Revisiting the Indifferentiability of PGV Hash Functions

Yiyuan Luo[1], Zheng Gong[2], Ming Duan[1], Bo Zhu[1] and Xuejia Lai[1]
[1]Department of Computer Science and Engineering
Shanghai Jiaotong University, China
luoyiyuan@sjtu.edu.cn
[2]Faculty of EEMCS
University of Twente, The Netherlands
z.gong@utwente.nl

March 4, 2009

Abstract

In this paper, first we point out some flaws in the existing indifferentiability simulations of the pf-MD and the NMAC constructions, and provide new differentiable attacks on the hash functions based these schemes. Afterthat, the indifferentiability of the 20 collision resistant PGV hash functions, which are padded under the pf-MD, the NMAC/HMAC and the chop-MD constructions, are reconsidered. Moreover, we disclose that there exist 4 PGV schemes can be differentiable from a random oracle with the pf-MD among 16 indifferentiable PGV schemes proven by Chang *et al*. Finally, new indifferentiability simulations are provided for 20 collision-resistant PGV schemes. The simulations exploit that 20 collision-resistant PGV hash functions, which implemented with the NMAC/HMAC and the chop-MD, are indifferentiable from a random oracle. Our result implies that same compression functions under MD variants might have the same security bound with respect to the collision resistance, but quite different in the view of indifferentiability.

1 Introduction

Cryptographic Hash Functions. Cryptographic hash function, which is defined as an admissible algorithm that uniformly maps arbitrary length inputs to fixed length outputs, is widely used as a pivotal primitive for ensuring the integrity of information. In nowadays, the popular design of cryptographic hash functions still follows the well-known Merkle-Damgård (MD) construction [12, 21], by iterating a compression function on an input message to realize a domain extension transform and yields a collision resistant hash function if the underlying compression function is. The primary security goal for cryptographic hash functions has historically been collision resistance. Unfortunately, hash functions have been used for all kinds of applications which the security requirements are not only satisfied by collision resistance, but also pseudo-randomness, and even to be a random oracle [2].

In recent years, the hash community starts to argue that the traditional Merkle-Damgård (MD) construction is not a good design in the security view as a random oracle [9]. Since the well-known extension attack allows one to take a value $H(x)$ for x, and then computes the value $H(x, |x|, y)$, where $|x|$ is the length of x and y is an arbitrary suffix. But this extension property is not allowed for any truly random oracle. For instance, even if the underlying compression function f is assumed to be a fixed-length random oracle, any hash function H^f under MD construction will unlikely to be indifferentiable with a random oracle. From those counter-examples, people realize that collision resistance alone is insufficient for the security of so many different applications of hash functions. For this reason, a rich literature analyzed the security of hash functions obtaining variable-input-length (VIL) from an ideal fixed-input-length (FIL) compression function, such as [1, 2, 3, 9, 17].

In practice, there exist two main approaches to design a compression function for an iterated hash function. One is to implicitly design a compression function by implicitly using the idea of block ciphers, which is called *dedicated* hash function. The other is to explicitly compose a compression function from block ciphers, which is called *block-cipher-based* hash function. By now, it seems still hard to design a dedicated compression function by witnessing the recent collision attacks on serval popular hash functions [23, 24]. The advantage of block-cipher-based hash functions is that one can conveniently choose an extensively studied block cipher (e.g., DES, IDEA, AES, etc) to construct a compression function, so that the design and implementation efforts could be minimized. Also the latest cryptanalysis on such a block cipher can be used to avoid the potential weakness in the compression function. Discussions of hash functions constructed from n-bit block ciphers are mainly divided into *single block length* (SBL) such as 64 PGV schemes [22], and *double block length* (DBL) such as MDC2 [6], where single and double are related to the output range of the underlying block cipher.

The original proposals of block-cipher-based hash functions usually focus on attacks, not formal proofs. As the development of provable security, some works have focused on the provable security of hash function based on block ciphers by modeling the underlying block cipher as a black box [5, 16]. In [5], Black *et al.* described a black-box analysis of all 64 PGV hash functions and proved that in the black box model, there exist 20 out of 64 PGV hash functions are collision resistant.

Indifferentiability Methodology. In TCC'04, Maurer *et al.* introduced a strong security notion called as indifferentiability [19] for a hash function based on a compression function which is an extension of the classical indistinguishability security notion. The advantage of the indifferentiability is that one can built a secure VIL-RO from smaller (FIL) idealized components(such as an ideal compression function or ideal cipher). In Crypto'05, Coron *et al.* first implemented the indifferentiability in analysis of hash functions and suggested four secure constructions [9], which were the prefix-free padding(pf-MD), the NMAC/HMAC and the chop construction(chop-MD). The compression function is viewed as a fixed-length random oracle or built from an ideal block cipher with Davies-Meyer structure. After that, several works followed to investigate the indifferentiability of a hash construction, such as [2, 3, 4, 7, 13, 14].

At Asiacrypt'06, Chang *et al.* presented a unified way to prove the indifferentiability for block-cipher-based hash functions [7]. They analyzed 20 collision resistant PGV hash functions with pf-MD and found there are sixteen schemes are indifferentiable from random oracle and other four schemes are differentiable in the ideal cipher model. In [15], Gong *et al.* provided a synthetic indifferentiability analysis of some block-cipher-based hash functions and claimed that all 20 collision resistant PGV schemes are indifferentiable from random oracle with the pf-MD, the NMAC/HMAC and the chop-MD constructions, where the length padding should be used in the constructions.

Our Contributions. In this paper, by using the indifferentiability methodology, we revisit the indifferentiability of hash functions with pf-MD, NMAC/HMAC and chop-MD construction when the compression function is based on collision resistant PGV structures. We find that there exist 8 PGV schemes are differentiable from random oracle with pf-MD, but indifferentiable from random oracle with NMAC/HMAC and chop-MD. And this give evidence that the four constructions are not the same in the view of the indifferentiability. In the analysis, we revise the flaws in Coron et al.[9] and Chang *et al.*[7]'s proofs of Davies-Meyer compression function with pf-MD and NMAC, which allow an adversary can implement differentiable attacks on them. Furthermore, we find that in the 16 collision resistant PGV hash functions which are proved indifferentiable from a random oracle in the ideal cipher model with pf-MD in Chang *et al.*'s analysis, there are still 4 are really differentiable. According to our analysis, although all of the 20 collision resistant PGV hash function with NMAC/HMAC and chop-MD are indifferentiable from a random oracle in the ideal cipher model, the chop-MD construction has a better indifferentiability bound in advance.

Organization. The organization of this paper is as follows. In Section 2, the notation of indifferentiability and some previous works are reviewed. In Section 3, formal methods of the indifferentiability of a hash function in the ideal cipher model are described. In Section 4, Coron *et al.*'s and Chang *et al.*'s proofs of indifferentiability of hash functions based on the Davies-Meyer structure with pf-MD and NMAC construction are described and flaws in their works are pointed out, and the right proofs for pf-MD and NMAC construction are given. In Section 5, the indifferentiability of 20 collision resistant PGV hash functions with pf-MD, NMAC/HMAC, chop-MD construction are revisited. Finally we draw a conclusion in Section 6.

Group-1 schemes					
Case	*PGV*	*Case*	*PGV*	*Case*	*PGV*
1	$E_{h_{i-1}}(m_i) \oplus m_i$	5	$E_{m_i}(h_{i-1}) \oplus h_{i-1}$	9	$E_{w_i}(m_i) \oplus m_i$
2	$E_{h_{i-1}}(w_i) \oplus w_i$	6	$E_{m_i}(w_i) \oplus w_i$	10	$E_{w_i}(h_{i-1}) \oplus h_{i-1}$
3	$E_{h_{i-1}}(m_i) \oplus w_i$	7	$E_{m_i}(h_{i-1}) \oplus w_i$	11	$E_{w_i}(m_i) \oplus h_{i-1}$
4	$E_{h_{i-1}}(w_i) \oplus m_i$	8	$E_{m_i}(w_i) \oplus h_{i-1}$	12	$E_{w_i}(h_{i-1}) \oplus m_i$

Table 2.1 Group-1 schemes in [5].

2 Preliminaries

2.1 Ideal Cipher Model and Random Oracle Model

Ideal cipher model, which is often called black box model as well, is a formal model for the security analysis of block-cipher-based hash functions. An ideal cipher is an ideal primitive that models a random block-cipher $E : \{0,1\}^k \times \{0,1\}^n \mapsto \{0,1\}^n$. Each key $k \in \{0,1\}^k$ defines a random permutation $E_k = E(k, \cdot)$ on $\{0,1\}^n$. An adversary is given forward or inverse queries to oracles E, when he makes a forward query to E with $(+, k, p)$, it returns the point c such that $E_k(p) = c$, when he makes an inverse query to E with $(-, k, c)$, it returns the point p such that $E_k(p) = c$.

