On the Design of Secure and Fast Double Block Length Hash Functions[*]

Zheng Gong[†], Xuejia Lai[‡] and Kefei Chen[‡]

[†]Distributed and Embedded Security Group, Faculty of EEMCS
University of Twente, Enschede, The Netherlands
z.gong@utwente.nl

[‡]Department of Computer Science and Engineering
Shanghai Jiaotong University, Shanghai, China
{lai-xj,chen-kf}@cs.sjtu.edu.cn

Abstract

In this work the security of double block length hash functions with rate 1, which are based on a block cipher with a block length of n bits and a key length of $2n$ bits, is reconsidered. Counter-examples and new attacks are presented on this general class of fast double block length hash functions, which disclose uncovered flaws in the necessary conditions given by Satoh *et al.* and Hirose. Preimage and second preimage attacks are presented on Hirose's two examples which were left as an open problem. Furthermore, all rate-1 hash functions in this general class are failed to be optimally (second) preimage resistant, and the necessary conditions are refined for ensuring this general class of hash functions to be optimally secure against collision attacks. In particular, two typical examples, which are designed under the refined conditions, are proven to be indifferentiable from a random oracle in the ideal cipher model. The security results are extended to a new class of double block length hash functions with rate 1, where the key length of one block cipher used in the compression function is equal to the block length, while the other is doubled.

Key words. Cryptanalysis, Block-cipher-based hash function, Double block length, Indifferentiability.

1 Introduction

Cryptographic hash function $H : \{0,1\}^* \rightarrow \{0,1\}^\ell$ is defined as a feasible algorithm which uniformly maps an arbitrary length input to a fixed length output. The design of today's cryptographic hash functions still follows the Merkle-Damgard (MD) structure [7, 20], by iterating a compression function on arbitrary inputs to obtain a domain extension transform. Under the MD structure, a hash function will be collision resistant if the underlying compression function is. In practice, most of hash functions are either explicitly or implicitly composed from block ciphers. The advantage of the block-cipher-based approach is that one can conveniently choose an extensively studied block cipher (e.g., DES, IDEA, AES, etc) to construct a compression function, and also the latest cryptanalysis results on such a block cipher can be used to avoid the potential weakness in the construction. Discussions of hash functions constructed from n-bit block ciphers are mainly divided into *single block length* (SBL) and *double block length* (DBL) hash functions, where *single* and *double* are related to the output range of the underlying block cipher. The motivation of double block length is to combine two n-bit block ciphers for a sufficient output range for collision resistance. One such algorithm is MDC-2, which was developed by Brachtl *et al.* [2] based on DES; and its generic construction is included as a standard in ISO/IEC 10118-2. It is believed that the complexities of (second) preimage and collision attacks on MDC-2 are about $2^{3n/2}$ and $2^{6n/5}$ [28], respectively. Generally, a DBL hash

[*]The first author acknowledges the financial support of SenterNovem for the ALwEN project, grant PNE07007. The authors are partially supported by NSFC under the grants 60573032, 90604036 and National 863 Projects 2007AA01Z456.

function $H : \{0,1\}^* \rightarrow \{0,1\}^{2n}$ is said to be *optimally secure*, if any adversary with non-negligible advantages for (second) preimage and collision attacks on the hash function must spend no less than the complexities of 2^{2n} and 2^n, respectively.

Although DBL hash function can extend the output range for collision resistance, an obvious consequence is a decrease in performance. The *rate* of a block-cipher-based hash function is defined as *the number of n-bit message blocks processed per encryption or decryption* for the measurement of the efficiency. E.g., the rate of MDC-2 is only 1/2, which implies that MDC-2 is at least twice as slow as the underlying block cipher. To improve the efficiency, many DBL hash functions with rate 1 had been proposed, such as [3, 11, 23, 29]. Unfortunately, some critical results disclosed the fact that those proposed schemes unlikely achieve optimal security. Knudsen *et al.* [15] presented the attacks on a large class of DBL hash functions with rate 1, such that the key length is equal to the block length of n bits (denoted by FDBL-I). In particular, Knudsen *et al.*'s attacks break the proposed schemes in [3, 11, 23].

Still, many advanced block ciphers (e.g., AES, RC5, Blowfish, etc) support variants of the key length motivates renewed interests in finding good ways to construct a secure and fast DBL hash function. Satoh *et al.* [26] presented some attacks on a general class of DBL hash functions with rate 1 where the key length is twice as the block length (denoted by FDBL-II), and broke the rate-1 construction in [29]. In particular, Satoh *et al.* described a necessary condition for rate-1 hash functions in FDBL-II to be optimally secure against preimage, second preimage and collision attacks. Lately, Hirose [9] made a new comment on Satoh *et al.*'s result [26] and showed that there exists a missed case in their analysis. Based on this comment, Hirose proposed two necessary conditions [9] for rate-1 hash functions in FDBL-II to be optimally collision resistant. Furthermore, two examples are left in [9] as an open problem to make it clear whether they are optimally secure. In the existing literature, there are some other DBL hash functions were proposed recently, such as [10, 18, 21, 22]. But all those constructions are lacking in performance since all of them are less than rate 1.

Our Contributions. Consider the security of rate-1 DBL hash functions where the key length is doubled to the block length, our contributions are three-fold. First, we present (second) preimage attacks on Hirose's two examples which are left as an open problem in [9]. Moreover, three counter-examples in FDBL-II are designed to disclose that Hirose's necessary conditions [9] for optimal collision resistance are still imprecise. Secondly, based on these attacks and counter-examples, we formally analyze the security of rate-1 hash functions in FDBL-II, and find although all rate-1 hash functions in FDBL-II are failed to be optimally (second) preimage resistant, there exists a subclass of rate-1 hash functions in FDBL-II that can be optimally collision resistant. The necessary conditions for rate-1 hash functions in FDBL-II to be optimally collision resistant are refined by the analysis. In particular, the indifferentiability analysis is given on two typical examples in FDBL-II that satisfy our refined necessary conditions, which implies they are optimally collision resistant in the ideal cipher model. Finally, the security results are extended to a new class of DBL hash functions with rate 1 (denoted by FDBL-III), where the key length of one block cipher used in the compression function is equal to the block length, while the other is doubled. The extended results show that all rate-1 DBL hash functions in FDBL-III are failed to be optimally secure. Prior to this paper, there is no rigorous analysis on the examples which are proposed by Satoh *et al.* [26] and Hirose [9] to ensure whether they are really optimally secure.

Organization. The remainder of this paper is organized as follows. In Section 2, definitions and the former results on DBL hash functions with rate 1 are reviewed. In Section 3, two concrete attacks are presented on Hirose's two examples, then counter-examples are given to show the fact that Hirose's two necessary conditions [9] are not precise for optimal collision resistance. Attacks are presented on FDBL-II to obtain precise conditions towards optimal security. Section 4 concludes the paper. Additionally, the indifferentiability analysis of typical examples in FDBL-II is given in Appendix A. Appendix B describes an extended security result on FDBL-III.

2 Preliminaries

In this section, some necessary notions and definitions are reviewed for the analysis throughout the paper. Let the symbol $\oplus$ be the bitwise exclusive OR. For binary sequences a and b, $a||b$ denotes their concatenation. Let IV be the initial value. For DBL hash functions, an arbitrary input message M can be represented as a concatenation of $2n$-bit length blocks such that $M = m_1||m_2||\cdots||m_t$, where $t = \lceil |M|/2n \rceil$ and $m_i = m_{i,1}||m_{i,2}, i \in \{0, t\}$. The function $Rank(\cdot)$ returns the rank of an input matrix. In this paper, length-padding on the last block of input message is implicitly used to avoid some trivial attacks. The same terminology and abbreviation in different definitions are the same meaning, except there are special claims in the context.

2.1 Block-Cipher-Based Hash Functions

Let κ, n, ℓ be integers. A *block cipher* is a keyed function $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$. For each $k \in \{0,1\}^\kappa$, the function $E_k(\cdot) = E(k, \cdot)$ denotes a permutation on $\{0,1\}^n$. If E is a block cipher then E^{-1} is its inverse, where $E_k^{-1}(y) = x$ such that $E_k(x) = y$. Let $\texttt{Bloc}(\kappa, n)$ be the family of all block ciphers $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$. A *block-cipher-based* hash function is a hash function $H : \{0,1\}^* \to \{0,1\}^\ell$ by implementing $E \in \texttt{Bloc}(\kappa, n)$ in the iteration of H. If $\ell = n$, then H is called a single block length (SBL) hash function, e.g., the PGV hash functions [24]. If $\ell = 2n$, then H is called a double block length (DBL) hash function, e.g., MDC-2 [2], Parallel-DM [11], and LOKI-DBH [3]. The *rate* is widely accepted to measure the efficiency of a block-cipher-based hash function, which is defined as follows.

Definition 1 *Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be a hash function and $E \in \texttt{Bloc}(\kappa, n)$ is a block cipher used in the compression function of H. If the compression function performs T times encryption or decryption of E to process totally ∂ bits long message block, the* rate *of the hash function H equals $\frac{\partial}{T \cdot n}$.*

Ideal Cipher Model. Ideal cipher model is a well-known security model for the analysis of block-cipher-based hash functions, which is dating back to Shannon [27] and has been frequently used for the analysis of various block-cipher-based hash functions [1, 10, 17, 24]. Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be a hash function and $E \in \texttt{Bloc}(\kappa, n)$ be a block cipher used in the iteration of H. In the ideal cipher model, E is assumed to be randomly selected from $\texttt{Bloc}(\kappa, n)$ [10]. Adversary $\mathcal{A}$ has accesses to the encryption oracle E and the decryption oracle E^{-1}. The i-th query-response is defined as a four-tuple $(\sigma_i, k_i, x_i, y_i)$ where $\sigma_i \in \{1, -1\}, k_i \in \{0,1\}^\kappa$ and $x_i, y_i \in \{0,1\}^n$. If $\sigma_i = 1$ then $\mathcal{A}$ asks (k_i, x_i) and gets response $y_i = E_{k_i}(x_i)$, otherwise he asks (k_i, y_i) and gets response $x_i = E_{k_i}^{-1}(y_i)$. Since $E_k(\cdot)$ is a permutation on $\{0,1\}^n$, it holds that

$$\Pr[E_{k_i}(x_i) = y_i] = \Pr[E_{k_i}^{-1}(y_i) = x_i] = \frac{1}{2^n - i + 1}.$$

In the ideal cipher model, one measures the complexity of an attack, on which finding a collision, preimage or second preimage, is based on the total number of encryptions and decryptions that the adversary queried. Generally, all repetition queries will be ignored, namely, if $\mathcal{A}$ makes a query on $E_k(x)$ and this returns y, then he will not repeat the query or ask the inverse $E_k^{-1}(y)$. Such trivial queries do not help at all from the adversaries point of view. The block cipher in this model is variously named “Shannon oracle model”, “Black-box model”, or “Ideal cipher model”. Since the last name is more often called, it will be used throughout the paper.

