leq q$, let C_j be the event that a colliding pair of non-matching inputs are found for F with the j-th pair of queries. Namely, it is the event that

$$(f(w_j), f(p(w_j)) \in \{(f(w_{j'}), f(p(w_{j'}))), (f(p(w_{j'})), f(w_{j'}))\}$$

for some $j' < j$, where w_j and $p(w_j)$ are the j-th pair of queries. Since both $f(w_j)$ and $f(p(w_j))$ are randomly selected by the oracle,

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

Let C be the event that a colliding pair of non-matching inputs are found for F with q pairs of queries. Then,

$$\Pr[\mathsf{C}] = \Pr[\mathsf{C}_2 \vee \mathsf{C}_3 \vee \cdots \vee \mathsf{C}_q] \leq \sum_{j=2}^{q} \Pr[\mathsf{C}_j] \leq \left(\frac{q}{2^n}\right)^2 \ .$$

For $1 \leq j \leq q$, let $\mathsf{C}_j^{\mathrm{m}}$ be the event that a colliding pair of matching inputs are found for F with the j-th pair of queries, that is, $f(w_j) = f(p(w_j))$. Thus,

$$\Pr[\mathsf{C}_j^{\mathrm{m}}] = \frac{1}{2^n} \ .$$

Let C^{m} be the event that a colliding pair of matching inputs are found for F with q pairs of queries. Then,

$$\Pr[\mathsf{C}^{\mathrm{m}}] = \Pr[\mathsf{C}_1^{\mathrm{m}} \vee \mathsf{C}_2^{\mathrm{m}} \vee \cdots \vee \mathsf{C}_q^{\mathrm{m}}] \leq \sum_{j=1}^{q} \Pr[\mathsf{C}_j^{\mathrm{m}}] = \frac{q}{2^n} \ .$$

Thus, if $q \leq 2^n$, then

$$\mathbf{Adv}_H^{\mathrm{coll}}(\mathcal{A}) \leq \Pr[\mathsf{C} \vee \mathsf{C}^{\mathrm{m}}] \leq \Pr[\mathsf{C}] + \Pr[\mathsf{C}^{\mathrm{m}}] \leq \left(\frac{q}{2^n}\right)^2 + \frac{q}{2^n} \ ,$$

which holds for any $\mathcal{A}$. $\square$

From Theorem 1, any constant probability of success in finding a collision of H requires $\Omega(2^n)$ queries.

A better bound can be obtained with more restricted permutations.

Theorem 2. *Let H be a hash function composed of a compression function F specified in Definition 1. Suppose that the permutation p is represented by $p(g,h,m) = (p_{\mathrm{cv}}(g,h), p_{\mathrm{m}}(m))$, where $p_{\mathrm{cv}} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ and $p_{\mathrm{m}} : \{0,1\}^b \to \{0,1\}^b$. Suppose that p_{cv} has no fixed points and that $p_{\mathrm{cv}}(g,h) \neq (h,g)$ for any (g,h). Then, for every $1 \leq q \leq 2^n$,*

$$\mathbf{Adv}_H^{\mathrm{coll}}(q) \leq 3\left(\frac{q}{2^n}\right)^2 .$$

Proof. Let $\mathcal{A}$ be a collision-finding algorithm of H with the oracle f. $\mathcal{A}$ asks q pairs of queries to f in total. Suppose that $\mathcal{A}$ finds a colliding pair m, m' of H. Then, it is easy to find a colliding pair of inputs for F without any additional queries. Moreover, a pair of inputs to F are either matching or non-matching, so are the colliding pair of inputs for F.

Let C be the event that a colliding pair of non-matching inputs are found for F with q pairs of queries. Then, as in the proof of Theorem 1,

$$\Pr[\mathsf{C}] \leq \left(\frac{q}{2^n}\right)^2 .$$

Suppose that a colliding pair of matching inputs are obtained for F from the collision of H found by $\mathcal{A}$. Let (g,h,m) and (g',h',m') be the colliding pair. Then, $(g,h) = p_{\mathrm{cv}}(g',h')$ (and $(g',h') = p_{\mathrm{cv}}(g,h)$). (g,h) and (g',h') are both outputs of F, or at most one of them is the initial value (g_0,h_0) of H since $(g,h) \neq (g',h')$. Thus, a pair of inputs w and w' are also found for F from the collision of H such that $F(w) = p_{\mathrm{cv}}(F(w'))$ or $F(w) = p_{\mathrm{cv}}(g_0,h_0)$.

