Indifferentiable Security Analysis of Popular Hash Functions with Prefix-free Padding

Donghoon Chang[1], Sangjin Lee[1], Mridul Nandi[2], and Moti Yung[3]

[1] Center for Information Security Technologies(CIST), Korea University, Seoul, Korea
{dhchang,sangjin}@cist.korea.ac.kr

[2] David R. Cheriton School of Computer Science, University of Waterloo, Canada
m2nandi@cs.uwaterloo.ca

[3] RSA Laboratories and Department of Computer Science, Columbia University, New York, USA
moti@cs.columbia.edu

Abstract. Understanding what construction strategy has a chance to be a good hash function is extremely important nowadays. In TCC'04, Maurer *et al.* [13] introduced the notion of indifferentiability as a generalization of the concept of the indistinguishability of two systems. In Crypto'2005, Coron *et al.* [5] suggested to employ indifferentiability in generic analysis of hash functions and started by suggesting four constructions which enable eliminating all possible generic attacks against iterative hash functions. In this paper we continue this initial suggestion and we give a formal proof of indifferentiability and indifferentiable attack for prefix-free MD hash functions (for single block length (SBL) hash and also some double block length (DBL) constructions) in the random oracle model and in the ideal cipher model. In particular, we observe that there are sixteen PGV hash functions (with prefix-free padding) which are indifferentiable from random oracle model in the ideal cipher model.

1 Introduction

The notion of indifferentiability was first introduced by Maurer *et al.* [13] and is a stronger notion than indistinguishability. For example, assume a cryptosystem $\mathcal{P}(\mathcal{G})$ based on a random oracle $\mathcal{G}$ is secure. Now, to prove the security of $\mathcal{P}(H^{\mathcal{F}})$ based on Merkle-Damgard (MD) hash function H where the underlying compression function is a random oracle, we need to prove something different than indistinguishability. In fact, we need to prove that $H^{\mathcal{F}}$ is indifferentiable (as was introduced in [13]) from a random oracle. Informally, $H^{\mathcal{F}}$ is indifferentiable from random oracle if there is no efficient attacker (or distinguisher) which can distinguish $\mathcal{F}$ and the hash function based on it from a random oracle R and an efficient simulator of $\mathcal{F}$. Here R is a random oracle with (finite) domain and range same as that of H. In case of Indistinguishability, the distinguisher only needs to tell apart H from $\mathcal{G}$ without any help of oracle $\mathcal{F}$. Thus, the notion of indifferentiability is important when we consider attacks on a cryptosystem based on some ideal primitive where the attacker has some access on the computation of the primitive. In the case of hash function $H^{\mathcal{F}}$, the attacker can also compute $\mathcal{F}$ as it is a random oracle which can be computed publicly. So this new notion is important for stronger attackers. If the attacker does not have that access (to the random oracle) then merely indistinguishability will suffice to preserve the security of the cryptosystem.

Recently, Coron *et al.* [5] suggested to employ the notion for analysis of hash functions and they proved that the classical MD iteration is not indifferentiable with random oracle when the underlying compression function is random oracle. They have also stated indifferentiability for prefix-free MD hash functions or some other definition of hash functions like HMAC, NMAC, chop-MD hash function. They also have stated indifferentiability for Davis-Meyer construction (which is one of the classical PGV construction [17]) in the ideal cipher model.

Our Results: In this paper we extend the use of indifferentiability in analyzing hash functions, and we present a proof methodology for determining indifferentiability. We discuss indifferentiability of several known hash constructions with the random oracle model including the prefix free MD hash function. We consider all collision secure PGV hash functions in the ideal cipher model [2] (there are twenty such hash functions). It is easy to check that under ideal cipher model the underlying compression function is not indifferentiable with random oracle. So we can not use the indifferentiability result directly for prefix-free MD hash function (where we need the underlying compression function as a random oracle). But we will show that out of twenty, sixteen hash functions with prefix free padding are indifferentiable from random oracle. We also prove the indifferentiability of some known Double length hash functions in the random oracle model for the underlying single length compression function. Finally, we will also show several differentiability attacks on block-cipher based on double length hash function namely, PBGV, LOKI-DBH, MDC2 etc.

Organization: The organization of this paper is as follows. In section 2, we define notations and describe the security notion of indifferentiability with some mathematical background and notations which will help to prove the security later. In section 3, we provide formal proofs of prefix-free single length MD hash functions, PGV hash functions, and double length hash function. Then, in section 4, we show the differentiability of some SBL and DBL hash functions. Finally we conclude.

2 Preliminaries and Related Work

In this section, we briefly describe random oracle and ideal cipher model and we review how the adversary works in these model. Then some designs of hash functions are stated.

2.1 Ideal Model and Iterated structure

Random Oracle Model : f is said to be a *random oracle* from X to Y if for each $x \in X$ the value of $f(x)$ is chosen randomly from Y. More precisely, $\Pr[f(x) = y \mid f(x_1) = y_1, f(x_2) = y_2, \ldots f(x_q) = y_q] = \frac{1}{M}$, where $x \notin \{x_1, \ldots, x_q\}$, $y, y_1, \cdots, y_q \in Y$ and $|Y| = M$. There is an equivalent way to look a random function: Consider $\mathsf{Map}(X \to Y)$, the set of all mappings from X to Y. f is said to be a random oracle if it is chosen uniformly from the set $\mathsf{Map}(X \to Y)$. The adversary $\mathcal{A}$ can only query f adaptively, say by inputting $x_1, \cdots x_q$, where q is the total number of queries. Let $y_1, \cdots y_q$ be the responses of these queries, i.e., $f(x_1) = y_1, \cdots, f(x_q) = y_q$. Since an adversary makes queries adaptively, the i^{th} query x_i only depends on previous query-responses (in short, q-r) $(x_1, y_1), \cdots, (x_{i-1}, y_{i-1})$ and on the random coins selected by the adversary.

Ideal Cipher Model : Ideal cipher model is the one dating back to Shannon [19] and used, e.g., in [7, 10, 20]. Let $\mathsf{Bloc}(\mathcal{K}, X) = \{E : \mathcal{K} \times X \to X; E(k, \cdot)$ is a permutation for each $k \in \mathcal{K}\}$. As above, a function E is chosen uniformly from the set $\mathsf{Bloc}(\mathcal{K}, X)$. As $E(k, \cdot)$ (we also use the notation $E_k(\cdot)$) is a permutation, an adversary $\mathcal{A}$ can have access to two oracles E and E^{-1}. Thus, the q-r's look like $(\sigma_1, k_1, x_1, y_1), \cdots, (\sigma_q, k_q, x_q, y_q)$, where $\sigma_i = \pm 1$ and $E_{k_i}(x_i) = y_i$, $i \leq i \leq q$. If $\sigma_i = 1$ then adversary makes E query with input (k_i, x_i) and response is y_i and if $\sigma_i = -1$ then adversary makes E^{-1} query with input (k_i, y_i) and response is x_i. Now one can check that, for each k, $E_k(\cdot)$ behaves like a random permutation (i.e., $\Pr[E_k(x) = y \mid E_k(x_1) = y_1, \ldots, E_k(x_q) = y_q] = \frac{1}{M-q}$, where $x \notin \{x_1, \ldots, x_q\}$, $y \notin \{y_1, \cdots, y_q\} \subseteq Y$ and $|Y| = M$) and for different choices of keys $k_1, \ldots, k_l$, $E_{k_1}(\cdot), \ldots, E_{k_l}(\cdot)$ are independently distributed. See [2] for more details and discussions about black-box models.