2.2 Security Definitions

Now we recall the definitions for the security analysis of block-cipher-based hash functions.

Attacks on hash functions. For block-cipher-based hash functions, there are three standard attacks which are called the collision attack, the preimage attack and the second preimage attack. A limitation is that the standard attacks only consider the situation that initial value IV is fixed. The four extended attacks include the situation that IV can be changed by the adversary.

Definition 2 *Let $H : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{Y}$ be a family of hash functions where $\mathcal{K} \in \{0,1\}^{\kappa}, \mathcal{Y} \in \{0,1\}^{\ell}$. Let M be a message belongs to message space $\mathcal{M} \in \{0,1\}^{*}$. By considering whether IV is fixed or not, three standard attacks and four extended attacks are defined as follows.*

1. *The preimage attack (Pre) is that given IV and h, find a message M such that $h = H(IV, M)$.*
2. *The free-start preimage attack ($fPre$) is that given IV and $H(IV, M)$, find IV', M' such that $H(IV', M') = H(IV, M)$.*
3. *The second preimage attack (Sec) is that given IV and a message M, find another message $M' \neq M$ such that $H(IV, M) = H(IV, M')$.*
4. *The free-start second preimage attack ($fSec$) is that given IV and a message M, find IV' and another message $M' \neq M$ such that $H(IV, M) = H(IV', M')$.*
5. *The collision attack ($Coll$) is that given an initial value IV, find $M \neq M'$ such that $H(IV, M) = H(IV, M')$.*
6. *The semi-free-start collision attack ($sfColl$) is that find an initial value IV and two different messages M, M' such that $H(IV, M) = H(IV, M')$.*
7. *The free-start collision attack ($fColl$) is that find IV, IV' and messages M, M' such that $(IV, M) \neq (IV', M')$ but $H(IV, M) = H(IV', M')$.*

The above attacks are from [14]. Similar definitions can be found in [15, 17]. Compare with the standard attacks, the extended attacks are also meaningful since they support a complete examination on minimizing potential flaws in a family of hash functions. It is easy to see that the free-start and the semi-free-start attacks are never harder than the attacks where IV is specified in advance. To rigorously analyze the security of a hash construction at the presence of an adaptive adversary, a widely-accepted approach will be recalled in below.

Indifferentiability Model. Cryptosystems are considered to be *computational equivalent* if no polynomial-time procedure can tell them apart. For the formal analysis, Maurer *et al.* [19] first introduced the notion of *indifferentiability*, which is formalized to "distinguish" whether a given object exists any computational inequivalent from a heuristic random oracle. The indifferentiability has been focussed on the question: what conditions should be imposed on the compression function $\mathcal{F}$ to ensure that the hash function $\mathcal{C}^{\mathcal{F}}$ satisfies the certain conditions of a random oracle. This approach is based on the fact that one of the problems in assessing the security of a hash function is caused by domain extension transform. It is clear that the weakness of $\mathcal{F}$ will generally result in weakness of $\mathcal{C}^{\mathcal{F}}$, but the converse might not be true in general. The indifferentiability between a hash function and a random oracle is a more rigorous *white-box* analysis which requires the examination of the internal structure of the hash function, while the traditional instantiation just implements a *black-box* analysis.

Definition 3 *A Turing machine $\mathcal{C}$ with oracle access to an ideal primitive $\mathcal{F}$ is said to be (t_D, t_S, q, ϵ)-indifferentiable from an ideal primitive $Rand$ if there exists a simulator $\mathcal{S}$, such that for any distinguisher $\mathcal{D}$ it holds the advantage of indifferentiability that:*

$$Adv(\mathcal{D}) = |\Pr[\mathcal{D}^{\mathcal{C},\mathcal{F}} = 1] - \Pr[\mathcal{D}^{Rand,\mathcal{S}} = 1]| < \epsilon,$$

where $\mathcal{S}$ has oracle access to $Rand$ and runs in polynomial time at most t_S, and $\mathcal{D}$ runs in polynomial time at most t_D and makes at most q queries. $\mathcal{C}^{\mathcal{F}}$ is said to be indifferentiable from $Rand$ if ϵ is a negligible function of the security parameter k (in polynomial time t_D and t_S).

It was proven that if $\mathcal{C}^{\mathcal{F}}$ is indifferentiable from $Rand$, then $\mathcal{C}^{\mathcal{F}}$ can instantiate $Rand$ in any cryptosystem and the resulting cryptosystem is at least as secure in the $\mathcal{F}$ model as in the $Rand$ model [19]. In the rest of the paper, the Turing Machine $\mathcal{C}$ will denote the construction of an iterated hash function and the ideal primitive $\mathcal{F}$ will represent the compression function of $\mathcal{C}$.

For block-cipher-based hash functions, the above definition needs to be slightly modified due to the underlying compression function should be analyzed in the ideal cipher model [4, 8]. In other words, if a block-cipher-based hash function $\mathcal{C}^{\mathcal{F}}$ is indifferentiable from a random oracle $Rand$ in the ideal cipher model, then $\mathcal{C}^{\mathcal{F}}$ can replace $Rand$ in any cryptosystem, while keeping the resulting system (with $\mathcal{C}^{\mathcal{F}}$) to remain secure in the ideal cipher model if the original system (with $Rand$) is secure in the random oracle model. Let E be the block cipher used in the compression function and E^{-1} is its inverse. Simulator $\mathcal{S}$ has to simulate both E and E^{-1} because every distinguisher $\mathcal{D}$ can access encryption and decryption oracles in the ideal cipher model. Therefore, distinguisher $\mathcal{D}$ obtains the following rules: either the block cipher E, E^{-1} is chosen at random and the hash function H is constructed from it, or the hash function H is chosen at random and the block cipher E, E^{-1} is implemented by a simulator $\mathcal{S}$ with oracle accesses to H. Those two ways to build up a hash function should be indifferentiable.

Similarly, Hirose also proposed the notion of *indistinguishability* on iterated hash functions [10], which is weaker than the notion of indifferentiability. It is easy to see that if a block-cipher-based hash function $\mathcal{C}^{E,E^{-1}}$ is indifferentiable from a random oracle in polynomial time bounds t_S, t_D with a negligible probability ϵ, then it is also *indistinguishable* in the same bound. For simplicity, one needs only to prove the indifferentiability instead of the both.

Since hash function plays a pivotal role in the real-life cryptographic applications (e.g., data or entity authentication, public-key encryption and digital signature), it is prudent to make a block-cipher-based hash function to be optimally secure against all seven attacks for the security of the applications, and also be indifferentiable from a random oracle in the ideal cipher model.

2.3 Results on Fast DBL Hash Functions

Here we briefly review the former results on the rate-1 DBL hash functions. By assuming the key length κ of block cipher $E \in \texttt{Bloc}(\kappa, n)$ used in the compression function is identical to the block length n, Knudsen *et al.* [15] presented attacks on this class of DBL hash functions with rate 1 (FDBL-I). The general form of this class is described as follows.

$$\begin{cases} h_i = E_A(B) \oplus C, \\ g_i = E_X(Y) \oplus Z. \end{cases} \tag{1}$$

For all rate-1 DBL hash functions with the form (1), (A, B, C) are linear combinations of the n-bit vectors $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$, and (X, Y, Z) are linear combinations of the n-bit vectors $(h_i, h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$.

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \underbrace{\begin{pmatrix} L_l & L_r \end{pmatrix}}_{L} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_{i,1} \\ m_{i,2} \end{pmatrix}, \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} R_l & R_r \end{pmatrix}}_{R} \cdot \begin{pmatrix} h_i \\ h_{i-1} \\ g_{i-1} \\ m_{i,1} \\ m_{i,2} \end{pmatrix}. \tag{2}$$

If h_i and g_i can be computed independently, then the construction is called *parallel*, otherwise is called *serial*. Knudsen *et al.* [15] proved that all rate-1 hash functions in FDBL-I are failed to be optimally secure against collision, preimage and second preimage attacks. The result is concluded by the following theorem [15].

Theorem 1 *For any rate-1 iterated hash function with the form* (1) *(FDBL-I), there exist preimage and second preimage attacks with complexities of about* 4×2^n*. Furthermore, there exists a collision attack with complexity of about* $3 \times 2^{3n/4}$*. For all but two classes of the hash functions, there exists a collision attack with complexity of about* $4 \times 2^{n/2}$*.*

In AES algorithm, the key length can be 128,192,256 bits while the block length is 128 bits. This property motivates renewed interests in finding constructions to turn such a block cipher into a secure and fast DBL hash function, where the key length is doubled to the block length. By considering the block cipher $E \in \mathtt{Bloc}(\kappa, n)$ where $\kappa = 2n$, Satoh *et al.* [26] proposed a new family of rate-1 DBL hash functions (FDBL-II), which can be defined by the general form in below.

$$\begin{cases} h_i = E_{A||B}(C) \oplus D, \\ g_i = E_{W||X}(Y) \oplus Z. \end{cases} \tag{3}$$

For all rate-1 hash functions defined by (3), both (A, B, C, D) and (W, X, Y, Z) are linear combinations of the n-bit vectors $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$. Those linear combinations can be represented as

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \underbrace{\begin{pmatrix} L_l & L_r \end{pmatrix}}_{L} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_{i,1} \\ m_{i,2} \end{pmatrix}, \begin{pmatrix} W \\ X \\ Y \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} R_l & R_r \end{pmatrix}}_{R} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_{i,1} \\ m_{i,2} \end{pmatrix}, \tag{4}$$

where L_l and L_r denote 4×2 binary submatrices of L. Let L_l^i and L_r^i denote the 3×2 submatrices of L_l and L_r such that the i-th row of L_l and L_r are deleted, respectively. Matrix L is said to be "exceptional" [26] if $Rank(L) = 4$ and $Rank(L_r^3) = Rank(L_r^4) = 2$.