As the ideal cipher model, the random oracle model(ROM) is also a method of developing provably secure cryptosystems. Simply says, A random oracle (RO) is an ideal primitive which provides a random output for each new query. Identical input queries are given the same answer. Recently, it was proven by Coron *et al.* [11] that the ideal cipher model is equivalent to the random oracle model by using the indifferentiability methodology.

2.2 PGV Hash Functions

At Crypto'93, Preneel, Govaerts and Vandewalle (PGV) [22] proposed a synthetic approach to design single block length hash function based on block ciphers. They considered the method of turning a block cipher $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ into a hash function $H : \{0,1\}^* \to \{0,1\}^n$ using a compression function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ derived from E. For a fixed n-bit constant v, PGV considered all 64 compression functions f of the form $f(h_{i-1}, m_i) = E_k(p) \oplus a$ where $k, p, a \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i, v\}$, where $w_i = h_{i-1} \oplus m_i$ and v is a constant. The hash function $H(m_1, \ldots, m_l)$ can subsequently be described as follows:

$$h_i = f(h_{i-1}, m_i), i = 1, 2, \ldots, l$$

Here f is the underlying compression function, h_0 is equal to a fixed initial value IV, $|m_i| = n$ for each $i \in [1 \cdots l]$ and h_l is the hashcode. Of the 64 such schemes, PGV regards 12 schemes as secure in the sense of both the preimage resistance and the collision resistance. Another 13 schemes they classified as backward-attackable, which means they are subject to a potential attack. The remaining 39 schemes are subject to fatal attacks. Afterthat, Black *et al.* [5] revisited all the 64 PGV schemes in the ideal cipher model. They proved that the 12 secure schemes that PGV had singled out remain secure in the black-box analysis, which are denoted as the Group-1 schemes (listed in Table 2.1). Additionally, there are 8 schemes are also secure after iteration, they denoted these 8 schemes as the Group-2 schemes (listed in Table 2.2).

2.3 Four Merkle-Damgård Variants

In [9], Coron *et al.* proposed four Merkle-Damgård variants such that the arbitrary length hash function H must behave as a random oracle when the fixed-length building block is viewed as a random oracle or an ideal block cipher, namely, the prefix-free padding, the NMAC/HMAC and the chop constructions. In this paper only compression function based on PGV schemes is considered. The four variants are described in Table 2.3.

Group-2 schemes					
Case	*PGV*	*Case*	*PGV*	*Case*	*PGV*
13	$E_{w_i}(m_i) \oplus v$	16	$E_{w_i}(h_{i-1}) \oplus v$	19	$E_{m_i}(w_i) \oplus v$
14	$E_{w_i}(m_i) \oplus w_i$	17	$E_{m_i}(h_{i-1}) \oplus m_i$	20	$E_{m_i}(w_i) \oplus m_i$
15	$E_{m_i}(h_{i-1}) \oplus v$	18	$E_{w_i}(h_{i-1}) \oplus w_i$		

Table 2.2 Group-2 schemes in [5].

pf-MD$^f(IV, M)$: $M = m_1 \| \cdots \| m_i, h_0 = IV_1$ For $i = 1$ to i do $h_i = f(g(m_i), h_{i-1})$ Return h_i	NMAC^{f_1,f_2} (IV_1, M) : $M = m_1 \| \cdots \| m_i, h_0 = IV$ For $i = 1$ to i do $h_i = f_1(m_i, h_{i-1})$ Return $f_2(h_i, IV_2)$
HMAC^f (IV, M) : $M = m_1 \| \cdots \| m_i, h_0 = f(0^n, IV)$ For $i = 1$ to i do $h_i = f(m_i, h_{i-1})$ Return $h_{i+1} = f(h_i, IV)$	chop-MD$_s^f$ (IV, M) : $M = m_1 \| \cdots \| m_i, h_0 = IV$ For $i = 1$ to i do $h_i = f(m_i, h_{i-1})$ Return the first $n - s$ bit of h_i

Table 2.3 Definitions of the four MD variants [9].[1]

The famous Davis-Meyer scheme is an instance of PGV schemes, which can be denoted as $f(h_{i-1}, m_i) = E_{m_i}(h_{i-1}) \oplus h_{i-1}$. In the pf-MD construction, the message $(m_1, \ldots, m_l)$ are guaranteed to be prefix-free. This is because prefix-free encoding enables to eliminate the message expansion attack on hash functions, such as extension attack on MAC. For example, if a MAC is built from a hash function like MAC$(k, m) = H(k \parallel m)$ where k is the secret key. Then this MAC scheme is completely insecure for any Merkle-Damgård construction(including Merkle-Damgård strengthening). That is to say, given MAC$(k, m) = H(k \parallel m)$, we can extend the message m with any single arbitrary block m' and obtain MAC$(k, m \parallel m') = H(k \parallel m \parallel m')$ without knowing the secret key k. If we apply a prefix-free encoding to a message and then call the hash function to get its hash value, we can eliminate the message expansion attack. In fact, NMAC/HMAC and chop-MD are the same as pf-MD by references to avoid the message expansion attack.

2.4 Indifferentiability

In this part, we recall the definition for indifferentiability[9, 19], which will be used in the following security analysis of PGV hash functions on the four MD variants.

Definition 1 *A Turing machine H with oracle access to an ideal primitive E is said to be (t_D, t_S, q, ϵ)-indifferentiable from an ideal primitive $\mathcal{F}$ if there exists a simulator S with oracle access to $\mathcal{F}$ and running in time at most t_S, such that for any distinguisher D it holds that:*

$$|Pr[D^{H,E} = 1] - Pr[D^{\mathcal{F},S} = 1]| < \epsilon$$

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

The role of the simulator is to simulate the ideal primitive E so that no distinguisher can tell whether it is interacting with H and E, or with $\mathcal{F}$ and S; In other words, the output of S should look consistent with what the

[1] $g(m_i)$ is the prefix-free padding, returns $1||m_i$ if m_i is the last block, else returns $0||m_i$. f_1, f_2 are two independent compression functions, IV_1, IV_2 are two distinct initial values.

distinguisher can obtain from $\mathcal{F}$. Note that the simulator does not see the distinguisher's queries to $\mathcal{F}$; however, it can call $\mathcal{F}$ directly when it is required for the simulation. Here the algorithm H will represent the construction of an iterative hash function based on E. The ideal primitive E will represent the underlying primitive used to build the hash function. In this paper, we assume E is an ideal block cipher. $\mathcal{F}$ is a random oracle with same domain and range as the hash function. In the case of ideal cipher model the distinguisher can access both E and E^{-1} oracles and the simulator has to simulate the both.

It was proven by Maurer *et al.* that if H^E is indifferentiable from $\mathcal{F}$, then H^E can replace $\mathcal{F}$ in any cryptosystem. The original theorem stated in below is a generic statement of the indifferentiability.

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

Coron *et al.* stated the indifferentiability of Davies-Meyer block cipher based construction with four MD variants in the ideal cipher model, the theorem is stated in [9] as follows.

Theorem 2 *The Davis-Meyer scheme is $f(h_{i-1}, m_i) = E_{m_i}(h_{i-1}) \oplus h_{i-1})$ pf-MD, chop-MD, NMAC and HMAC are (t_D, t_S, q, ϵ)-indifferentiable from a random oracle in the ideal cipher model. For any t_D ,with $t_S = O(q^2)$, with $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$ for pf-MD, $\epsilon = 2^{-s} \cdot l^2 \cdot O(q^2)$ for chop-MD, $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$ for NMAC and HMAC, where l is the maximum length of a query made by the distinguisher D.*

It was observed that Coron *et al.*'s bound of chop-MD is not tight. In [8], Chang and Nandi presented an improved indifferentiability security bound for chop-MD and stated the following theorem:

Theorem 3 *The chop-MD construction is $(t_D, t_S, q, \sigma, \epsilon)$-indifferentiable from a random oracle, in the random oracle model for the compression function, for any t_D, with $t_S = l \cdot O(q^2)$ and $\epsilon = \frac{(3(n-s)+1)q_2+(n-s)q_1}{2^s} + \frac{q}{2^{n-s-1}} + \frac{\sigma^2}{2^{n+1}} = O(\frac{nq}{2^s} + \frac{q}{2^{n-s}} + \frac{\sigma^2}{2^n})$, where $q = q_1 + q_2$ is the total number of queries and σ is the total number of queried message blocks.*

3 Proofs of Indifferentiability of PGV Hash Functions

It is easy to see that any PGV compression functions are not indifferentiable from a random oracle [18]. But when the initial value IV is fixed, then there exist some PGV hash functions are indifferentiable from random oracle. To prove a scheme indifferentiable from a random oracle is not trivial. In Coron *et al.*'s paper [9], the proof of indifferentiability involved two steps. First, a simulator is built to simulate the task of the ideal cipher. Secondly, they showed that the view of any distinguisher in the random oracle model, with oracle access to the actual random oracle and the ideal cipher simulator, didn't differ from its view in the ideal cipher model, with oracle access to the RO construction and the ideal cipher, by more than a negligible amount. Each proofs of indifferentiability consisted of a hybrid argument that presented a sequence of mutually indistinguishable games starting in the random oracle model, with the RO $\mathcal{F}$ and the ideal cipher simulator S(denoted by $S^{\mathcal{F}}$), leading up to the ideal cipher model, with the RO construction and the ideal cipher E (denoted by H^E). To prove the indifferentiability of a construction, they played six games and the proof is complicated.

