Security Analysis of a 2/3-rate Double Length Compression Function in The Black-Box Model

Mridul Nandi[1], Wonil Lee[2], Kouichi Sakurai[2], and Sangjin Lee[3]

[1] Applied Statistics Unit,
Indian Statistical Institute, Kolkata, India
mridul_r@isical.ac.in
[2] Faculty of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, Japan
wonil@itslab.csce.kyushu-u.ac.jp
sakurai@csce.kyushu-u.ac.jp
[3] Center for Information Security Technologies (CIST),
Korea University, Seoul, Korea sangjin@cist.korea.ac.kr

Abstract. In this paper, we propose a 2/3-rate double length compression function and study its security in the black-box model. We prove that to get a collision attack for the compression function requires $\Omega(2^{2n/3})$ queries, where n is the single length output size. Thus, it has better security than a most secure single length compression function. This construction is more efficient than the construction given in [8]. Also the three computations of underlying compression functions can be done in parallel. The proof idea uses a concept of computable message which can be helpful to study security of other constructions like [8], [14], [16] etc.

1 Introduction

A hash function is a function from an arbitrary domain to a fixed domain. Hash functions have been popularly used in digital signatures schemes, public key encryption, message authentication codes etc. To have a good digital signature schemes or public key encryption, it is required that hash function should be collision resistant or preimage resistant. Intuitively, for a collision resistant hash function H it is hard to find two different inputs $X \neq Y$ 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. Besides this condition, one should define hash function on an arbitrary domain. Usually, 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 an arbitrary domain by iterating the compression function several times. The most popular method is known as MD-method [2], [15] with the classical iterations. We first pad the input message by some strings and the string representing the length so that the length of the padded message becomes multiple of m and it avoids some trivial attacks. Now 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 $H^f(h_0, \cdot) : (\{0,1\}^m)^* \to \{0,1\}^n$ is 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** $h_l$
```

There are many constructions of the underlying compression functions e.g. SHA-family i.e. SHA-0, SHA-1, SHA-256 [17], MD-family i.e. MD-4, MD-5, RIPEMD [5] [19] etc. There are several collision attacks [3] [4] [10] [21] on some of these compression functions. Also people tried to design a compression function from a block cipher known as PGV hash functions [18]. In [1], [13], the security of the PGV hash functions were studied in the black box model of the underlying block cipher.

Nowadays, people are also interested in designing a bigger size hash function to make the birthday attack infeasible. One can do it by just constructing a compression function like SHA-512. The other way to construct it from a smaller size compression function. In the later case one can study the security level of the bigger hash function assuming some security level of underlying compression functions. People also try to use block ciphers to extend the output size. There are many literatures where the double block length hash function were studied e.g. [7], [8], [11], [12], [16], [20] etc.

1.1 Motivation and Our Results.

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. Thus to make birthday attack infeasible we need to construct a compression function with larger size output. In this paper, we construct a double length compression function from a single length compression function or a block cipher. We use three invocations of independent single length compression functions or block ciphers to hash two message blocks. Thus, the rate of the compression function is 2/3. We also prove the security level is $\Omega(2^{2n/3})$ and prove the bound is tight by showing an attack on this compression function with complexity $O(2^{2n/3})$.

2 Preliminaries

2.1 Some Results on Probability Distribution.

In this paper we will be interested in random variables taking values on $\{0,1\}^n$ for some integer $n > 0$. A random variable X is uniformly distributed over the set $\{0,1\}^n$ if $\Pr[X = x] = 1/2^n$ for all $x \in \{0,1\}^n$. We use the notation $X \sim U_n$ to denote a uniform random variable X. We say random variables $X_1, \cdots, X_k$ are independent if the joint distribution of $(X_1, \cdots, X_k)$ is the product of marginal distributions of X_i's. So if $X_1, \cdots, X_k$ are independent and $X_i \sim U_n$ for all i, then $\Pr[X_1 = x_1, \cdots, X_k = x_k] = 1/2^{nk}$ for all $x_i \in \{0,1\}^n$. We describe this case by the notation $(X_1, \cdots, X_k) \models U_n$. In this case, it is easy to see that $X_1 || \cdots || X_k \sim U_{nk}$ i.e. uniformly distributed over the set $\{0,1\}^{nk}$. The n-bit string $0 \cdots 0$ (known as a zero string) is denoted by $\mathbf{0}$. For a binary vector $l = (l_1, \cdots, l_k) \in Z_2^k$, l^T denotes the transpose vector of l. Given a set of k random variables $X = (X_1, \cdots, X_k)$, $X \cdot l^T = l_1 X_1 \oplus \cdots \oplus l_k X_k$, where $0X = \mathbf{0}$ and $1X = X$. For a binary matrix $L_{k \times r} = [l_1^T, \cdots, l_r^T]$, $X \cdot L$ denotes the random vector $(X \cdot l_1^T, \cdots, X \cdot l_r^T)$. Now we state a simple fact from the probability theory.