Furthermore, Satoh *et al.* [26] presented attacks on FDBL-II when the compression function does not satisfy the *exceptional* property.

Theorem 2 *Consider a rate-1 iterated hash function with the form* (3) *(FDBL-II), if* L *or* R *is not exceptional, there exist preimage, second preimage and collision attacks with complexities of about* 4×2^n*,* 3×2^n *and* $3 \times 2^{n/2}$*, respectively.*

In particular, Satoh *et al.* [26] presented attacks on a subclass of rate-1 DBL hash functions in FDBL-II. We note that the proposed scheme by Yi and Lam [29] is a paradigm with respect to this subclass.

Theorem 3 *For a subclass of rate-1 double block length hash functions in FDBL-II with the compression function:*

$$\begin{cases} h_i = E_{A||B}(C) \oplus D, \\ g_i = E_{A||B}(C) \oplus F, \end{cases} \tag{5}$$

where (A, B, C, D, F) *are linear combinations of* $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$ *and* $E \in \mathtt{Bloc}(2n, n)$*, there exist (second) preimage and collision attacks with complexities of about* 2×2^n *and* $2 \times 2^{n/2}$*, respectively.*

Inherited from the above works, Hirose [9] gave a new comment on Satoh *et al.*'s result [26]. The comment shows there exist rate-1 DBL hash functions in FDBL-II whose compression functions are not *exceptional* but still no meaningful collision attacks can be found. For convincing of this comment, an example without the *exceptional* property was first proposed in [9] as follows.

HDBL-1: Let HDBL-1:$\{0,1\}^* \to \{0,1\}^{2n}$ be a double block length hash function and $E \in \mathtt{Bloc}(2n, n)$ is the block cipher used in the compression function. The compression function has the following definition:

$$\begin{cases} h_i = E_{m_{i,1}||m_{i,2}}(h_{i-1} \oplus g_{i-1}) \oplus h_{i-1} \oplus g_{i-1}, \\ g_i = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus h_{i-1}, \end{cases} \tag{6}$$

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}}_{L} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_{i,1} \\ m_{i,2} \end{pmatrix}, \begin{pmatrix} W \\ X \\ Y \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}}_{R} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_{i,1} \\ m_{i,2} \end{pmatrix}. \tag{7}$$

Moreover, the other example with the *exceptional* property but extremely simple was also proposed by Hirose in [9].

HDBL-2: Let HDBL-2:$\{0,1\}^* \rightarrow \{0,1\}^{2n}$ be a double block length hash function and $E \in \mathtt{Bloc}(2n, n)$ is the block cipher used in the compression function. The compression function has the following definition:

$$\begin{cases} h_i = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus g_{i-1}, \\ g_i = E_{m_{i,1}||m_{i,2}}(g_{i-1}) \oplus h_{i-1}, \end{cases} \tag{8}$$

$$\begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}}_{L} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_{i,1} \\ m_{i,2} \end{pmatrix}, \begin{pmatrix} W \\ X \\ Y \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}}_{R} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_{i,1} \\ m_{i,2} \end{pmatrix}. \tag{9}$$

Both HDBL-1 and HDBL-2 are the instances of FDBL-II. Let (a, b, c, d) and (w, x, y, z) be the values of (A, B, C, D) and (W, X, Y, Z) which are used in the computations of h_i and g_i, respectively. Satoh *et al.* [26] assumed that if an adversary can randomly choose triple (a, b, c) such that $c = \alpha \cdot a \oplus \beta \cdot b$ where $\alpha, \beta \in \{0, 1\}$, then he can compute $d = E_{a||b}(c) \oplus h_i$. Nonetheless, Hirose [9] found if $c = \alpha \cdot a \oplus \beta \cdot b \oplus d$, the adversary cannot compute d by $E_{a||b}(c) \oplus h_i$. Therefore, besides the condition that both L and R are *exceptional*, a new condition for rate-1 hash functions in FDBL-II to be optimally collision resistant was defined by Hirose [9] as follows.

Definition 4 *For any rate-1 iterated hash function in FDBL-II, if it is optimally collision resistant, then it must be in one of the two types:*

1. *Both* L *and* R *are exceptional,*
2. *Rank*$(L) =$ *Rank*$(R) = 3$*,* $c \oplus d = \lambda_1 a \oplus \lambda_2 b$ *and* $y \oplus z = \lambda_3 w \oplus \lambda_4 x$*, for some* $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \{0, 1\}$*, and the upper right* 2×2 *submatrices of* L *and* R *are both non-singular.*

It was claimed in [9] that the above conditions are not sufficient but just necessary for the property of optimal collision resistance. The two *probably secure* examples (HDBL-1 and HDBL-2) were left as an open problem to check if they are really optimally secure.

3 Security Analysis of FDBL-II

In this section, the security of rate-1 hash functions in FDBL-II is reconsidered. A synthetic analysis is presented which exploits the fact that the former results [9, 26] on the security of FDBL-II are still imprecise. First, two concrete attacks are presented to prove that both HDBL-1 and HDBL-2 are failed to be optimally preimage and second preimage resistant. Next, three counter-examples are described to disclose Hirose's conditions for optimally collision resistant are failed in some uncovered cases. Finally, based on the examples and new attacks, the necessary conditions for rate-1 hash functions in FDBL-II to be optimally secure are refined.

3.1 Attacks on Hirose's Two Examples

Originally, Satoh *et al.* [26] suggested that any rate-1 hash function in FDBL-II will not to be optimally secure, if its compression function does not satisfy the *exceptional* property. Towards this approach, Hirose [9] made a comment on Satoh *et al.*'s result, and said there exist optimally collision resistant hash functions in FDBL-II whose compression functions do not satisfy the *exceptional* property. Moreover, Hirose proposed two examples in FDBL-II (HDBL-1 and HDBL-2, described in Section 2.3) which are *probably secure* against the collision attack. HDBL-2 satisfies the *exceptional* property while HDBL-1 does not, and both of them satisfy Hirose's two necessary conditions in Definition 4. Here we present two concrete attacks on Hirose's two examples which prove they both failed to be optimally (second) preimage resistant.

Theorem 4 *Let HDBL-1 be a hash function defined by the form* (6),

$$\begin{cases} h_i = E_{m_{i,1}||m_{i,2}}(h_{i-1} \oplus g_{i-1}) \oplus h_{i-1} \oplus g_{i-1}, \\ g_i = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus h_{i-1}, \end{cases}$$

then there exists a (second) preimage attack on the hash function with complexity of about $4 \times 2^{3n/2}$.

Proof. By using the idea of the *meet-in-the-middle* attack [17], a preimage attack on the HDBL-1 hash function proceeds as follows.

1. For the preimage attack on (h_i, g_i), an adversary $\mathcal{A}$ chooses an arbitrary message $M = m_1||m_2||\cdots||m_{i-2}$, and by computing the values of (h_{i-2}, g_{i-2}) iteratively from the initial value $IV = h_0||g_0$.
2. Backward step:
   (a) $\mathcal{A}$ tries 2^n operations to find a pair (m_i, c) where $h_i = E_{m_i}(c) \oplus c = E_{m_{i,1}||m_{i,2}}(c) \oplus c$.
   (b) $\mathcal{A}$ chooses 2^n values of h_{i-1} where $c = h_{i-1} \oplus g_{i-1}$. Due to randomness of the outputs of E, $\mathcal{A}$ can find a value of h_{i-1} satisfies $g_i = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus h_{i-1}$.
   (c) $\mathcal{A}$ repeats q_1 times of the above step to obtain q_1 values of $(m_{i,1}, m_{i,2}, h_{i-1}, g_{i-1})$.
3. Forward step: $\mathcal{A}$ chooses q_2 values of m_{i-1}, computes q_2 values of (h'_{i-1}, g'_{i-1}) from $(m_{i-1}, h_{i-2}, g_{i-2})$.

The attack succeeds if some (h_{i-1}, g_{i-1}) and some (h'_{i-1}, g'_{i-1}) are matched. Since the quantities in the meet-in-the-middle attack are $2n$-bit long, the successful probability $\Pr[Pre]$ equals

$$\begin{aligned} \Pr[Pre] &= (1 - \frac{q_1}{2^{2n}}) \cdot (1 - \frac{q_1}{2^{2n} - 1}) \cdots (1 - \frac{q_1}{2^{2n} - q_2}) \\ &\geq (1 - \frac{q_1}{2^{2n} - q_2})^{q_2}. \end{aligned} \tag{10}$$

The complexity of the above attack is the larger value between $2^n \times q_1$ and q_2. For a non-negligible probability in the lowest complexity, it follows that

$$\begin{cases} 2^n \times q_1 = q_2, \\ q_1 \times q_2 = 2^{2n} - q_2. \end{cases} \tag{11}$$

Consequently, it holds that $q_1 \approx 2^{n/2}$ and $q_2 \approx 2^{3n/2}$, then the probability

$$\begin{aligned} \Pr[Pre] &\geq (1 - \frac{2^{n/2}}{2^{2n} - 2^{3n/2}})^{2^{3n/2}} = (1 - \frac{2^{n/2}}{2^{3n/2} \cdot (2^{n/2} - 1)})^{2^{3n/2}} \\ &\approx (1 - \frac{1}{2^{3n/2}})^{2^{3n/2}} \approx 1 - e^{-1} \approx 0.39. \end{aligned} \tag{12}$$

It is easy to see that both the forward step and the backward step require $2 \times 2^{3n/2}$ operations. Thus the total complexity of the attack is $4 \times 2^{3n/2}$. We note that a second preimage attack can be deduced by using the same method. So the theorem holds. □

Similar to HDBL-1, a (second) preimage attack can be found in the HDBL-2 hash function as well. The attack is described in the following theorem.

