Designs of Efficient Secure Large Hash Values

Mridul Nandi
Applied Statistics Unit
Indian Statistical Institute, Kolkata, India
mridul_r@isical.ac.in

April 20, 2005

Abstract

A double length hash function is a $2n$-bit hash function based on an n-bit compression function. To increase the security level, designs of good double length hash functions are important. In this paper we construct a class of maximally secure double length hash functions in random oracle model based on some good permutations. This class contains recently proposed double length hash functions [16, 25]. We also propose an efficient double length hash function and study its security level in the random oracle model. We prove that any attack algorithm in the random oracle model needs $\Omega(2^n/(s^2n^s))$ time complexity, where s is some parameter related to the rate of the hash function. Thus there is a trade-off between the efficiency and security. We use the notion of computable message [**?**] to make the security analysis of proposed hash functions. We also see that the security analysis of hash functions based on random permutations and hash functions based on random functions are very much related.

1 Introduction

A *hash function* is an easily computable function from the set of all finite binary strings, $\{0,1\}^*$, to a set of binary strings of some fixed length, $\{0,1\}^n$. Hash functions have been popularly used in digital signatures schemes [8, 14], public key encryption schemes [6, 38] message authentication codes [22] etc. To construct secure digital signature schemes, public key encryption schemes etc., people use collision resistant hash functions or preimage resistant hash functions. Intuitively, a collision resistant hash function is a hash function, $H(\cdot)$, where it is hard to find two different inputs $X \neq Y$ (known as a *collision pair*) such that $H(X) = H(Y)$. In case of preimage resistant hash function, given a random image it is hard to find an inverse of that image.

1.1 Design of a Hash Function

Usually, a hash function is designed in two steps. We first design a small domain compression function and then we extend the domain into the arbitrary domain, $\{0,1\}^*$.

1. In the first step a compression function, $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$, is either defined from scratch or based on other primitives, e.g. Block Cipher. The most popular hash functions from scratch are SHA family (SHA-1, SHA-256 [31, 32]), MD-family (MD5, RIPEMD [12, 34, 35]), TIGER [1] etc. B. Preneel, R. Govaerts, and J. Vandewalle [33] classified block cipher based compression functions. Let $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher then a

compression function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ (here, $m = n$) can be defined in the following way :

$$f(h, x) = E_a(b) \oplus c, \text{ where } a, b, c \in \{h, x, h \oplus x, v\},$$

Here, v is a fixed n-bit string and $|h| = |x| = n$. These sixty-four compression functions are known as PGV-compression functions. Some of the PGV compression functions were proposed earlier. For example, Davis-Meyer [?] compression function, $E_x(h) \oplus h$, Miyaguchi [?] and Preneel [?] compression function, $E_x(h) \oplus x \oplus h$.

2. The second step is a method of domain extension of compression functions. In the classical method i.e. MD-method (Merkle-Damgård) [8, 28] the input message, M, is padded unambiguously. The binary representation of length of the message is also padded. The padding rule makes the input size multiple of m and at the same time, it rules out some trivial attack. Let the padded message be $M' = m_1 || \cdots || m_l$, where $|m_i| = m$, $1 \le i \le l$ and let h_0 be a fixed initial value. Define the hash output of the message M as follows;

$$H(M) = h_l, \text{ where } h_i = f(h_{i-1}, m_i),\ 1 \le i \le l.$$

1.2 Motivation and Our Contribution

The collision attack and the preimage attack are the most popular types of attack for a hash function. The birthday attack requires $O(2^{n/2})$ or $O(2^n)$ complexity to find a collision or to find a preimage respectively on an n-bit hash function. The complexity means the number of computations of the underlying compression function f. The birthday attack on the compression function based on a block cipher can be practically feasible, since they have small output size (eg., $n = 64$ or 128). Thus, in this paper we are interested in the following problem.

Problem : Given a compression function, $f : \{0,1\}^{n+m} \to \{0,1\}^n$ (or s compression functions $f_1, \cdots, f_s : \{0,1\}^{n+m} \to \{0,1\}^n$), how to design a compression function $F : \{0,1\}^N \to \{0,1\}^{2n}$, where $N > 2n$ and a hash function $H : \{0,1\}^* \to \{0,1\}^{2n}$.

The compression function, $F : \{0,1\}^N \to \{0,1\}^{2n}$, is termed as a *double length* compression function and the hash function $H : \{0,1\}^* \to \{0,1\}^{2n}$ is known as a *double length* hash function. Designing a secure double length compression function would be sufficient to construct a secure double length hash function. The classical hash function is as secure as the underlying compression function. The most natural and efficient construction of a double length hash function is the concatenated hash function $H||G$, where H and G are two classical n bit hash functions based on the compression function, $f(\cdot)$, with two different initial values. H and G also can be based on two different compression functions f_1 and f_2. This concatenated hash functions had been popularly used in many industries. Recently, A. Joux [17] showed that there is a collision attack on the concatenated hash function in time complexity $O(n2^{n/2})$. There were several attempts to construct a secure block cipher based double length compression functions. Most of these have several attacks much better than the birthday attack [15, 20, 19, 25, 16, 37].

In this paper we design several new double length hash functions and compute their security level and the rate. Our first design is a generalization of Lucks's [25] and Hirose's [16] construction. Given a permutations $p(\cdot)$ on the set of all N-bit strings and a compression function $f : \{0,1\}^N \to \{0,1\}^n$, define $F(X) = f(X)||f(p(X))$. We show that the double length function F is maximally secure provided the permutation p does not have any fixed point (see Sect. **??**). Thus, we have a class of maximally secure double length hash functions. Next, we designed an efficient double length hash function. This construction is very much similar to the concatenated hash function except the mixing up the intermediate hash values. We show the collision security (or preimage security) level of the double length hash function $\Omega(2^{2n/3})$ (or $\Omega(2^{4n/3})$).

Organization of the paper : We first give some basic results and definitions. We define the random oracle model and black box model. We define a rate function or rate which is a measurement of efficiency of a hash function. This definition differ from the definition taken in [37, 19]. But all these definitions are equivalent. We also state some recent works. Next we state our new class of double length hash function and study the security level in the random oracle model. We also describe an efficient double length hash function and study it's security property. Finally, we give some idea of future works and conclude.