Iterated Hash Function : Now given a function $F : Y \times B \to Y$, one can define an iterated function $F^* : Y \times B^* \to Y$ as follows :

$$F^*(x, m_1, m_2, \cdots, m_l) = F(\cdots F(x, m_1), \cdots, m_\ell), m_i \in B, x \in Y$$

where $B^* = \cup_{i \geq 0} B^i$. Let $\mathcal{M}$ be a message space (finite) and $g : \mathcal{M} :\to B^*$ be any function called a padding rule. Then the MD-Hash function based on a compression function F, a fixed initial value $\mathsf{IV} \in Y$ and a padding rule $g(\cdot)$ is $\mathsf{MD}_g^F(M) = F^*(\mathsf{IV}, g(M))$. A padding rule is called a prefix-free if $M_1 \neq M_2 \Rightarrow g(M_1)$ is not a prefix of $g(M_2)$. Coron *et al.* [5] considered prefix-free MD iteration and suggested indifferentiability from random oracle model.

Given a compression function $F : Y \times B \to Y$, one can also define a wide compression function $W : Y' \times B' \to Y'$, where Y' is a bigger set than Y. For example, in case of a double length compression function $Y' = Y \times Y$. An example of a general class of double length compression functions due to Nandi [15] is as follows : $W(x_1, x_2, m) = F(x_1 \parallel x_2, m) \parallel F(p(x_1 \parallel x_2), m)$, where $x_1, x_2 \in Y, m \in B', F : Y \times (Y \times B') \to Y$ and p is a permutation on $Y \times Y$ so that it does not have any fixed point (y is called fixed point of p if $p(y) = y$).

2.2 Known Results on Indifferentiability

In this section we give a brief introduction of indifferentiability and state some known results on it.

Definition 1. [5] A Turing machine C with oracle access to an ideal primitive $\mathcal{F}$ is said to be $(t_D, t_S, q, \varepsilon)$ indifferentiable from an ideal primitive $\mathcal{G}$ if there exists a simulator S such that for any distinguisher D it holds that :

$$|\Pr[D^{C,\mathcal{F}} = 1] - \Pr[D^{\mathcal{G},S} = 1] < \varepsilon$$

The simulator has oracle access to $\mathcal{G}$ and runs in time at most t_S. The distinguisher runs in time at most t_D and makes at most q queries. Similarly, $C^{\mathcal{F}}$ is said to be (computationally) indifferentiable from $\mathcal{G}$ if ε is a negligible function of the security parameter k (for polynomially bounded t_D and t_S).

In this paper, we will mainly consider $C = H^{\mathcal{F}}$, where H is MD (or prefix-free MD) hash function based on the random oracle model (or ideal cipher model) $\mathcal{F}$ and $\mathcal{G}$ is a random oracle with same domain and range as the hash function. In case of ideal cipher model the distinguisher can access both $\mathcal{F}$ and $\mathcal{F}^{-1}$ oracles and the simulator has to simulate both.

The following Theorem [13] due to Maurer *et al.* is related to this paper. We explain the theorem for random oracle model of hash functions. Suppose a hash function (in some design of iteration) H based on a random oracle (or an ideal cipher) $\mathcal{F}$ is indifferentiable from a random oracle $\mathcal{G}$. Then a cryptosystem $\mathcal{P}$ based on the random oracle $\mathcal{G}$ is at least as secure as the cryptosystem $\mathcal{P}$ based on the hash function H in the random oracle model (or an ideal cipher model) $\mathcal{F}$. Here, $\mathcal{F}$ is the underlying compression function of H (or block-cipher in case of block cipher based hash function). The original theorem as stated below is a more general statement.

Theorem 1. *[13] Let $\mathcal{P}$ be a cryptosystem with oracle access to an ideal primitive $\mathcal{G}$. Let H be an algorithm such that $H^{\mathcal{F}}$ is indifferentiable from $\mathcal{G}$. Then cryptosystem $\mathcal{P}$ is at least as secure in the $\mathcal{F}$ model with algorithm H as in the $\mathcal{G}$ model.*

Coron *et al.* stated the indifferentiability of prefix free MD construction in random oracle (or in ideal cipher model in the case of block-cipher based construction). In [5] the following theorems are stated.

Theorem 2. *[5] The prefix-free MD construction is $(t_D, t_S, q, \varepsilon)$-indifferentiable from a random oracle, in the random oracle model for the compression function, for any t_D, with $t_S = \ell \cdot O(q^2)$ and $\varepsilon = 2^{-n} \cdot \ell^2 \cdot O(q^2)$, where ℓ is the maximum length of a query made by the distinguisher D.*

Theorem 3. *The Davis-Meyer Hash function (based on the compression function $f(x, m) = E_m(x) \oplus x$ and a prefix free padding g) MD_g^f is $(t_D, t_S, q, \varepsilon)$-indifferentiable from a random oracle, in the ideal cipher model, for any t_D, with $t_S = \ell \cdot O(q^2)$ and $\varepsilon = 2^{-n} \cdot \ell^2 \cdot O(q^2)$, where ℓ is the maximum length of a query made by the distinguisher D.*

2.3 Adversary in the random oracle model

A binary relation $\mathcal{R}$ on $(X \times B, X)$ is a subset of $X \times B \times X$. A relation is called *functional relation* (or *partial functional relation*) if for each $(x, m) \in X \times B$ there exists at most one $y \in X$ such that $(x, m, y) \in \mathcal{R}$. Thus, a partial functional relation is uniquely characterized by a partial function $f : X \times B \to X$ (a partial function may have some points on domain where the functional value is not defined). Now given a relation $\mathcal{R}$ on $(X \times B) \times X$, one can define a *functional closure* relation $\mathcal{R}^*$ on $(X \times B^*) \times X$ which is a minimal relation containing $\mathcal{R}$ such that following are true:

1. $(x_1, M_1, x_2),\ (x_2, M_2, x_3) \in \mathcal{R}^* \Longrightarrow (x_1, M_1 \parallel M_2, x_3) \in \mathcal{R}^*$.
2. $(x_1, M_1 \parallel M_2, x_3),\ (x_1, M_1, x_2) \in \mathcal{R}^* \Longrightarrow (x_2, M_2, x_3) \in \mathcal{R}^*$.

Thus, if $\mathcal{R}$ corresponds to a partial function $f : X \times B \to X$, then $\mathcal{R}^*$ corresponds to the partial function f^* which is obtained from the partial function f iteratively. Sometimes, we use a more appealing notation $x_1 \to_{M_1} x_2 \in \mathcal{R}$ (or $x_1 \to_{M_1} x_2$ when the relation is clear from the context) to denote that $(x_1, M_1, x_2) \in \mathcal{R}^*$. Thus, in terms of this notation, $\mathcal{R}^*$ is the minimal relation containing $\mathcal{R}$ with the following conditions:

1. If $x_1 \to_{M_1} x_2 \to_{M_2} x_3$, then $x_1 \to_{M_1 \parallel M_2} x_3$ (transitive property).
2. If $x_1 \to_{M_1} x_2$ and $x_1 \to_{M_1 \parallel M_2} x_3$, then $x_2 \to_{M_2} x_3$ (substitute property).