Suppose that $(g,h) = F(w)$ and $(g',h') = F(w')$. Then, a pair of w and w' are non-matching since $(g,h) = p_{\mathrm{cv}}(g',h') \neq (h',g')$.

For $1 \leq j \leq q$, let $\hat{\mathsf{C}}_j^{\mathrm{m}}$ be the event that, for the j-th pair of queries w_j and $p(w_j)$,

$$F(w_j) \in \{p_{\mathrm{cv}}(g_0,h_0)\} \cup \bigcup_{1 \leq j' < j} \{p_{\mathrm{cv}}(F(w_{j'})), p_{\mathrm{cv}}(F(p(w_{j'})))\}$$

or

$$F(p(w_j)) \in \{p_{\mathrm{cv}}(g_0,h_0)\} \cup \bigcup_{1 \leq j' < j} \{p_{\mathrm{cv}}(F(w_{j'})), p_{\mathrm{cv}}(F(p(w_{j'})))\} .$$

Thus,

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

Let $\hat{\mathsf{C}}^{\mathrm{m}} = \hat{\mathsf{C}}_1^{\mathrm{m}} \vee \cdots \vee \hat{\mathsf{C}}_q^{\mathrm{m}}$. Then,

$$\Pr[\hat{\mathsf{C}}^{\mathrm{m}}] \leq \sum_{j=1}^{q} \Pr[\hat{\mathsf{C}}_j^{\mathrm{m}}] \leq \sum_{j=1}^{q} \frac{2(2j-1)}{2^{2n}} = 2\left(\frac{q}{2^n}\right)^2 \enspace .$$

Thus,

$$\mathbf{Adv}_H^{\mathrm{coll}}(\mathcal{A}) \leq \Pr[\mathsf{C} \vee \hat{\mathsf{C}}^{\mathrm{m}}] \leq \Pr[\mathsf{C}] + \Pr[\hat{\mathsf{C}}^{\mathrm{m}}] \leq 3\left(\frac{q}{2^n}\right)^2 \enspace ,$$

for $1 \leq q \leq 2^n$, which holds for any $\mathcal{A}$. □

For $q < 2^{n-1}$, Theorem 2 gives a smaller upper bound than Theorem 1. The difference between their upper bounds is significant. For example, let $n = 128$ and $q = 2^{80}$. Then, the upper bound of Theorem 1 is about 2^{-48}, while the upper bound of Theorem 2 is less than 2^{-94}.

Example 1. Here is an example of the permutation p satisfying the conditions given in Theorem 2:

$$p(g, h, m) = (g \oplus c_1, h \oplus c_2, m) \enspace ,$$

where c_1 and c_2 are distinct constants in $\{0,1\}^n$.

3.3 Indistinguishability in the Iteration

We introduce a new security notion which is called indistinguishability in the iteration.

Definition. Let F be a compression function specified in Definition 1. The following experiment `DistinguishCF`$(\mathcal{A}, F)$ is introduced to quantify the indistinguishability in the iteration of F. The adversary $\mathcal{A}$ is a distinguishing algorithm of F. $\mathcal{A}$ has an oracle $\mathcal{O}$. In this experiment, a randomly chosen bit $d \in \{0,1\}$ is given to $\mathcal{O}$ first. If $d = 1$, then $\mathcal{O}$ chooses $f \in \boldsymbol{F}_{2n+b,n}$ randomly in advance. Then, $\mathcal{O}$ returns $F(w) = (f(w), f(p(w)))$ to each query w from $\mathcal{A}$. If $d = 0$, then $\mathcal{O}$ chooses $R \in \boldsymbol{F}_{2n+b,2n}$ randomly in advance. Then, $\mathcal{O}$ returns $R(w)$ to each query w from $\mathcal{A}$. $\mathcal{A}$ makes a chosen message attack and tries to tell whether $\mathcal{O}$ uses F or R. However, $\mathcal{A}$ is only allowed to select his j-th query $w_j = (w_j^{(1)}, w_j^{(2)}, w_j^{(3)})$ from