2 Preliminaries and Related Works

In this section we give a brief introduction of random function and random permutations. We also state the behavior or an adversary in the random oracle model or in the black-box model. We also illustrate some related and recent works in designing double length hash or compression functions.

2.1 (Independent) Random Functions and Permutations.

Random Function. A random function $f : D \to R$ taking values as a random variable such that for any $x \in D$, $f(x)$ has uniform distribution on R and for any $k > 0$ and k distinct elements $x_1, \cdots x_k \in D$, the random variables $f(x_1), \cdots, f(x_k)$ are independently distributed.

Independent Function. We say a family of functions $f_1, \cdots, f_s : D \to R$ are independent if for any s subsets $\{x_1^1, \cdots, x_{k_1}^1\}, \cdots, \{x_1^s, \cdots, x_{k_s}^s\}$, the random vectors $(f_1(x_1^1), \cdots, f_1(x_{k_1}^1)), \cdots, (f_s(x_1^s), \cdots, f_s(x_{k_s}^s))$ are independently distributed. We say $f_1, f_2, \cdots f_s : D \to R$ are independent random functions if they are random functions and independent too.

(Independent) Random Permutation. A permutation $E : D \to D$ is said to be a random permutation if for any $k > 0$ and k distinct elements $x_1, \cdots, x_k \in D$, the random variable $f(x_k)$ condition on $f(x_1) = y_1, \cdots, f(x_{k-1}) = y_{k-1}$ is uniformly distributed over the set $D - \{y_1, \cdots, y_{k-1}\}$. Obviously $f(x_1), \cdots, f(x_k)$ are not independently distributed. We say a family of permutations $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a random permutation if for each $K \in \{0,1\}^k$, $E(K, \cdot)$ is a random permutation and for each $s > 0$, and s distinct elements $K_1, \cdots K_s$, $E(K_1, \cdot), \cdots, E(K_s, \cdot)$ are independent function.

Proposition 1 *If $f : \{0,1\}^{n+k} \to \{0,1\}^m$ is a random function then the family of functions $\{f_s\}_{s \in \{0,1\}^k}$, $f_s : \{0,1\}^n \to \{0,1\}^m$ defined by $f_s(x) = f(x||s)$, where $|x| = n$, are independent random functions. In particular, if $f : \{0,1\}^N \to \{0,1\}^n$ is a random function then $f_0, f_1 : \{0,1\}^{N-1} \to \{0,1\}^n$, $f_i(X) = f(i||X)$, $|X| = N - 1$ and $i = 0, 1$ are two independent random functions.*

2.2 The model of the adversary

In this paper mainly we assume the underlying compression function is a random function unless it is stated clearly. If the compression function is based on a block-cipher then we assume the block cipher is a random permutations. In these model the adversary plays in some particular ways.

When we assume that the compression function, $f(\cdot)$, is a random function then the adversary made several queries to know the output values of f. Thus he choose $x_1, \cdots x_q$ adaptively and got responses $y_1, \cdots, y_q$, where $y_i = f(x_i)$. We can think that y_i as a realization of the random variable $f(x_i)$ which is observed by the adversary. Define the complete list of query-response pairs $((x_1, y_1), \cdots, (x_q, y_q))$ by the *view* of the adversary. Any output produced by the adversary should only depend on the view. Moreover, if the adversary is finding collision for a hash function, $H(\cdot)$, based on the compression function, $f(\cdot)$, and it outputs a pair of distinct messages $M \neq N$ then the values of $H(M)$ and $H(N)$ should be computed from the view. When we have two or more compression functions $f_1, f_2, \cdots$ we have a set of lists of pairs $\{((x_1^1, y_1^1), \cdots, (x_q^1, y_q^1)), ((x_1^2, y_1^2), \cdots, (x_q^2, y_q^2)), \cdots\}$ where the first member is the view due to the random compression function f_1 and so on. This set will be called as a view of the adversary.

In case of block cipher based construction, an adversary has access to oracles E and E^{-1}. For E-query he gives (a, x) and got response y such that $E_a(x) = y$. Similarly for E^{-1} query. Here, the list of triples $((a_1, x_1, y_1), \cdots, (a_q, x_q, y_q))$ will be called as a *view* of the adversary. Again we follow the similar conventions. Firstly, an adversary does not ask any oracle query in which the response is already known. Secondly, if M is one of the output(s) produced by an adversary, then the adversary should make necessary E/E^{-1} queries to compute $H(M)$ during the whole query process. Note that these conventions are important to make the discussion easy and to prove the security. These assumptions are meaningful as any adversary $\mathcal{A}$ not obeying these conventions can easily be modified to obtain an adversary $\mathcal{A}'$ having similar computational complexity that obeys these conventions and has the same advantage as $\mathcal{A}$.

2.3 Recent Works on Double Length Hash Functions

The classical iteration is the most popular method to construct a single length hash function from a compression function. To design a double length hash function one can use the simple method of concatenation of two independent classical hash functions. Let $H^f(IV, \cdot)$ be the classical hash function with the initial value IV and the compression function f. The simplest method to design a double length hash function is $H^{f_1}(IV_1, M) \mid\mid H^{f_2}(IV_2, M)$. Recently, A. Joux [17] observed that the concatenated hash function is not secure. For any two classical hash functions H and G, $H||G$ has collision attack in time $\mathrm{O}(n2^{n/2})$ (see [17] for more detail). He showed a $2^{n/2}$-way collision on H can be found in time complexity $\mathrm{O}(n.2^{n/2})$. A K-way collision is a K-set $\{M_1.\cdots, M_K\}$ such that $H(M_1) = \cdots = H(M_K)$. Now, find $M \neq N$ (by birthday paradox) from that multicollision set such that $G(M) = G(N)$ and thus we have $H(M)||G(M) = H(N)||G(N)$.

1. Given a compression function $f : \{0,1\}^N \to \{0,1\}^n$ with $N > 2n+1$, S. Hirose [16] defined a compression function $F : \{0,1\}^{N-1} \to \{0,1\}^{2n}$, where $F(X) = f_0(X)||f_1(X)$ and $|X| = N-1$. The functions f_0 and f_1 are defined like in the section 3.3. In his paper, he used block cipher $E : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^n$, to construct a secure compression function $f : \{0,1\}^{3n} \to \{0,1\}^n$. One of the example is $f(x, y, z) = E_{x||y}(z) \oplus z$, where $|x| = |y| = |z| = n$.