Later Chang *et al.* presented a formal method to prove the indifferentiability for many designs of hash functions with pf-MD construction which was in fact the same to Coron *et al.*'s proof. Since Chang *et al.*'s proof is more mathematical and formal, we adopt their method in our analysis. Here we describe Chang *et al.*'s proof on pf-MD in below.

Let D be a distinguisher and S be a simulator for the formal analysis of indifferentiability. By following Definition 1, D is interacting with two cryptosystems $(\mathcal{O}_1, \mathcal{O}_2)$, where either $(\mathcal{O}_1, \mathcal{O}_2) = (H, E)$ or $(\mathcal{O}_1, \mathcal{O}_2) = (\mathcal{F}, S)$. The distinguisher's goal is to distinguish which scenario it involves after the queries to $(\mathcal{O}_1, \mathcal{O}_2)$. $H : \mathcal{M} \to \mathcal{Y}$ denotes a hash function constructed from a block-cipher $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ where $\mathcal{M} \in \{0,1\}^*$ and $\mathcal{Y} \in \{0,1\}^n$. $\mathcal{F}$ is a random oracle which has the same domain and range with H. h_i denotes the hash value of the i-th query. Let $r_i \leftarrow (h_{i-1} \xrightarrow{m_i} h_i)$ be the i-th query-response obtain from the query to the oracle $\mathcal{O}_2$ where $m_i \in \{0,1\}^n$. $\mathcal{R}_i = (r_1, \cdots, r_i)$ denotes the query-response set on the oracles $\mathcal{O}_2$ after the i-th query. Let $r'_i \leftarrow (IV \xrightarrow{M} h_i)$ be the i-th query-response to the oracles $\mathcal{O}_1$ where $M \in \mathcal{M}$. $\mathcal{R}'_i = (r'_1, \cdots, r'_i)$ denotes the query-response set on the oracles $\mathcal{O}_1$ after the i-th query. A *functional closure* $\mathcal{R}^*$ on $\mathcal{R}$ is the set with the following properties.

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

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

The $\mathcal{O}_1$-query inputs an arbitrary length message and outputs a fixed length hash value, while the $\mathcal{O}_2$-query inputs a fixed length key and plaintext or ciphertext and outputs the corresponding ciphertext or plaintext, respectively. The details of the two categories of queries are described in below.

- **Query on $\mathcal{O}_1 = H$ or $\mathcal{O}_1 = \mathcal{F}$:**
  - For the i-th query on $\mathcal{O}_1$, distinguisher D selects an arbitrary length message $M_i \in \mathcal{M}$. The response of $\mathcal{O}_1$ is $h_i = H(IV, M_i)$ or $h_i = \mathcal{F}(M_i)$ where $h_i \in \mathcal{Y}$.
  - Let $\mathcal{R}'_i = \mathcal{R}'_{i-1} \cup (IV \xrightarrow{M_i} h_i)$ be the query-response set on the oracles $\mathcal{O}_1$ after the i-th query. The query-response set $\mathcal{R}'_q$ is the complete view of distinguisher D on the oracles $\mathcal{O}_1$ after the maximum q queries. Note that the simulator S never see the distinguisher's queries to $\mathcal{O}_1$.

- **Query on $\mathcal{O}_2 = E$ or $\mathcal{O}_2 = S$:**
  - For the i-th forward query on $\mathcal{O}_2$, distinguisher D queries $(+, k_i, p_i)$ where $k_i, p_i \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i, v\}$ and the response is $c_i = E_{k_i}(p_i)$ or $c_i = S(k_i, p_i)$, where $c_i \in \{0,1\}^n$. By computing the hash value h_i from the tuple (k_i, p_i, c_i), the i-th query-response set $\mathcal{R}_i = \mathcal{R}_{i-1} \cup (h_{i-1} \xrightarrow{m_i} h_i)$.
  - For the i-th inverse query on $\mathcal{O}_2$, distinguisher D queries $(-, k_i, c_i)$ where $k_i \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i, v\}$ and $c_i \in \{0,1\}^n$ and the response is $p_i = E^{-1}_{k_i}(c_i)$ or $p_i = S^{-1}(k_i, c_i)$, where $p_i \in \{0,1\}^n$. By computing h_{i-1}, h_i from the tuple (k_i, p_i, c_i), the i-th query-response set $\mathcal{R}_i = \mathcal{R}_{i-1} \cup (h_{i-1} \xrightarrow{m_i} h_i)$.
  - Let $\mathcal{R}_q$ be the query-response set of the oracle $\mathcal{O}_2$ after the maximum q queries. According to the transitive and substitute properties of $\mathcal{R}_q$, the functional closure $\mathcal{R}^*_q$ is the complete view of distinguisher D on the oracles $\mathcal{O}_2$. Here the simulator S also has this view.

When D interacts with $(\mathcal{F}, S)$, the simulator should simulate the ideal cipher E perfectly except a negligible probability. When D makes queries to the oracle $(\mathcal{O}_1, \mathcal{O}_2)$, there may be some bad events happen, and the distinguisher D can exploit these bad events to decide which scenario it is in. If bad events don't happen, the distinguisher can never distinguish which scenario it is in except for a negligible probability.

In Chang *et al.*'s indifferentiability analysis, E_1, E_2 are the bad events when D interacts with (H, E) and $(\mathcal{F}, S)$, respectively. The oracles (H, E) and $(\mathcal{F}, S)$ are identically distributed in the past view of the distinguisher when E_1, E_2 do not happen. $Adv(\mathcal{D})$ is the measure of the maximal advantage of indifferentiability over all distinguishers $\mathcal{D}$. For brevity, D_1 denotes the event $\mathcal{D}^{H,E} = 1$ and D_2 denotes the event $\mathcal{D}^{\mathcal{F},S} = 1$. Let the function $max()$ returns the largest value of inputs, The advantage of D is given in [7] as follows.

$$Adv(\mathcal{D}) = |Pr[\mathcal{D}_1] - Pr[\mathcal{D}_2]| \leq 2 \times max(Pr[E_1], Pr[E_2]).$$

Now the proof of indifferentiability of a scheme is clear. First, one should construct a simulator S such that D interacting with $(\mathcal{F}, S)$ is indifferentiable with (H, E). Next, one must calculate the upper bound of the probability of the differentiable events, when D interacts with $(\mathcal{F}, S)$ and (H, E) respectively. Finally, one can deduce the maximal advantage of the differentiability over all distinguishers D.

4 Flaws in Previous Indifferentiability Analysis of the Davies-Meyer Scheme.

The Davies-Meyer scheme is a well-known construction in the design of compression function based on block ciphers, which also belongs to 20 collision resistant PGV structures. It is also used implicitly implemented in the constructions of MD5 and SHA-1. Coron *et al.*'s full paper [9] presented the detailed proof of the indifferentiability of the pf-MD, the chop-MD and NMAC based on the Davies-Meyer scheme. Chang *et al.* [7] also proposed a proof of the indifferentiability of pf-MD, which uses the Davis-Meyer scheme as the underlying compression function. Unfortunately, we find that there exist some flaws in Coron *et al.*'s proofs of pf-MD and NMAC, and also Chang *et al.*'s proof of the pf-MD such that a new type distinguisher can implement differentiable attacks on the Davies-Meyer scheme while extends its domain by using the pf-MD and the NMAC construction. This section will be divided into three parts. In the first part, Coron *et al*'s and Chang *et al*'s simulators for pf-MD and NMAC are recalled. In the second part, new differentiable attacks on these simulators are presented. Finally, according to our new attacks, the indifferentiability simulations for the Davies-Meyer scheme with pf-MD and NMAC are refined in the third part.

4.1 Previous Simulators of pf-MD and NMAC

Coron *et al.*'s and Chang *et al.*'s simulators of pf-MD and NMAC based on Davies-Meyer structure are described in the appendix A. When these simulators are built, then the advantages of the distinguishers can be calculated by using the method in [9] or [7]. In the next part, we will show how to differentiable attack these simulations and refine the simulations to against this type of attacks.

4.2 A New Type of Differentiable Attacks on the Simulations of pf-MD and NMAC.

In this part, some differentiable attacks are presented to disclose the fact that the plausible simulations(which are recalled in Appendix A) will be failed in the ideal cipher model. After pointed out the attacks, the simulations and the proofs for pf-MD and NMAC are refined to avoid the above attacks. The following distinguishers demonstrate how to attack Coron *et al*'s and Chang *et al*'s simulators.

Attack on the Simulations of pf-MD.

The following distinguisher can distinguish (H, E) and $(\mathcal{F}, S)$ with a non-negligible probability when the simulator behaves as Coron *et al.*'s and Chang *et al.*'s simulator of pf-MD construction.