Theorem 5 *Let HDBL-2 be a hash function defined by the form* (8),

$$\begin{cases} h_i = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus g_{i-1}, \\ g_i = E_{m_{i,1}||m_{i,2}}(g_{i-1}) \oplus h_{i-1}, \end{cases}$$

then there exists a (second) preimage attack on the hash function with complexity of about $4 \times 2^{3n/2}$.

Proof. A (second) preimage attack on the HDBL-2 hash function proceeds as follows.

1. For the preimage attack on (h_i, g_i), $\mathcal{A}$ chooses an arbitrary message $M = m_1||m_2||\cdots||m_{i-2}$, and by computing the values of (h_{i-2}, g_{i-2}) iteratively from the initial value $IV = h_0||g_0$.
2. Backward step:
   (a) $\mathcal{A}$ randomly chooses 2^n values of $(m_{i,1}, m_{i,2}, h_{i-1})$, then computes 2^n values of g_{i-1} where $g_{i-1} = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus h_i$.
   (b) $\mathcal{A}$ repeats the above step $2^{n/2}$ times. Due to randomness of the outputs of E, $\mathcal{A}$ obtains $2^{n/2}$ values of (m_i, h_{i-1}, g_{i-1}) yield the fixed value (h_i, g_i).
3. Forward step: $\mathcal{A}$ randomly chooses $2^{3n/2}$ values of m_{i-1}, then computes $2^{3n/2}$ values of (h'_{i-1}, g'_{i-1}) from $(m_{i-1}, h_{i-2}, g_{i-2})$.

The attack succeeds if some (h_{i-1}, g_{i-1}) and some (h'_{i-1}, g'_{i-1}) are matched. Since the quantities in the meet-in-the-middle attack are $2n$ bits, same to the equations (10), (11) and (12) in the attack of HDBL-1, the successful probability $\Pr[Pre]$ equals 0.39 as well. Consequently, the complexity of the (second) preimage attack is also about $4 \times 2^{3n/2}$. So the theorem holds. □

Since HDBL-1 and HDBL-2 satisfy Type 2 and Type 1 conditions in Definition 4 respectively, the above attacks disclose the point that there might exist uncovered flaws in the former security results of rate-1 hash functions in FDBL-II which are given by Satoh *et al.* [26] and Hirose [9]. Heuristically, we propose three counter-examples, which do not satisfy Hirose's two necessary conditions but still no efficient collision attack can be found, to support this considerable point.

First we give two examples in FDBL-II, which do not satisfy Type 2 condition defined in Definition 4, such that $c \oplus d = \lambda_1 a \oplus \lambda_2 b$ and $y \oplus z = \lambda_3 w \oplus \lambda_4 x$, for some $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \{0, 1\}$.

Example 1:

$$\begin{cases} h_i = E_{m_{i,1}\oplus m_{i,2}\oplus h_{i-1}||m_{i,2}\oplus g_{i-1}}(m_{i,1} \oplus h_{i-1}) \oplus m_{i,2} \oplus g_{i-1}, \\ g_i = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus h_{i-1}. \end{cases} \tag{13}$$

Example 2:

$$\begin{cases} h_i = E_{m_{i,1}\oplus m_{i,2}\oplus h_{i-1}||m_{i,2}\oplus g_{i-1}}(m_{i,1} \oplus m_{i,2} \oplus h_{i-1}) \oplus m_{i,1} \oplus h_{i-1}, \\ g_i = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus h_{i-1}. \end{cases} \tag{14}$$

The third example does not satisfy Type 2 condition that the upper right 2×2 submatrices of L and R are both non-singular.

Example 3:

$$\begin{cases} h_i = E_{m_{i,1}||h_{i-1}}(m_{i,2} \oplus g_{i-1}) \oplus m_{i,2} \oplus g_{i-1}, \\ g_i = E_{m_{i,1}||m_{i,2}}(h_{i-1}) \oplus h_{i-1}. \end{cases} \tag{15}$$

By implementing the similar methods which were used in the indifferentiability analysis of some popular block-cipher-based hash functions [4, 8], here we present an indifferentiability analysis on two typical examples in FDBL-II as well. Let distinguisher $\mathcal{D}$ can access two cryptosystems $(\mathcal{O}_1, \mathcal{O}_2)$ where $\mathcal{O}_1 = (H, E, E^{-1})$ and $\mathcal{O}_2 = (Rand, S, S^{-1})$. Let $r_i \leftarrow ((h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i))$ be the i-th query-response to the oracles $\{E, E^{-1}, S, S^{-1}\}$ where $m_i \in \{0,1\}^{2n}$. $\mathcal{R}_i = (r_1, \cdots, r_i)$ denotes the query-response set on the oracles $\{E, E^{-1}, S, S^{-1}\}$ after the i-th query. Let $r'_i \leftarrow (IV \xrightarrow{M_i} (h_i, g_i))$ be the i-th query-response to the oracles $\{H, Rand\}$, where $M_i \in \mathcal{M}$. $\mathcal{R}'_i = (r'_1, \cdots, r'_i)$ denotes the query-response set on the oracles $\{H, Rand\}$ after the i-th query. A *functional closure* $\mathcal{R}^*$ on $\mathcal{R}$ is the set with the following properties.

1. If $(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i), (h_i, g_i) \xrightarrow{m_{i+1}} (h_{i+1}, g_{i+1}) \in \mathcal{R}_{i+1}$, then $(h_{i-1}, g_{i-1}) \xrightarrow{m_i||m_{i+1}} (h_{i+1}, g_{i+1}) \in \mathcal{R}^*_{i+1}$.

2. If $(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i), (h_{i-1}, g_{i-1}) \xrightarrow{m_i||m_{i+1}} (h_{i+1}, g_{i+1}) \in \mathcal{R}_{i+1}$, then $(h_i, g_i) \xrightarrow{m_{i+1}} (h_{i+1}, g_{i+1}) \in \mathcal{R}^*_{i+1}$.

The algorithm $Pad(\cdot)$ denotes the indifferentiable padding rules, e.g., the prefix-free padding, HMAC/NMAC and the chop construction, which were analyzed in [5, 6, 8]. For brevity, we note that all of the examples are implicitly implemented with one of those padding rules. To avoid some trivial attacks, the last block contains the length of input. First we give the following theorem which establishes the indifferentiability result of *Example 1*.

Theorem 6 *The rate-1 hash function defined by (13) is (t_D, t_S, q, ϵ)-indifferentiable from a random oracle in the ideal cipher model with the prefix-free padding and the HMAC/NMAC construction, for any distinguisher $\mathcal{D}$ in polynomial time bound t_D, with $t_S = 2l \cdot O(q)$ and the advantage $\epsilon = 2^{-n+2} \cdot l \cdot O(q)$, where l is the maximum length of a query made by $\mathcal{D}$ and $l \cdot q \leq 2^{n-1}$.*

Proof. First we give a simulation to prove that *Example 1* (denoted by H in the following simulation) is indifferentiable from a random oracle $Rand$ in the ideal cipher model with complexity of $l \cdot q \leq 2^{n-1}$.

- $Rand$**-Query.** For the i-th query $M_i \in \mathcal{M}$ on $Rand$, if M_i is a repetitive query, the oracle $Rand$ retrieves $r'_j \leftarrow (IV \xrightarrow{M_i} (h_j, g_j))$ where $r'_j \in \mathcal{R}'_{i-1}, j \leq i-1$, then returns $Rand(M_i) = (h_j, g_j)$. Else $Rand$ randomly selects a hash value $(h_i, g_i) \in \mathcal{Y}$ and updates $\mathcal{R}'_i = \mathcal{R}'_{i-1} \cup \{IV \xrightarrow{M_i} (h_i, g_i)\}$, then returns $Rand(M_i) = (h_i, g_i)$.

- $\{S, S^{-1}\}$**-Query.** To answer the distinguisher $\mathcal{D}$'s encryption and decryption queries, the simulator $\mathcal{S}$ proceeds as follows.