$$\left\{ (w^{(1)}, w^{(2)}, w^{(3)}) \;\middle|\; (w^{(1)}, w^{(2)}) \in \bigcup_{\ell=0}^{j-1} (v_\ell^{(1)}, v_\ell^{(2)}) \wedge w^{(3)} \in \{0,1\}^b \right\} \enspace ,$$

where $(v_\ell^{(1)}, v_\ell^{(2)})$ is $\mathcal{O}$'s answer to the ℓ-th query for $\ell \geq 1$ and $(v_0^{(1)}, v_0^{(2)})$ is some fixed initial value of a hash function H. F is assumed to be used only in the iteration of H.

```
DistinguishCF(𝒜, F)
  d ←R {0,1};
  d' ←R 𝒜^{𝒪(d)};
  if d = d' return 1; else return 0;
```

Let $\mathbf{Succ}_F^{\text{ind-it}}(\mathcal{A})$ be the probability that $\texttt{DistinguishCF}(\mathcal{A}, F)$ returns 1. Without loss of generality, it can be assumed that $\mathbf{Succ}_F^{\text{ind-it}}(\mathcal{A}) \geq 1/2$ because the probability that $d = d'$ is $1/2$ even if $\mathcal{A}$ chooses d' randomly. It can also be assumed that $\mathcal{A}$ does not ask the same query twice. Let

$$\mathbf{Adv}_F^{\text{ind-it}}(\mathcal{A}) \stackrel{\text{def}}{=} \mathbf{Succ}_F^{\text{ind-it}}(\mathcal{A}) - 1/2 \ .$$

Definition 4. *For $q \geq 1$, let*

$$\mathbf{Adv}_F^{\text{ind-it}}(q) = \max_{\mathcal{A}} \left\{ \mathbf{Adv}_F^{\text{ind-it}}(\mathcal{A}) \right\} \ ,$$

where $\mathcal{A}$ makes at most q queries to $\mathcal{O}$.

As long as $\mathbf{Adv}_F^{\text{ind-it}}(q)$ is small enough, the compression function F behaves like a random function in the iterated hash function. The following theorem presents an upper bound on $\mathbf{Adv}_F^{\text{ind-it}}(q)$ with additional restriction on the permutation p.

Theorem 3. *Let F be a compression function specified in Definition 1. Suppose that the permutation p is represented by $p(g, h, m) = (p_{\text{cv}}(g, h), p_{\text{m}}(m))$, where $p_{\text{cv}} : \{0,1\}^{2n} \to \{0,1\}^{2n}$ and $p_{\text{m}} : \{0,1\}^b \to \{0,1\}^b$. Suppose that p_{cv} has no fixed points. Then, for every $1 \leq q \leq 2^n$,*

$$\mathbf{Adv}_F^{\text{ind-it}}(q) \leq \frac{1}{2} \left(\frac{q}{2^n} \right)^2 \ .$$

Proof. Let $\mathcal{A}$ be the optimal distinguishing algorithm for F which makes q queries. Let w_j be $\mathcal{A}$'s j-th query to $\mathcal{O}$ and $T = \{w_j \mid 1 \leq j \leq q\} \cap \{p(w_j) \mid 1 \leq j \leq q\}$. Suppose that $d = 1$. Then, $\mathcal{O}$ returns $F(w_j) = (f(w_j), f(p(w_j)))$ for w_j. If $T = \phi$, then F is completely indistinguishable from R. It is because each one of $f(w_j)$ and $f(p(w_j))$ for $1 \leq j \leq q$ appears only once and it is chosen randomly by $\mathcal{O}$.