Distinguisher D can access to oracles $(\mathcal{O}_1, \mathcal{O}_2)$ where $(\mathcal{O}_1, \mathcal{O}_2)$ is (H, E) or $(\mathcal{F}, S)$.

1. D selects a message M such that $g(M) = m$ where $|m| = n$, then he makes the query M to $\mathcal{O}_1$ and receives h.
2. D makes an inverse query $(-, m, h \oplus IV)$ to $\mathcal{O}_2$ and receives IV^*.
3. If $IV = IV^*$ output 1, otherwise output 0.

If the D outputs 1, then $(\mathcal{O}_1, \mathcal{O}_2)$ is (H, E), otherwise $(\mathcal{F}, S)$. Since receiving an inverse query by the first time and there does not exist $IV \xrightarrow{M'} h_{i-1} \in \mathcal{R}^*_{i-1}$, the simulator S^{-1} can output the right IV with a negligible probability 2^{-n}, such that

$$Adv(D) = |Pr[D^{H,E} = 1] - Pr[D^{\mathcal{F},S} = 1]| = 1 - 2^{-n}.$$

The reason why this attack can be succeed is that Coron *et al.* didn't consider the scenario when the distinguisher makes an inverse query to the simulator and the goal of the distinguisher is to receive a value he already knows. So the response of the simulator S can't be random for each inverse query. Chang *et al.* may observe Coron *et al.*'s flaw in pf-MD since their simulator is different from Coron *et al.*'s. Their correction has avoided attacks which involve queries which the length are at least two blocks . But they didn't consider the scenario that an attack which applied in only one block length and the distinguisher's goal is to receive the initial value IV. We can see the distinguisher can distinguish (H, E) from $(\mathcal{F}, S)$ with an overwhelming probability. The similar attack can be extended to Coron *et al.*'s simulator of NMAC.

Attack on Coron *et al*'s Simulation of NMAC.

The following distinguisher can distinguish (H, E) and $(\mathcal{F}, S)$ with a non-negligible probability when the simulator behaves as Coron *et al*'s simulator of NMAC construction.

Distinguisher D can access to oracles $(\mathcal{O}_1, \mathcal{O}_2)$ where $(\mathcal{O}_1, \mathcal{O}_2)$ is $(H, \{E1, E2\})$ or $(\mathcal{F}, \{S1, S2\})$.

1. D selects a message m where $|m| = n$, then he makes the query m to $\mathcal{O}_1$ and receives h.
2. D makes a forward query $(1, +, m, IV_1)$ to $\mathcal{O}_2$ and receives c_1, then he gets $h_1 = IV_1 \oplus c_1$.
3. D makes an inverse query $(2, -, h_1, h \oplus IV_2)$ to $\mathcal{O}_2$ and receives IV_2^*.
4. If $IV_2 = IV_2^*$ output 1, otherwise output 0.

If D outputs 1, then $(\mathcal{O}_1, \mathcal{O}_2)$ is $(H, \{E1, E2\})$, otherwise it is $(\mathcal{F}, \{S1, S2\})$. Since the inverse is never queried before, the simulator $S2$ can output the right IV_2 with a negligible probability of 2^{-n}, whilst

$$Adv(D) = |Pr[D^{H,E1,E2} = 1] - Pr[D^{\mathcal{F},S1,S2} = 1]| = 1 - 2^{-n}.$$

Hence, the distinguisher D can distinguish $(H, \{E1, E2\})$ from $(\mathcal{F}, \{S1, S2\})$ with an overwhelming probability.

4.3 Corrections

Though there are some flaws in simulators mentioned above, they can be corrected easily. In fact, all problems are from the inverse queries of the last block of a message. So the simulator's response to an inverse query to the last block needs to be treated with caution. Now corrections for each of the simulators mentioned above are given in below.

1. Corrections on Coron *et al.*'s and Chang *et al.*'s simulator of pf-MD.
    - For the i-th query $(-, k_i, c_i)$ on S where $k_i = m_i$:
        (a) If $\exists h_{j-1} \xrightarrow{m_i} (h_{j-1} \oplus c_i) \in \mathcal{R}_{i-1}$ for $j < i$, this is a repetition query, S returns h_{j-1}.
        (b) Else S runs $\mathcal{F}(m_i)$ and obtains the response h. If $h \oplus c_i = IV$, then returns IV and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{IV \xrightarrow{m_i} h\}$.
        (c) Else for each $IV \xrightarrow{M'} h_{i-1} \in \mathcal{R}^*_{i-1}$ and $g(M) = M' \parallel m_i$, runs $\mathcal{F}(M) = h_i$. If $h_i \oplus h_{i-1} = c_i$, returns h_{i-1} and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$
        (d) Else S randomly selects an intermediate value $h'_{i-1} \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h'_{i-1} \xrightarrow{m_i} c_i \oplus h'_{i-1}\}$, then returns h'_{i-1}.
2. Corrections on Coron *et al.*'s simulator of NMAC.

- For the j-th query $(2, -, k_j, c_j)$ on $S2$ where $k_j = m_j$:
  (a) If $\exists h_{k-1} \xrightarrow{m_j} (h_{k-1} \oplus c_j) \in \mathcal{Q}_{j-1}$ where $k < j$, this is a repetition query, S returns h_{k-1}.
  (b) Else If $\exists IV_1 \xrightarrow{M} (k_j) \in \mathcal{R}_i^*$ where $\mathcal{R}_i^*$ is the simulator's view of the past queries on $S1$ and then S runs $\mathcal{F}(M)$ and gets h. If $IV_2 \oplus h = c_j$, S updates $\mathcal{Q}_j = \mathcal{Q}_{j-1} \cup \{IV_2 \xrightarrow{m_j} h\}$, then returns IV_2.
  (c) Else S randomly selects an intermediate value $h'_{j-1} \in \{0,1\}^n$ and updates $\mathcal{Q}_j = \mathcal{Q}_{j-1} \cup \{h'_{j-1} \xrightarrow{m_j} c_j \oplus h'_{j-1}\}$, then returns h'_{j-1}.

When these simulators are corrected, then the advantage of any distinguisher can be calculated as in [9] or [7]. It is easy to see that the time complexity of the simulator and the advantage of any distinguishers are not affected. Thus one can easily obtain the following corollary.

Corollary 1 *The Davis-Meyer scheme with pf-MD, chop-MD, NMAC and HMAC are (t_D, t_S, q, ϵ)-indifferentiable from a random oracle in the ideal cipher model. For any t_D ,with $t_S = O(q^2)$, with $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$ for pf-MD, $\epsilon = 2^{-s} \cdot l^2 \cdot O(q^2)$ for chop-MD, $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$ for NMAC and HMAC, where l is the maximum length of a query made by the distinguisher D.*

In [15], Gong *et al.* also provided an indifferentiability analysis of 20 PGV schemes with pf-MD and claimed that all 20 schemes are indifferentiable from random oracles with prefix-free padding (the length padding is also implemented). There is an obvious error in their simulators that the simulators needed to record the distinguisher's queries to the random oracle $\mathcal{F}$. In fact, the simulator can never have the record of the distinguisher's queries, which can be derived from the definition of indifferentiability.

5 Indifferentiability Analysis of PGV Hash Functions

Due to the new flaws disclosed in the our analysis, the indifferentiability of PGV schemes with pf-MD, NMAC/HMAC and chop-MD are reconsidered in this section. Based on our analysis of pf-MD, the necessary conditions for a PGV hash construction to be indifferentiable from a random oracle are analyzed. Filtered by those necessary conditions, there are only twelve schemes survived in 64 PGV schemes, which include eight of the Group-1 and four of the Group-2 schemes. [5].

At AsiaCrypt'06, Chang *et al.*[7] presented an indifferentiability security analysis of these schemes with pf-MD. They claimed that there are 4 schemes among 20 collision-resistant PGV schemes are differentiable from random oracle with pf-MD. And the remaining 16 schemes are indifferentiable from a random oracle with pf-MD. The four insecure schemes(in the sense of indifferentiability with pf-MD) are case 1, 2, 3 and 4 of the Group-1 schemes. Here we find that in the remaining 16 schemes, there are another four schemes are differentiable from random oracle with pf-MD. These four schemes are case 15, 17, 19 and 20 from the Group-2 schemes.

When analyze these 20 collision resistant PGV hash function for NMAC/HMAC and chop-MD construction, we found all of them are indifferentiable from a random oracle in the ideal cipher model, and the chop-MD construction has the better indifferentiability security bound than NMAC/HMAC construction. This exploits that the four MD variants are not the same in the sense of indifferentiability. According to our synthetic analysis, we exploit the fact that in 20 PGV collision resistant constructions, there exist schemes that are differentiable from random oracle for the pf-MD construction, but are indifferentiable from random oracle for the NMAC/HMAC and chop-MD construction, while the chop-MD construction has the better indifferentiability security bound. This fact gives the evidence that the four popular MD variants, namely pf-MD, NMAC/HMAC, the chop construction, are not the same in the sense of indifferentiability.