    1. For the i-th query $(1, k_i, x_i)$ on S:
        (a) If $\exists IV \xrightarrow{M} (h_{i-1}, g_{i-1}) \in \mathcal{R}^*_{i-1}$, $\mathcal{S}$ computes $Pad(M) = m_i = m_{i,1}||m_{i,2}$, and then
            i. if $k_i = m_{i,1} \oplus m_{i,2} \oplus h_{i-1}||m_{i,2} \oplus g_{i-1}$ and $x_i = m_{i,1} \oplus h_{i-1}$, $\mathcal{S}$ runs $Rand(M)$ and obtains the response (h_i, g_i), updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$, then returns $y_i = h_i \oplus k_{i,2}$;
            ii. if $k_i = m_{i,1}||m_{i,2}$ and $x_i = h_{i-1}$, $\mathcal{S}$ runs $Rand(M)$ and obtains the response (h_i, g_i), and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$, then returns $y_i = g_i \oplus x_i$.
        (b) Else $\mathcal{S}$ randomly selects $(h_i, g_i, h_{i-1}, g_{i-1})$, computes $m_{i,2} = k_{i,2} \oplus g_{i-1}$ and $m_{i,1} = k_{i,1} \oplus m_{i,2} \oplus h_{i-1}$, then adds the tuple $(1, k'_i, x'_i, y'_i)$ as $x'_i = h_{i-1}$, $y'_i = g_i \oplus x_i$ and $k'_i = m_{i,1}||m_{i,2}$, and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$, then returns $y_i = h_i \oplus m_{i,2} \oplus g_{i-1}$.
    2. For the i-th query $(-1, k_i, y_i)$ on S^{-1}:
        (a) If $\exists IV \xrightarrow{M} (h_{i-1}, g_{i-1}) \in \mathcal{R}^*_{i-1}$, $\mathcal{S}$ computes $Pad(M) = m_i = m_{i,1}||m_{i,2}$, and then
            i. if $k_i = m_{i,1} \oplus m_{i,2} \oplus h_{i-1}||m_{i,2} \oplus g_{i-1}$, $\mathcal{S}$ runs $Rand(M)$ and obtains the response (h_i, g_i). And then, if $y_i = h_i \oplus m_{i,2} \oplus g_{i-1}$, $\mathcal{S}$ updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$ and returns $x_i = m_{i,1} \oplus h_{i-1}$;
            ii. if $k_i = m_{i,1}||m_{i,2}$, $\mathcal{S}$ runs $Rand(M)$ and obtains the response (h_i, g_i). And then, if $y_i = g_i \oplus h_{i-1}$, $\mathcal{S}$ updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$ and returns $x_i = h_{i-1}$.
        (b) Else $\mathcal{S}$ randomly selects (g_i, h_{i-1}, g_{i-1}), computes $h_i = y_i \oplus k_{i,2}$, $m_{i,2} = k_{i,2} \oplus g_{i-1}$ and $m_{i,1} = k_{i,1} \oplus m_{i,2} \oplus h_{i-1}$, then adds the tuple $(1, k'_i, x'_i, y'_i)$ as $x'_i = h_{i-1}$, $y'_i = g_i \oplus x_i$ and $k'_i = m_{i,1}||m_{i,2}$, and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$, then returns $x_i = h_i \oplus m_{i,2} \oplus g_{i-1}$.

Before stating the indifferentiability result of *Example 1*, the probability of the indifferentiable events on *Example 1* can be obtained from the above simulation.

Lemma 1 *In double block length hash functions with the form (13), it holds that* $\Pr[Pre] = 2^{-(3n+4)/2} \cdot l \cdot O(q)$ *and* $\Pr[Coll] = 2^{-n+1} \cdot l \cdot O(q)$*, where* l *is the maximum number of length in a hash query.*

Proof. In case of $\mathcal{O}_2 = (Rand, S, S^{-1})$, the total number of choices is $l \cdot q$, where l is the maximum number of length in a hash query. For every $2 \leq j \leq l \cdot q$, let $Coll_j$ be the collision event that a pair of inputs yield a same output after the j-th queries. Namely, for some $j' < j$, it follows that

$$(h_j, g_j) = (h_{j'}, g_{j'}) \; or \; h_j = g_j,$$

which is equivalent to

$$(y_j \oplus k_{j,2}, y'_j \oplus x'_j) = (y_{j'} \oplus k_{j',2}, y'_{j'} \oplus x'_{j'}) \; or \; (y_j \oplus k_{j,2} = y'_j \oplus x'_j).$$

Since (h_i, g_i) are randomly selected by the simulator $\mathcal{S}$ from the range $\{0,1\}^n$ where $i \in \{1, 2, \cdots, l \cdot q\}$, the probability that the collision event happens after the j-th queries is as follows.

$$\Pr[Coll_j] \leq \frac{(j-1)}{(2^n - (j-1)) \cdot (2^n - (j-1))} + \frac{1}{2^n}.$$

Let $Coll$ be the collision event that a pair of inputs yield a same output after the maximum q times queries. Thus, if $l \cdot q \leq 2^{n-1}$,

$$\begin{aligned} \Pr[Coll] &= \Pr[Coll_1 \vee Coll_2 \vee \cdots \vee Coll_{l \cdot q}] \leq \sum_{j=2}^{l \cdot q} \Pr[Coll_j] \\ &\leq \sum_{j=2}^{l \cdot q} (\frac{j-1}{(2^n - (j-1)) \cdot (2^n - (j-1))} + \frac{1}{2^n}) \\ &\leq \frac{\sum_{j=2}^{l \cdot q}(j-1)}{(2^n - 2^{n-1}) \cdot (2^n - 2^{n-1})} + \frac{l \cdot q}{2^n} \\ &\leq \frac{(1 + l \cdot q) \cdot (l \cdot q)}{2^{2n-1}} + \frac{l \cdot q}{2^n} \leq \frac{2^{n-1}(l \cdot q) + (l \cdot q) + 2^{n-1}(l \cdot q)}{2^{2n-1}} \approx \frac{l \cdot q}{2^{n-1}}. \end{aligned} \tag{16}$$

From the preimage attack on HDBL-1 in Theorem 4, it is easy to see the probability of the preimage events Pre is

$$\begin{aligned} \Pr[Pre] &= \Pr[Pre_1 \vee Pre_2 \vee \cdots \vee Pre_{l \cdot q}] \leq \sum_{j=1}^{l \cdot q} \Pr[Pre_j] \\ &\leq \sum_{j=1}^{l \cdot q} (\frac{1}{4 \times 2^{3n/2}}) \leq \frac{l \cdot q}{4 \times 2^{3n/2}}. \end{aligned} \tag{17}$$

Consequently, the probability of the indifferentiable event Bad [4, 8] is

$$\Pr[Bad] = 2 \times Max(\Pr[Coll], \Pr[Pre]) = 2 \times \Pr[Coll] = 2^{-n+2} \cdot l \cdot O(q).$$

By implementing the advantage of indifferentiability in keyed hash function [4, 8], similar results can be easily deduced in the HMAC/NMAC construction. So the theorem follows. □

The following theorem establishes the indifferentiability result of HDBL-1. Since the proof method is similar to Theorem 6, it is omitted here for the ease of readers. A detailed proof can be found in Appendix A.

Theorem 7 *The rate-1 hash function defined by (6) is (t_D, t_S, q, ϵ)-indifferentiable from a random oracle in the ideal cipher model with the prefix-free padding and the HMAC/NMAC construction, for any distinguisher $\mathcal{D}$ in polynomial time bound t_D, with $t_S = 2l \cdot O(q)$ and the advantage $\epsilon = 2^{-n+2} \cdot l \cdot O(q)$, where l is the maximum length of a query made by $\mathcal{D}$ and $l \cdot q \leq 2^{n-1}$.*

According to the above indifferentiability analysis, it is easy to see that *Example 1* and HDBL-1 are also optimally collision resistant in the ideal cipher model (which are proven by Lemma 1 and Lemma 4 in Appendix A respectively). Based on the above results and the improved bound of the chopDBL construction [5], one can easily obtain similar indifferentiability results of *Example 1* and HDBL-1 with the chop construction.

Theorem 8 *The rate-1 hash functions defined by (6) and (13) are (t_D, t_S, q, ϵ)-indifferentiable from a random oracle in the ideal cipher model with the chop construction, for any distinguisher $\mathcal{D}$ in polynomial time bound t_D, with $t_S = l \cdot O(q^2)$ and the advantage $\epsilon = O(\frac{(2n-s)q}{2^s} + \frac{(lq)^2}{2^{2n+1}} + \frac{lq}{2^{2n-s-1}})$, where the chopped bit size is s and l is the maximum length of a query made by $\mathcal{D}$ and $l \cdot q \leq 2^{n-1}$.*

Proof. Since both *Example 1* and HDBL-1 are DBL hash functions such that the length of any internal hash value is $2n$ bits, this wide-pipe property [18] are good at resisting Joux's r-multicollision attack [12] and Kelsey-Schiner second preimage attack [13]. From Lemma 1, it is easy to see that *Example 1* has the probability of the collision event is $\Pr[Coll] \leq \frac{(lq)^2}{2^{2n-1}}$ in the ideal cipher model. Thus we have $\Pr[Coll^s] \leq \frac{(lq)^2}{2^{2n-s-1}}$ where the chopped bit size is s. By using the similar simulation in Theorem 6 and an improved indifferentiability bound of the chopDBL hash function [5], one can easily deduce the indifferentiability result of *Example 1* with the chop construction in the ideal cipher model as follows.

Let $Bad_i, i = \{1, 2\}$ be the set of the indifferentiable events on the two cryptosystems $\mathcal{O}_1 = (H, E, E^{-1})$ and $\mathcal{O}_2 = (Rand, S, S^{-1})$, respectively. The oracles $\{H, E, E^{-1}\}$ and $\{Rand, S, S^{-1}\}$ are identically distributed in the past view of the distinguisher and Bad_i does not occur. If $\mathcal{D}$ is a distinguisher then we write $Adv(\mathcal{D})$ as a measure of the maximal advantage of indifferentiability over all distinguishers $\mathcal{D}$. For brevity, D_1 denotes the event $\mathcal{D}^{H,E,E^{-1}} = 1$ and D_2 denotes the event $\mathcal{D}^{Rand,S,S^{-1}} = 1$. By using the Strong Interpolation Theorem in [5], the advantage of indifferentiability on *Example 1* with the chop construction is at most

$$\begin{aligned}
Adv(\mathcal{D}) &= |Pr[\mathcal{D}^{H,E,E^{-1}} = 1] - Pr[\mathcal{D}^{Rand,S,S^{-1}} = 1]| \\
&= |(Pr[D_1 \cap Bad_1] + Pr[D_1 \cap \neg Bad_1]) - (Pr[D_2 \cap Bad_2] + Pr[D_2 \cap \neg Bad_2])| \\
&= |(Pr[D_1 \cap Bad_1] - (Pr[D_2 \cap Bad_2]) + Pr[D_1 \cap \neg Bad_1] - Pr[D_2 \cap \neg Bad_2])| \\
&\leq |Pr[D_1 \cap \neg Bad_1] - Pr[D_2 \cap \neg Bad_2]| + |Pr[D_1 \cap Bad_1] - Pr[D_2 \cap Bad_2]|.
\end{aligned}$$

