On the Design of Secure Double Block Length Hash Functions with Rate 1*

Zheng Gong, Xuejia Lai and Kefei Chen
Department of Computer Science and Engineering
Shanghai Jiaotong University, China
neoyan@sjtu.edu.cn,{lai-xj,chen-kf}@cs.sjtu.edu.cn

Abstract

This paper reconsiders the security of the rate-1 double block length hash functions, which based on a block cipher with a block length of n-bit and a key length of $2n$-bit. Counter-examples and new attacks are presented on this general class of double block length hash functions with rate 1, which disclose there exist uncovered flaws in the former analysis given by Satoh *et al.* and Hirose. Preimage and second preimage attacks are designed to break Hirose's two examples which were left as an open problem. Some refined conditions are proposed for ensuring this general class of the rate-1 hash functions to be optimally secure against the collision attack. In particular, two typical examples, which designed under the proposed conditions, are proven to be indifferentiable from the 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 one block cipher used in the compression function has the key length is equal to the block length, while the other is doubled.

Key words. Cryptanalysis, Block cipher, Hash construction, Double block length, Indifferentiability.

*This paper is supported by NSFC under the grants 60573032, 90604036 and National 863 Projects 2006AA01Z422

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[6, 18], by iterating a compression function on the input message to realize a domain extension transform. The 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 scheme is that one can conveniently choose an extensively studied block cipher (e.g., DES, IDEA, AES, etc) to construct the compression function, and also the latest cryptanalysis results on such a block cipher can be used to avoid the potential weakness in the scheme. 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 block cipher that used in the hash function. The motivation of double block length is to combine two n-bit block ciphers to obtain 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 general construction to an arbitrary block cipher 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^n, respectively. A DBL hash function H is said to be *optimally secure*, if any adversary with non-negligible successful probability must spend the computation costs *no less than* publicly-accepted upper bounds of brute force attacks, namely, the complexities of (second) preimage and collision attacks on MD structure hash functions are about 2^{2n} and 2^n, with respect to the pigeonhole principle and the birthday paradox, respectively.

Although DBL hash function can extend the range for collision resistance, a consequent disadvantage 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, 10, 21, 26]. Unfortunately, some critical results showed the fact that those proposed schemes unlikely achieve optimally secure. In [14], Knudsen *et al.* 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 n-bit (FDBL-I for short). In particular, the attacks break the proposed schemes in [3, 10, 21]. Still, many advanced block ciphers (e.g., AES, RC5, Blowfish, etc) support variants of key length motivates renewed interest in finding good ways to construct a secure and fast DBL hash function. Many instructive examples were proposed recently, e.g., [9, 16, 19, 20]. But all these schemes are less than rate 1, which means they are not fast enough.

In [24], Satoh *et al.* presented some attacks on a general class of DBL hash functions with rate 1 where the key length as twice as the block length (FDBL-II for short), and broke the rate-1 scheme in [26]. In particular, Satoh *et al.* described a necessary condition for the rate-1 hash functions in FDBL-II to be optimally secure against preimage, second preimage and collision attacks. Recently in [8], Hirose gave a comment on Satoh *et al.*'s result[24] and showed that there exists a missed case in their analysis. Based on this comment, two necessary conditions for the rate-1 hash functions in FDBL-II to be optimally collision resistant are given by Hirose in [8]. Furthermore, two examples are left in [8] as an open problem to make it clear whether they are optimally secure.

