A Class of secure Double Length Hash Functions

Mridul Nandi

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

Abstract. In this paper we constructed a class of double length hash functions which are maximally secure i.e. the birthday attack is the best possible attack. Recently, Joux [6] in Crypto-04 showed a multicollision attack on the classical iterated hash function which can be used to get the collision on the concatenated double length hash functions. Very recently, Lucks [10] also designed a double-pipe hash which is secure against any multicollision attack and Hirose [5] designed a double block length collision resistance hash functions which are based on a secure block-cipher. Here, we study closely to their papers [5], [10] and constructed a class of secure double length hash functions.

1 Introduction

Theoretically one can define hash function by any function $f : D \rightarrow R$ where, $|D| > |R|$. But in practice, one consider hash function $f : \{0,1\}^{n+m} \rightarrow \{0,1\}^n$, $m > 0$. It has many applications in cryptography such as digital signature schemes, public key encryption schemes, message authentication codes etc. To guarantee the security of these schemes hash function should satisfy some security assumptions. There are many known security assumptions e.g. collision resistance, pre-image resistance, 2nd pre-image resistance and so on. Also one need hash function defined on an arbitrary domain. To design an arbitrary domain hash function one first design a fixed domain hash function $f : \{0,1\}^{n+m} \rightarrow \{0,1\}^n$ (also known as a compression function) and then extend the domain to arbitrary domain by iterating the compression function several times. This method is known as MD-method [2], [11]. Given a message M first append 10^i so that the length of the message is a multiple of m and then append the binary representation of the length of the message. This padding method is also known as *MD-strengthening.* So, for some fixed initial value $h_0 \in \{0,1\}^n$ and a padded input $M = m_1||\cdots||m_l \in (\{0,1\}^m)^*$ where, $|m_i| = m$, the hash function $H^f(h_0, \cdot) : (\{0,1\}^m)^* \rightarrow \{0,1\}^n$ can be defined as follow :

```
Algorithm $H^f(h_0, m_1||\ldots||m_l)$
For $i = 1$ to $l$
$h_i = f(h_{i-1}, m_i)$
Return $h_l$
```

1.1 Type of Attacks on compression functions and Hash Functions.

Let $f : \{0,1\}^{m+n} \rightarrow \{0,1\}^n$ be a compression function. We can define the following attacks on this compression function.

1. **preimage attack :** Given y find x such that $f(x) = y$.
2. **2nd preimage attack :** Given x find $x' \neq x$ such that $f(x') = f(x)$.
3. **collision attack :** Find $x \neq x'$ such that $f(x') = f(x)$.

Let $H^f(IV, \cdot) : \{0,1\}^* \rightarrow \{0,1\}^n$ be a hash function based on a compression function $f(\cdot)$ with an initial value $IV \in \{0,1\}^n$. The most popularly attacks are the following :

free-start (2nd) preimage attack: For a given output $y \in \{0,1\}^n$ find $IV \in \{0,1\}^n$, $x \in \{0,1\}^*$ such that $H^f(IV, x) = y$. In case of 2nd pre-image attack given x' find $IV \in \{0,1\}^n$ and $x \in \{0,1\}^*$ such that $H^f(IV, x) = H^f(IV, x')$ and $x \neq x'$.

(2nd) preimage attack : Given $IV \in \{0,1\}^n$ and output $y \in \{0,1\}^n$ find $x \in \{0,1\}^*$ such that $H^f(IV, x) = y$. Similarly one can define 2nd preimage attack.

(free-start) collision attack : Find $x, x' \in \{0,1\}^*$ and $IV, IV' \in \{0,1\}^n$ such that $(IV, x) \neq (IV', x)$ and $H^f(IV', x') = H^f(IV, x)$. In case of collision attack the initial value IV is fixed and given.