Intuitively, to obtain a maximum probability of $\varepsilon_1 = |Pr[D_1 \cap \neg Bad_1] - Pr[D_2 \cap \neg Bad_2]|$, we can choose a lower bound of $Pr[D_2 \cap \neg Bad_2]$ and an uppper bound of $Pr[D_1 \cap \neg Bad_1]$. If $s < n$, the adversary can guess the chopped s bits of the outputs of $\mathcal{O}_1$ to implement an extension attack on $\mathcal{O}_2$ [6]. Due to randomness of the outputs of E, E^{-1} and the improved bound of chopDBL [5], we have

$$\varepsilon_1 \leq \frac{(2n-s)q}{2^s} + \frac{(lq)^2}{2^{2n+1}}.$$

To obtain a maximum probability of $\varepsilon_2 = |Pr[D_1 \cap Bad_1] - Pr[D_2 \cap Bad_2]|$, Chang and Nandi [5] proved that the upper bound of ε_2 is a r-multicollision among lq uniformly and independently chosen $(2n-s)$ bits. Based on the probability $\Pr[Coll^s] \leq \frac{(lq)^2}{2^{2n-s-1}}$ and Joux's multicollision attack [12], if we choose $r = 2n - s$, it is easy to see that

$$\begin{aligned}
\varepsilon_2 &\leq \frac{\binom{lq}{r}}{2^{(2n-s)(r-1)}} \\
&\leq (lq/2^{2n-s-1})^r \leq lq/2^{2n-s-1}.
\end{aligned}$$

By combining the above results, we obtain the indifferentiability of *Example 1* with the chop construction in the ideal cipher model as follows.

$$\begin{aligned} Adv(\mathcal{D}) &\leq |Pr[D_1 \cap \neg Bad_1] - Pr[D_2 \cap \neg Bad_2]| + |Pr[D_1 \cap Bad_1] - Pr[D_2 \cap Bad_2]| \\ &\leq \varepsilon_1 + \varepsilon_2 \leq \frac{(2n-s)q}{2^s} + \frac{(lq)^2}{2^{2n+1}} + \frac{lq}{2^{2n-s-1}}. \end{aligned}$$

Since HDBL-1 with the chop construction has the same wide-pipe property and the probability of the collision event (Lemma 4 in Appendix A), the proof of the indifferentiability of HDBL-1 will be identical to *Example 1.* So the theorem follows. □

From the above concrete attacks and the counter-examples, it is easy to see that Hirose's two necessary conditions are still imprecise for rate-1 hash functions in FDBL-II to be optimally secure against preimage, second preimage and collision attacks. A more rigorous analysis is required to exploit the certain conditions which should be imposed on FDBL-II for optimal security.

3.2 The Exact Security of FDBL-II

Although Hirose [9] made a comment that the attacks presented by Satoh *et al.* [26] are infeasible for some hash functions in FDBL-II, as is expected even the underlying compression function unlikely satisfies the exceptional property. E.g., HDBL-1 is a counter-example that supports this comment. According to the three counter-examples which are described in Section 3.1, Hirose's conditions [9] become inaccurate as well. Moreover, Since HDBL-2 is an instance of FDBL-II with the exceptional property, the two concrete attacks on HDBL-1 and HDBL-2 show the fact that the result given by Satoh *et al.* [26] can not imply the optimal security. The exact security of rate-1 hash functions in FDBL-II is reconsidered through the following attacks. First generic attacks are presented.

Theorem 9 *For any rate-1 hash functions in FDBL-II with the form (3), if T operations are required to find a block $m_i = m_{i,1}||m_{i,2}$ for any given value of (h_{i-1}, g_{i-1}), such that the resulting four-tuple $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$ yields the fixed value for h_i (or g_i or $h_i \oplus g_i$), then there exist collision, preimage, and second preimage attacks on the hash function with complexities of about $(T+3) \times 2^{n/2}$, $(T+3) \times 2^n$, and $(T+3) \times 2^n$, respectively.*

Proof. An adversary $\mathcal{A}$ starts the attacks by choosing an arbitrary message $M = m_1||m_2|| \cdots ||m_{i-2}$, and by computing the values of (h_{i-2}, g_{i-2}) iteratively from the initial value $IV = h_0||g_0$. The initial operations for the values of (h_{i-2}, g_{i-2}) can be ignored if $i \ll 2^{n/2}$.

For (second) preimage attacks, $\mathcal{A}$ searches for two blocks m_{i-1} and m_i such that the fixed hash value (h_i, g_i) is hit. First, $\mathcal{A}$ computes the pair (h_{i-1}, g_{i-1}) from the given values (h_{i-2}, g_{i-2}) and $(m_{i-1,1}, m_{i-1,2})$. Next, $\mathcal{A}$ finds a block $(m_{i,1}, m_{i,2})$ such that the resulting four-tuple $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$ yields the fixed value for h_i(or g_i or $h_i \oplus g_i$). This step costs T times of encryption or decryption. Finally, $\mathcal{A}$ computes the value of g_i (or h_i) from the tuple $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$. If the value is not hit, $\mathcal{A}$ will repeat the above steps at most 2^n times. Due to randomness of the outputs, the probability of finding the (second) preimage in the above procedure is non-negligible. The total complexity of these (second) preimage attacks is about $(T+3) \times 2^n$.

For collision attacks, $\mathcal{A}$ searches for a pair of the blocks (m_{i-1}, m_i) and (m'_{i-1}, m'_i) yields the same hash value (h_i, g_i). First, $\mathcal{A}$ chooses a value of h_i. Then $\mathcal{A}$ proceeds $2^{n/2}$ times in the same way as the preimage attack. Due to the birthday paradox, the probability of finding the collision in the above procedure is non-negligible. The total complexity of these collision attacks is about $(T+3) \times 2^{n/2}$. So the theorem holds. □

Subsequently, the attacks that simultaneously break optimal collision and (second) preimage resistances are described as follows.

Lemma 2 *For any rate-1 hash function in FDBL-II with the form (3), if the rank of L (or R) is less than three, then there exist collision, preimage, and second preimage attacks on the hash function with complexities of about $4 \times 2^{n/2}$, 3×2^n, and 3×2^n, respectively.*

Proof. Consider the general form of FDBL-II. Since the rank of L (or R) is at most two and h_i (or g_i) depends on a subspace of $(m_{i,1}, m_{i,2}, h_{i-1}, g_{i-1})$, it follows that an adversary has at least one dimensional of freedom to find the values of $m_{i,1}$ (or $m_{i,2}$ or $m_{i,1} \oplus m_{i,2}$) yields the given hash value (h_i, g_i). Based on the attacks defined by Theorem 9, it is easy to prove that $T \simeq 0$ in the (second) preimage attack, and $T \simeq 1$ in the collision attack. So the lemma holds. □

Lemma 3 *For any rate-1 hash function in FDBL-II with the form (3), if the rank of L_r^3 (or L_r^4 or R_r^3 or R_r^4) is less than two, then there exist collision, preimage, and second preimage attacks on the hash function with complexities of about $4 \times 2^{n/2}$, 3×2^n, and 3×2^n, respectively.*

Proof. Consider the general form of FDBL-II. If either the rank of L_r^3 or L_r^4 is less than two, then the key $A||B$ of $E_{A||B}(C)$ (or $E^{-1}_{A||B}(h_i \oplus D)$) depends on one dimensional of $(m_{i,1}, m_{i,2})$ (or $m_{i,1} \oplus m_{i,2}$). Let (a, b, c, d) be the values of (A, B, C, D) used in the computations of h_i. By computing $d = E_{a||b}(c) \oplus h_i$ (in case of $Rank(L_r^4) < 2$) or $c = E^{-1}_{a||b}(d \oplus h_i)$ (in case of $Rank(L_r^3) < 2$), an adversary can decide the value of $m_{i,1}$ (or $m_{i,2}$) from the hash values of $(h_{i-1}, g_{i-1}, h_i, g_i)$. Based on the attacks in Theorem 9, it is easy to prove that $T \simeq 0$ in the (second) preimage attack, and $T \simeq 1$ in the collision attack. Same result holds if the rank of R_r^3 or R_r^4 is less than two. □

Furthermore, the attacks that just break the property of optimal collision or (second) preimage resistance are described as follows.

Theorem 10 *For any rate-1 hash function in FDBL-II with the form (3), if both the second column of L and the first column of R are zero column vectors, then there exists a collision attack on the hash function with complexity of about $O(n \cdot 2^{n/2})$.*

Proof. Consider the general form of FDBL-II. Because the second column of L and the first column of R are zero column vectors, so h_i does not depend on g_{i-1} and g_i does not depend on h_{i-1} in mutual. It is easy to see the hash value (h_i, g_i) is simply computed from a concatenation of two parallel SBL hash functions. Due to Joux's multicollision attack [12], we can find $2^{n/2}$ different messages yield the same hash value h_i with complexity of about $O(n \cdot 2^{n/2})$, which implies at least one pair of messages yield the same hash value g_i with a non-negligible probability. So the theorem follows. □

Theorem 11 *For any rate-1 hash function in FDBL-II with the form (3), there exists a (second) preimage attack on the hash function with complexity of about $4 \times 2^{3n/2}$.*

Proof. Consider the general form of FDBL-II. Let (a, b, c, d) be the values of (A, B, C, D) used in the computations of h_i. If the rank of L or R is less than three, then the result follows from Lemma 2; If the rank of L or R is greater or equal three, an adversary $\mathcal{A}$ starts the attacks by choosing an arbitrary message $M = m_1||m_2|| \cdots ||m_{i-2}$, and by computing the values of (h_{i-2}, g_{i-2}) iteratively from the given initial value $IV = h_0||g_0$.

1. Backward step:
   - If the rank of L is three, $\mathcal{A}$ randomly chooses 2^n values of (a, b, c, d) which satisfy the linear combination of L, then $\mathcal{A}$ tries 2^n values of (a, b, c, d) to find a tuple (a, b, c, d) yields the given value $h_i = E_{a||b}(c) \oplus d$;
   - If the rank of L is four, $\mathcal{A}$ randomly chooses 2^n values of (a, b, c), then $\mathcal{A}$ computes 2^n values of d where $d = E_{a||b}(c) \oplus h_i$. $\mathcal{A}$ tries to find at least one tuple (h_{i-1}, g_{i-1}, m_i) from (a, b, c, d) that satisfies the equation.
   - $\mathcal{A}$ repeats the above step $2^{n/2}$ times. Due to randomness of the outputs, $\mathcal{A}$ can obtain $2^{n/2}$ values of (m_i, h_{i-1}, g_{i-1}) yield the fixed value (h_i, g_i).