2. In [25], a compression function, $F : \{0,1\}^N \to \{0,1\}^{2n}$ had been defined from a secure compression function $f : \{0,1\}^N \to \{0,1\}^n$ with $N > 2n$. $F(H', H'', M) = f(H', H'', M)||f(H'', H', M)$,

where $|H'| = |H''| = n$ and $|M| = N - 2n$. It is easy to find a collision attack on this compression function with time complexity $O(2^{n/2})$. In his paper a hash function outputting n-bits was constructed based on two secure underlying compression functions. He proved that it was secure against multicollision attack. But we can prove something more. In particular, the double length hash function is secure against collision attack.

3. There were many other attempts [20, 19, 21, 37, 15] to construct an efficient double length hash function based on a block cipher. Unfortunately, most of them have collision attacks better than the birthday attack. We describe some of them later.

4. Recently, Nandi *et. al.* [] designed a double length compression function based on three independent compression functions $f_1, f_2, f_3 : \{0,1\}^{2n} \to \{0,1\}^n$. The double length compression function F on $3n$-bits is defined as follows;

$$F(x_1, x_2, x_3) = (f_1(x_1, x_2) \oplus f_2(x_2, x_3)) || (f_3(x_1, x_3) \oplus f_2(x_2, x_3)), \text{ where}$$
$$|x_1| = |x_2| = |x_3| = n.$$

   It was shown that the compression function has collision security $\Omega(2^{2n/3})$ in the random oracle model.

3 New Designs of Double Length Compression Functions

In this section, we design several double length compression functions. We study their collision and preimage security in the random oracle model of the underlying compression functions. Thus, we have underlying compression functions $f_1, f_2, \cdots f_k : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$. We design a double length compression function, $F : \{0,1\}^N \to \{0,1\}^{2n}$, based on $f_1, f_2, \cdots f_k$. We define a measurement of the efficiency of the compression function, $F(\cdot)$, called the rate function of F. Roughly, it says the number of message blocks are hashed per underlying compression function. By a message block, we mean the size of hashed message in the underlying compression functions. Thus, a message block has size m, since $f_1, f_2, \cdots f_k : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$.

Definition 2 (Rate Function)
Let a double length compression function, F, is based on $f_1, \cdots, f_k$. Define the rate function of F by $\frac{N-2n}{m \times s}$, where s is the number of invocations of all f_i's are needed to compute $F(X)$, $X \in \{0,1\}^N$.

Since F is a compression function, $N > 2n$. Thus, the rate function is always positive. When the rate function is a constant, we only use the term "rate" instead of rate function.

Example 3 *The underlying double length compression function of the concatenated hash function $H^{f_1} || H^{f_2}$ is $F(x_1, x_2, x_3) = f_1(x_1, x_3) || f_2(x_2, x_3)$, where $|x_1| = |x_2| = n$ and $|x_3| = m$. The rate function of F is* $1/2$.

Example 4 *Let $F(X) = f_1(X) || f_2(X)$ be a compression function with domain $\{0,1\}^{n+m}$. Here, the rate function is $\frac{n+m-2n}{2m} = \frac{1}{2} - \frac{n}{2m}$. Obviously, we need to assume that $m > n$. When $m = 2n$, i.e. $f_1, \cdots, f_k : \{0,1\}^{3n} \to \{0,1\}^n$, the rate of the compression function is $\frac{1}{4}$.*

Example 5 *The compression function described in [] (also in Sect. 3.3) hash rate function* $1/3$.

3.1 A Class of Double Length Compression Functions

Now, we define a class of double length compression functions. This class contains newly proposed double length compression functions in [25, 16]. Let $f : \{0,1\}^{n+m} \to \{0,1\}^n$ be a compression function with $m > n$ and $p_1, p_2 : \{0,1\}^{n+m} \to \{0,1\}^{n+m}$ are some *simple permutations*. A permutation, p, is called a *simple permutation* if the both permutations, p and p^{-1}, are easy to compute. Define a double length compression function, $f^{p_1,p_2}(X) = f(p_1(X)||f(p_2(X))$, where $|X| = n + m$. If we take $Y = p_1(X)$ and $p = p_2 \circ p_1^{-1}$, we can write $f^{p_1,p_2}(X) = f^{id,p}(Y)$ where $id(\cdot)$ is the identity permutation. Since p_1 and p_2 are simple permutations, it is enough to study the security properties of $f^{id,p}$ instead of studying f^{p_1,p_2}. We write, f^p instead of $f^{id,p}$ and we say, f^p is a double length compression function based on a permutation, p. Here, we fix a compression function, f, and define a class of double length compression functions

$$\mathcal{C} = \{f^p : p \text{ is a simple permutation on } \{0,1\}^{n+m}\}$$

All these compression function have rate $\frac{1}{2} - \frac{n}{2m}$. In this section, we study the security properties of the compression functions from the class, $\mathcal{C}$, in the random oracle model of f. We show that a double length compression function is secure, provided the permutation satisfies some conditions. We first start with the example of compression function considered in [25].

Example 6 *[25] $p(H_1, H_2, M) = H_2||H_1||M$, where $|H_1| = |H_2| = n$ and $|M| = m - n$. Thus $f^p(H_1, H_2, M) = f(H_1, H_2, M)||f(H_2, H_1, M)$.*

Collision and Preimage Attack : By using the birthday attack, find H, G and M_1, M_2, such that $(H, M_1) \neq (G, M_2)$ and $f(H, H, M_1) = f(G, G, M_2)$. Now, it is easy to check that $f^p(H, H, M_1) = f^p(G, G, M_2)$. Here, we need $\mathrm{O}(2^{n/2})$ many queries for the birthday attack. If an image $y||y$ is given to the adversary where $|y| = n$ then by using the birthday attack, find H, M such that $f(H, H, M) = y$. Now, $f^p(H, H, M) = y||y$ and complexity of this attack is $\mathrm{O}(2^n)$. The reason for having the above attacks is that the permutation p has many fixed points.