5.1 Indifferentiability of PGV Hash Functions with pf-MD

Here we use the indifferentiability methodology to revisit PGV schemes with the pf-MD construction. We analyze the properties of 64 PGV schemes and find the necessary conditions for a PGV schemes to be indifferentiable from a random oracle. The necessary conditions are described as follows. First we present the theorem with respect to the compression function which is not a collision resistant PGV scheme.

Theorem 4 *A hash function H built from any PGV scheme $h_i = f(h_{i-1}, m_i)$ with pf-MD is differentiable from a random oracle if H is not collision resistant.*

The proof is given in Appendix B.1. Based on Theorem 4, it is easy to see that 44 out of the total 64 PGV schemes are not collision resistant, thus they are differentiable from random oracle with pf-MD.

Theorem 5 *A hash function H built from any PGV construction $h_i = f(h_{i-1}, m_i)$ with pf-MD is differentiable from a random oracle if $(h_i, m_i) \Rightarrow h_{i-1}$. That is to say, it is trival to deduce h_{i-1} from (h_i, m_i) with access to the block cipher. For example, $h_i = E_{m_i}(h_{i-1})$, if we know the value of (h_i, m_i), then $h_{i-1} = E^{-1}_{m_i}(h_i)$.*

The proof is given in Appendix B.2. Based on Theorem 5, the 4 PGV schemes, which are case 15, 17, 19 and 20 of the Group-2 schemes, are differentiable from a random oracle.

Theorem 6 *A hash function H built from any PGV schemes $h_i = f(h_{i-1}, m_i)$ with pf-MD is differentiable from a random oracle if given (h_{i-1}, k, c) where $k \in \{h_{i-1}, v\}$ is the key to the block cipher E and c is a linear combination of $\{h_{i-1}, m_i, h_i, v\}$ and the cipher text of the block cipher E, it is infeasible to deduce m_i without access to the block cipher. For example, if $h_i = E_{h_{i-1}}(m_i) \oplus m_i$, then $k = h_{i-1}$ and $c = h_i \oplus m_i$, from the triple $(h_{i-1}, h_{i-1}, h_i \oplus m_i)$, it is infeasible to deduce m_i without access to E.*

The proof is given in Appendix B.3. Based on theorem 6, the 4 PGV schemes, which are case 1, 2, 3, 4 of the group-1 schemes, are differentiable from a random oracle. From the the above analysis, one can easily get the following corollary.

Corollary 2 *A hash function H built from the PGV compression function $h_i = f(h_{i-1}, m_i)$ with pf-MD is differentiable from a random oracle if it satisfies one of the following conditions.*

- ***A.*** *The hash function H is not collision resistant.*
- ***B.*** *$(h_i, m_i) \Rightarrow h_{i-1}$. That is to say, it is trival to deduce h_{i-1} from (h_i, m_i) with access to the block cipher.*
- ***C.*** *Given (h_{i-1}, k, c) where $k \in \{h_{i-1}, v\}$ is the key to the block cipher E and c is a linear combination of $\{h_{i-1}, m_i, h_i, v\}$ and the cipher text of the block cipher E, it is infeasible to deduce m_i without access to the block cipher.*

The case 15, 17, 19, 20 of the group-2 schemes(see table 1.2) satisfy the condition B, and the case 1, 2, 3, 4 of the group-1 schemes(see table 1.1) satisfy the condition C, so they are differentiable from a random oracle with pf-MD construction. Those 8 differentiable schemes are listed in Table C.1.

Since the necessary conditions for the indifferentiability of a PGV structure with the pf-MD construction are given, it is easy to analyze a construction by checking if it satisfies any one of the conditions mentioned above. If anyone of these conditions holds, then the PGV scheme is differentiable from a random oracle with the pf-MD construction. After checking these conditions for every 64 PGV construction, there are only 12 PGV schemes are secure against differentiable attack with pf-MD construction, which are listed in table C.2. The following theorem is proven in Appendix B.4.

Theorem 7 *The twelve PGV schemes, which are list in table C.2, are (t_D, t_S, q, ϵ) indifferentiable from a random oracle in the ideal cipher model. For any t_D, with $t_S = l \cdot O(q^2)$, with $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$ for pf-MD, where l is the maximum length of a query made by the distinguisher D.*

5.2 Indifferentiability of PGV Hash Functions with NMAC/HMAC

In the above analysis, there are only 12 of the 20 collision-resistant PGV schemes are indifferentiable from random oracle with pf-MD construction. In this part we will show it is not the same in the analysis of NMAC/HMAC construction. For brevity, we only analyze the NMAC construction. The results can be easily extended to the HMAC construction because HMAC is a special case of NMAC. In our analysis, all of 20 collision-resistant PGV constructions are indifferentiable from random oracle with NMAC/HMAC construction, which implies that the NMAC/HMAC construction is better than the pf-MD construction.

Furthermore, we will show even if a collision resistant PGV construction satisfies condition B or C in corollary 2, it can be indifferentiable from random oracle with NMAC/HMAC construction. For simplicity, we only show the case 15 from group-2 schemes(table 2.2) satisfies condition B, but is indifferentiable from a random oracle for the NMAC construction. For other cases, one can make a similar analysis and the proof of the indifferentiability will be deduced similarly.

Lemma 1 *The collision resistant PGV compression function $h_i = E_{m_i}(h_{i-1})$ which satisfies condition B in theorem 3 is (t_D, t_S, q, ϵ) indifferentiable from a random oracle in the ideal cipher model. For any t_D, with $t_S = O(q^2)$, with $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$ for NMAC, where l is the maximum length of a query made by the distinguisher D.*

Lemma 1 is proven in Appendix B.5. In fact, for any one of the 20 collision resistant PGV constructions, one can build the similar simulator with NMAC/HMAC construction such that any distinguisher fails. Since the proof of the indifferentiability for each PGV scheme is similar to the proof of Lemma 1, we have the following theorem.

Theorem 8 *The 20 collision resistant PGV schemes are (t_D, t_S, q, ϵ) indifferentiable from a random oracle in the ideal cipher model. For any t_D, with $t_S = O(q^2)$, with $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$ for NMAC/HMAC, where l is the maximum length of a query made by the distinguisher D.*

5.3 Indifferentiability of PGV Hash Functions with chop-MD

In this part the indifferentiability of chop-MD for the 20 collision resistant PGV schemes will be analyzed. We show that all the 20 collision resistant PGV schemes are indifferentiable from random oracle in the ideal cipher model for the chop-MD construction. In [10], Coron *et al.* analyzed the indifferentiability of chop-MD based on the Davies-Meyer construction. They had the following lemma:

Lemma 2 *The Merkle-Damgård construction with truncated output chop-MD$_s^E$ based on the Davies-Meyer construction applied to an ideal cipher $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is (t_D, t_S, q, ϵ) indifferentiable from a random oracle $F : \{0,1\}^* \to \{0,1\}^{n-s}$ in the ideal cipher model for E, for any t_D and $t_S = l \cdot O(q^2)$, with $\epsilon = 2^{-s} \cdot l^2 \cdot O(q^2)$.*

Coron *et al.*'s bound of chop-MD is not very tight. In [20], Maurer and Tessaro firstly presented a prefix-free chop-MD construction which has indifferentiability security beyond the birthday barrier. Later, Chang and Nandi presented an improved indifferentiability security bound for chop-MD which stated in theorem 3. Though Chang and Nandi's improved indifferentiability security bound is proved when looks the compression function as a random oracle, their proof of the security bound can be applied in the ideal cipher model when the compression function is based on Davies-Meyer structure. Some collision resistant PGV schemes satisfy condition B or C in theorem 2 can be indifferentiable from random oracle for chop-MD in the ideal cipher model. Take the PGV scheme $h_i = E_{h_{i-1}}(m_i) \oplus m_i$ as an example, if $n = 2s$, we can build the following distinguisher:

Distinguisher D can access to oracles $(\mathcal{O}_1, \mathcal{O}_2)$ where $(\mathcal{O}_1, \mathcal{O}_2)$ is (chop-MD$_s^E$, E) or $(\mathcal{F}, S)$.

1. D selects a message M such that $g(M) = m$ where $|m| = n$, then makes the query M to $\mathcal{O}_1$ and receives h.
2. For each h' from 0 to $2^s - 1$, D makes an inverse query $(-, IV, m \oplus (h \parallel h'))$ to $\mathcal{O}_2$ and receives m'.
3. If there exist an m' such that $m' = m$, D output 1, otherwise output 0.

Since the simulator never knows the right message m, it gives the right response only with probability 2^{-s} after $q = 2^s$ queries. After queried q times to $\mathcal{O}_2$,

$$Adv(D) = |Pr[D^{H,E,E^{-1}} = 1] - Pr[D^{\mathcal{F},S,S^{-1}} = 1]| = \frac{q}{2^s} - \frac{q}{2^{2s}} \approx \frac{q}{2^s}.$$

It is obvious that the advantage of the distinguisher is less than the birthday bound, and this advantage is less than Chang and Nandi's improved security bound and so that this type of differentiable attack fails. The result can be extended to other 19 collision resistant PGV schemes. For any one of 20 collision resistant PGV schemes, the following simulator can be built such that the advantage of any distinguisher is in Chang and Nandi's improved bound.