Let D be a distinguisher (or an adversary) in the indifferentiable attack. He has an access to two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. In this scenario, either $(\mathcal{O}_1, \mathcal{O}_2) = (H, f)$ or $(\mathcal{O}_1, \mathcal{O}_2) = (\mathsf{Rand}, S)$, where $H = \mathsf{MD}_g^f$ (prefix free MD hash function with fixed initial value IV), S is any simulator, f and Rand are random oracles from $X \times B$ to X and from $\mathcal{M}$ to X respectively. Distinguisher is making successive queries of $\mathcal{O}_1$ or $\mathcal{O}_2$. Suppose the i^{th} query is an $\mathcal{O}_1$ query with the message $M \in \mathcal{M}$ and the response of the query is h (say), then we write $r_i = \mathsf{IV} \to_{g(M)} h$. Otherwise, $r_i = h_1 \to_m h_2$ for $\mathcal{O}_2$ query (h_1, m) with response h_2. Let $\mathcal{R}_i = \{r_1, \cdots, r_i\}$ be the relation characterizing the query-response after the i^{th} query and $\mathcal{R}_i^*$ be the functional closure of $\mathcal{R}_i$ characterizing the view of the distinguisher after i^{th} query. Thus, $\mathcal{Q} = (\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_q)$ be the complete query-response tuple and $\mathcal{V} = (\mathcal{R}_1^*, \mathcal{R}_2^*, \cdots, \mathcal{R}_q^*)$ be the complete view of the distinguisher D, where q is the total number of queries. Now we define some terminology which will be useful in this context.

1. Define *support* of a relation $\mathcal{R}_i$ by a subset of X, $\mathsf{Supp}(\mathcal{R}_i) = \{h : h \to_m h_1 \in \mathcal{R}_i\} \cup \{h : h_1 \to_m h \in \mathcal{R}_i\} \cup \{\mathsf{IV}\}$. Note that, $\mathsf{Supp}(\mathcal{R}_i) = \mathsf{Supp}(\mathcal{R}_i^*)$.
2. We say, r_i is a *trivial query* if $r_i \in \mathcal{R}_{i-1}^*$. Since g is a prefix-free padding, r_i can be trivial query only if any one of the following holds :
   (a) $r_i = \mathsf{IV} \to_{g(M)} h_\ell$, where $\mathsf{IV} = h_0 \to_{m_1} h_1 \to_{m_2} \ldots h_{\ell-1} \to_{m_\ell} h_\ell \in \mathcal{R}_{i-1}^*$ and $g(M) = m_1 \parallel \ldots \parallel m_\ell$.
   (b) $r_i = h_{\ell-1} \to_{m_\ell} h_\ell$, where $\mathsf{IV} = h_0 \to_{m_1} h_1 \to_{m_2} \ldots h_{\ell-1}$, $\mathsf{IV} \to_{g(M)} h_\ell \in \mathcal{R}_{i-1}^*$ and $g(M) = m_1 \parallel \ldots \parallel m_\ell$.

(c) r_i is a repetition query i.e. $r_i = r_j$ for some $j < i$. For simplicity, we can assume that there is no repetition query as distinguisher's point of view it doest not help anything.

3. We say $\mathcal{V}$ is not *collision free* (or in short $\neg$ CF) if for some i, $r_i = h \rightarrow_M h'$ is non trivial and $h' \in \mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{h\}$.

3 Security Analysis

In this section, we explain how to obtain a formal proof of indifferentiability of prefix-free single length or double length or block-cipher based MD hash functions. Let E be an event which is only a function of the view of the distinguisher. In this case we consider complement of the collision-free event ($\neg$ CF). Thus, there are events E_1 and E_2 for E when D interact with (H, f) and (Rand, S), respectively. If this event is defined carefully so that

1. (H, f) and (Rand, S) are identically distributed conditioned on the past view of the distinguisher and E does not occur, and
2. if $\Pr[E_1], \Pr[E_2] \leq \mathsf{max}$, where max is some negligible function.

Because of item 1, $\Pr[D^{H,f} \rightarrow 1 \mid \neg\ \mathsf{E}_1] = \Pr[D^{R,S} \rightarrow 1 | \neg \mathsf{E}_2]$. Then, one can show the indifferentiability of H with the random oracle model. More precisely,

$$\begin{aligned}
\mathsf{Adv}(D) &= \mid \Pr[D^{H,f} \rightarrow 1] - \Pr[D^{R,S} \rightarrow 1] \mid \\
&= \mid \Pr[D^{H,f} \rightarrow 1 \mid \mathsf{E}_1] \times \Pr[\mathsf{E}_1] + \Pr[D^{H,f} \rightarrow 1 \mid \neg\mathsf{E}_1] \times \Pr[\neg\mathsf{E}_1] \\
&\quad - \Pr[D^{R,S} \rightarrow 1 \mid \mathsf{E}_2] \times \Pr[\mathsf{E}_2] - \Pr[D^{R,S} \rightarrow 1 \mid \neg\mathsf{E}_2] \times \Pr[\neg\mathsf{E}_2] \mid \\
&\leq \mathsf{max} \times \mid \Pr[D^{H,f} \rightarrow 1 \mid \mathsf{E}_1] - \Pr[D^{R,S} \rightarrow 1 \mid \mathsf{E}_2] \mid \\
&\quad + \Pr[D^{H,f} \rightarrow 1 \mid \neg\mathsf{E}_1] \times \mid \Pr[\neg\ \mathsf{E}_1] - \Pr[\neg\ \mathsf{E}_2] \mid \qquad \cdots\cdots (1) \\
&= \mathsf{max} \times \mid \Pr[D^{H,f} \rightarrow 1 \mid \mathsf{E}_1] - \Pr[D^{R,S} \rightarrow 1 \mid \mathsf{E}_2] \mid \\
&\quad + \Pr[D^{H,f} \rightarrow 1 \mid \neg\mathsf{E}_1] \times \mid \Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2] \mid \qquad \cdots\cdots (2) \\
&\leq \mathsf{max} \times \mid \Pr[D^{H,f} \rightarrow 1 \mid \mathsf{E}_1] - \Pr[D^{R,S} \rightarrow 1 \mid \mathsf{E}_2] \mid \\
&\quad + \mathsf{max} \times \Pr[D^{H,f} \rightarrow 1 \mid \neg\mathsf{E}_1] \\
&\leq 2 \times \mathsf{max}
\end{aligned}$$

In (1), $\Pr[D^{H,f} \rightarrow 1 \mid \neg\ \mathsf{E}_1] = \Pr[D^{R,S} \rightarrow 1 | \neg\mathsf{E}_2]$ and in (2), $\Pr[\neg\ \mathsf{E}_2] - \Pr[\neg\ \mathsf{E}_1] = \Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]$. Thus we have,

$$\mathsf{Adv}(D) \leq 2 \times \mathsf{max}\{\mathsf{Pr}[E_1], \mathsf{Pr}[E_2]\} \qquad \cdots\cdots (3)$$

Similarly, if H is based on the block cipher E, we have three set of oracles (H^E, E, E^{-1}) or $(\mathsf{Rand}, S, S^{-1})$. Then we can proceed as like above.