Besides these attacks one can consider a generalization of collision attack which is known as multicollision attack. Although the security of multicollision attack has very limited applications in cryptographic protocol it has importance to find a collision attack as in [4], [6], [7], [8], [13].

r-way collision attack or multicollision attack : Given $IV \in \{0,1\}^n$ find a set $\{x_1, \ldots, x_r\}$ (*multicollision set*) such that $g(IV, x_1) = \cdots = g(IV, x_r)$. Similarly one can define r-way (2nd) preimage attack.

In the case of classical iterated hash function with MD-strengthening, *the security of free-start attack on H^f is equivalent to that of collision resistance of f*. Similar results also hold for (2nd) preimage. It is easy to observe that free-start attack is much easier than that of without free-start attack. To construct a collision resistance hash function it is enough to construct a collision resistant compression function. Also there are possibilities that the underlying compression function is not collision resistance but the hash function is secure against collision attack.

1.2 Complexity of attacks in the random oracle model.

A function $g : D \rightarrow R$ is said to be a random function (in other word, g is modelled as a random oracle) if for each $k > 0$ and a subset $\{x_1, \ldots, x_k\} \subseteq D$,

$g(x_1), \cdots, g(x_k)$ are independently and uniformly distributed on the set R. So, only way to know the value of $g(x)$ is to make a query of $g(\cdot)$ with input x even if the value of $g(x_1), \cdots, g(x_k)$ are known with $x_i \neq x$. The complexity of an attack in the random oracle model of g is the number of queries of g.

1. Complexity of the birthday attack for (free-start) preimage resistance and (free-start) 2nd preimage attack is $\mathrm{O}(|R|)$ or $\mathrm{O}(2^n)$ (when $R = \{0,1\}^n$).

2. Complexity of the birthday attack for collision attack is $\mathrm{O}(|R|^{1/2})$ or $\mathrm{O}(2^{n/2})$ (when $R = \{0,1\}^n$).

3. Complexity of the birthday attack for r-way collision attack is $\mathrm{O}(|R|^{(r-1)/r})$ or $\mathrm{O}(2^{(r-1)n/r})$ (when $R = \{0,1\}^n$). For r-way (2nd) preimage attack it has complexity $\mathrm{O}(\mathrm{r}.|R|)$ or $\mathrm{O}(r.2^n)$ (when $R = \{0,1\}^n$).

Here we model the underlying compression function $f : \{0,1\}^{n+m} \rightarrow \{0,1\}^n$ as a random oracle and based on that we have either a compression function F or a hash function H. When we study an attack on F or H the complexity is the number of queries of f to be required.

2 Double Length Compression Function

We will construct a double length compression function from one or two single length compression functions. If a single length compression function has output size n then that of double length compression function is $2n$. For the smaller size hash function the birthday attack can be feasible. So to make birthday infeasible we need to construct a compression function with larger size output. One way to do that design a compression function with larger size output from scratch. The other way is to design a larger size compression function from a smaller size compression function and prove its security level using the security level of the underlying compression function. We will be interested to the second method. For designing a hash function we will use the classical MD-method.

2.1 Using two independent single length compression functions

Let $C_1, C_2 : \{0,1\}^N \rightarrow \{0,1\}^n$ be two independent compression functions with $N > 2n$. Define a compression function $C : \{0,1\}^N \rightarrow \{0,1\}^{2n}$ by $C(X) = C_1(X)||C_2(X)$. A collision on C reduces to simultaneous collisions on C_1 and C_2 i.e. for $X \neq Y$, $C(X) = C(Y)$ implies $C_1(X) = C_1(Y)$ and $C_2(X) = C_2(Y)$. Two random functions C_1 and C_2 are said to be independent if the output distributions are independent.