Our Contributions. Consider the security of the rate-1 double block length hash functions where the key length is double to the block length, our contributions are three-folds. First, we present (second) preimage attacks on Hirose's two examples which are left as an open problem in [8]. Three counter-examples in FDBL-II are designed to disclose that Hirose's necessary conditions[8] for optimal collision resistance are still not precise. Secondly, based on new attacks and counter-examples, we formally analyze the security of the rate-1 hash functions in FDBL-II, and find there exists a subclass in FDBL-II can be optimally collision resistant. But all the rate-1 hash functions in FDBL-II are failed to be optimally (second) preimage resistant. The necessary conditions for the rate-1 hash functions in FDBL-II to be optimally collision resistant are refined by the analysis. In particular, the indifferentiability analysis are given on two typical examples under our refined conditions, which imply 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 (FDBL-III for short), where one block cipher has the key length equal to the block length, while the other is doubled in the compression function. The extended results show that all the rate-1 DBL hash functions in this general class (FDBL-III) are failed to be optimally secure. Prior to this paper, there is no rigorous analysis

on the half-baked cases proposed by Satoh *et al.*[24] and Hirose[8] 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[8] for optimal collision resistance are not precise. Attacks are presented on FDBL-II to obtain precise conditions towards optimal security. Section 4 concludes the paper. Additionally, the indifferentiability analysis of two typical examples are given in Appendix A. Appendix B describes an extended result on a new class of the rate-1 DBL hash functions.

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 looked as a concatenation of the $2n$-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 abbreviations 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 numbers. 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 $E : \{0,1\}^\kappa \times \{0,1\}^n \to \{0,1\}^n$. To avoid trivial iteration attacks, we assume that any block cipher $E \in \texttt{Bloc}$ has no *fixed-point* such that $E_k(x) = k$ or x or $E_k^{-1}(y) = y$ or k. A *block-cipher-based* hash function is a

hash function $H : \{0,1\}^* \to \{0,1\}^\ell$ by implementing $E \in \mathtt{Bloc}(\kappa, n)$ in the compression function of H. If $\ell = n$, then H is called a single block length (SBL) hash function, e.g., the PGV hash functions[22]. If $\ell = 2n$, then H is called a double block length (DBL) hash function, e.g., MDC-2[2], Parallel-DM[10], 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 $E \in \mathtt{Bloc}(\kappa, n)$ 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 ℓ bits long message block, the rate of the hash function H equals $\frac{\ell}{T \cdot n}$.*

Ideal Cipher Model. Ideal cipher model is a well-known model for the security analysis of block-cipher-based hash functions, which is dating back to Shannon [25] and has been frequently used for the security analysis of various hash functions[1, 9, 15, 22]. Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be a hash function and $E \in \mathtt{Bloc}(\kappa, n)$ be a block cipher used in the compression function of H. Adversary $\mathcal{A}$ is given access 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

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

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 asked. 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 does not help anything at the view of the adversary. 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 $H : \mathcal{K} \times \mathcal{M} \to \mathcal{Y}$ 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, find IV' and M such that $h = H(IV', M)$.*
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 [13]. Similar definitions can be found in [15]. 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 function. 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 function

at the presents of adversary, a widely-accepted approach will be recalled in below.

Indifferentiability Model. Objects are considered to be *computational equivalent* if no polynomial-time procedure can tell them apart. In [17], Maurer *et al.* 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 the 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 is not 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 is proven in [17] 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. 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, 7]. 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 access to H. Those two ways to build up a hash function should be indifferentiable.

Similarly, Hirose proposed the notion of *indistinguishability* on iterated hash functions in [9], 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), we stress that an *ideal* block-cipher-based hash function must be optimally secure against the 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, Knudsen *et al.* [14] 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 hash functions defined by (1), (A, B, C) are linear combinations of the n-bit vectors $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$, (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*. In [14], Knudsen *et al.* proved that all the rate-1 hash functions in FDBL-I are failed to be optimally secure against collision, preimage and second preimage attacks. The result is given by the following theorem in [14].

Theorem 1 *For the 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 128,196,256-bit while the block length is 128-bit. This property motivates interests in finding schemes to turn such a block cipher into a secure and fast DBL hash function, where the key length are longer than the block length. By considering the block cipher $E \in \mathtt{Bloc}(\kappa, n)$ where $\kappa = 2n$, Satoh *et al.*[24] proposed a new family of the rate-1 DBL hash functions (FDBL-II), which is defined by the general form as follows.