3.1 Indifferentiability of prefix free single length MD hash functions

Now we define a simulator S which simulates f so that no distinguisher can distinguish (R, S) with (H, f), where R and f are assumed to be random oracles and H is the prefix-free hash function based on f.

Simulator : The simulator keeps the relations $(\mathcal{R}_1, \ldots, \mathcal{R}_{i-1})$. Initially, $\mathcal{R}_0 = \emptyset$. On the i^{th} query (h_i, x_i), the response of S is as follow

1. If $\exists\ \mathsf{IV} \to_N h_i \in \mathcal{R}_{i-1}, g(M) = N \parallel x_i$, then run $\mathsf{Rand}(M)$ and obtain the response h^*. $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_i \to_{x_i} h^*\}$ and return h^*. For more than one choices of M, return a random string h^* (this will never happen if $(\mathcal{R}_1, \ldots, \mathcal{R}_q)$ is collision-free).
2. Else return a random string h^* and $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_i \to_{x_i} h^*\}$.

If distinguisher is making at most q queries then one can design the above simulator so that it runs in time $O(\ell q)$. In the worst case, simulator has to back track to initial value to check whether condition (1) is satisfied or not and this is needed at most $O(\ell q)$ time. Note that in [5] time complexity for simulator is $O(\ell q^2)$.

Distribution of oracles : Here, we study the conditional distribution of all oracles given the past view of the distinguisher and the collision-freeness of the view.

Let $\mathcal{Q}_i$ be the set of all query-response after i queries. Let CF_1 and CF_2 denote that the complete view $\mathcal{V}$ is collision free (CF) in case of (H, f) and (Rand, S) queries, respectively. Given $\mathcal{Q}_{i-1} \wedge \mathsf{CF}$, the i^{th} query r_i is a trivial query in (H, f) if and only if so is in (Rand, S) and the response of the trivial query is uniquely determined by the previous view. So, output of H or S is same as output of Rand or S respectively. So assume that r_i is not a trivial query.

Lemma 1. *Given $\mathcal{Q}_{i-1} \wedge \mathsf{CF}$, the conditional distribution of H, f, Rand and S on i^{th} query (h_i, x_i) is uniformly distributed on the set $X \setminus (\mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{h_i\})$ provided it is not a trivial query ($h_i = \mathsf{IV}$ for $\mathcal{O}_1$ oracle query).*

Proof. In case of Rand and S, as CF_2 is not true the output is drawn randomly outside the set $\mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{h_i\}$. In case of Rand query M, since r_i is a nontrivial query, $\mathsf{Rand}(M)$ hash has not been queried before even by the simulator. So, condition on CF_2 the distribution of $\mathsf{Rand}(M)$ is uniformly distributed on the set $X \setminus (\mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{\mathsf{IV}\})$. In case of S query (h_i, x_i) query, the output is not random only if it is trivial query (where the case (1) of the simulator occurs and for the corresponding message M $\mathsf{Rand}(M)$ has been queried before by the distinguisher). So it is true for both Rand and S. Now we will prove it for H.

Let $S_i = \mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{h_i\}$. If we can prove that for all $a \neq a' \notin S_i$, $\Pr[H(M) = a \mid \mathcal{Q}_{i-1} \wedge \mathsf{CF}_1] = \Pr[H(M) = a' \mid \mathcal{Q}_{i-1} \wedge \mathsf{CF}_1]$ then we are done since for all other choices of a the probability is zero because of condition of CF_1. Given a and a', Let $A = \{f : X \times B \to X : H^f(M) = a \wedge f \text{ satisfies } \mathcal{Q}_{i-1}\}$. Similarly define A' for a'. Now one can define a bijection ϕ between A and A' in the following way.

1. If $f \in A$ then $\phi(f)(h, x) = f(h, x)$ if $\{f(h, x), h\} \cap \{a, a'\} = \phi$

2. $\phi(f)(a, x) = f(a', x)$ if $f(a', x) \notin \{a, a'\}$. Similarly, $\phi(f)(a', x) = f(a, x)$ if $f(a, x) \notin \{a, a'\}$.
3. If $h \notin \{a, a'\}$ but $f(h, x) = a$ then $\phi(f)(h, x) = a'$. Similarly, $f(h, x) = a'$ then $\phi(f)(h, x) = a$.
4. There are four other possibilities i.e.
   (a) if $f(a, x) = a$ then $\phi(f)(a', x) = a'$.
   (b) if $f(a, x) = a'$ then $\phi(f)(a', x) = a$.
   (c) if $f(a', x) = a$ then $\phi(f)(a, x) = a'$.
   (d) if $f(a', x) = a'$ then $\phi(f)(a, x) = a$.

Now it is easy to check that $\phi(f)$ is well defined and it belongs to A'. Here, we mainly interchange the role of a and a' in all possible cases of input and output keeping other values the same. Thus, given $H^f(M) = a$, we should have $H^{\phi(f)}(M) = a'$ keeping all other equalities fixed (in $\mathcal{Q}_{i-1}$). Now it is also easy to check that this is a bijection as we can define the inverse function similarly. Thus, $\mid A \mid = \mid A' \mid$ and hence the probabilities are equal. We can prove similarly for the distribution of f. So we skip the proof of this. ∎

Now we bound the probability of collision events for both cases.

Lemma 2. $\Pr[\neg \mathit{CF}_1] = O(\frac{l^2 q^2}{2^n})$ *and* $\Pr[\neg \mathit{CF}_2] = O(\frac{q^2}{2^n})$*, where l is the maximum number of blocks in H-query and $|X| = 2^n$.*

Proof. We first assume that there is no trivial query. If it is there, then we have less probability as it does not help in collision. Now we compute the probability where all outputs (including the intermediate hash values for different messages) and inputs of f are distinct. Now any choices of input-outputs satisfying the above give all different inputs to f. Thus, the probability of any such choice is $1/2^{nq'}$, where q' is the total number of inputs of f. Number of choices of above tuples is at least $(|X| - 1)(|X| - 3) \cdots (|X| - 2q' + 1)$. Thus, $\Pr[\mathsf{CF}_1] = (|X| - 1)(|X| - 3) \cdots (|X| - 2q' + 1)/2^{nq'} = 1 - O(\frac{l^2 q^2}{2^n})$. In case of $\mathsf{Pr}[\mathsf{CF}_2]$, the probability is $O(\frac{q^2}{2^n})$ as output of simulator and Rand is random except for nontrivial query. As nontrivial can not make collision we have the above collision probability. ∎

Combining the lemmas and Equation (3) we obtain the following main theorem of this section.