Proposition 1. *If C_1 and C_2 are two independent random oracles then finding collision on C requires $\Omega(2^n)$ many queries of C_1 and C_2.*

Proof. If the adversary asks q many queries to C_1 and C_2 then he can compute the values of C for at most q inputs say $X_1, \cdots, X_q$. For any two inputs X_i and X_j with $i \neq j$, we have,

$$\Pr[C(X_i) = C(X_j)] = \Pr[C_1(X_i) = C_1(X_j), C_2(X_i) = C_2(X_j)] = 1/2^{2n}.$$

So total probability of getting collision is bounded by $q(q-1)/2^{2n+1}$, where $q(q-1)/2$ is the number of pairs (X_i, X_j) with $i \neq j$. To have the non-negligible probability we need $q = \Omega(2^n)$. □

2.2 Using two independent double key length block-ciphers

There are many secure ways to construct a compression function from a block-cipher [9], [12]. Let $E^{(i)} : \{0,1\}^{2n} \times \{0,1\}^n \rightarrow \{0,1\}^n$ be two independent block ciphers with key-length $2n$ bits, $i = 1, 2$. One can define a compression function $C_i : \{0,1\}^{3n} \rightarrow \{0,1\}^n$ where, $C_i(x, y, z) = E^{(i)}_{x||y}(z) \oplus z$ and finally define C by $C_1||C_2$. Here, we assume that $E^{(1)}(k, \cdot)$ and $E^{(2)}(k, \cdot)$ are independent random permutations. Similar to the compression function let us assume that an adversary can make at most q queries of $E^{(1)}, {E^{(1)}}^{-1}$ and $E^{(2)}, {E^{(2)}}^{-1}$. Now note the following :

1. From one query of $E^{(i)}$ or ${E^{(i)}}^{-1}$ he can make exactly one computation of C and hence, for at most q inputs $X_1, \cdots, X_q$, $C(X_1), \cdots, C(X_q)$ can be computed.
2. For any X_i, $C_1(X_i)$ and $C_2(X_i)$ are independently distributed over a set of size at least $2^n - q$. If $q \leq 2^{n-1}$, $C(X_i)$ can take any n-bit with probability at most $1/2^{n-1}$

So probability of getting collision after q queries is bounded by $q(q-1)/2^{2n-1}$. So again we need $q = \Omega(2^n)$ to have non-negligible probability of getting collision. In case of $q \geq 2^{n-1}$ it is also $\Omega(2^n)$.

Remark 1. The rate of the double length compression function is 1/2 as it uses two invocations of $E(\cdot)$ to hash a single n-block message.

Remark 2. Similar result holds if we consider $C(x, y, z) = E_{x||y}(z) \oplus z \oplus y$. We only need that

1. x, y, z should determine uniquely the computation of $E(\cdot)$ and hence from a set of q queries of E or E^{-1} at most q values of C will be known.
2. the output of $C(x, y, z)$ is randomly distributed over a large set in the black-box model of the block cipher. In fact, if for any single computation it is randomly distributed on the set $\{0,1\}^n$ then after q queries the value of C will be uniformly distributed over a set of size at least $2^n - q$.

So there are many block cipher based rate-1/2 double length schemes [5] which are maximally secure.

Remark 3. One can define two independent compression functions from a single compression function by changing one bit in a position. For example, $C_1(X) = C'(0||X)$ and $C_2(X) = C'(1||X)$. Here, $C_i : \{0,1\}^{N-1} \to \{0,1\}^n$ whereas $C : \{0,1\}^N \to \{0,1\}^n$. In case of block cipher based compression function we do the same thing for a single key bit.

2.3 Using a single length compression function.

In this section we study the Lucks's construction [10] and generalize the class of secure double length hash function. The construction proposed by Lucks [10] has also been proposed independently by Finney [3] in a mailing list. Let $C : \{0,1\}^N \to \{0,1\}^n$ be a compression function, $N > 2n$. One can define a compression function by $C(p_1(X))||C(p_2(X))$ where, p_1 and p_2 are some permutations on N-bits. Without loss of generality one can assume p_1 as an identity function and $p = p_2$ (otherwise think as a function of $p_1(X)$ and take $p = p_2(p_1^{-1}(\cdot))$). So we have a compression function $C^p(X) = C(X)||C(p(X))$ where,

- $C : \{0,1\}^N \to \{0,1\}^n$.
- $p : \{0,1\}^N \to \{0,1\}^N$, a permutation.