Definition 7 *X is called a fixed point of a function $p(\cdot)$, if $p(X) = X$. We write $\mathcal{F}_p$ for the set of all fixed points of p.*

In the above example, $\mathcal{F}_p = \{H||H||M : |H| = n, |M| = m - n\}$ is the set of fixed points of the permutation p and $|\mathcal{F}_p| > 2^n$. Thus, one can find a collision (or a preimage) on the compression function, f, from the fixed point set. Similarly, for any permutation, p, with $|\mathcal{F}_p| > 2^n$, we can apply the birthday attack on $\mathcal{F}_p$ for the compression function, $f(\cdot)$. Let $X \neq Y$ have been returned by the birthday attack algorithm, such that $f(X) = f(Y)$. Thus, $f^p(X) = f(X)||f(p(X)) = f(Y)||f(p(Y)) = f^p(Y)$ since $p(X) = X$ and $p(Y) = Y$. Thus, we have a collision attack with complexity $\mathrm{O}(2^{n/2})$ on all double length compression functions f^p with $|\mathcal{F}_p| > 2^n\}$. In the light of the above discussion, one should use a permutation, p, which does not have many fixed points. In fact, there are many permutations where the set of fixed points are the empty set. We give two classes of examples of that kind, in below.

Example 8 *Let A be a non-zero N-bit string. Then define a permutation $p : \{0,1\}^N \to \{0,1\}^N$ such that $p(X) = X \oplus A$. In particular, if $A = 11\cdots1$ then $p(M) = \overline{M}$, where $\overline{M}$ is the bit-wise complement of M. It is easy to check that $\mathcal{F}_p$ is the empty set.*

Example 9 *We can think any N-bit string by an integer modulo 2^N. Let $p(X) = X + A(\text{mod } 2^N)$ where $A \neq 0$. For simplicity, we also use the notation $X + A$ to denote the modulo addition $X + A(\text{mod } 2^N)$. Note that, $p(X) \neq X$ for all X. Moreover, if $A \neq 2^{N-1}$ then $p(p(X)) = X + 2A \neq X$. Thus, the set of fixed point for $p \circ p$ (in notation, p^2) is also empty.*

Suppose, f^p is a double length compression function based on a permutation, p, where $\mathcal{F}_p$ is the empty set. Then a collision, $f^p(X) = f^p(Y)$ with $X \neq Y$ implies $f(X) = f(Y)$ and $f(p(X)) = f(p(Y))$, where $X \neq Y$. Thus, $\{X, Y\}$ and $\{p(X), p(Y)\}$ are collision sets of f. Now, we have the following two cases.

- **Case-1 :** $\{X, Y\} = \{p(X), p(Y)\}$. Since p does not have any fixed point, we have $Y = p(X)$ and $X = p(Y)$. Thus, we should have a collision set $\{X, p(X)\}$, where $p(X) \neq X$ and $p(p(X)) = X$. Let $\Omega(K_1(n))$ (or in short K_1) be the complexity of the best attack for the above event.
- **Case-2 :** $\{X, Y\} \neq \{p(X), p(Y)\}$. Let $\Omega(K_2(n))$ (or in short K_2) be the complexity of the best attack.

Thus a collision on f^p reduces to the one of the above two events and hence the complexity of best collision attack is $\min\{K_1, K_2\}$. If p^2 does not have any fixed point then we can exclude the first case also. We summarize the above discussion into following proposition.

Proposition 10 *The complexity of the best collision attack on f^p is* $\min\,\{\Omega(K_1(n)), \Omega(K_2(n))\}$ *where p is a permutation with no fixed point and K_1 and K_2 are defined as above. Moreover, if the permutation, p^2, does not have any fixed point (like in the Example 2) then the best collision attack on f^p is $\Omega(K_2(n))$.*

Now we give some evidences why K_1 and K_2 would be large for a good compression function, f. Suppose an adversary tries to find two collision sets $\{X, Y\} \neq \{p(X), p(Y)\}$. After finding a collision set $\{X, Y\}$, he does not have any freedom to choose for the second collision set and he is forced to check whether $\{p(X), p(Y)\}$ is a collision set or not. Thus K_2 would be large and may be almost same to 2^n for a good underlying compression function. Next, an adversary tries to find a related collision set $\{X, p(X)\}$. After fixing one message X, $p(X)$ is completely determined (and also vice-versa) and hence the adversary has to check equality of two values, $f(X)$ and $f(p(X))$, instead of comparing several values like in the birthday attack. Thus we would expect that K_2 to be large. In the random oracle model of f, we can prove that, $K_1(n) = K_2(n) = 2^n$

Theorem 11 *Under the assumption of the random oracle model of f, $K_1(n) = K_2(n) = 2^n$. Thus, for any permutation p where $\mathcal{F}_p$ is the empty set, any attack algorithm finding collision requires $\Omega(2^n)$ many queries of f in the random oracle model of f.*

The second part of theorem is immediate from Proposition **??**. Before proving the first part of Theorem, we first introduce a new notion called *computable message*. A set of pairs, $\{(x_1, y_1), \cdots, (x_q, y_q)\}$ is called *view* of a function, f, if $f(x_i) = y_i$, $1 \leq i \leq q$. Similarly, $\mathcal{Q} = (\mathcal{Q}_1, \cdots, \mathcal{Q}_k)$ is called view of $f_1, \cdots, f_k$, where $\mathcal{Q}_i$ is the view of f_i. Intuitively, a computable message, X of F is a message so that $F(X)$ can be computed from a set of query-response pairs of the underlying compression functions.

Definition 12 (Computable Message)

Let the double length compression function, F, be based on the compression functions, $f_1, \cdots, f_k$. Let $\mathcal{Q}_j = \{(x_1^j, y_1^j), \cdots, (x_{q_j}, y_{q_j})\}$ be the view *of f_j, $1 \le j \le k$. Let $\mathcal{Q} = (\mathcal{Q}_1, \cdots, \mathcal{Q}_k)$ be the* view *of the underlying compression functions $f_1, \cdots, f_k$. We say, an input X is* computable message *of F with respect to the view $\mathcal{Q}$, if the value of $F(X)$ can be computed from $\mathcal{Q}$.*

For example, when $F = f^p$, an input X is computable message of F with respect to $\{(x_1, y_1), \cdots, (x_q, y_q)\}$, view of f, if $X = x_i$ and $p(X) = x_j$ for some $i, j \in [1, q]$. Thus, $f^p(X) = f(x_i)||f(x_j) = y_i||y_j$, which can be computed from $\mathcal{Q}$.