Theorem 4. *Prefix-free single length MD hash functions in a fixed-size random oracle model is (t_D, t_S, q, ϵ)-indifferentiable from a random oracle, for any t_D, with $t_S = l \cdot \mathcal{O}(q)$ and $\epsilon = 2^{-n+1} \cdot l^2 \cdot \mathcal{O}(q^2)$, where l is the maximum length of a query made by the distinguisher D.*

3.2 Indifferentiability of prefix free PGV hash functions

Now we consider all collision secure PGV hash functions. We will show, in the prefix-free mode, that sixteen (indexed by $1 \sim 16$ in table 1 of Appendix A) out of twenty are also indifferentiable with random oracle. Others (indexed by $17 \sim 20$ in table 1 of Appendix A) are not indifferentiable from random oracle. It is easy to check that any PGV compression functions are not indifferentiable with random oracle.

Thus, we can not apply the previous theorem directly. First we consider the previous example $f(h_{i-1}, m_i) = E_{m_i}(h_{i-1}) \oplus h_{i-1}$. Coron *et al.* also considered this example and stated indifferentiability in [5]. We will define a simulator which simulates both E and E^{-1}. On query $(1, \cdot, \cdot)$ it simulates E and on query $(-1, \cdot, \cdot)$ it simulates E^{-1}.

Simulator Like the previous simulator, it also keeps the relations $(\mathcal{R}_1, \ldots, \mathcal{R}_{i-1})$. Initially, $\mathcal{R}_0 = \emptyset$. Let $\{P_x\}_{x \in X}$ be a family of random permutation. Now the response of S is as follow:

1. On query $(1, h_i, x_i)$,
   (a) If $\mathsf{IV} \rightarrow_N h_i$ and $g(M) = N \parallel x_i$ then run $\mathsf{Rand}(M)$ and obtain the response h^*. Return $h^* \oplus h_i$ and $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_i \rightarrow_{x_i} h^*\}$ (otherwise behave randomly and similar to previous simulator this does not occur if collision-free occurs).
   (b) Else return $P_{x_i}(h_i) = h^*, \mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_i \rightarrow_{x_i} h^* \oplus h_i\}$.
2. On query $(-1, y_i, x_i)$,
   (a) For each $\mathsf{IV} \rightarrow_N h$ such that $g(M) = N \parallel x_i$, run $\mathsf{Rand}(M) = h^*$. If $h^* \oplus h = y_i$, return h and $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_i \rightarrow_{x_i} h^*\}$. If there is more than one such M we say the event BAD occurs and return randomly.
   (b) Else return $P_{x_i}^{-1}(y_i) = h$ (say) and $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_i \rightarrow_{x_i} h^* \oplus h_i\}$. .

The time complexity of the simulator is $O(lq^2)$. The worst case occurs to search all choices of $\mathsf{IV} \rightarrow_M h$ in the case of S^{-1} query. We define the COLL as defined in previously or BAD occurs. Let D be a distinguisher keeping relations $\mathcal{R}_i$ and $\mathcal{R}_i^*$. Note that, $(E_x(y) = z \Leftrightarrow h \rightarrow_m h') \Longleftrightarrow m = x, h = y$ and $h' = z \oplus y$. Now for a random permutation either z or y is chosen randomly.

1. For E query, define $S_i = \mathsf{Supp}(\mathcal{R}_i) \oplus h_i \cup P_{x_i}[i]$, where $P_x[i]$ is the set of all images of P_x obtained from P_x or P_x^{-1}-query till i^{th} query of the distinguisher.
2. For E^{-1} query, $S_i = \mathsf{Supp}(\mathcal{R}_i) \cup (\mathsf{Supp}(\mathcal{R}_i) \oplus y_i) \cup P_{x_i}^{-1}[i]$, where $P_x^{-1}[i]$ is the set of all images of P_x^{-1}.
3. Define, $W_i = \{h \oplus h^* : \mathsf{IV} \rightarrow_M h \rightarrow_m h^* \in \mathcal{R}_{i-1}^*$ and $M \parallel m = g(X)$ for some $X\}$. This set is related to the BAD event.
4. Finally we define, $Z_i = S_i \cup W_i \cup \{h_i\}$ (for R query $h_i = \mathsf{IV}$, for E^{-1} query we can ignore $\{h_i\}$).

Now we say that $\mathcal{V}_i$ is not collision-free if for for some $j \leq i$, the output of O_2 oracle (in j^{th} query) is in W_i and it is not a trivial query. This definition is a modified definition of previous collision-free. Here we change the collision set to W_i. Similar to the previous results we have the following lemma and main theorem of this section.

Lemma 3. *The conditional distribution of $H, E, E^{-1}\mathsf{Rand}, S$ and S^{-1} on i^{th} query, given $\mathcal{Q}_{i-1} \wedge \mathsf{CF}$ is uniformly distributed on the set $X \setminus W_i$ provided it is not a trivial query, where $h_i = \mathsf{IV}$ for $\mathcal{O}_1$ query or (h_i, x_i) be the query for $\mathcal{O}_2$. In case of trivial query all distribution are degenerated.*

Proof. If the query is non-trivial query and collision free is true then Rand, S, S^{-1}, E and E^{-1} are uniformly distributed on the set $X \setminus W_i$. In case of H^E, the hash function, we can prove that $\Pr[H^E(M_i) = a_1] = \Pr[H^E(M_i) = a_2]$, where $a_1, a_2 \in X \setminus W_i$. While we count all possible functions E, we interchange the roll of a_1 and a_2 in the inputs and outputs of E (as in H^f). We skip the detail of the proof as it is similar to Lemma 1.

If collision free is true the response of trivial query is completely determined by the past view (for all possible oracles). For example, if it is S^{-1} query then note that there are not more than one choice of M (or h, see case (1)) as BAD events is included in the event $\neg$ CF. Thus, there is exactly one h which is completely determined by the past view and this is the response of this query. Other cases also can be checked. ■

Lemma 4. $\Pr[\neg \mathsf{CF}_1] = O(\frac{l^2 q^2}{2^n})$ *and* $\Pr[\neg \mathsf{CF}_2] = O(\frac{q^2}{2^n})$*, where l is the maximum number of blocks in H-query.*

Proof. The proof of the lemma is similar to lemma 2 except when BAD event occurs. For each query it will happen with probability $O(q/2^n)$ as $R(M) \oplus h = y_i$ has probability $1/2^n$ and there can be at most 2^n such M's. ■

Theorem 5. *Prefix-free single length MD hash functions in a fixed-size random oracle model is (t_D, t_S, q, ϵ)-indifferentiable from a random oracle, for any t_D, with $t_S = l \cdot \mathcal{O}(q^2)$ and $\epsilon = 2^{-n+1} \cdot l^2 \cdot \mathcal{O}(q^2)$, where l is the maximum length of a query made by the distinguisher D.*

Indifferentiability of sixteen PGV hash functions.

Now we consider all collision secure PGV hash functions. We will show, in the prefix-free mode, that sixteen (indexed by $1 \sim 16$ in table 1 of Appendix A) out of twenty are also indifferentiable with random oracle. Others (indexed by $17 \sim 20$ in table 1 of Appendix A) are not indifferentiable from random oracle. Till now we have shown for the case-1 of Appendix A. For other cases one can make similar analysis. For example, $h_i = f(h_{i-1}, m_i) = E_{w_i}(m_i) \oplus h_{i-1}$. So, $(E_k(x) = y \Leftrightarrow h \to_m h') \Longleftrightarrow m = x, h = x \oplus k$ and $h' = k \oplus x \oplus y$. One can also define the simulator for other PGV functions similarly. The proof of the indifferentiability will follow similarly.