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

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_r are

deleted, respectively. Matrix L is said to be "exceptional" if $Rank(L) = 4$ and $Rank(L_r^3) = Rank(L_r^4) = 2$.

In [24], Satoh *et al.* stated attacks on FDBL-II when the compression function does not satisfy the *exceptional* property.

Theorem 2 *For the 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.*[24] presented attacks on a subclass of the rate-1 DBL hash functions in FDBL-II. We stress that the proposed scheme in [26] is a paradigm with respect to this subclass.

Theorem 3 *For a subclass of the 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) is linear combinations of $(h_{i-1}, g_{i-1}, m_{i,1}, m_{i,2})$ and $E \in \texttt{Bloc}(2n, n)$, there exist (second) preimage and collision attacks with complexities of about 2×2^n and $2 \times 2^{n/2}$, respectively.

In [8], Hirose gave a comment on the analysis by Satoh *et al.*[24]. The comment shows there exist the 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 proposed in [8] as follows.

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

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

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

Furthermore, the other example with the *exceptional* property was also proposed in [8].

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

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

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

Both HDBL-1 and HDBL-2 are the instances of FDBL-II. Let (a, b, c, d) be the values of (A, B, C, D) used in the computations of h_i. In [24], the adversary chooses random triple (a, b, c) such that $c = \alpha \cdot a \oplus \beta \cdot b$ where $\alpha, \beta \in \{0, 1\}$, then computes $d = E_{a||b}(c) \oplus h_i$. Hirose 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 both L and R are exceptional, a new condition for the rate-1 hash functions in FDBL-II to be optimally collision resistant is defined by Hirose in [8].

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.*

In [8], Hirose claimed that the above conditions are not sufficient but just necessary for the property of optimal collision resistance. It was left as an open problem if the two *probably secure* examples (HDBL-1 and HDBL-2) are really optimally secure.

3 Security Analysis of FDBL-II

In this section, the security of the rate-1 hash functions in FDBL-II is reconsidered. A synthetic analysis is presented which exploits the fact that the former results[8, 24] on the security of FDBL-II are still not accurate. 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, which 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 the rate-1 hash functions in FDBL-II to be optimally secure are refined.

3.1 Attacks on Hirose's Two Examples

In [24], Satoh *et al.* 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[8] gave 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 defined in Definition 4. Here we present two concrete attacks on Hirose's two examples which prove they are 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, 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 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. Forward 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 the pigeonhole principle, $\mathcal{A}$ 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 forward step to obtain q_1 values of $(m_{i,1}, m_{i,2}, h_{i-1}, g_{i-1})$.

3. Backward 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

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

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 constructed 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 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. Forward 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 the pigeonhole principle, $\mathcal{A}$ obtains $2^{n/2}$ values of (m_i, h_{i-1}, g_{i-1}) yield the fixed value (h_i, g_i).
3. Backward step: $\mathcal{A}$ 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 $2n$-bit long, 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. $\square$

Since HDBL-1 and HDBL-2 satisfy Type 2 and Type 1 in Definition 4, respectively, which are the two necessary conditions defined by Hirose in [8]. The above attacks disclose the point that maybe there exist uncovered flaws in the former security results on the rate-1 hash functions in FDBL-II which are given by Satoh *et al.*[24] and Hirose[8]. Heuristically, we present three counter-examples, which do not satisfy Hirose's necessary conditions but still no efficient collision attack can be found, to support this considerable point.

First we give two examples of the rate-1 hash functions in FDBL-II, which do not satisfy Type 2 condition 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:

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

We give the following theorem which establishes the indifferentiability of Example 1. The omitted proof can be found in Appendix A.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, the NMAC/HMAC, and the chop 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 $l \cdot q \leq 2^{n-1}$.*

The following theorem establishes the indifferentiability of HDBL-1. The omitted proof can be found in Appendix A.2. Based on the indifferentiability analysis, one can find that both Example 1 and HDBL-1 are optimally collision resistant in the ideal cipher model. By using the similar analysis, the proofs can be extended to other examples.