2. Forward step: $\mathcal{A}$ randomly chooses $2^{3n/2}$ values of m_{i-1}, then computes $2^{3n/2}$ values of (h'_{i-1}, g'_{i-1}) from $(m_{i-1}, h_{i-2}, g_{i-2})$.

It is easy to see the attack will succeed with a non-negligible probability from the equation (12). The total complexity is about $4 \times 2^{3n/2}$. So the theorem follows. □

We stress that both HDBL-1 and HDBL-2 are failed to be optimally (second) preimage resistance due to Theorem 11. The complexity of the generic second preimage attack, which was proposed by Kelsey and Schneier in [13], can be asymptotically smaller than ours. But their attack needs an unpractical long message, which makes it become less attractive. E.g., for $2n$-bit hash functions, the generic second preimage attack requires a 2^x-bit long message with about $x \times 2^{n+1} + 2^{2n-x+1}$ complexity. Thus the generic second preimage attack with the complexity of about $O(2^{3n/2})$ requires a $2^{n/2}$-bit long message.

Based on the above results, necessary conditions for rate-1 hash functions in FDBL-II to be optimally secure are refined as follows. It is easy to see that the same result similarly follows in the serial situation of FDBL-II.

Corollary 1 *For any rate-1 hash functions in FDBL-II, if the compression function matches one of the following two conditions:*

1. *The rank of L or R is less than three;*
2. *The rank of L_r^3(or L_r^4 or R_r^3 or R_r^4) is less than two,*

then there exist collision , preimage and second preimage preimage attacks with a non-negligible successful probability must spend the complexities of about $O(2^{n/2})$, $O(2^n)$ and $O(2^n)$, respectively. Furthermore, if both the second column of L and the first column of R are zero column vertexes, then there exists a collision attack on the hash function with complexity of about $O(n \cdot 2^{n/2})$. For all of the rate-1 hash functions in FDBL-II, there exist preimage and second preimage attacks with a non-negligible successful probability must spend the same complexity of about $O(2^{3n/2})$.

Based on the attacks on FDBL-I and FDBL-II, a fully negative result is extended to a new class of DBL hash functions with rate 1 (denoted by FDBL-III), where one block cipher has the key length equal to the block length, while the other is doubled. For brevity, details can be found in Appendix B.

4 Conclusion

In this paper, the security of FDBL-II has been reconsidered and the necessary conditions for optimally collision resistant are refined. It is proven that all rate-1 hash functions in FDBL-II are failed to be optimally (second) preimage resistant. Moreover, the indifferentiability analysis supported that there exist paradigms in FDBL-II which can be indifferentiable from a random oracle in the ideal cipher model, and also they are optimally collision resistant. These cryptanalysis results give a complete view to the rate-1 DBL hash functions based on existed block ciphers, which are helpful for the design of secure and fast DBL hash functions. In practice, AES algorithm can be simply implemented in hardware circuits, i.e., a fully AES-based cryptosystem on chip (uses AES as block cipher, while uses the proposed constructions as hash function) is meaningful.

Since key length will definitely impact the efficiency, e.g., AES encrypts 20% slower for 192-bit keys and 40% slower for 256-bit keys. The definition of the hash rate is not appropriate for the new designs of double block (or multi-block) length hash functions. Generally, a rate-1 DBL hash function in FDBL-I cannot directly compare to such one in FDBL-II. To solve this inaccuracy, a new preferable concept should be defined instead of the hash rate for the measurement of the efficiency. At FSE 2008, Knudsen roughly presented a new definition on the hash rate, which takes into account the key schedule and the block length as well [16]. We think Knudsen's new definition is still imprecise, since the key length is ignored. E.g., the performances of hash functions, which are based on different block ciphers with different key lengths but same key schedules and block lengths, will apparently be inequivalent. Future work is to summarize a generic proof on block-cipher-based hash functions with variants of block and key length through a preferable definition on the hash rate.

Acknowledgments. We would like to thank many anonymous reviewers for their helpful comments that improved the presentation of this paper.

References

[1] J. Black, P. Rogaway and T. Shrimpton. Black-Box Analysis of the Black-Cipher-Based Hash-Function Constructions from PGV. In *Advances in Cryptology-CRYPTO'02*, LNCS 2442, pp. 320-335, 2002.

[2] B.O. Brachtl, D. Coppersmith, M.M. Hyden, S.M. Matyas, C.H. Meyer, J. Oseas, S. Pilpel and M. Schilling. *Data Authentication Using Modification Detection Codes Based on a Public One Way Encryption Function.* U.S. Patent Number 4,908,861, March 13, 1990.

[3] L. Brown, J. Pieprzyk, and J. Seberry. LOKI-a cryptographic primitive for authentication and secrecy applications. In J. Seberry and J. Pieprzyk (Eds): *Advances in Cryptology-AusCrypt'90*, LNCS 453, pp. 229-236, Springer-Verlag, Berlin, 1990.

[4] D. H. Chang, S. J. Lee, M. Nandi and M. Yung. Indifferentiable Security Analysis of Popular Hash Functions with Prefix-Free Padding. In X. Lai and K. Chen (Eds): *ASIACRYPT 2006*, LNCS 4284, pp. 283-298, 2006.

[5] D. H. Chang and M. Nandi. Improved Indifferentiability Security Analysis of chopMD Hash Function. In K. Nyberg (Ed.): *FSE 2008*, LNCS 5086, pp. 429-443, 2008.

[6] J. S. Coron, Y. Dodis, C. Malinaud and P. Puniya. Merkle-Damgard Revisited: How to Construct a Hash Function. In *Advances in Cryptology-CRYPTO'05*, LNCS 3621, pp. 21-39, 2005.

[7] I. Damgard. A Design Principle for Hash Functions, In *Advances in Cryptology-Cyrpto'89*, LNCS 435, pp. 416-427, 1989.

[8] Z. Gong, X. Lai, and K. Chen. A Synthetic Indifferentiability Analysis of Some Block-Cipher-Based Hash Functions. *Designs, Codes and Cryptography*, Springer. 48:3, Sept 2008.

[9] S. Hirose. A Security Analysis of Double-Block-Length Hash Functions with the Rate 1. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E89-A, NO.10, pp. 2575-2582, Oct 2006.

[10] S. Hirose. Some Plausible Constructions of Double-Block-Length Hash Functions. In *FSE 2006*, LNCS 4047, pp. 210-225, 2006.

[11] W. Hohl, X. Lai, T. Meier, and C. Waldvogel. Security of iterated hash function based on block ciphers. In *CRYPTO'93*, LNCS 773, pp. 379-390, 1993.

[12] A. Joux. Multicollisions in iterated hash functions, Application to cascaded constructions. In *Crypto 2004*, LNCS 3152, pp. 306-316, 2004.

[13] J. Kelsey and B. Schneier. Second Preimages on n-Bit Hash Functions for Much Less than $2n$ Work. In *EUROCRYPT 2005*, LNCS 3494, pp. 474-490, 2005.

[14] L.R. Knudsen. Block Ciphers-Analysis, Design and Applications. *Ph. D. thesis*, Aarthus University, 1994.

[15] L. R. Knudsen, X. Lai, and B. Preneel. Attacks on fast double block length hash functions. *Journal of Cryptology*, 11(1):59-72, 1998.

[16] L. R. Knudsen. Hash Functions and SHA-3. Invited talk in *FSE 2008*.

[17] X. Lai and J. L. Massey. Hash Functions Based on Block Ciphers. In *Advances in Cryptology-Eurocrypt'92*, LNCS 658, pp. 55-70, 1993.

[18] S. Lucks. A Failure-Friendly Design Principle for Hash Functions. In *ASIACRYPT 2005*, LNCS 3788, pp. 474-494, 2005.

[19] U. Maurer, R. Renner, and C. Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In *Theory of Cryptography - TCC 2004*, LNCS 2951, pp. 21-39, 2004.

[20] R.C. Merkle. One way hash functions and DES, *Advances in Cryptology-Crypto'89*, LNCS 435, pp. 428-446, 1989.

[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 *INDOCRYPT 2005*, LNCS 3797, pp. 77-89, 2005.

[23] B. Preneel, A, Bosselaers, R. Govaerts and J. Vandewalle. Collision-free Hash-functions Based on Blockcipher Algorithms. In *Proceeding of 1989 International Carnahan Conference on Security Technology*, pp. 203-210, 1989.

[24] B. Preneel, R. Govaerts and J. Vandewalle. Hash functions based on block ciphers: A synthetic approach. In *Advances in Cryptology-CRYPTO'93*, LNCS 773, pp. 368-378, 1994.

[25] P. Rogaway and T. Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance and Collision Resistance. In *FSE 2004*, LNCS 3017, pp. 371-388, 2004.

[26] T. Satoh, M. Haga, and K. Kurosawa. Towards Secure and Fast Hash Functions. *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E82-A, NO.1, pp. 55-62, Jan 1999.

[27] C. Shannon. Communication theory of secrecy systems. Bell Systems Techincal Journal, 28(4): pp. 656-715, 1949.

[28] J. P. Steinberger. The Collision Intractability of MDC-2 in the Ideal-Cipher Model. In *EUROCRYPT 2007*, LNCS 4515, pp. 34-51, 2007.

[29] X. Yi and K.Y. Lam. A New Hash Function Based on Block Cipher. In *ACISP'97 Information Security and Privacy*, LNCS 1270, pp. 139-146, Springer-Verlag, 1997.

A. Proof of Theorem 7

Here we give an indifferentiability analysis on HDBL-1 (described in Section 2.3), which is a typical rate-1 hash functions in FDBL-II as well. The proof is also related to the collision resistance of HDBL-1.