Simulator:

1. For the i-th query $(+, k_i, p_i)$ on S where $k_i, p_i \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i\}$, h_{i-1} and m_i can be deduced from (k_i, p_i):
   (a) If $\exists h_{i-1} \xrightarrow{m_i} h_i \in \mathcal{R}_{i-1}$, then this is a repetition query, deduces c_i from (h_{i-1}, h_i, m_i), S returns c_i.
   (b) Else if $\exists IV \xrightarrow{M'} h_{i-1} \in \mathcal{R}^*_{i-1}$ and $g(M) = M' \parallel m_i$, S runs $\mathcal{F}(M)$ and obtains the response h_i, randomly choose a s-bit string h', updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} (h_i \parallel h')\}$, then deduces c_i from $\{h_{i-1}, m_i, (h_i \parallel h'), v\}$ and returns c_i;
   (c) Else S randomly selects a hash value $h_i \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then deduces c_i from $\{h_{i-1}, m_i, h_i, v\}$ and returns c_i.
2. For the i-th query $(-, k_i, c_i)$ on S where $k_i \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i\}$:
   (a) If $\exists h_{i-1} \xrightarrow{m_i} h_i \in \mathcal{R}_{i-1}$ where k_i, c_i can be deduced from (h_{i-1}, m_i, h_i) , then this is a repetition query, S deduces p_i from (h_{i-1}, m_i, h_i), then returns the p_i.
   (b) Else S randomly selects a message $h_{i-1} \in \{0,1\}^n$, deduces m_i, h_i from $\{h_{i-1}, k_i, c_i\}$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then returns h_{i-1}.

For anyone of the 20 collision PGV schemes, we can calculate the advantage of any distinguisher using the method explained in [8]. So combined our analysis of PGV schemes and Chang and Nandi's improved bound. We get the following theorem:

Theorem 9 *The chop-MD$_s^E$ construction based on anyone of 20 collision resistant collision PGV schemes is $(t_D, t_S, q, \sigma, \epsilon)$ indifferentiable from a random oracle, in the ideal cipher model for any t_D, with $t_S = l \cdot O(q^2)$ and $\epsilon = O(\frac{nq}{2^s} + \frac{q}{2^{n-s}} + \frac{\sigma^2}{2^n})$, where q is the total number of queries and σ is the total number of message blocks queried.*

The above theorem shows that the distinguisher needs at least $2^s/(3s+1)$ query complexity to have an indifferentiability attack when $n = 2s$. In [8], the result implies the chop-MD hash function is almost optimally secure with respect to second preimage and multicollision attack. Note that it doesn't improve the security bound for resisting collisions to chop-MD, but does improve the bound for indifferentiability in the ideal cipher model.

6 Conclusion

The indifferentiability of 20 collision resistant PGV hash functions for pf-MD, NMAC/HMAC and chop-MD construction are revisited. It is shown that the indifferentiability is really a method to verify the security of a construction. There are some schemes can be differentiable from random oracle with pf-MD, but are indifferentiable from random oracle with NMAC/HMAC and chop-MD construction. Our results exploit that the four Merkle-Damgård variants are not the same in the sense of the indifferentiability. And the later two constructions are better than pf-MD. Since the pf-MD construction has lower input domain and the chop-MD construction has lower output range, the NMAC/HMAC would be a better choice for practice use. We also suggest that one should take care of the proof of the indifferentiability of a construction, since some flaws have been found in previous works.

References

[1] Andreeva, E., Neven, G., Preneel, B., Shrimpton, T.: Seven-property-preserving hashing: ROX. In: Kurosawa, K. (ed.) *ASIACRYPT'2007*. LNCS 4833, pp. 130-146. Springer, 2007.

[2] Bellare, M., Ristenpart, T.: Multi-property-preserving hash domain extension: The EMD transform. In: Lai, X., Chen, K. (eds.) *ASIACRYPT'2006*. LNCS 4284, pp. 299-314. Springer, 2006.

[3] M. Bellare and T. Ristenpart. Hash Functions in the Dedicated-key Setting: Design Choices and MPP Transforms. In: *ICALP'07*, LNCS 4596, pp. 339-410. Springer, 2007.

[4] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche: On the indifferentiability of the sponge construction. In: Smart, N. (ed.) *EUROCRYPT'2008*. LNCS 4965, pp. 181-197. 2008.

[5] J. Black, P. Rogaway, and T. Shrimpton. Black-box analysis of the blockcipher- based hash function constructions from PGV. In *Crypto2002*, LNCS 2442, pp. 320-335. Springer, 2002.

[6] 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.

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

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

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

[10] J. S. Coron, Y. Dodis, C. Malinaud and P. Puniya. Merkle-Damgard Revisited: How to Construct a Hash Function (Full Version). In *http://people.csail.mit.edu/dodis/ps/merkle.ps.* 2007. A preliminary version was accepted by *Crypto'05*, LNCS 3621, pp. 21-39. 2005.

[11] J. S. Coron, J. Patarin, and Y. Seurin. The random oracle model and the ideal cipher model are equivalent. In D. Wagner(ed.), *CRYPTO'2008*, LNCS 5157, pp. 1-20. Springer, 2008.

[12] I. Damgard. A Design Principle for Hash Functions, In:*Cyrpto'89*, LNCS 435, pp. 416-427. Springer, 1989.

[13] Y. Dodis, L. Reyzin, R. L. Rivest and E. Shen. Indifferentiability of Permutation-Based Compression Functions and Tree-Based Modes of Operation, with Applications to MD6. *FSE'09*, Appear soon.

[14] Y. Dodis, T. Ristenpart, and T. Shrimpton. Salvaging Merkle-Damgard for Practical Applications. In: *Euro-Crypt'09*, LNCS 5479, pp. 371-388. Springer.

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

[16] S. Hirose. Some Plausible Constructions of Double-Length Hash Functions. In: *FSE'06*, LNCS 4047, pp. 210-225. Springer, 2006.

[17] S. Hirose, J. Park, and A. Yun. A Simple Variant of the Merkle-Damgard Scheme with a Permutation. In: *ASIACRYPT'07*, LNCS vol. 4833, pp. 113-129. Springer, 2007.

[18] H. Kuwakado , M. Morii: Indifferentiability of single-block-length and rate-1 compression functions. IEICE Trans Fundamentals, vol.e90-A, pp. 2301-2308. 2007.

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

[20] U. Maurer and S. Tessaro. Domain Extension of Public Random Functions: Beyond the Birthday Barrier. In: Menezes, A. (ed.) *CRYPTO'2007*. LNCS 4622, pp. 187-204. Springer, 2007

[21] R.C. Merkle. One way hash functions and DES, In: *Crypto'89*, LNCS 435, pp. 428-446. Springer, 1989.

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

[23] X. Wang, Y. Yin and H. Yu. Finding Collision in the Full SHA-1. In: *CRYPTO'05*, LNCS 3621, pp. 17-36. Springer, 2005.

[24] X. Wang and H. Yu. How to Break MD5 and Other Hash Functions. In: *EUROCRYPT'05*, LNCS 3494, pp. 19-35. Springer, 2005.

A Previous Simulators of pf-MD and NMAC

Coron *et al.*'s and Chang *et al.*'s simulators of pf-MD and NMAC based on Davies-Meyer structure are described as follows:

Coron *et al.*'s Simulation of pf-MD.

The simulator S accepts either forward ideal cipher queries, $(+, k_i, p_i)$, or inverse ideal cipher queries, $(-, k_i, c_i)$, such that $k_i \in \{0,1\}^n$ and $p_i, c_i \in \{0,1\}^n$. In either case, the simulator S responses with a n-bit string that is interpreted as $E_{k_i}(p_i)$ in the case of a forward query $(+, k_i, p_i)$ and as $E_{k_i}^{-1}(c_i)$ in the case of an inverse query. The simulator keeps the relations $(\mathcal{R}_1, \ldots, \mathcal{R}_{i-1})$. To answer the distinguisher D's forward and inverse queries, the simulator S responses as follows.

1. For the i-th query $(+, k_i, p_i)$ on S where $k_i = m_i$ and $p_i = h_{i-1}$:
   (a) If $\exists h_{i-1} \xrightarrow{m_i} h_i \in \mathcal{R}_{i-1}$, then this is a repetition query which the response is already known. S returns $c_i = h_i \oplus h_{i-1}$.
   (b) Else if $\exists IV \xrightarrow{M'} h_{i-1} \in \mathcal{R}^*_{i-1}$ and $g(M) = M' \parallel m_i$, S runs $\mathcal{F}(M)$ and obtains the response h_i, updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then returns $c_i = h_i \oplus h_{i-1}$;
   (c) Else S randomly selects a hash value $h_i \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then returns $c_i = h_i \oplus h_{i-1}$.
2. For the i-th query $(-, k_i, c_i)$ on S where $k_i = m_i$:
   (a) If $\exists h_{j-1} \xrightarrow{m_i} (h_{j-1} \oplus c_i) \in \mathcal{R}_{i-1}$ for $j < i$, then this is a repetition query. S returns h_{j-1}.
   (b) Else S randomly selects a message $h'_{i-1} \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h'_{i-1} \xrightarrow{m_i} c_i \oplus h'_{i-1}\}$, then returns h'_{i-1}.