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, the NMAC/HMAC, and the chop 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* $l \cdot q \leq 2^{n-1}$*.*

From the concrete attacks and the counter-examples, it is easy to see that Hirose's two necessary conditions (at least) are not precise for the 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 discover the certain conditions which should be imposed on FDBL-II for the property of the optimal security.

3.2 The Exact Security of FDBL-II

In [8], a comment is shown that the attacks given by Satoh *et al.*[24] do not work on 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. Due to the three counter-examples which are described in Section 3.1, Hirose's conditions[8] 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.*[24] do not directly imply the optimal security. The exact security of the rate-1 hash functions in FDBL-II is reconsidered through the following attacks. First generic attacks are presented.

Theorem 8 *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 the pigeonhole principle, 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 the optimal collision and the (second) preimage resistances are described as follows.

Lemma 1 *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 8, 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 2 *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 $E_{A||B}^{-1}(h_i \oplus D)$) 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 $c = E_{a||b}^{-1}(d \oplus h_i)$ (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 8, 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 either the rank of R_r^3 or R_r^4 is less than two. □

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

Theorem 9 *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 separate hash functions. Due to Joux's multicollision attack[11], 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 10 *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 1; If the rank of L or R is greater or equal three, adversary $\mathcal{A}$ start the attacks by choosing an arbitrary messages $M = m_1||m_2||\cdots||m_{i-2}$, and by computing the values of (h_{i-2}, g_{i-2}) iteratively from the given initial value $IV = h_0||g_0$.

1. Forward step: $\mathcal{A}$ randomly chooses 2^n values of (a, b, c). If the rank of L is three (assume d is a linear combination of a, b, c), then $\mathcal{A}$ obtains a tuple (a, b, c) yields the given value $h_i = E_{a||b}(c) \oplus c$; If the rank of L is four, then $\mathcal{A}$ computes 2^n values of d where $d = E_{a||b}(c) \oplus h_i$. Due to the pigeonhole principle, $\mathcal{A}$ can find at least one tuple (h_{i-1}, g_{i-1}, m_i) from (a, b, c, d) that satisfies the equation.

2. $\mathcal{A}$ repeats the above step $2^{n/2}$ times. Due to the pigeonhole principle, $\mathcal{A}$ obtains $2^{n/2}$ values of (m_i, h_{i-1}, g_{i-1}) yield the fixed value (h_i, g_i).

3. Backward step: $\mathcal{A}$ 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 due to 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 10. The complexity of the generic second preimage attack, which was proposed by Kelsey and Schneier in [12], can be smaller than ours asymptotically. But their attack needs an unpractical long message, which makes it become less attractive. E.g., for $2n$-bit hash functions, a 2^x-bit long message with about $x \times 2^{n+1} + 2^{2n-x+1}$ work. It is easy to see that a second preimage attack with the complexity of about $O(2^{3n/2})$ requires a $2^{n/2}$-bit message.

Based on the above results, necessary conditions for the 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 ranks 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 $O(2^{3n/2})$.

4 A New Class of Fast DBL Hash Functions

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

Definition 5 *Let $E \in \texttt{Bloc}(\kappa, n)$ 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.*

$$\begin{cases} h_i = E_A(B) \oplus C, \\ g_i = E_{W||X}(Y) \oplus Z. \end{cases} \tag{16}$$

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

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \underbrace{\begin{pmatrix} L_l & L_r \end{pmatrix}}_{L} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_i^1 \\ m_i^2 \end{pmatrix}, \begin{pmatrix} W \\ X \\ Y \\ Z \end{pmatrix} = \underbrace{\begin{pmatrix} R_l & R_r \end{pmatrix}}_{R} \cdot \begin{pmatrix} h_{i-1} \\ g_{i-1} \\ m_i^1 \\ m_i^2 \end{pmatrix}, \tag{17}$$

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

Lemma 3 *For any rate-1 hash function in FDBL-III with the form (16), 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 4 *For any rate-1 hash function in FDBL-III with the form (16), 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 5 *For any rate-1 hash function in FDBL-III with the form (16), 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 [14], it is easy to obtain the following lemma.