In Lucks's paper [10] he considered the permutation $p(x||y||z) = (y||x||z)$ where, $|x| = |y| = |z| = n$. If $p(\cdot)$ is not a permutation then the compression function C^p is weak in structure as one can find two different inputs $X \neq Y$ such that $p(X) = p(Y)$ and hence $C(p(X)) = C(p(Y))$. So one can find collision on one part of the output very easily. Now we show that if the permutation p satisfies a condition then the function C^p is secure against collision attack under some reasonable assumptions of C.

Assumption 1 *The minimum complexity for finding $X \in \{0,1\}^N$ such that $C(X) = C(p(X))$ where, $p(X) \neq X$ is $\Omega(2^n)$.*

Assumption 2 *It is hard (minimum complexity $\Omega(2^n)$) to find $X \neq Y$ such that $C(X) = C(Y)$ and $C(p(X)) = C(p(Y))$ where, $\{X, Y\} \neq \{p(X), p(Y)\}$.*

The assumptions say that it is hard to find a related collision pair or two collision pairs which are related. The above assumptions can be verified easily if we assume C as a random function. From a set of q queries one can get $\mathrm{O}(q)$ many pairs of the form $(X, p(X))$ and for fixed X, $\Pr[C(X) = C(p(X))] = 1/2^n$ provided $p(X) \neq X$. Similarly, from a set of q queries one can get $\mathrm{O}(q^2)$ many pairs of the form (X, Y) and for fixed X and Y, $\Pr[C(X) = C(Y), C(p(X)) = C(p(Y))] = 1/2^{2n}$ provided $\{X, Y\} \neq \{p(X), p(Y)\}$.

Definition 1. *Let $f : S \to S$ be some function. An element $x \in S$ is said to be a fixed point of f if $f(x) = x$. We denote the set of fixed points of f by $\mathcal{F}_f = \{x; f(x) = x\}$.*

The next theorem says the compression function C^p is maximally collision resistance if the permutation p does not have many fixed points.

Theorem 1. *For any permutation p where $\mathcal{F}_p$ is small enough to find a collision on C (i.e. $|\mathcal{F}_p| << 2^{n/2}$) then*

1. *under the assumption 1 and 2, finding a collision of C^p requires $\Omega(2^n)$ queries of C.*
2. *under the random oracle model of C finding a collision of C^p requires $\Omega(2^n)$ queries of C.*

Proof. Let $C^p(X) = C^p(Y)$ where, $X \neq Y$. So we have 2-way collision sets $\{X, Y\}$ and $\{p(X), p(Y)\}$. Now we have three possible cases.

- **Case-1 :** $Y = p(X)$ and $X = p(Y)$. In this case we have a collision set $\{X, p(X)\}$ where $p(X) \neq X$. So by assumption 1 the complexity for finding collision of C^p is $\Omega(2^n)$.
- **Case-2 :** $p(X) = X$ and $p(Y) = Y$ i.e. $X \neq Y \in \mathcal{F}_p$. But this can not be true as $\mathcal{F}_p$ is small enough to find a collision on C (otherwise (X, Y) will be a collision pair of C).
- **Case-3 :** $\{X, Y\} \neq \{p(X), p(Y)\}$. By assumption 2 again we need $\Omega(2^n)$ queries.