Chang *et al.*'s Simulation of pf-MD

Generally speaking, Chang *et al.*'s simulator is the same as Coron *et al.*'s except for the inverse query. To answer the distinguisher D's forward and inverse queries, the simulator S responses as follows.

1. For the i-th query $(+, k_i, p_i)$ on S where $k_i = m_i$ and $p_i = h_{i-1}$: S behaves the same as Coron *et al.*'s simulator.

2. For the i-th query $(-, k_i, c_i)$ on S where $k_i = m_i$:

   (a) If $\exists h_{j-1} \xrightarrow{m_i} (h_{j-1} \oplus c_i) \in \mathcal{R}_{i-1}$ for $j < i$, this is a repetition query. S returns h_{j-1}.

   (b) Else for each $IV \xrightarrow{M'} h_{i-1} \in \mathcal{R}_{i-1}$ and $g(M) = M' \parallel m_i$, run $\mathcal{F}(M) = h_i$. If $h_i \oplus h_{i-1} = c_i$, return h_{i-1} and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$

   (c) Else S randomly selects a message $h'_{i-1} \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h'_{i-1} \xrightarrow{m_i} c_i \oplus h'_{i-1}\}$, then returns h'_{i-1}.

Coron *et al.*'s Simulation of NMAC.

The NMAC construction NMAC^{E1,E2} essentially applies the Davies-Meyer construction using the block cipher $E1$ to the input $m_1 \parallel \ldots \parallel m_l$ to get the final output h_l. It then applies another independent the Davies-Meyer construction using $E2$ to this output h_l. For simplicity the output length n of $E1$ is the same as the key length of $E2$. And one use IV_1 for the Davies-Meyer construction applied to $E1$, and use IV_2 for the Davies-Meyer construction with $E2$.

The simulator gets forward/inverse queries for either of the block ciphers $E1$ and $E2$. Thus the queries that simulator S responds to are as follows:

1. $(1, +, k_i, p_i)$: A forwards $E1$ query ,where $(k_i, p_i) \in \{0,1\}^n \times \{0,1\}^n$. The expected response is $E1_{k_i}(p_i)$.
2. $(1, -, k_i, c_i)$: A inverses $E1$ query ,where $(k_i, c_i) \in \{0,1\}^n \times \{0,1\}^n$. The expected response is $E1^{-1}_{k_i}(c_i)$.
3. $(2, +, k_i, p_i)$: A forwards $E2$ query ,where $(k_i, p_i) \in \{0,1\}^n \times \{0,1\}^n$. The expected response is $E2_{k_i}(p_i)$.
4. $(2, -, k_i, c_i)$: A inverses $E2$ query ,where $(k_i, c_i) \in \{0,1\}^n \times \{0,1\}^n$. The expected response is $E2^{-1}_{k_i}(c_i)$.

The simulator S also maintains the relations $(\mathcal{R}_1, \ldots, \mathcal{R}_{i-1})$ and $(\mathcal{Q}_1, \ldots, \mathcal{Q}_{j-1})$ where $(\mathcal{R}_1, \ldots, \mathcal{R}_{i-1})$ records the triples that obtained from queries on $E1$ and $(\mathcal{Q}_1, \ldots, \mathcal{Q}_{j-1})$ records the triples that obtained from queries on $E2$. To answer the distinguisher D's forward and inverse queries on $E1$ or $E2$, the simulator S should simulate $E1, E2$ as $S1, S2$ and responses as follows.

- **Query on** $S1$:

  1. For the i-th query $(1, +, k_i, p_i)$ on $S1$ where $k_i = m_i$ and $p_i = h_{i-1}$:
     (a) If $\exists h_{i-1} \xrightarrow{m_i} h_i \in \mathcal{R}_{i-1}$, then this is a repetition query. S returns $c_i = h_i \oplus h_{i-1}$.
     (b) Else S randomly selects a hash value $h_i \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then returns $c_i = h_i \oplus h_{i-1}$.
  2. For the i-th query $(1, -, k_i, c_i)$ on $S1$ where $k_i = m_i$:
     (a) If $\exists h_{j-1} \xrightarrow{m_i} (h_{j-1} \oplus c_i) \in \mathcal{R}_{i-1}$ where $j < i$, S returns h_{j-1}.
     (b) Else S randomly selects a message $h'_{i-1} \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h'_{i-1} \xrightarrow{m_i} c_i \oplus h'_{i-1}\}$, then returns h'_{i-1}.

- **Query on** $S2$:

  1. For the j-th query $(2, +, k_j, p_j)$ on $S2$ where $k_j = m_j$ and $p_j = h_{j-1}$:
     (a) If $\exists h_{j-1} \xrightarrow{m_j} h_j \in \mathcal{Q}_{j-1}$, then this is a repetition query. $S2$ returns $c_j = h_j \oplus h_{j-1}$.
     (b) Else if $\exists IV_1 \xrightarrow{M'} m_j \in \mathcal{R}^*_i$ and $p_j = IV_2$, S runs $\mathcal{F}(M' \parallel m_j)$ and obtains the response h_j, updates $\mathcal{Q}_j = \mathcal{Q}_{j-1} \cup \{IV_2 \xrightarrow{m_j} h_j\}$, then returns $c_j = IV_2 \oplus h_j$.
     (c) Else S randomly selects a hash value $h_j \in \{0,1\}^n$ and updates $\mathcal{Q}_j = \mathcal{Q}_{i-1} \cup \{h_{j-1} \xrightarrow{m_j} h_j\}$, then returns $c_j = h_j \oplus h_{j-1}$.

2. For the j-th query $(2, -, k_j, c_j)$ on $S2$ where $k_j = m_j$:
   (a) If $\exists h_{k-1} \xrightarrow{m_j} (h_{k-1} \oplus c_j) \in \mathcal{Q}_{j-1}$ where $k < j$, S returns h_{k-1}.
   (b) Else S randomly selects a message $h'_{j-1} \in \{0,1\}^n$ and updates $\mathcal{Q}_j = \mathcal{Q}_{j-1} \cup \{h'_{j-1} \xrightarrow{m_j} c_j \oplus h'_{j-1}\}$, then returns h'_{j-1}.

B Proofs

B.1 Proof of Theorem 4

The distinguisher D accesses to oracles $(\mathcal{O}_1, \mathcal{O}_2)$ where $(\mathcal{O}_1, \mathcal{O}_2)$ is (H, E) or $(\mathcal{F}, S)$. If it is easy to find a collision (M, M') such that $H(M) = H(M')$ when makes queries to E, D can query M and M' to $\mathcal{O}_1$ and receive the responses. If the responses are different, then D is interacting with $(\mathcal{F}, S)$, otherwise it is interacting with (H, E). Then we have

$$Adv(D) = |Pr[D^{H,E} = 1] - Pr[D^{\mathcal{F},S} = 1]| = 1 - 2^{-n}.$$

Since the advantage is non-negligible, so the construction is differentiable from a random oracle. □

B.2 Proof of Theorem 5

If a PGV scheme satisfies $(h_i, m_i) \Rightarrow h_{i-1}$, then we know the key k_i to the block cipher E must be a linear combination of $\{m_i, v\}$ and c_i is a linear combination of $\{h_i, m_i, v\}$, here v is a constant, then we can build the following distinguisher D such that any simulator fails.

Distinguisher D can access to oracles $(\mathcal{O}_1, \mathcal{O}_2)$ where $(\mathcal{O}_1, \mathcal{O}_2)$ is (H, E) or $(\mathcal{F}, S)$.

1. D selects a message M, M' such that $g(M) = (m_1 \parallel m_2)$ and $g(M') = (m_1 \parallel m'_2)$ where $m_2 \neq m'_2$ and $|m_1| = |m_2| = |m'_2| = n$, then makes the query M to $\mathcal{O}_1$ and receives h_2 and the query M' to $\mathcal{O}_1$ and receives h'_2.
2. D computes (k_2, c_2) from (m_2, h_2) and (k'_2, c'_2) from (m'_2, h'_2), then makes an inverse query $(-, k_2, c_2)$ to $\mathcal{O}_2$ and receives p_2 and computes h_1 from (m_2, k_2, h_2, p_2) , then makes an inverse query $(-, k'_2, c'_2)$ to $\mathcal{O}_2$ and receives p'_2 and computes h'_1 from (m'_2, k'_2, h'_2, p'_2).
3. If $h_1 = h'_1$ output 1, otherwise output 0.