Let Empty be the event that $T = \phi$. Then,

$$\begin{aligned}
\mathbf{Succ}_F^{\text{ind-it}}(\mathcal{A}) &= \Pr[d = d'] = \Pr[d = d' \wedge \mathsf{Empty}] + \Pr[d = d' \wedge \neg\mathsf{Empty}] \\
&= \Pr[d = d' \mid \mathsf{Empty}] \Pr[\mathsf{Empty}] + \Pr[d = d' \mid \neg\mathsf{Empty}] \Pr[\neg\mathsf{Empty}] \\
&\leq \frac{1}{2} + \Pr[\neg\mathsf{Empty}] \ .
\end{aligned}$$

Let v_j be the initial value if $j = 0$ and the answer of $\mathcal{O}$ to the j-th query by $\mathcal{A}$ if $j \geq 1$. For $1 \leq j \leq q$, let C'_j be the event that $v_j \in \{p_{\text{cv}}(v_\ell) \mid 0 \leq \ell \leq j - 1\}$. Then,

$$\Pr[\mathsf{C}'_j] \leq \frac{j}{2^{2n}} \ .$$

For $1 \le q \le 2^n$,

$$\Pr[\neg\mathsf{Empty}] \le \Pr[\mathsf{C}'_1 \vee \cdots \vee \mathsf{C}'_{q-1}] \le \sum_{j=1}^{q-1} \Pr[\mathsf{C}'_j] \le \frac{1}{2}\left(\frac{q}{2^n}\right)^2$$

which implies that $\mathbf{Adv}_F^{\text{ind-it}}(q) \le (q/2^n)^2/2$. □

4 DBL Hash Function in the Ideal Cipher Model

4.1 Compression Function

In this section, the collision resistance of a DBL hash function composed of a compression function using a block cipher is analyzed. The compression function specified in the following definition is considered.

Definition 5. *Let $F : \{0,1\}^{2n} \times \{0,1\}^b \to \{0,1\}^{2n}$ be a compression function such that $(g_i, h_i) = F(g_{i-1}, h_{i-1}, m_i)$, where $g_i, h_i \in \{0,1\}^n$ and $m_i \in \{0,1\}^b$. F consists of a $(n, n+b)$ block cipher e as follows:*

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

where $\|$ represents concatenation and $c \in \{0,1\}^n - \{0^n\}$ is a constant.

The compression function in Definition 5 is also shown in Fig. 1. It is one of the compression functions specified in Definition 1 and its f and p are specified as follows:

$$f(g_{i-1}, h_{i-1}, m_i) = e(h_{i-1} \| m_i, g_{i-1}) \oplus g_{i-1} \quad ,$$
$$p(g_{i-1}, h_{i-1}, m_i) = (g_{i-1} \oplus c, h_{i-1}, m_i) \quad .$$

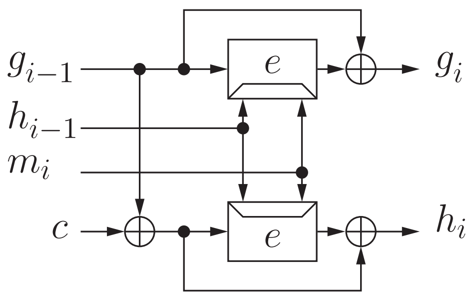

Fig. 1. A compression function considered in Sect. 4.2

F requires two invocations of e to produce an output. However, these two invocations need only one key scheduling of e. If F is implemented using the AES with 192-bit key-length, then $n = 128$, $b = 64$ and the rate is $1/4$. If implemented using the AES with 256-bit key-length, then $n = b = 128$ and the rate is $1/2$.

4.2 Collision Resistance

Let F be a compression function specified in Definition 5. Two queries to the oracles e and e^{-1} in total are required to compute the output of F for an input. It is easy to see from Fig. 1 that a query to e or e^{-1} and the corresponding reply for F_U (F_L) uniquely determine the query to e for F_L (F_U). Moreover, these two queries are only used to compute the outputs of F for a matching pair of inputs. Thus, it is assumed that a pair of queries to e, e^{-1} required to compute an output of F are asked at a time.

Definition. The following experiment $\texttt{FindColHF}(\mathcal{A}, H)$ is similar to the one in Sect. 3 except that the adversary $\mathcal{A}$ is a collision-finding algorithm with the oracles e, e^{-1}.