1. On query $(1, k_i, x_i)$ i.e. $E_{k_i}(x_i)$,
   (a) If $\mathsf{IV} \to_N h_i$ and $g(M) = N \parallel x_i$ then run $\mathsf{Rand}(M)$ and obtain the response h^*. Return $h^* \oplus k_i \oplus x_i$ and $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{(k_i \oplus x_i) \to_{x_i} h^*\}$ (otherwise behave randomly and similar to previous simulator this does not occur if collision-free occurs).
   (b) Else return $P_{k_i}(x_i) = h^*, \mathcal{R}_i = \mathcal{R}_{i-1} \cup \{k_i \oplus x_i \to_{x_i} h^* \oplus k_i \oplus x_i\}$.

2. On query $(-1, k_i, y_i)$, i.e., $E_{k_i}^{-1}(y_i)$
   (a) For each $\mathsf{IV} \rightarrow_N h$ such that $g(M) = N \parallel k_i \oplus h$, run $\mathsf{Rand}(M) = h^*$. If $h^* \oplus h = y_i$, return $h \oplus k_i$ and $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h \rightarrow_{k_i \oplus h} h^*\}$. If there is more than one such M we say the event BAD occurs and return randomly.
   (b) Else return $P_{k_i}^{-1}(y_i) = h$ (say) and $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h \oplus k_i \rightarrow_{x_i} h^* \oplus h \oplus k_i\}$.

3.3 Indifferentiability of Double Length hash functions

Now we consider the double length construction. A $2n$-bit hash value $x_l = (h_l, g_l)$ is computed from κl-bit message $(m_1, m_2, \cdots, m_l)$ as follows. For $i = 1, 2, \cdots, l$, $F(x_{i-1}, m_i) = (h_i, g_i)$ such that

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

where p is a permutation on $2n$ bits and p has no fixed point and $p(g, h) \neq (h, g)$ for any h, g. Further we assume that $p^2(\cdot)$ is an identity permutation. One example would be $p(x) = \overline{x}$, where $\overline{x}$ is the bitwise complement. We define an equivalence relation, $w \equiv w^*$ if $w = p(w^*)$ or $w = w^*$. Like previous simulator we define the simulator as follows:

Simulator : The simulator keeps the relations $(\mathcal{R}_1, \ldots, \mathcal{R}_{i-1})$. Initially, $\mathcal{R}_0 = \emptyset$. On the i^{th} query (h_i, g_i, x_i), the response of S is as follow:

1. If the ith query is same as a previous query, output same output of the previous query.
2. Else if $\exists$ $\mathsf{IV} \rightarrow_N h||g \in \mathcal{R}_{i-1}, g(M) = N \parallel x_i$ where $h||g \equiv h_i||g_i$, then run $\mathsf{Rand}(M)$ and obtain the response $h^*||g^*$. For more than one choices of M, return a random string $h^*||g^*$ (this will never happen if $(\mathcal{R}_1, \ldots, \mathcal{R}_q)$ is collision-free).
   (a) If $h||g = h_i||g_i$ then return h^*.
   (b) If $h||g = p(h_i||g_i)$ then return g^*.
   (c) If $(p(h_i||g_i), x_i)$ has been queried before then
      i. If $h||g = h_i||g_i$ then $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h||g \rightarrow_{x_i} h^*||g^*\}$.
      ii. If $h||g = p(h_i||g_i)$ then $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h||g \rightarrow_{x_i} g^*||h^*\}$.
3. Else return a random string h^*. If $(p(h_i||g_i), x_i)$ has been queried before and response is g^* then $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_i||g_i \rightarrow_{x_i} h^*||g^*\} \cup \{p(h||g) \rightarrow_{x_i} g^*||h^*\}$.

If distinguisher is making q queries at most then one can design the above simulator so that it runs in time $O(lq)$. In the worst case simulator has to back track to initial value to check whether condition (1) is satisfied or not and this needs at most $O(lq)$ time. Similar to previous results we have the following lemma and main theorem of this section. Similar to prefix free MD construction, we can define *support* and *collision free*.

1. Define *support* of a relation $\mathcal{R}_i$ by a subset of X, $\mathsf{Supp}(\mathcal{R}_i) = \{h||g, p(h||g) : h||g \rightarrow_m h_1||g_1 \in \mathcal{R}_i\} \cup \{h||g, p(h||g) : h_1||g_1 \rightarrow_m h||g \in \mathcal{R}_i\} \cup \{\mathsf{IV}\}$. Note that, $\mathsf{Supp}(\mathcal{R}_i) = \mathsf{Supp}(\mathcal{R}_i^*)$.
2. We say, r_i is a *trivial query* if $r_i \in \mathcal{R}_{i-1}^*$. Since g is a prefix-free padding, r_i can be trivial query only if any one of the following holds :
   (a) $r_i = \mathsf{IV} \rightarrow_{g(M)} h_\ell||g_\ell$, where $\mathsf{IV} = h_0||g_0 \rightarrow_{m_1} h_1||g_1 \rightarrow_{m_2} \ldots h_{\ell-1}||g_{\ell-1} \rightarrow_{m_\ell} h_\ell||g_\ell \in \mathcal{R}_{i-1}^*$ and $g(M) = m_1 \parallel \ldots \parallel m_\ell$.
   (b) $r_i = h_{\ell-1}||g_{\ell-1} \rightarrow_{m_\ell} h_\ell$ or $p(h_{\ell-1}||g_{\ell-1}) \rightarrow_{m_\ell} g_\ell$, where $\mathsf{IV} = h_0||g_0 \rightarrow_{m_1} h_1||g_1 \rightarrow_{m_2} \ldots h_{\ell-1}||g_{\ell-1}$, $\mathsf{IV} \rightarrow_{g(M)} h_\ell||g_\ell \in \mathcal{R}_{i-1}^*$ and $g(M) = m_1 \parallel \ldots \parallel m_\ell$.
   (c) r_i is a repetition query i.e. $r_i = r_j$ for some $j < i$. For simplicity we can assume that there is no repetition query as distinguisher's point of view it doest not help anything.
3. We say $\mathcal{V}$ is not *collision free* (or in short $\neg$ CF) if for some i one of followings hold :
   (a) In case of $\mathcal{O}_1$ query : $r_i = h_i||g_i \rightarrow_M h'||g'$ is non trivial and $h'||g' \in \mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{h_i||g_i\}$.
   (b) In case of $\mathcal{O}_2$ query : $r_i = h_i||g_i \rightarrow_{m_i} h'$ is non trivial and $\mathcal{O}_2(p(h_i||g_i), x_i) = g'$ has been queried before and $h'||g'$ or $g'||h' \in \mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{h_i||g_i\} \cup \{p(h_i||g_i)\}$.