Proof of Theorem ??: Since f is a random oracle, for $\{X, Y\} \ne \{p(X), p(Y)\}$ we have,

$$
\begin{aligned}
&\Pr[\ f(X) = f(Y) \text{ and } f(p(X)) = f(p(Y))\ ] \\
&= \Pr[\ f(p(X)) = f(p(Y))\ \mid\ f(X) = f(Y)\ ] \times \Pr[\ f(X) = f(Y)\ ] \\
&= \Pr[\ f(p(X)) = f(p(Y))\ ] \times \Pr[\ f(X) = f(Y)\ ] \\
&= 1/2^{2n}.
\end{aligned}
$$

The second equality holds because the compression function f is assumed to be a random function and $\{X, Y\} \ne \{p(X), p(Y)\}$. If an adversary can ask at most q many queries then he can have at most q many computable messages and hence at most $\binom{q}{2}$ 2-sets $\{X, Y\}$. Hence the probability that the adversary finds $X \ne Y$ with $\{X, Y\} \ne \{p(X), p(Y)\}$ such that $f(X) = f(Y)$ and $f(p(X)) = f(p(Y))$ is at most $\binom{q}{2}/2^{2n}$. Thus, to have a significant success probability, q should be $\Omega(2^n)$.

Similarly, from a set of q queries one can get $\mathrm{O}(q)$ many pairs of the form $(X, p(X))$ and for fixed X, $\Pr[f(X) = f(p(X))] = 1/2^n$ provided $p(X) \ne X$. Thus success probability is at most $q/2^n$ and hence $q = \Omega(2^n)$ to have significant success probability. Thus, both the assumptions 1 and 2 are true in the random oracle model of f. ∎

Remark 13 *We can relax the condition that the set of fixed point of the permutation is the empty set. We can choose a permutation where $|\mathcal{F}_p| << 2^{n/2}$ so that there are no two elements $X \ne Y \in \mathcal{F}_p$ such that $p(X) = p(Y)$. Then also, we can prove the same statement.*

3.2 A class of secure double length hash functions

Till now, we have proved that a double length compression function, f^p, based on a permutation having no fixed point, is maximally secure provided the underlying compression function, f, satisfies some reasonable assumptions or f is assumed to be a random oracle. Thus, the classical hash function based on this double length compression function is also maximally secure. We have also seen in Example **??** that if the permutation, $p(\cdot)$, has many fixed points then there is a collision attack (also preimage) better than the birthday attack on the double length compression function based on the permutation p. But, it may happen that the classical hash function based on this insecure compression function can be secure. In Example **??**, if we start the iteration with an initial value satisfying some condition (described momentarily) then the hash functions becomes maximally secure. Recall that, $f^p(H_1, H_2, M) = f(H_1, H_2, M)||f(H_2, H_1, M)$, where M is a message block and H_1 and H_2 are chaining variables of the classical iteration. If we start with an initial value $H_1^0||H_2^0$ such that $H_1^0 \ne H_2^0$ then it is hard to find a message block M such that $f(H_1^0, H_2^0, M) = H_1^1||H_2^1$ where $H_1^1 = H_2^1$ because there are 2^n many outputs with $H_1^1 = H_2^1$ where as the total number of possible outputs is 2^{2n}. Thus, the complexity would be roughly $\Omega(2^n)$. Also it would be hard to find $H_1 \neq H_2$ and $G_1 \neq G_2$ and M such that $f^p(H_1, H_2, M) = f^p(G_1, G_2, M)$. Now we define a *good permutation* and prove the maximal security of the hash function based on a good permutation.

Definition 14 (Good Permutation)

Let p be a permutation on the set of $(n+m)$-bits strings. Define $\mathcal{F}_p[2n] = \{Z \in \{0,1\}^{2n} : \exists M \in \{0,1\}^{m-n}$ such that $Z||M \in \mathcal{F}_p$ \}. It is a projection of $\mathcal{F}_p$ onto the the first $2n$-bits of it. We say the permutation, $p(\cdot)$, is good *if $2^{2n}/|\mathcal{F}_p[2n]| = \Omega(2^n)$. In other words, $|\mathcal{F}_p[2n]| << 2^n$.*

Now we define the following attack. Find M and $H \notin \mathcal{F}_p[2n]$, such that $f^p(H, M) \in \mathcal{F}_p[2n]$ where, $|M| = m - n$ and $|H| = 2n$. Let the complexity of the best attack be $\Omega(K_3(n))$ (or in short K_3).

Proposition 15 *In the random oracle model, $K_3(n) = 2^n$ provided the permutation $p(\cdot)$ is good.*

Proof. We have already seen that after q many queries the adversary can have at most q many computable message for f^p. Given a computable message $H||M$ with $H \notin \mathcal{F}_p[2n]$, we have $p(H||M) \neq H||M$ (see the definition of $\mathcal{F}_p[2n]$ in Definition 3.3) and hence $f^p(H||M)$ is uniformly distributed over the set $\{0,1\}^{2n}$. But $|\mathcal{F}_p[2n]| < 2^n$ since the permutation $p(\cdot)$ is good. Thus we have, $\Pr[f^p(H||M) \in \mathcal{F}_p[2n]] \leq 1/2^n$. Since we have at most q computable message the success probability of the adversary is less than $q/2^n$. This proves the fact that $K_3(n) = 2^n$ under the random oracle model of $f(\cdot)$. ■

Theorem 16 *The classical hash function, H^{f^p}, based on a good permutation and an initial value $H_0 \notin \mathcal{F}_p[2n]$ has collision security* $\min\{K_1, K_2, K_3\}$. *Thus, in the random oracle model of f, H^{f^p} is maximally secure against collision attack for a good permutation $p(\cdot)$.*

Proof. Let (M, M') be a collision on H^{f^p} and $H_0 \notin \mathcal{F}_p[2n]$. We denote H_i and G_i for internal hash values while computing the final hash value for messages $M = M_1||M_2\cdots$ and $M' = M'_1||M'_2\cdots$ respectively. Now we have one of the following :

1. There is an i such that $H_i \notin \mathcal{F}_p[2n]$ but $f^p(H_i||M_{i+1}) \in \mathcal{F}_p[2n]$ or there is a j such that $G_j \notin \mathcal{F}_p[2n]$ but $f^p(G_j||M'_{j+1}) \in \mathcal{F}_p[2n]$.
2. There are $H_i, G_j \notin \mathcal{F}_p[2n]$ with $H_i \neq G_j$ such that $f(H_i, M_{i+1}) = f(G_j, M'_{j+1})$ and $f(p(H_i, M_{i+1})) = f(p(G_j, M'_{j+1}))$. Let $X = H_i||M_{i+1}$ and $Y = G_j||M'_{j+1}$. Since $H_i \neq G_j$, $X \neq Y$. Also we have $X, Y \notin \mathcal{F}_p$ since $H_i, G_j \notin \mathcal{F}_p[2n]$. Thus either $\{X, Y\} \neq \{p(X), p(Y)\}$ or $\{X, p(X)\}$ is a collision set for the compression function f.