Since the simulator doesn't know whether the two inverse queries lead to a same internal value, the simulator S can output the right response only with probability 2^{-n},

$$Adv(D) = |Pr[D^{H,E,E^{-1}} = 1] - Pr[D^{\mathcal{F},S,S^{-1}} = 1]| = 1 - 2^{-n}$$

This is not negligible. So the construction is differentiable from a random oracle. □

B.3 Proof of Theorem 6

In this case, the following distinguisher is built.

Distinguisher D can access to oracles $(\mathcal{O}_1, \mathcal{O}_2)$ where $(\mathcal{O}_1, \mathcal{O}_2)$ is (H, E) or $(\mathcal{F}, S)$.

1. D selects a message M such that $g(M) = m$ where $|m| = n$, then makes the query M to $\mathcal{O}_1$ and receives h.
2. D computes (k, c) from (h, m, IV, v), then makes an inverse query $(-, k, c)$ to $\mathcal{O}_2$ and receives p, then computes m' from (IV, k, c, p).
3. If $m = m'$ output 1, otherwise output 0.

Since the simulator never knows the right message m, it gives the right response only with probability 2^{-n},

$$Adv(D) = |Pr[D^{H,E,E^{-1}} = 1] - Pr[D^{\mathcal{F},S,S^{-1}} = 1]| = 1 - 2^{-n}.$$

So the construction is differentiable from a random oracle. □

B.4 Proof of Theorem 7

The Davies-Meyer construction(case 5) has been shown to be indifferentiable from random oracle with pf-MD. For the other 11 cases, we can make similar analysis. Thus, we can define a general simulator for these 12 PGV functions. The simulator is defined as follows:

Simulator:

1. For the i-th query $(+, k_i, p_i)$ on S where $k_i, p_i \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i\}$, we can deduce h_{i-1} and m_i from (k_i, p_i):
   (a) If $\exists h_{i-1} \xrightarrow{m_i} h_i \in \mathcal{R}_{i-1}$, then this is a repetition query. S deduces c_i from $\{h_{i-1}, m_i, h_i\}$ and returns c_i.
   (b) Else if $\exists IV \xrightarrow{M'} h_{i-1} \in \mathcal{R}^*_{i-1}$ and $g(M) = M' \parallel m_i$, S runs $\mathcal{F}(M)$ and obtains the response h_i, updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then deduces c_i from $\{h_{i-1}, m_i, h_i, v\}$ and returns c_i;
   (c) Else S randomly selects a hash value $h_i \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then deduce c_i from $\{h_{i-1}, m_i, h_i, v\}$ and returns c_i.
2. For the i-th query $(-, k_i, c_i)$ on S where $k_i \in \{h_{i-1}, m_i, h_{i-1} \oplus m_i\}$:
   (a) For each M' such that $IV \xrightarrow{M'} h_{i-1} \in \mathcal{R}^*_{i-1}$($M'$ can be the empty string, in that case, $h_{i-1} = IV$), deduce m_i from $\{h_{i-1}, k_i\}$. If $\exists M$ such that $g(M) = M' \parallel m_i$, runs $\mathcal{F}(M)$ and obtains the response h'_i. At the same time, we can deduce h_i from $\{h_{i-1}, m_i, c_i\}$ for each PGV scheme.
   (b) If $h_i = h'_i$, S returns the corresponding plaintext which belongs to $\{h_{i-1}, m_i, h_{i-1} \oplus m_i\}$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$.
   (c) Else S randomly selects a message $h_{i-1} \in \{0,1\}^n$, deduce m_i, h_i from $\{h_{i-1}, k_i, c_i\}$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then returns h_{i-1}.

By using Theorem 4 and Theorem 5 in [7], or Theorem 4.1 in [10], we can compute $t_S = l \cdot O(q^2)$ and $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$, where l is the maximum length of a query made by the distinguisher D. □

B.5 Proof of Lemma 1

The NMAC^{E_1,E_2} applies this compression function using the block cipher $E1$ to the input $m_1 \parallel \ldots \parallel m_l$ to get the final output h_l, then applies another independent compression function using $E2$ to this output h_l. We can build the following simulator:

Simulator:

- **Query on** $S1$:

  1. For the i-th query $(1, +, k_i, p_i)$ on $S1$ where $k_i = m_i$ and $p_i = h_{i-1}$:
     (a) If $\exists h_{i-1} \xrightarrow{m_i} h_i \in \mathcal{R}_{i-1}$, then this is a repetition query. S returns h_i.
     (b) Else S randomly selects a hash value $h_i \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h_{i-1} \xrightarrow{m_i} h_i\}$, then returns h_i.
  2. For the i-th query $(1, -, k_i, c_i)$ on $S1$ where $k_i = m_i$:
     (a) If $\exists h_{j-1} \xrightarrow{m_i} (c_i) \in \mathcal{R}_{i-1}$ where $j < i$, S returns h_{j-1}.
     (b) Else S randomly selects a message $h'_{i-1} \in \{0,1\}^n$ and updates $\mathcal{R}_i = \mathcal{R}_{i-1} \cup \{h'_{i-1} \xrightarrow{m_i} c_i\}$, then returns h'_{i-1}.

- **Query on** $S2$:

  1. For the j-th query $(2, +, k_j, p_j)$ on $S2$ where $k_j = m_j$ and $p_j = h_{j-1}$:
     (a) If $\exists h_{j-1} \xrightarrow{m_j} h_j \in \mathcal{Q}_{j-1}$, then this is a repetition query. $S2$ returns $c_j = h_j$.
     (b) Else if $\exists IV_1 \xrightarrow{M} m_j \in \mathcal{R}_i^*$ and $p_j = IV_2$, S runs $\mathcal{F}(M)$ and obtains the response h_j, updates $\mathcal{Q}_j = \mathcal{Q}_{j-1} \cup \{IV_2 \xrightarrow{m_j} h_j\}$, then returns $c_j = h_j$.
     (c) Else S randomly selects a hash value $h_j \in \{0,1\}^n$ and updates $\mathcal{Q}_j = \mathcal{Q}_{i-1} \cup \{h_{j-1} \xrightarrow{m_j} h_j\}$, then returns $c_j = h_j$.
  2. For the j-th query $(2, -, k_j, c_j)$ on $S2$ where $k_j = m_j$:
     (a) If $\exists h_{k-1} \xrightarrow{m_j} (c_j) \in \mathcal{Q}_{j-1}$ where $k < j$, this is a repetition query, S returns h_{k-1}.
     (b) Else If $\exists IV_1 \xrightarrow{M} (k_j) \in \mathcal{R}_i^*$ then S runs $\mathcal{F}(M)$ and gets h. If $h = c_j$, S updates $\mathcal{Q}_j = \mathcal{Q}_{j-1} \cup \{IV_2 \xrightarrow{m_j} h\}$, then returns IV_2.
     (c) Else S randomly selects a message $h'_{j-1} \in \{0,1\}^n$ and updates $\mathcal{Q}_j = \mathcal{Q}_{j-1} \cup \{h'_{j-1} \xrightarrow{m_j} c_j\}$, then returns h'_{j-1}.

It is easy to show the distinguisher which was succeeding in the pf-MD will fail in the NMAC construction. $t_S = l \cdot O(q^2)$ and $\epsilon = 2^{-n} \cdot l^2 \cdot O(q^2)$ are calculated according the proof of Lemma A.8 in [10] , where l is the maximum length of a query made by the distinguisher D. □

C Tables

Case	*PGV*	*Case*	*PGV*	*Case*	*PGV*
1	$E_{h_{i-1}}(m_i) \oplus m_i$	4	$E_{h_{i-1}}(w_i) \oplus m_i$	19	$E_{m_i}(w_i) \oplus v$
2	$E_{h_{i-1}}(w_i) \oplus w_i$	15	$E_{m_i}(h_{i-1}) \oplus v$	20	$E_{m_i}(w_i) \oplus m_i$
3	$E_{h_{i-1}}(m_i) \oplus w_i$	17	$E_{m_i}(h_{i-1}) \oplus m_i$		

Table C.1 Eight differentiable PGV schemes with pf-MD. $w_i = h_{i-1} \oplus m_i$, v is a constant.

Case	*PGV*	*Case*	*PGV*	*Case*	*PGV*
5	$E_{m_i}(h_{i-1}) \oplus h_i$	9	$E_{w_i}(m_i) \oplus m_i$	13	$E_{w_i}(m_i) \oplus v$
6	$E_{m_i}(w_i) \oplus w_i$	10	$E_{w_i}(h_{i-1}) \oplus h_{i-1}$	14	$E_{w_i}(m_i) \oplus w_i$
7	$E_{m_i}(h_{i-1}) \oplus w_i$	11	$E_{w_i}(m_i) \oplus h_{i-1}$	16	$E_{w_i}(h_{i-1}) \oplus v$
8	$E_{m_i}(w_i) \oplus h_{i-1}$	12	$E_{w_i}(h_{i-1}) \oplus m_i$	18	$E_{w_i}(h_{i-1}) \oplus w_i$

Table C.2 Twelve Indifferentiable PGV schemes with pf-MD. $w_i = h_{i-1} \oplus m_i$, v is a constant.