Lemma 6 *For any rate-1 hash function in FDBL-III with the form (16), 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 upper security bounds of the rate-1 hash functions in FDBL-III. From the bounds, one can see all of the 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 (16), there exist collision, preimage and second preimage attacks on the hash function with complexities of about $O(2^{3n/4})$, $O(2^n)$ and $O(2^n)$.*

Theorem 11 *The rate-1 hash function in FDBL-III with the form (16) is (t_D, t_S, q, ϵ)-indifferentiable from a random oracle in the ideal cipher model with the prefix-free padding, the NMAC/HMAC, and the chop construction, for any distinguisher $\mathcal{D}$ in polynomial time bound t_D, with $t_S = 2l \cdot O(q)$ and the advantage $\epsilon = 2^{-3n+4/4} \cdot l \cdot O(q)$, where l is the maximum length of a query made by $\mathcal{D}$ and $l \cdot q \leq 2^{n-1}$.*

5 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 of the 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 implies 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 to design secure and fast DBL hash functions. Due to the key length will definitely impact the efficiency, the definition of the hash rate is not appropriate any more,V for the new designs of double block (or multi-block) length hash functions. E.g, a rate-1 DBL hash function in FDBL-I cannot directly compare to such one in FDBL-II. To solve this problem, we suggest a new preferable definition *hash capacity* for the measurement.

Definition 6 *Let $H : \{0,1\}^* \to \{0,1\}^\ell$ be a hash function and $E \in \mathtt{Bloc}(\kappa, n)$ 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 ℓ bits long message block, the* capacity *of the hash function H equals*

$$\mathrm{cap} = \frac{\ell/n}{T \cdot \kappa/n} = \frac{\ell}{T \cdot \kappa}.$$

The above definition is admissible since AES encrypts 20% slower for 192-bit keys and 40% slower for 256-bit keys. 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 schemes as hash function) is meaningful. Future work is to summarize a generic proof on block-cipher-based hash functions with variants of block and key length through the definition of *hash capacity* .

References

[1] J. Black, P. Rogaway and T. Shrimpton. Black-Box Analysis of the Black-Cipher-Based Hash-Function Constructions from PGV. *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, editors, *Advances in Cryptology-Proc. 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. X. Lai and K. Chen(Eds): *ASIACRYPT 2006*, LNCS 4284, pp. 283-298, 2006.

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

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

[7] Z. Gong, X. Lai, and K. Chen. A Synthetic Indifferentiability Analysis of Some Block-Cipher-Based Hash Functions. *Designs, Codes and Cryptography*. Accepted, Online Availiable. Springer.

[8] S. Hirose. A Security Analysis of Double-Block-Length Hash Functions with the Rate 1. *IEICE Trans. Fundamentals*, Vol. E89-A, NO.10, pp. 2575-2582, Oct 2006.

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

[10] 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.

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

[12] 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.

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

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

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

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

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

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

[19] M. Nandi. Design of Iteration on Hash Functions and Its Cryptanalysis. *PhD thesis*, Indian Statistical Institute, 2005.

[20] M. Nandi. Towards optimal double-length hash functions. *INDOCRYPT 2005*, LNCS 3797, pages 77C89, 2005.

[21] 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.

[22] 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.

[23] 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.