In the first case, the adversary requires K_3 many queries of f whereas in the second case the adversary requires $\min\{K_2, K_3\}$ many queries of f. Thus the adversary needs $\min\{K_1, K_2, K_3\}$ complexity. By Propositions **??**, H^{f^p} is maximally secure under the random oracle model of f. ■

3.3 An efficient double length hash function

Let a compression function $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ with $m \geq n$. To understand the design in a simpler way, we assume that $m = n$, that is $f : \{0,1\}^{2n} \to \{0,1\}^n$. For $i > 0$, define $f^{(i)} : \{0,1\}^{(i+1)n} \to \{0,1\}^n$ by using the classical iteration. Thus, for $x_0||\cdots||x_i$ with $|x_j| = n$, $0 \leq j \leq i$ and $h_0 = x_0$.

$$f^{(i)}(x_0||\cdots||x_i) = h_i, \text{ where, } h_j = f(h_{j-1}, x_j),\ 1 \le j \le i.$$

We say $f^{(i)}$ by the *i-iterated compression function*. Now we can observe that the multicollision on this compression function is not as easy as the classical hash function, since we restrict the number of message blocks Any r^i-way collision on $f^{(i)}$ reduces to at least r-way collision on the underlying compression function f (by using pigeon-hole principle). Thus, if we assume that $(r+1)$-way collision on f is infeasible then we can have at most r^i-way collision on $f^{(i)}$. Recall that, in the random oracle model of f, r-way collision requires $\Omega(2^{n(r-1)/r})$ queries. Now we summarize this by the following lemma.

Lemma 17 *(r^i+1)-way collision on $f^{(i)}$ reduces to at least $(r+1)$-way collision on f. In particular, when f is a random function, the complexity of (r^i+1)-way collision attack on $f^{(i)}$ is $\Omega(2^{nr/(r+1)})$ and the complexity of n^i+1-way collision attack on $f^{(i)}$ is $\Omega(2^n)$.*

Like the concatenation of two independent hash functions we can define the concatenation of two independent i-iterated compression functions. Thus, given two independent compression functions, f_1 and f_2, we can define a double length compression function, $F^{(i)}(X) = f_1^{(i)}(X)||f_2^{(i)}(X)$, $|X| = n(i+1)$. Obviously, in this construction, we need to assume $i \ge 2$. Otherwise, for $i = 1$, it does not compress the input. Now we can study the security property of this concatenated compression function in the random oracle model.

Lemma 18 *If f is a random function then for any two distinct $(i+1)$-block inputs X and Y, $Pr[f^{(i)}(X) = f^{(i)}(Y)] \le i/2^n$. If f_1 and f_2 are two independent random functions then $Pr[F^{(i)}(X) = F^{(i)}(Y)] = i^2/2^{2n}$.*

Proof. Let j be the round number where collision of f occurs. Call this event by C_j. Thus, $f^{(i)}(X) = f^{(i)}(Y)$ implies $\cup_{j=1}^{i} C_j$. Now, $\Pr[C_j] \le 1/2^n$ (?). Thus, $\Pr[\cup_{j=1}^{i} C_j] \le i/2^n$.

$$\begin{aligned}\Pr[F^{(i)}(X) = F^{(i)}(Y)] &= \Pr[f_1^{(i)}(X) = f_1^{(i)}(Y), f_2^{(i)}(X) = f_2^{(i)}(Y)] \\ &= \Pr[f_1^{(i)}(X) = f_1^{(i)}(Y)] \times \Pr[f_2^{(i)}(X) = f_2^{(i)}(Y)] \\ &= i^2/2^{2n}.\end{aligned}$$

The second equality follows from the fact that f_1 and f_2 are independent random functions and the last equality is immediate from Lemma 18. ∎

Thus to find the collision probability for any adversary we need to compute the number of pairs (X, Y) he can get from any possible set of queries. Note that the adversary should compute the F-values of both X and Y. Now we state the computable message which means the message whose hash value can be computed from the set of queries the adversary made. We fix $i \ge 2$.

Definition 19 (Computable message) *Let $\mathcal{Q}_j$ be the set of query response tuples for the random function f_j, $j = 1, 2$. X is said to be a computable message for $f_j^{(i)}$ (also for $F^{(i)}$) with respect to $\mathcal{Q}_j$ if the value of $f_j^{(i)}(X)$ (or $F^{(i)}$) can be computed from $\mathcal{Q}_j$ (or $\mathcal{Q}_1 \cup \mathcal{Q}_2$ respectively).*

More precisely, if $X = x_0||\cdots||x_i$ then X is computable for $f_1^{(i)}$ with respect to $\mathcal{Q}_1$ if $(x_0||x_1, h_1)$, $(h_1||x_2, h_2)$,$\cdots$,$(h_{i-1}||x_i, h_i) \in \mathcal{Q}_1$. Thus the $f_1^{(i)}$-value of X is h_i. Similarly one can define computable messages for $f_2^{(i)}$. A message X is computable with respect to $\mathcal{Q}_1 \cup \mathcal{Q}_2$ for the compression function $F^{(i)}$, if X is computable for both $f^{(i)}$ with respect to $\mathcal{Q}_j$, $j = 1, 2$.