Lemma 5. *Given $\mathcal{Q}_{i-1} \wedge \mathsf{CF}$, the conditional distribution of H, f, Rand and S on i^{th} query is uniformly distributed on the set $X \setminus (\mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{h_i||g_i\})$ provided it is not a trivial query, where $h_i||g_i = \mathsf{IV}$ for $\mathcal{O}_1$ query or $(h_i||g_i, x_i)$ be the query for $\mathcal{O}_2$.*

Proof. Given $a(= a_1 \parallel a_2)$ and $a'(= a_1' \parallel a_2') \notin X \setminus (\mathsf{Supp}(\mathcal{R}_{i-1}) \cup \{h_i||g_i\})$, Let $A = \{f : X \times B \rightarrow X : H^f(M) = a \wedge f$ satisfies $\mathcal{Q}_{i-1}\}$. Similarly define A' for a'. Similar to prefix free MD construction, we can define a bijection ϕ between A and A' similar to the Lemma 5.

1. If $f \in A$ then $\phi(f)(b, x)||\phi(f)(p(b), x) = f(b, x)||f(p(b), x)$ if $\{f(b, x)||f(p(b), x), b\} \cap \{a, a'\} = \phi$
2. $\phi(f)(a, x)||\phi(f)(p(a), x) = f(a', x)||f(p(a'), x)$ if $f(a', x)||f(p(a'), x) \notin \{a, a'\}$. Similarly, $\phi(f)(a', x)||\phi(f)(p(a'), x) = f(a, x)||f(p(a), x)$ if $f(a, x)||f(p(a), x) \notin \{a, a'\}$.
3. If $b \notin \{a, a'\}$ but $f(b, x)||f(p(b), x) = a$ then $\phi(f)(b, x)||\phi(f)(p(b), x) = a'$. Similarly, $f(b, x)||f(p(b), x) = a'$ then $\phi(f)(b, x)||\phi(f)(p(b), x) = a$.
4. There are four other possibilities i.e.
   (a) if $f(a, x)||f(p(a), x) = a$ then $\phi(f)(a', x)||\phi(f)(p(a'), x) = a'$.
   (b) if $f(a, x)||f(p(a), x) = a'$ then $\phi(f)(a', x)||\phi(f)(p(a'), x) = a$.
   (c) if $f(a', x)||f(p(a'), x) = a$ then $\phi(f)(a, x)||\phi(f)(p(a), x) = a'$.
   (d) if $f(a', x)||f(p(a'), x) = a'$ then $\phi(f)(a, x)||\phi(f)(p(a), x) = a$.

Now it is easy to check that $\phi(f)$ is well defined and it belongs to A'. Here, we mainly interchange the roll of a and a' in all possible cases of input and output keeping others same. Thus, given $H^f(M) = a$, we should have $H^{\phi(f)}(M) = a'$ keeping all other equalities fixed (in $\mathcal{Q}_{i-1}$). Now it is also easy to check that this is a bijection as we can define the inverse function similarly. Thus, $\mid A \mid = \mid A' \mid$ and hence the probabilities are equal. We can prove similarly for the distribution of f. So we skip the proof of this. ∎

Now we bound the probability of collision events for both cases.

Lemma 6. $\Pr[\neg \mathsf{CF}_1] = O(\frac{l^2 q^2}{2^{2n}})$ *and* $\Pr[\neg \mathsf{CF}_2] = O(\frac{q^2}{2^{2n}})$*, where* l *is the maximum number of blocks in* H*-query and* $|X| = 2^{2n}$*.*

Proof. The proof is also similar to the Lemma 2. So we skip the proof. ∎

Theorem 6. *Let* F *be above double length hash function. Then for any prefix-free function* g*,* MD_g^F *in a single-size random oracle model is* (t_D, t_S, q, ϵ)*-indifferentiable from a random oracle, for any* t_D*, with* $t_S = l \cdot \mathcal{O}(q)$ *and* $\epsilon = 2^{-2n+1} \cdot l^2 \cdot \mathcal{O}(q^2)$*, where* l *is the maximum length of a query made by the distinguisher* D*.*

4 Attack on Some SBL and DBL Hash Functions

In this section we define PGV and PBGV hash functions. We give some indifferentiable attacks on some of these hash functions. We show only attacks with one-block padded message. More than one block, we can attack similarly.

The Preneel-Govaerts-Vandewalle (PGV) schemes [17]

Let x_0 be the initial value and $\kappa = N$. E is N-bit block cipher with an N-bit key. An N-bit hash value x_l is computed from κl-bit message $(m_1, m_2, \cdots, m_l)$ as follows. For $i = 1, 2, \cdots, l$,

$$F(x_{i-1}, m_i) = x_i = E_a(b) \oplus c$$

where $a, b, c \in \{x_{i-1}, m_i, v, x_{i-1} \oplus m_i\}$. Here, v is a constant.

Among 20 collision resistant PGV schemes, even we use prefix-free padding g, we show that 4 schemes are differentiable from random oracle. 4 schemes are $F_1(h_{i-1}, m_i) = E_{h_{i-1}}(m_i) \oplus m_i$, $F_2(h_{i-1}, m_i) = E_{h_{i-1}}(m_i \oplus h_{i-1}) \oplus m_i \oplus h_{i-1}$, $F_3(h_{i-1}, m_i) = E_{h_{i-1}}(m_i) \oplus m_i \oplus h_{i-1}$, and $F_4(h_{i-1}, m_i) = E_{h_{i-1}}(m_i \oplus h_{i-1}) \oplus m_i$. Here, we consider F_1. Similarly, we can show the insecurity of other 3 cases.

- distinguisher D can access to oracles $(\mathcal{O}_1, \mathcal{O}_2)$ where $(\mathcal{O}_1, \mathcal{O}_2)$ is (H, E, E^{-1}) or $(\mathsf{Rand}, S, S^{-1})$.
  - make a random query M such that $g(M) = m$ and $|m| = n$. then give the query M to oracle $\mathcal{O}_1$ and receive z.

- make an inverse query $(-1, x_0, z \oplus m)$ to $\mathcal{O}_2$ and receive m^*.
- if $m = m^*$ output 1, otherwise 0.
- Since any simulator S can know random m only with probability 2^{-n},

$$|\Pr[D^{H,E,E^{-1}} = 1] - \Pr[D^{R,S,S^{-1}} = 1]| = 1 - 2^{-n}$$

This is not negligible. So $\mathsf{MD}_g^{F_1}$ is differentiable from random oracle.

The Preneel-Bosselaers-Govaerts-Vandewalle (PBGV) scheme [16]

Let $x_0 = (h_0, g_0)$ be initial value and $N = 2n$ and $\kappa = N$. E is N-bit block cipher with an N-bit key. A N-bit hash value $x_l = (h_l, g_l)$ is computed from κl-bit message $m = (m_1, m_2, \cdots, m_l)$ where $m_i = (m_{i,1}, m_{i,2})$ and $|m_{i,1}| = |m_{i,2}| = n$. For $i = 1, 2, \cdots, l$, $F(x_{i-1}, m_i) = (h_i, g_i)$ is defined as follows.

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

The following is the indifferentiable attack for the PBGV scheme.