- $Rand$-**Query.** For the i-th $Rand$-query $M_i \in \mathcal{M}$, if M_i is a repetitive query, the oracle $Rand$ retrieves $r'_j \leftarrow (IV \xrightarrow{M_i} (h_j, g_j))$ where $r'_j \in \mathcal{R}'_{i-1}, j \leq i-1$, then returns $Rand(M_i) = (h_j, g_j)$. Else $Rand$ randomly selects a hash value $(h_i, g_i) \in \mathcal{Y}$ and updates $\mathcal{R}'_i = \mathcal{R}'_{i-1} \cup \{IV \xrightarrow{M_i} (h_i, g_i)\}$, then returns $Rand(M_i) = (h_i, g_i)$.

- $\{S, S^{-1}\}$**-Query.** To answer the distinguisher $\mathcal{D}$'s encryption and decryption queries, the simulator $\mathcal{S}$ proceeds as follows.

  1. For the i-th query $(1, k_i, x_i)$ on S:

     (a) If $\exists IV \xrightarrow{M} (h_{i-1}, g_{i-1}) \in \mathcal{R}^*_{i-1}$, $\mathcal{S}$ computes $Pad(M) = m_i = m_{i,1}||m_{i,2}$, and then

       i. if $k_i = m_{i,1}||m_{i,2}$ and $x_i = h_{i-1} \oplus g_{i-1}$, $\mathcal{S}$ runs $Rand(M)$ and obtains the response (h_i, g_i), updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$, then returns $y_i = h_i \oplus x_i$;

       ii. if $k_i = m_{i,1}||m_{i,2}$ and $x_i = h_{i-1}$, $\mathcal{S}$ runs $Rand(M)$ and obtains the response (h_i, g_i), and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$, then returns $y_i = h_i \oplus x_i$.

     (b) Else $\mathcal{S}$ randomly selects (h_i, g_i, g_{i-1}), computes $m_{i,1} = k_{i,1}$, $m_{i,2} = k_{i,2}$ and $h_{i-1} = x_i \oplus g_{i-1}$, then adds the tuple $(1, k'_i, x'_i, y'_i)$ as $x'_i = g_{i-1}$, $y'_i = g_i \oplus x'_i \oplus h_{i-1}$ and $k'_i = k_i$, and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$, then returns $y_i = h_i \oplus x_i$.

  2. For the i-th query $(-1, k_i, y_i)$ on S^{-1}:

     (a) If $\exists IV \xrightarrow{M} (h_{i-1}, g_{i-1}) \in \mathcal{R}^*_{i-1}$, $\mathcal{S}$ computes $Pad(M) = m_i = m_{i,1}||m_{i,2}$, and then

       i. if $k_i = m_{i,1}||m_{i,2}$, $\mathcal{S}$ runs $Rand(M)$ and obtains the response (h_i, g_i). And then, if $y_i = h_i \oplus h_{i-1} \oplus g_{i-1}$, $\mathcal{S}$ updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$ and returns $x_i = h_{i-1} \oplus g_{i-1}$;

       ii. if $k_i = m_{i,1}||m_{i,2}$, $\mathcal{S}$ runs $Rand(M)$ and obtains the response (h_i, g_i). And then, if $y_i = g_i \oplus h_{i-1}$, $\mathcal{S}$ updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$ and returns $x_i = h_{i-1}$.

     (b) Else $\mathcal{S}$ randomly selects (g_i, h_{i-1}, g_{i-1}), computes $h_i = y_i \oplus g_{i-1}$, $m_{i,1} = k_{i,1}$ and $m_{i,2} = k_{i,2}$, then adds the tuple $(1, k'_i, x'_i, y'_i)$ as $x'_i = g_{i-1}$, $y'_i = g_i \oplus x'_i \oplus h_{i-1}$ and $k'_i = k_i$, and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(h_{i-1}, g_{i-1}) \xrightarrow{m_i} (h_i, g_i)\}$, then returns $x_i = h_{i-1} \oplus g_{i-1}$.

Before stating the indifferentiability result of HDBL-1, a simple lemma is proven from the above simulation.

Lemma 4 *In double block length hash functions defined by (6), it holds that* $\Pr[Pre] = 2^{-(3n+4)/2} \cdot l \cdot O(q)$ *and* $\Pr[Coll] = 2^{-n+2} \cdot l \cdot O(q)$*, where l is the maximum number of length in a hash query.*

Proof. In case of $\mathcal{O}_2 = (Rand, S, S^{-1})$, the total number of choices is $l \cdot q$, where l is the maximum number of length in a hash query. For every $2 \leq j \leq l \cdot q$, let $Coll_j$ be the collision event that a pair of inputs yield a same output after the j-th queries. Namely, for some $j' < j$, it follows that

$$(h_j, g_j) = (h_{j'}, g_{j'}) \ or \ h_j = g_j,$$

which is equivalent to

$$(y_j \oplus x_j, y'_j \oplus x'_j) = (y_{j'} \oplus x_{j'}, y'_{j'} \oplus x'_{j'}) \ or \ (y_j \oplus x_j = y'_j \oplus x'_j).$$

Since (h_i, g_i) are randomly selected by the simulator $\mathcal{S}$ from the range $\{0,1\}^n$ where $i \in \{1, 2, \cdots, l \cdot q\}$, the probability that the above event happens after the j-th queries is as follows.

$$\Pr(Coll_j) \leq \frac{(j-1)}{(2^n - (j-1)) \cdot (2^n - (j-1))} + \frac{1}{2^n}.$$

Let $Coll$ be the collision event that a pair of inputs yield a same output after the maximum q times queries. By implementing the same idea on the proof of *Example 1*, if $l \cdot q \leq 2^{n-1}$, it is easy to find that $\Pr[Coll] \leq \frac{l \cdot q}{2^{n-1}}$. Similarly, the probability of the preimage event Pre is $\Pr[Pre] \leq \frac{l \cdot q}{2^{(3n+4)/2}}$.

Consequently, the probability of the indifferentiable events Bad is

$$\Pr[Bad] = 2 \times Max(\Pr[Coll], \Pr[Pre]) = 2 \times \Pr[Coll] = 2^{-n+2} \cdot l \cdot O(q).$$

By implementing the advantage of indifferentiability in keyed hash function [4, 8], similar results can be easily deduced in the HMAC/NMAC construction. So the theorem follows. □

From the indifferentiability analysis of *Example 1* and HDBL-1, we believe that many rate-1 hash functions in FDBL-II, which obey Corollary 1, can be indifferentiable from a random oracle in the ideal cipher model. Furthermore, if both the ranks of L and R are three, the indifferentiability analysis might imply a formal proof in the ideal cipher model, since the simulator $\mathcal{S}$ might be able to simulate the responses of the encryption and decryption from the query (k_i, x_i) and (k_i, y_i), respectively.

B. A New Class of Fast DBL Hash Functions

Based on FDBL-I and FDBL-II, a new class of fast DBL hash functions (FDBL-III) can be extended as follows. Hash functions in FDBL-III can be constructed on a block cipher $E \in \mathtt{Bloc}(\kappa, n)$ with variants of key length where $\kappa = n$ or $\kappa = 2n$.

Definition 5 *Let $E \in \mathtt{Bloc}(\kappa, n)$ be a block cipher with variants of key length where $\kappa = n$ or $\kappa = 2n$. A new class of DBL hash functions with rate 1 (denoted by FDBL-III) can be constructed as follows.*

$$\begin{cases} h_i = E_A(B) \oplus C, \\ g_i = E_{W||X}(Y) \oplus Z. \end{cases} \tag{18}$$

Both (A, B, C) and (W, X, Y, Z) are linear combinations of the n-bit vectors $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$. Those linear combinations can be represented as

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \underbrace{\begin{pmatrix} L_l & L_r \end{pmatrix}}_{L} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_i^1 \\ m_i^2 \end{pmatrix}, \begin{pmatrix} W \\ X \\ Y \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} R_l & R_r \end{pmatrix}}_{R} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_i^1 \\ m_i^2 \end{pmatrix}. \tag{19}$$

By implementing the similar attacks on FDBL-I and FDBL-II, one can easily derive the following attacks on FDBL-III.

Lemma 5 *For any rate-1 hash function in FDBL-III with the form (18), if the rank of L(or R) is less than three, then there exist collision, preimage, and second preimage attacks on the hash function with complexities of about* $4 \times 2^{n/2}$, 3×2^n, *and* 3×2^n, *respectively.*

Lemma 6 *For any rate-1 hash function in FDBL-III with the form (18), if the rank of L_r^2(or L_r^3 or R_r^3 or R_r^4) is less than two, then there exist collision, preimage, and second preimage attacks on the hash function with complexities of about* $4 \times 2^{n/2}$, 3×2^n, *and* 3×2^n, *respectively.*

Lemma 7 *For any rate-1 hash function in FDBL-III with the form (18), there exist free-start collision and free-start (second) preimage attacks on the hash function with complexities of about* $2 \times 2^{n/2}$ *and* 2×2^n, *respectively.*

The above lemmas are extended from the similar attacks on FDBL-II, so we omitted the proofs here. In particular, based on Knudsen *et al.* result on FDBL-I [15], it is easy to obtain the following lemma.

Lemma 8 *For any rate-1 hash function in FDBL-III with the form (18), then there exist (second) preimage attacks on the hash function with the complexity of about* 4×2^n. *Furthermore, if the rank of L_l^2 and L_l^3 are two, then there exists a collision attack on the hash function with complexity of about* $3 \times 2^{3n/4}$, *else there exists a collision attack with complexity of about* $4 \times 2^{n/2}$.

Consequently, the following corollary gives the security bounds of rate-1 hash functions in FDBL-III. From the results, one can see all rate-1 hash functions in FDBL-III are failed to be optimally secure against collision, second preimage and preimage attacks. Same result can be obtained in the serial mode of FDBL-III.

Corollary 2 *For any rate-1 hash function H in FDBL-III with the form (18), there exist collision, preimage and second preimage attacks on the hash function with complexities of about $O(2^{3n/4})$, $O(2^n)$ and $O(2^n)$, respectively.*