Let q be the number of queries. We assume that $q = o(2^n)$. Thus there is no n-way collision on both f_1 and f_2. Note that, the complexity of n-way collision on a random function is $\Omega(2^{n(n-1)/n}) = \Omega(2^n)$. Thus we can have at most n^{i-1}-way collision on $f_1^{(i-1)}$ or $f_2^{(i-1)}$. The number of computable messages for $F^{(i)}$ is at most qn^{i-1}. Thus, total number of pairs of the form (X, Y) where $X \neq Y$ are $(i+1)$-block inputs and both X and Y are computable messages is at most $q^2 n^{2(i-1)}/2$. Thus, probability that we have a collision among these pairs is bounded by $i^2 q^2 n^{2(i-1)}/2^{2n+1}$. To have non-negligible probability we need $q = \Omega(2^n/i^2 n^{i-1})$. Thus we have the following theorem :

Theorem 20 *If f_1 and f_2 are two independent random functions then the complexity for finding a collision on $F^{(i)}$ requires $\Omega(2^n/(i^2 n^{i-1}))$ queries.*

Remark 21 *If we look the proof more closely then we can find a better security bound. If $q = O(2^{n(r-1)/r})$ for some r (determined later) then we do not have any $(r+1)$-way collision on f_i and hence the number of computable message for $F^{(i)}$ is at most $r^{i-1}.q$. Thus to find a collision on $F^{(i)}$ we need $q = 2^n/r^{i-1}$. Thus, we can choose r such that $2^{n(r-1)/r} = 2^n/r^{i-1}$. Thus, $r \log r = n/(i-1)$ and denote that r by r_0. Thus, the security bound for collision attack on $F^{(i)}$ is $\Omega(2^n/r_0^{i-1})$.*

Efficiency of the compression function. The rate function of the compression function, $F^{(i)}$, is $((i+1)n - 2n)/2ni = \frac{1}{2} - \frac{1}{2i}$. Thus, the rate of the compression function is close to 1/2 provided i is large. So we have a trade-off between the security level and the efficiency.

Now we define a double length hash function $H^s : (\{0,1\}^n)^* \to \{0,1\}^{2n}$, $s \geq 2$. We can define the hash function on arbitrary domain by applying some standard padding rule. Let $M = m_1||\cdots||m_l$ be l-block message. Let $l = (s-1).b + r$, where $0 \leq r < s-1$. Thus, we divide the message $M = M_1||\cdots||M_b||M_{b+1}$, where $|M_i| = (s-1)n$, $1 \leq i \leq b$ and $|M_{b+1}| = rn$. In case of $r = 0$ we do not have any message block M_{b+1}. Let H_0 be an initial two block message that is $|H_0| = 2n$. Now define the hash function $H^s(H_0, M)$ as follows;

Algorithm $H^s(H_0, m_1||\ldots||m_l)$
$H_i = F^s(H_{i-1}, M_i)$, $i = 1$ to b
If $r > 0$ **then** $H = F^{r+1}(H_b, M_{q+1})$
If $r = 0$ **then** $H = H_b$
Return H

Thus, the hash function is the classical iterated hash function by using two underlying compression functions $F^{(s)}$ and $F^{(r+1)}$. Thus any collision on $H^{(s)}$ reduces to the collision on one of the compression function. But from theorem 20 we know that collision on $f^{(i)}$ is infeasible and hence the hash function is secure against collision attack. More precisely we have the following theorem :

Theorem 22 *For any $s \geq 2$, collision on $H^{(s)}$ requires $\Omega(2^n/s^2n^{s-1})$ complexity.*

3.3.1 (2nd) Preimage security of the new hash function.

Similar to the previous section we can study the (2nd) preimage security. Recall that we say a message X is computable from the set of queries $\mathcal{Q}$ if $f^{(i)}(X)$ can be computed from the set $\mathcal{Q}$. We already observed that if q is the maximum number of queries and at most r-way collision is possible then we can have $q.r^{i-1}$ computable messages. Now given M, $F^{(i)}(M)$ is a $2n$-bit random string. We have already observed that $\Pr[F(M) = F(N)] = i^2/2^{2n}$, where $M \neq N$. So, if $q = o(2^n)$ then the number of computable message for N is at most $n^{i-1}.q$. Thus, there will be a computable message $N \neq M$ such that $F^{(i)}(M) = F^{(i)}(N)$ is bounded by $qn^{i-1}/i^2 2^{2n}$. Thus complexity for any attack algorithm of 2nd preimage attack is $\Omega(2^{2n}/i^2n^{i-1})$. The security level for preimage attack is same as that of 2nd preimage.

4 Conclusion

This paper deals with several new double length compression functions. We first introduce a class of double length compression function which contains recently known constructions [16, 25]. We study their security level in the random oracle model. Finally, we designed a double length compression function with rate close to 1/2 (the rate of concatenated hash function). The design is very much similar to the concatenated hash functions. It has almost maximal security level.

References

[1] R. Anderson, E. Biham, *Tiger: A new fast hash function.* Fast Software Encryption, LNCS 1039, D. Gollmann, Ed., Springer-Verlag, 1996, pp.89-97.

[2] M. Bellare and T. Kohno. *Hash function balance and its impact on birthday attacks.* Advances in Cryptology - Eurocrypt'04, Lecture Notes in Computer Science Vol. 3027, Springer-Verlag, 2004.

[3] E. Biham, R. Chen, A. Joux, P. Carribault, W. Jalby and C. Lemuet. *Collisions of SHA-0 and Reduced SHA-1.* To be appeared in Eurocrypt-05, 2005.

[4] J. Black, M. Cochran and T. Shrimpton. *On the Impossibility of Highly Efficient Blockcipher-Based Hash Functions.* To appear in Eurocrypt-05, 2005. ePrint Archive, 2004. Available at http://eprint.iacr.org/2004/062.

[5] J. Black, P. Rogaway, and T. Shrimpton. *Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV.* Advances in Cryptology - Crypto'02, Lecture Notes in Computer Science, Vol. 2442, Springer-Verlag, pp. 320-335, 2002.

[6] R. Cramer and V. Shoup. *Using Hash Functions as a Hedge against Chosen Ciphertext Attack.* Advances in Cryptology - Eurocrypt'00, Lecture Notes in Computer Science, Vol. 2442, Springer-Verlag, pp. 275-288 , 2000.

[7] J. Daemen and V. Rijmen. *The Design of Rijndael: AES.* The Advanced Encryption Standard. Springer, 2002.