– distinguisher D can access to oracles $(\mathcal{O}_1, \mathcal{O}_2)$ where $(\mathcal{O}_1, \mathcal{O}_2)$ is (H, E, E^{-1}) or $(\mathsf{Rand}, S, S^{-1})$.
  - make a random query M such that $g(M) = m_1 = m_{1,1} || m_{1,2}$ and $|m_1| = 2n$. Then give the query M to oracle $\mathcal{O}_1$ and receive $x_1 = (h_1, g_1)$.
  - make an inverse query $(-1, m_{1,2} \oplus h_0 \oplus g_0 \oplus g_1, m_{1,1} \oplus h_0)$ to $\mathcal{O}_2$ and receive out.
  - if $out = m_{1,2} \oplus g_0$ output 1, otherwise 0.
  - Since any simulator S can know random $m_{1,2}$ only with probability 2^{-n},

$$|\Pr[D^{H,E,E^{-1}} = 1] - \Pr[D^{R,S,S^{-1}} = 1]| = 1 - 2^{-n}$$

This is not negligible. So MD_g^{F} is differentiable from random oracle.

By using the same idea one can find indifferentiability attack on QG-I, LOKI DBH, MDC-2 and some of the Hirose's double length hash constructions.

5 Conclusion

As hash function is at times a popular candidate for approximation of a random oracle, the notion of indifferentiability is important to study. In this paper we have studied many known designs of hash function in term of indifferentiability. Some of them are secure and against some of them we have found attack. So there are many designs, for example sixteen PGV hash functions, which are secure beyond the collision security. This paper also presents an unified way to prove the indifferentiability for many designs of hash functions. Finally we note that there are still many designs whose security analysis in the view of indifferentiability are open.

Acknowledgement

The first author was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF- 2005-213-C00005).

References

1. M. Bellare and P. Rogaway. Random Oracles Are Practical : A Paradigm for Designing Efficient Protocols. In *1st Conference on Computing and Communications Security*, ACM, pages 62–73. 1993.
2. J. Black, P. Rogaway, and T. Shrimpton. Black-box analysis of the block-cipher-based hash function constructions from PGV. In *Advances in Cryptology-Crypto'2002*, volume **2442** of *Lecture Notes in Computer Science*, pages 320–335. Springer-Verlag, 2002.
3. 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.
4. L. Brown, J. Pieprzyk and J. Seberry. LOKI - A Cryptographic Primitive for Authentication and Secrecy Applications. In *Advances in Cryptology-Auscrypt'1990*, volume **453** of *Lecture Notes in Computer Science*, pages 229–236. Springer-Verlag, 1990.
5. J. S. Coron, Y. Dodis, C. Malinaud and P. Puniya. Merkle-Damgard Revisited: How to Construct a Hash Function. In *Advances in Cryptology-Crypto'2005*, volume **3621** of *Lecture Notes in Computer Science*, pages 430–448. Springer-Verlag, 2005.
6. I. B. Damgard. A design principle for hash functions. In *Advances in Cryptology-Crypto'1989*, volume **435** of *Lecture Notes in Computer Science*, pages 416–427. Springer-Verlag, 1989.
7. S. Even, and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. In *Advances in Cryptology-Asiacrypt'1991*, volume **739** of *Lecture Notes in Computer Science*, pages 210–224. Springer-Verlag, 1992.
8. S. Hirose. Provably Secure Double-Block-Length Hash Functions in a Black-Box Model. In *ICISC'2004*, volume **3506** of *Lecture Notes in Computer Science*, pages 330–342. Springer-Verlag, 2005.
9. S. Hirose. Some Plausible Constructions of Double-Block-Length Hash Functions. To appear in proceeding of FSE' 2006.
10. J. Kilian, and P. Rogaway. How to protect DES against exhaustive key search. In *Journal of Cryptology*, **14(1)**:17-35, 2001, Earlier version in CRYPTO' 96.
11. X. Lai and J. L. Massey. Hash Functions Based on Block Ciphers. In *Advances in Cryptology-Eurocrypt'1992*, volume **658** of *Lecture Notes in Computer Science*, pages 55–70. Springer-Verlag, 1993.
12. Stefan Lucks. A Failure-Friendly Design Principle for Hash Functions. In *Advances in Cryptology-Asiacrypt'2005*, volume **3788** of *Lecture Notes in Computer Science*, pages 474–494. Springer-Verlag, 2005.

13. U. Maurer, R. Renner and C. Holenstein. Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology. In *TCC'2004*, volume **2951** of *Lecture Notes in Computer Science*, pages 21–39. Springer-Verlag, 2004.
14. R. C. Merkle. One way hash functions and DES. In *Advances in Cryptology-Crypto'1989*, volume **435** of *Lecture Notes in Computer Science*, pages 428–446. Springer-Verlag, 1990.
15. Mridul Nandi. Towards Optimal Double-Length Hash Functions. In *Indocrypt'2005*, volume **3797** of *Lecture Notes in Computer Science*, pages 77–89. Springer-Verlag, 2005.
16. B. Preneel, A. Bosselaers, R. Govaerts and J. Vandewalle. Collision-free Hashfunctions Based on Blockcipher Algorithms. Proceedings of 1989 International Carnahan Conference on Security Technology, pages 203–210.
17. B. Preneel, R. Govaerts and J. Vandewalle. Hash Functions based on Block Ciphers : A Synthetic approach. In *Advances in Cryptology-Crypto'1993*, volume **773** of *Lecture Notes in Computer Science*, pages 368–378. Springer-Verlag, 1994.
18. J. J. Quisquater and M. Girault. 2n-bit Hash Functions Using n-bit Symmetric Block Cipher Algorithms. In *Advances in Cryptology-Eurocrypt'1989*, volume **434** of *Lecture Notes in Computer Science*, pages 102–109. Springer-Verlag, 1990.
19. C. Shannon. Communication theory of secrecy systems. Bell Systems Technical Journal, **28(4)**: pages 656–715, 1949.
20. R. Winternitz. A secure one-way hash function built from DES. In *Proceedings of the IEEE Symposium on Information Security and Privacy*, pages 88–90, 1984.

Appendix A : Table of twenty PGV Hash Functions

Case	PGV	Case	PGV
1	$E_{m_i}(h_{i-1}) \oplus h_{i-1}$	11	$E_{m_i}(h_{i-1}) \oplus v$
2	$E_{m_i}(w_i) \oplus w_i$	12	$E_{w_i}(h_{i-1}) \oplus v$
3	$E_{m_i}(h_{i-1}) \oplus w_i$	13	$E_{m_i}(h_{i-1}) \oplus m_i$
4	$E_{m_i}(w_i) \oplus h_{i-1}$	14	$E_{w_i}(h_{i-1}) \oplus w_i$
5	$E_{w_i}(m_i) \oplus m_i$	15	$E_{m_i}(w_i) \oplus v$
6	$E_{w_i}(h_{i-1}) \oplus h_{i-1}$	16	$E_{m_i}(w_i) \oplus m_i$
7	$E_{w_i}(m_i) \oplus h_{i-1}$	17	$E_{h_{i-1}}(m_i) \oplus m_i$
8	$E_{w_i}(h_{i-1}) \oplus m_i$	18	$E_{h_{i-1}}(w_i) \oplus w_i$
9	$E_{w_i}(w_i) \oplus v$	19	$E_{h_{i-1}}(m_i) \oplus w_i$
10	$E_{w_i}(m_i) \oplus w_i$	20	$E_{h_{i-1}}(w_i) \oplus m_i$

Table 1. 20 Collision Resistant PGV Hash Functions in the Ideal Cipher Model. ($w_i = m_i \oplus h_{i-1}$)