$\texttt{FindColHF}(\mathcal{A}, H)$
  $e \leftarrow_{\mathrm{R}} \boldsymbol{B}_{n,n+b}$;
  $(m, m') \leftarrow_{\mathrm{R}} \mathcal{A}^{e,e^{-1}}$;
  $\texttt{if}\ m \neq m' \wedge H(m) = H(m')\ \texttt{return}\ 1\texttt{;}\ \texttt{else return}\ 0\texttt{;}$

Let $\mathbf{Adv}_H^{\mathrm{coll}}(\mathcal{A})$ be the probability that $\texttt{FindColHF}(\mathcal{A}, H)$ returns 1. The probability is taken over the uniform distribution on $\boldsymbol{B}_{n,n+b}$ and random choices of $\mathcal{A}$.

Definition 6. *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 pairs of queries to e, e^{-1} in total.

Without loss of generality, it is assumed that $\mathcal{A}$ asks at most only once on a triplet of a key, a plaintext and a ciphertext obtained by a query and the corresponding reply.

Analysis. The following theorem shows the collision resistance of a hash function composed of F in Definition 5.

Theorem 4. *Let H be a hash function composed of the compression function F specified in Definition 5. Then, for every $1 \leq q \leq 2^{n-2}$,*

$$\mathbf{Adv}_H^{\mathrm{coll}}(q) \leq 3 \left(\frac{q}{2^{n-1}} \right)^2 .$$

The proof of Theorem 4 is given in the appendix.

5 Concluding Remark

In this article, some plausible constructions have been proposed for DBL hash functions.

Recently, Pramstaller and Rijmen presented a collision attack on the scheme in Sect. 4 with DESX as an underlying block cipher [24]. Their result does not contradict Theorem 4. It is a warning that we should be careful when we choose an underlying block cipher. It also shows a limitation of the random oracle/ideal cipher model. Related topics are discussed in [1, 9].

Acknowledgements. The author would like to thank the anonymous reviewers for their helpful comments.

References

1. J. Black. The ideal-cipher model, revisited: An uninstantiable blockcipher-based hash function. Cryptology ePrint Archive, Report 2005/210, 2005. http://eprint.iacr.org/. Also appear in this proceedings.
2. J. Black, M. Cochran, and T. Shrimpton. On the impossibility of highly efficient blockcipher-based hash functions. In *EUROCRYPT 2005 Proceedings, Lecture Notes in Computer Science 3494*, pages 526–541, 2005.
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, Lecture Notes in Computer Science 2442*, pages 320–335, 2002.
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. J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-damgård revisited: How to construct a hash function. In *CRYPTO 2005 Proceedings, Lecture Notes in Computer Science 3621*, pages 430–448, 2005.
6. I. Damgård. Collision free hash functions and public key signature schemes. In *EUROCRYPT '87 Proceedings, Lecture Notes in Computer Science 304*, pages 203–216, 1988.
7. P. Gauravaram, W. Millan, and L. May. CRUSH: A new cryptographic hash function using iterated halving technique. In *Proceedings of Cryptographic Algorithms and their Uses 2004*, pages 28–39, 2004.
8. M. Hattori, S. Hirose, and S. Yoshida. Analysis of double block length hash functions. In *Proceedings of the 9th IMA International Conference on Cryptography and Coding, Lecture Notes in Computer Science 2898*, pages 290–302, 2003.
9. S. Hirose. Secure block ciphers are not sufficient for one-way hash functions in the Preneel-Govaerts-Vandewalle model. In *Proceedings of the 9th Selected Areas in Cryptography (SAC 2002), Lecture Notes in Computer Science 2595*, pages 339–352, 2002.
10. S. Hirose. Provably secure double-block-length hash functions in a black-box model. In *Proceedings of the 7th Internatinal Conference on Information Security and Cryptology (ICISC 2004), Lecture Notes in Computer Science 3506*, pages 330–342, 2005.