[8] I. B. Damgȧrd. *Collision Free Hash Functions and Public Key Signature Schemes.* Advances in Cryptology - Eurocrypt'87, Lecture Notes in Computer Sciences, Vol. 304, Springer-Verlag, pp. 203-216, 1987.

[9] C. Debaert and H. Gilbert. RIPEMD^L *and* RIPEMD^R *improved variants of MD4 are not collision free.* Fast Software Encryption- 2002, no. 2355, Lecture Notes in Computer Science, pp. 52-65, Springer- Verlag, 2002.

[10] H. Dobbertin. *Cryptanalysis of MD4.* Fast Software Encryption, Cambridge Workshop. Lecture Notes in Computer Science, vol 1039, D. Gollman ed. Springer-Verlag, 1996.

[11] H. Dobbertin. *Cryptanalysis of MD5*, Rump Session of Eurocrypt'96, 1996. http//www.iacr.org/conferences/ec96/rump/index.html.

[12] H. Dobbertin, A. Bosselaers and B. Preneel. *RIPEMD-160: A Strengthened Version of RIPEMD*, Fast Software Encryption. Lecture Notes in Computer Science 1039, D. Gollmann, ed., Springer-Verlag, 1996.

[13] H. Finney. *More problems with hash functions.* The cryptographic mailing list, 24 Aug 2004. Available at http://lists.virus.org/cryptography-0408/msg00124.html.

[14] S. Goldwasser, S. Micali and R. Rivest. *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks*, SIAM Journal of Computing, Vol17, No2, pp. 281-308, April 1998.

[15] M. Hattori, S. Hirose and S. Yoshida. *Analysis of Double Block Lengh Hash Functions.* 9th IMA International Conference Cryptographi and Coding, 2003, Lecture Notes in Computer Science, vol-2898.

[16] S. Hirose. *Provably Secure Double-Block-Length Hash Functions in a Black-Box Model*, 7th International Conference on Information Security and Cryptology, 2004.

[17] A. Joux. *Multicollision on Iterated Hash Function.* Advances in Cryptology - Crypto'04, Lecture Notes in Computer Science vol-3152.

[18] L. Knudsen. *Some properties of an FSE 2005 Hash Proposal.* Cryptology ePrint Archive, 2005. Available at http://eprint.iacr.org/2005/082.

[19] L. R. Knudsen and B. Preneel. *Hash Functions Based on Block Ciphers and Quaternary Codes.* Asiacrypt'96, pp-77-90.

[20] L. Knudsen, X. Lai and B. Preneel. *Attacks on fast double block length hash functions.* Journal of Cryptology, vol-11, no-1, winter, 1998.

[21] L. Knudsen and B. Preneel. *Construction of Secure and Fast Hash Functions Using Nonbinary Error-Correcting Codes*. IEEE transactions on information theory, VOL-48, NO. 9, Sept-2002.

[22] H. Krawczyk, M. Bellare and R. Canetti. *HMAC: Keyed-Hashing for Message Authentication*. Internet RFC 2104, February 1997.

[23] W. Lee, M. Nandi, P. Sarkar, D. Chang, S. Lee and K. Sakurai *A Generalization of PGV-Hash Functions and Security Analysis in Black-Box Model*. Information Security and Privacy: 9th Australasian Conference, ACISP'04, Lecture Notes in Computer Science, vol-3108, 2004.

[24] W. Lee, M. Nandi, P. Sarkar, D. Chang, S. Lee and K. Sakurai *PGV-style Block-Cipher-Based Hash Families and Black-Box Analysis* . IEICE transaction, vol-E88-A, no.1, Jan, 2005, pp-39-48.

[25] S. Lucks. *Design principles for Iterated Hash Functions*. ePrint Archive Report, 2004. Available at http://eprint.iacr.org/2004/253.

[26] A. J. Menezes, P. van Oorschot and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press ISBN: 0-8493-8523-7, October 1996.

[27] T. Matsuo and K. Kurosawa. *On Parallel Hash Functions Based on Block-Cipher*. Information Security and Privacy: 8th Australasian Conference, ACISP'03, Lecture Notes in Computer Science, vol-2727, pp-510-521, 2003.

[28] R. Merkle. *One Way Hash Functions and DES*. Advances in Cryptology - Crypto'89, Lecture Notes in Computer Sciences, Vol. 435, Springer-Verlag, pp. 428-446, 1989.

[29] M. Nandi, W. Lee, K. Sakurai and S. Lee. *Security Analysis of a 2/3-rate Double Length Compression Function in The Black-Box Model*. FSE'05, 2005.

[30] M. Nandi and D. R. Stinson. *Multicollision Attacks on Generalized Hash Functions*. Cryptology ePrint Archive, 2004. Available at http://eprint.iacr.org/2004/330.

[31] National Institute of Standards, FIPS 180-1, Secure Hash Standard. April 1995.

[32] NIST/NSA. *FIPS 180-2 Secure Hash Standard*. August, 2002. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf.

[33] B. Preneel, R. Govaerts and J. Vandewalle. *Cryptographically secure hash functions: an overview*. ESAT Internal Report, K. U. Leuven, 1989.

[34] R. L. Rivest. *MD4 message digest algorithm*. Advances in Cryptology, Proceedings Crypto'90, LNCS 537, S. Vanstone, Ed., Springer-Verlag, 1991, pp. 303-311.

[35] R. L. Rivest *The MD5 message digest algorithm*. Available online : http://www.ietf.org/rfc/rfc1321.txt.

[36] P. Sarkar. *Domain Extender for Collision Resistant Hash Functions: Improving Upon Merkle-Damgard Iteration*. ePrint Archive Report, 2002. Available at http://eprint.iacr.org/2003/173.

[37] T. Satoh, M. Haga and K. Kurosawa. *Towards Secure and Fast Hash Functions*. IEICE Trans. VOL. E82-A, NO. 1 January, 1999.

[38] V. Shoup. *Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack*. SIAM Journal of Computing 33:167-226, 2003.

[39] D. R. Stinson. *Cryptography : Theory and Practice*, Second Edition, CRC Press, Inc.

[40] D. R. Stinson. *Some observations on the theory of cryptographic hash functions.* Eprint Archive Report, 2001. Available at http://eprint.iacr.org/2001/020/.

[41] X. Wang and D. Feng and X. Lai and H. Yu. *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD.*