So, in the all cases the complexity for finding collision of C^p is $\Omega(2^n)$. We have already noted that the assumptions 1 and 2 are true in the random oracle model of C and hence the second part of the theorem is immediate from the first part. □

Remark 4. The assumptions made for the permutation $p(\cdot)$ is necessary as one can find a collision attack on C^p if $p(\cdot)$ does not satisfy the condition. So, p is a permutation such that $|\mathcal{F}_p| \geq 2^{n/2}$. Then find $X \neq Y \in \mathcal{F}_p$ such that $C(X) = C(Y)$. Then obviously, $C^p(X) = C(X)||C(p(X)) = C(X)||C(X) = C(Y)||C(Y) = C^p(Y)$.

Also there are many permutations which satisfy the condition. Here we state few examples of that.

Example 1. Let $p(H'||H''||M) = H'||H''||\overline{M}$ where $\overline{M}$ is the bit-wise complement of M and $|H'| = |H''| = n$. Note $M \neq \overline{M}$ for all M. So $\mathcal{F}_p = \emptyset$.

Example 2. Let $p_i(X) = X_1|| \cdots ||X_{i-1}||\overline{X_i}||X_{i+1}|| \cdots ||X_N$. Here, X_i denotes the i^{th} bit of X. It is easy to check that, $\mathcal{F}_{p_i} = \emptyset$. In fact, we can take bit-wise complement for any number of positions of X.

Example 3. Now consider the example in [10]. $p(H'||H''||M) = H''||H'||M$. Here, $\mathcal{F}_p = \{H'||H''||M : H' = H''\}$. So, $|\mathcal{F}_p| = 2^{2n}$ which is large enough to find a collision.

2.4 Using a double key length block cipher.

Now we will study the security of the compression function based on a secure block cipher. Let $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher with $2n$-bit keys and we will assume $E(\cdot)$ as a black-box. Now consider the same definition we have already taken i.e. $C(x,y,z) = E_{x||y}(z) \oplus z$ and $C^p(X) = C(X)||C(p(X))$ for some permutation $p(\cdot)$. To prove that C^p is maximally secure against collision attack it is enough to prove the assumptions 1 and 2 and take some permutation p with a small set of fixed points.

As we have already noted in the remark 3 that from a set of q queries of E or E^{-1} we have at most q values of C and each one is randomly distributed over a set of size at least 2^{n-1} provided $q \leq 2^{n-1}$.

1. So after q queries of E or E^{-1} we can have $\mathrm{O}(q)$ inputs say $X_1, \cdots, X_q$, where $C(X_1), \cdots, C(X_q)$ can be computed. Also there can be at most $\mathrm{O}(q)$ pairs of the form $(X, p(X))$. For each pair $\Pr[C(p(X)) = C(X)] \leq 1/2^{n-1}$ if $q \leq 2^{n-1}$ and $p(X) \neq X$. This proves the assumption 1.

2. Also we can have $\mathrm{O}(q^2)$ pairs of the form (X, Y). So, for each pair (X_i, X_j) with $i \neq j$ we have, $\Pr[C(X_i) = C(X_j), C(p(X_i)) = C(p(X_j))] \leq q^2/2^{2n-2}$ if $q \leq 2^{n-1}$ and $\{X, Y\} \neq \{p(X), p(Y)\}$. So probability of finding a collision is bounded by $q^2/2^{2n-2}$ which proves assumption 2.

3 Double Length Hash Function

From the previous section we know several double length compression functions which are maximally secure. Collision resistance of the compression function implies the free-start collision resistance of the extended hash function. So, all hash function based on these compression function are also maximally secure. Now one can ask whether are there any compression functions which are not secure but the hash functions based on that compression function are secure against collision attack? As free-start collision resistance of extended hash function is equivalent to the collision resistance of the underlying compression function we only here will be looking for collision resistance of hash functions with fixed initial value. As in the previous section we have a double length compression function C^p for a single length compression function C and a permutation $p(\cdot)$.