[24] T. Satoh, M. Haga, and K. Kurosawa. Towards Secure adn Fast Hash Functions. *IEICE Trans. Fundamentals*, Vol. E82-A, NO.1, pp. 55-62, Jan, 1999.

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

[26] 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. Indifferentiability Analysis of The Examples in FDBL-II

Based on the synthetic analysis of block-cipher-based hash functions[4, 7], here we present the indifferentiability analysis on two typical examples in FDBL-II. Let distinguisher $\mathcal{D}$ can access to 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. 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]. For brevity, we note that all of the examples are implicitly implemented with one of those padding rules.

A.1 Proof of Theorem 6

First we give a simulation to prove that Example 1 (defined in Section 3.1) is indifferentiable from a random oracle.

- *Rand*-**Query.** For the i-th query $M_i \in \mathcal{M}$ on $Rand$, if M_i is a repetition query, the simulator $\mathcal{S}$ 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 $\mathcal{S}$ 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 7 *In double block length hash functions defined by (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), where $i \in \{1, 2, \cdots, l \cdot q\}$, is randomly and uniformly selected by the simulator $\mathcal{S}$ in the range $\{0,1\}^n$, 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. Thus, if $l \cdot q \leq 2^{n-1}$,

$$\begin{aligned}
\Pr[Coll] = \Pr[Coll_1 \vee Coll_2 \vee \cdots \vee Coll_{l \cdot q}] &\leq \sum_{j=2}^{l \cdot q} \Pr[Coll_j] \\
&\leq \sum_{j=2}^{l \cdot q} \left(\frac{j-1}{(2^n - (j-1)) \cdot (2^n - (j-1))} + \frac{1}{2^n}\right) \\
&\leq \frac{\sum_{j=2}^{l \cdot q} (j-1)}{(2^n - 2^{n-1}) \cdot (2^n - 2^{n-1})} + \frac{l \cdot q}{2^n} \\
&\leq \frac{(1 + l \cdot q) \cdot (l \cdot q)}{2^{2n-1}} + \frac{l \cdot q}{2^n} \leq \frac{2^{n-1}(l \cdot q) + (l \cdot q) + 2^{n-1}(l \cdot q)}{2^{2n-1}} \approx \frac{l \cdot q}{2^{n-1}}
\end{aligned} \tag{18}$$

From the preimage attack on FDBL-II in Theorem 10, it is easy to see the probability of the preimage events Pre is

$$\begin{aligned}
\Pr[Pre] = \Pr[Pre_1 \vee Pre_2 \vee \cdots \vee Pre_{l \cdot q}] &\leq \sum_{j=1}^{l \cdot q} \Pr[Pre_j] \\
&\leq \sum_{j=1}^{l \cdot q} \left(\frac{1}{4 \times 2^{3n/2}}\right) \leq \frac{l \cdot q}{4 \times 2^{3n/2}}
\end{aligned} \tag{19}$$

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[7], similar results can be easily deduced in keyed mode. □

Based on the above analysis, Theorem 6 follows on Example 1. We stress that the analysis implies a formal proof in the ideal cipher model as well. By using the above method, one can extend the similar results on Example 2 and Example 3, which are also defined in Section 3.1.

A.2 Proof of Theorem 7

Now 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.

- *Rand*-**Query.** For the i-th *Rand*-query $M_i \in \mathcal{M}$, if M_i is a repetition query, the simulator $\mathcal{S}$ 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 $\mathcal{S}$ 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 8 *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

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

which is equivalent to

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

Since (h_i, g_i), where $i \in \{1, 2, \cdots, l \cdot q\}$ is randomly and uniformly selected by the simulator $\mathcal{S}$ in the range $\{0,1\}^n$, 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[7], similar results can be easily deduced in keyed mode. □

From the above analysis, Theorem 7 follows on HDBL-1. We believe many of the 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 rank of L and R are three, the indifferentiability analysis implies a formal proof in the ideal cipher model, since the simulator $\mathcal{S}$ can simulate the response 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