11. W. Hohl, X. Lai, T. Meier, and C. Waldvogel. Security of iterated hash functions based on block ciphers. In *CRYPTO '93 Proceedings, Lecture Notes in Computer Science 773*, pages 379–390, 1994.
12. L. Knudsen and F. Muller. Some attacks against a double length hash proposal. In *ASIACRYPT 2005 Proceedings, Lecture Notes in Computer Science 3788*, pages 462–473, 2005.
13. L. Knudsen and B. Preneel. Hash functions based on block ciphers and quaternary codes. In *ASIACRYPT '96 Proceedings, Lecture Notes in Computer Science 1163*, pages 77–90, 1996.
14. L. Knudsen and B. Preneel. Fast and secure hashing based on codes. In *CRYPTO '97 Proceedings, Lecture Notes in Computer Science 1294*, pages 485–498, 1997.
15. 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.
16. L. R. Knudsen, X. Lai, and B. Preneel. Attacks on fast double block length hash functions. *Journal of Cryptology*, 11(1):59–72, 1998.
17. X. Lai and J. L. Massey. Hash function based on block ciphers. In *EUROCRYPT '92 Proceedings, Lecture Notes in Computer Science 658*, pages 55–70, 1993.
18. S. Lucks. A failure-friendly design principle for hash functions. In *ASIACRYPT 2005 Proceedings, Lecture Notes in Computer Science 3788*, pages 474–494, 2005.
19. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.
20. R. C. Merkle. One way hash functions and DES. In *CRYPTO '89 Proceedings, Lecture Notes in Computer Science 435*, pages 428–446, 1990.
21. M. Nandi. *Design of Iteration on Hash Functions and Its Cryptanalysis*. PhD thesis, Indian Statistical Institute, 2005.
22. M. Nandi. Towards optimal double-length hash functions. In *Proceedings of the 6th International Conference on Cryptology in India (INDOCRYPT 2005), Lecture Notes in Computer Science 3797*, pages 77–89, 2005.
23. M. Nandi, W. Lee, K. Sakurai, and S. Lee. Security analysis of a 2/3-rate double length compression function in the black-box model. In *Proceedings of the 12th Fast Software Encryption (FSE 2005), Lecture Notes in Computer Science 35571*, pages 243–254, 2005.
24. N. Pramstaller and V. Rijmen. A collision attack on a double-block-length hash proposal. Cryptology ePrint Archive, Report 2006/116, 2006. http://eprint.iacr.org/.
25. B. Preneel, R. Govaerts, and J. Vandewalle. Hash functions based on block ciphers: A synthetic approach. In *CRYPTO '93 Proceedings, Lecture Notes in Computer Science 773*, pages 368–378, 1994.
26. T. Satoh, M. Haga, and K. Kurosawa. Towards secure and fast hash functions. *IEICE Transactions on Fundamentals*, E82-A(1):55–62, 1999.

A Proof of Theorem 4

Let $\mathcal{A}$ be a collision-finding algorithm of H with oracles e, e^{-1}. $\mathcal{A}$ asks q pairs of queries to e, e^{-1} in total.

Since $g_i = e(h_{i-1} \| m_i, g_{i-1}) \oplus g_{i-1}$, g_i depends both on the plaintext and the ciphertext of e and one of them is fixed by a query and the other is determined randomly by the answer from the oracle. Thus, g_i is randomly determined by the answer. h_i is also randomly determined by the other answer.

Let $(k_{1,j} \| k_{2,j}, x_j, y_j)$ and $(k_{1,j} \| k_{2,j}, x_j \oplus c, z_j)$ be the triplets of e obtained by the j-th pair of queries and the corresponding answers.

For every $2 \le j \le q$, let C_j be the event that a colliding pair of non-matching inputs are found for F with the j-th pair of queries. Namely, it is the event that, for some $j' < j$,

$$F(x_j, k_{1,j}, k_{2,j}) = F(x_{j'}, k_{1,j'}, k_{2,j'}) \text{ or } F(x_{j'} \oplus c, k_{1,j'}, k_{2,j'})$$

or

$$F(x_j \oplus c, k_{1,j}, k_{2,j}) = F(x_{j'}, k_{1,j'}, k_{2,j'}) \text{ or } F(x_{j'} \oplus c, k_{1,j'}, k_{2,j'}) \ ,$$

which is equivalent to

$$(y_j \oplus x_j, z_j \oplus x_j \oplus c) = (y_{j'} \oplus x_{j'}, z_{j'} \oplus x_{j'} \oplus c) \text{ or } (z_{j'} \oplus x_{j'} \oplus c, y_{j'} \oplus x_{j'}) \ .$$