Definition 2. *Let p be a permutation on N- bits.*

1. *Define $\mathcal{F}_p[2n] = \{Z \in \{0,1\}^{2n} : \exists\ M \in \{0,1\}^{N-2n}$ such that $Z||M \in \mathcal{F}_p\ \}$. It is a projection of $\mathcal{F}_p$ onto the the first $2n$-bits of it.*
2. *We say the permutation p is* good *if $2^{2n}/|\mathcal{F}_p[2n]| = \Omega(2^n)$.*

Assumption 3 *The complexity of any algorithm which finds M and $H_0 \notin \mathcal{F}_p[2n]$ such that $C^p(H_0, M) \in \mathcal{F}_p[2n]$ has complexity $\Omega(2^n)$.*

The above assumption can be verified under random oracle model of C and if $2^{2n}/|\mathcal{F}_p[2n]| = \Omega(2^n)$ i.e. the permutation $p(\cdot)$ is *good*. The permutation in the example-3 in the previous section is good. Now we can state the following theorem :

Theorem 2. *If Assumptions 1,2 and 3 are satisfied then any collision finding algorithm for H^{C^p} requires time complexity $\Omega(2^n)$. So, in the random oracle model of C, H^{C^p} is maximally secure against collision attack.*

Proof. Let (M, N) ba a collision on H^{C^p} and $H_0 \notin \mathcal{F}_p[2n]$. We denote H_i and G_i for internal hash value while computing the final hash value for message M and N respectively. Now we have one of the following :

1. there is i such that $H_i \notin \mathcal{F}_p[2n]$ but $C^p(H_i||M) \in \mathcal{F}_p[2n]$, for some M.

2. There are $H_i \neq G_i \notin \mathcal{F}_p[2n]$ but there is M such that $C(H_i, M) = C(G_i, M)$ and $C(p(H_i, M)) = C(p(G_i, M))$. Here, H_i and G_i denote the internal hash value for the collision pair M and N respectively.

In the second case we reduces to the collision on C^p which is infeasible because of assumption 1 and 2. In the first case it is still infeasible by assumption 3. □

Let $E : \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher with $2n$-bit key and we will assume $E(\cdot)$ as a black-box. Now consider the same definition we have already taken i.e. $C(x, y, z) = E_{x||y}(z) \oplus z$ and $C^p(X) = C(X)||C(p(X))$ for some good permutation $p(\cdot)$. To prove that the hash function as defined is secure against collision attack it is enough to prove the assumptions 1,2 and 3 under black-box model of $E(\cdot)$. We have already proved that assumption 1 and 2. Assumption 3 is also true as output of C is randomly distributed over a large set of size at least 2^{n-1} provided $q \leq 2^{n-1}$ and because of $H_0 \notin \mathcal{F}_p[2n]$, $H_0||M \neq p(H_0||M)$.

4 Multicollision security of collision resistance Hash function.

Joux's Multicollision attack. In a recent paper by Joux [6], it was shown that there is a 2^r-way collision attack for the classical iterated hash function with complexity $\mathrm{O}(r2^{n/2})$ which is much less than $\Omega(2^{\frac{n(2^r-1)}{2^r}})$ (the complexity for random oracle model, see the section 1.2. The idea of his attack is to find first r successive collisions and then combine those collisions independently to get 2^r-way collision.

$$\begin{aligned} f(h_0, m_1^1) = f(h_0, m_1^2) &= h_1, \\ f(h_1, m_2^1) = f(h_1, m_2^2) &= h_2, \\ &\vdots \\ f(h_{r-1}, m_r^1) = f(h_{r-1}, m_r^2) &= h_r. \end{aligned}$$

Now it is easy to check that, $H^f(m_1^{i_1}||\cdots||m_r^{i_r}) = h_r$ where, $i_1, \cdots, i_r \in \{1,2\}$. So, we have 2^r-way collision by finding only r successive 2-way collisions.

Application of Multicollision.

In the same paper by Joux [6], it was shown that how multicollision can be used. Let $H : D \to \{0,1\}^n$ be some hash function which has $2^{n/2}$-way multicollision with time complexity $q(n)$. Then for any other function $G : D \to \{0,1\}^n$, $H(\cdot)||G(\cdot)$ has collision in the time complexity $q(n)$. So if $q(n)$ is lower than 2^n (which is the complexity for birthday attack on $H||G$) then we have a collision-attack for $H||G$ with complexity $q(n)$ assuming one query for one computation of $G(\cdot)$. Basically, we will search a collision for $G(\cdot)$ from the multicollision set of $H(\cdot)$ and a collision is guaranteed in the random oracle model if the search space has desirable size (here $2^{n/2}$). So it is not desirable to use a hash function having multicollision for extending the size of the output.

4.1 Multicollision secure Hash Functions.

In [10] Lucks constructed a double-pipe hash function which is secure against multicollision attack. The idea of the construction is to first construct a secure double length hash function H^{C^p} (for some permutation p) and then compress it into a single length output by applying another compression function. the multicollision attack has very limited applications in cryptography except to find a collision on bigger length hash functions. As we have already showed the double length hash function used by Lucks is secure against collision attack so we may not need to study the multicollision security of the single length hash function. So the last invocation of the compression function is not important in the view of multicollision security.

Also it can be shown that if we consider any one part of the final hash value of double length hash function then it is secure against multicollision attack provided the underlying double length compression function is collision secure. So, consider the the hash function $H'(M) = [H^{C^p}(M)]_L$ where, $[X]_L$ denotes the first n bits of $2n$-bit X. Any r-way collision on H' can be reduced to a collision on C^p or r-way collision on C. So under the assumption 1 and 2 and the permutation p does not have many fixed point then H' defined as above is secure against multicollision attack.

5 Conclusion

In this paper we have discussed the security of a class of double block length hash functions. This class includes the construction proposed by [5], [10]. In fact, it almost includes all possible definitions of double length compression function from a single length compression function having rate 1/2. The security study is generic in nature. We have also pointed out the multicollision security of a single length hash function constructed from a double length compression function depends on the collision security of the double length compression function. As the multicollision security is important only (it has other very limited application) for getting collision, there is no need to study the multicollision security of the construction defined like [10]. Even if we use this hash function we can ignore the last invocation as it would be still multicollision secure.

References

1.
2. I. B. Damgård. *A design principle for hash functions*, Advances in Cryptology - Crypto'89, Lecture Notes in Computer Sciences, Vol. 435, Springer-Verlag, pp. 416-427, 1989.
3. H. Finney. *More problems with hash functions.* The cryptographic mailing list. 24 Aug 2004. http://lists.virus.org/cryptography-0408/msg00124.html.
4. M. Hattori, S. Hirose and S. Yoshida. *Analysis of Double Block Lengh Hash Functions.* Cryptographi and Coding 2003, LNCS 2898.
5. S. Hirose. *Provably Secure Double-Block-Length Hash Functions in a Black-Box Model*, to appear in ICISC-04.
6. A. Joux. *Multicollision on Iterated Hash Function.* Advances in Cryptology, CRYPTO 2004, Lecture Notes in Computer Science 3152.
7. L. Knudsen, X. Lai and B. Preneel. Attacks on fast double block length hash functions. *J.Cryptology, vol 11 no 1, winter 1998.*
8. 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.*
9. 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.* Lecture Notes in Computer Science, ACISP-2003.
10. S. Lucks. *Design principles for Iterated Hash Functions*, e-print server : http://eprint.iacr.org/2004/253.
11. 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.
12. B. Preneel, R. Govaerts, and J. Vandewalle. *Hash functions based on block ciphers:A synthetic approach*, Advances in Cryptology-CRYPTO'93, LNCS, Springer-Verlag, pp. 368-378, 1994.
13. T. Satoh, M. Haga and K. Kurosawa. *Towards Secure and Fast Hash Functions.* IEICE Trans. VOL. E82-A, NO. 1 January, 1999.