Thus,

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

Let C be the event that a colliding pair of non-matching inputs are found for F with q pairs of queries. Then,

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

Suppose that a colliding pair of matching inputs are obtained for F from the collision of H found by $\mathcal{A}$. Let (g, h, m) and (g', h', m') be the colliding pair of F. Then, $(g, h) = (g' \oplus c, h')$. (g, h) and (g', h') are both outputs of F or at most one of them is the initial value (g_0, h_0) of H. Thus, a pair of inputs w and w' are also found for F from the collision of H such that

$$(F_U(w), F_L(w)) = (F_U(w') \oplus c, F_L(w')) \text{ or } (g_0 \oplus c, h_0) \ .$$

Suppose that $(F_U(w), F_L(w)) = (F_U(w') \oplus c, F_L(w'))$. Then, a pair of w and w' are non-matching since

$$(F_U(w), F_L(w)) = (F_U(w') \oplus c, F_L(w')) \ne (F_L(w'), F_U(w')) \ .$$

For $1 \le j \le q$, let $(k_{1,j} \| k_{2,j}, x_j, y_j)$ and $(k_{1,j} \| k_{2,j}, x_j \oplus c, z_j)$ be the pair of triplets of e obtained by the j-th pair of queries and the corresponding answers. Let $\hat{\mathsf{C}}^{\mathrm{m}}_j$ be the event that $F(x_j, k_{1,j}, k_{2,j}) \in V$ or $F(x_j \oplus c, k_{1,j}, k_{2,j}) \in V$, where

$$V = \{(g_0 \oplus c, h_0)\} \cup \bigcup_{1 \le j' < j} \{(F_U(x_{j'}, k_{1,j'}, k_{2,j'}) \oplus c, F_L(x_{j'}, k_{1,j'}, k_{2,j'}))\} \cup \bigcup_{1 \le j' < j} \{(F_U(x_{j'} \oplus c, k_{1,j'}, k_{2,j'}) \oplus c, F_L(x_{j'} \oplus c, k_{1,j'}, k_{2,j'}))\} \ .$$

Thus,

$$\Pr[\hat{\mathsf{C}}^{\mathrm{m}}_j] \le \frac{2(2j-1)}{(2^n-(2j-2))(2^n-(2j-1))} \le \frac{2(2j-1)}{(2^n-(2j-1))^2} \enspace .$$

Let $\hat{\mathsf{C}}^{\mathrm{m}} = \hat{\mathsf{C}}^{\mathrm{m}}_1 \vee \cdots \vee \hat{\mathsf{C}}^{\mathrm{m}}_q$. Then,

$$\Pr[\hat{\mathsf{C}}^{\mathrm{m}}] \le \sum_{j=1}^{q} \Pr[\hat{\mathsf{C}}^{\mathrm{m}}_j] \le \sum_{j=1}^{q} \frac{2(2j-1)}{(2^n-(2j-1))^2} \enspace .$$

Thus, if $q \le 2^{n-2}$, then

$$\begin{aligned}
\mathbf{Adv}^{\mathrm{coll}}_H(\mathcal{A}) &\le \Pr[\mathsf{C} \vee \hat{\mathsf{C}}^{\mathrm{m}}] \le \Pr[\mathsf{C}] + \Pr[\hat{\mathsf{C}}^{\mathrm{m}}] \\
&\le \sum_{j=2}^{q} \frac{2(j-1)}{(2^n-(2j-1))^2} + \sum_{j=1}^{q} \frac{2(2j-1)}{(2^n-(2j-1))^2} \\
&\le \sum_{j=1}^{q} \frac{6j-4}{(2^n-(2j-1))^2} \\
&\le \frac{3q^2-q}{(2^{n-1})^2} \le 3\left(\frac{q}{2^{n-1}}\right)^2
\end{aligned}$$

which holds for any $\mathcal{A}$. $\square$