Proposition 1. *If $X = (X_1, \cdots, X_k) \models U_n$ then for any vector $l \in Z_2^k$ with $l \neq 0$, the random variable $X \cdot l^T \sim U_n$. For any matrix $L_{k \times r}$ with rank $r (\leq k)$, the random vector $X \cdot L \models U_r$.*

Example 1. Take $r = 2$ and $k = 3$. Let $l_1 = (0,1,1)$ and $l_2 = (1,1,0)$ then $X \cdot L = (X_2 \oplus X_3, X_1 \oplus X_2)$, where $X = (X_1, X_2, X_3) \models U_n$. By the above Proposition 1, both $X_2 \oplus X_3$ and $X_1 \oplus X_2$ are independently and uniformly distributed on $\{0,1\}^n$ since the matrix $L = [l_1^T, l_2^T]$ has rank 2.

2.2 (Independent) Random Functions and Permutations.

A random function $f : D \to R$ taking values as random variable satisfy the following conditions

1. for any $x \in D$, $f(x)$ has uniform distribution on R.
2. 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.

More precisely, one can not construct a single function which is a random function. Consider a class of functions $Func^{D \to R}$ which consists of all function from D to R. When one says that f is a random function it means that f is drawn randomly from $Func^{D \to R}$. However, to study some security property one assume a single function as a random function. Although, it is not theoretically possible this can be meaningful for some types of adversary who only query the function f and do not explore the internal structure of f. We say two functions f_1 and f_2 from D to R are independent random functions if they are random functions and for any $k, l > 0$ and k distinct elements $x_1^1, \cdots, x_k^1 \in D$ and l distinct elements $x_1^1, \cdots, x_l^1 \in D$ the random variables $f_1(x_1^1), \cdots, f_1(x_k^1), f_2(x_1^2), \cdots, f_2(x_l^2)$ are independently distributed. Similarly one can define that f_1, f_2 and f_3 are independent random functions and so on.

Similarly one can define a 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 $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$

by a family of permutations if for each $K \in \{0,1\}^k$, $E(K,\cdot)$ is a permutation on n-bit strings. We say a family of permutations $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ 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.

2.3 Some Attacks on Hash/Compression Functions

In this paper, we mainly study the collision resistant hash function but for the sake of completeness, we want to state the preimage resistance also. Given a compression function $f : \{0,1\}^N \rightarrow \{0,1\}^n$, it is called collision resistant if it is hard to find two inputs $x \neq y$ such that $f(x) = f(y)$. It is said to be a preimage resistant compression function if given a random $y \in \{0,1\}^n$, it is hard to find x such that $f(x) = y$. In the case of a random function f, the best attack is birthday attack which takes $\mathrm{O}(2^{n/2})$ or $\mathrm{O}(2^n)$ queries of f for collision or preimage attack, respectively. For the hash function based on a compression function, we can similarly define collision and preimage attack. But, here the initial value of the hash function is fixed and given to the adversary before starting the attack. There are also free-start collision and preimage attack where the adversary can choose the initial value. It can be easily shown that the free start attack on hash function is equivalent to the corresponding attack on the underlying compression function.

3 A New Double Length Compression Function

Let $f_i : \{0,1\}^{2n} \rightarrow \{0,1\}^n$ be independent random functions, $i = 1, 2, 3$. Define, $F : \{0,1\}^{3n} \rightarrow \{0,1\}^{2n}$, where $F(x,y,z) = (f_1(x,y) \oplus f_2(y,z)) \mid\mid (f_2(y,z) \oplus f_3(z,x))$ with $|x| = |y| = |z| = n$. We also write $F = F_1 \mid\mid F_2$, where $F_1(x,y,z) = f_1(x,y) \oplus f_2(y,z)$ and $F_2(x,y,z) = f_2(y,z) \oplus f_3(z,x)$ (see Figure 1).

Theorem 1. *$(F(x_1,y_1,z_1), F(x_2,y_2,z_2)) \models U_{2n}$, $(x_1,y_1,z_1) \neq (x_2,y_2,z_2)$. In particular, $\forall M \neq N$ and Z, $Pr[F(M) = F(N)] = \frac{1}{2^{2n}}$ and $Pr[F(M) = Z] = \frac{1}{2^{2n}}$.*

Proof. Let $M = (x_1,y_1,z_1) \neq (x_2,y_2,z_2) = N$. Assume that $x_1 \neq x_2$, $y_1 = y_2 = y$ (say), and $z_1 = z_2 = z$ (say). For the other cases, we can prove the result similarly. To prove that $(F(M), F(N)) \models U_{2n}$, it is enough to prove that $(F_1(M), F_2(M), F_1(N), F_2(N)) \models U_n$. Since f_1, f_2 and f_3 are independent random functions, $f_1(x_1,y), f_1(x_2,y), f_2(y,z), f_3(z,x_1)$ and $f_3(z,x_2)$ are independently distributed. Thus, by Proposition 1 (in Section 2.1) we know that $f_1(x_1,y) \oplus f_2(y,z)$, $f_3(z,x_1) \oplus f_2(y,z)$, $f_1(x_2,y) \oplus f_2(y,z)$ and $f_3(z,x_2) \oplus f_2(y,z)$ are independently distributed. So we have proved the proposition. □

3.1 The Model for Adversary and Computable Message.

In this subsection, we state briefly how an adversary works in the random oracle model. Adversary can ask the oracles f_1, f_2 and f_3 i.e. he can submit (a,b) to any

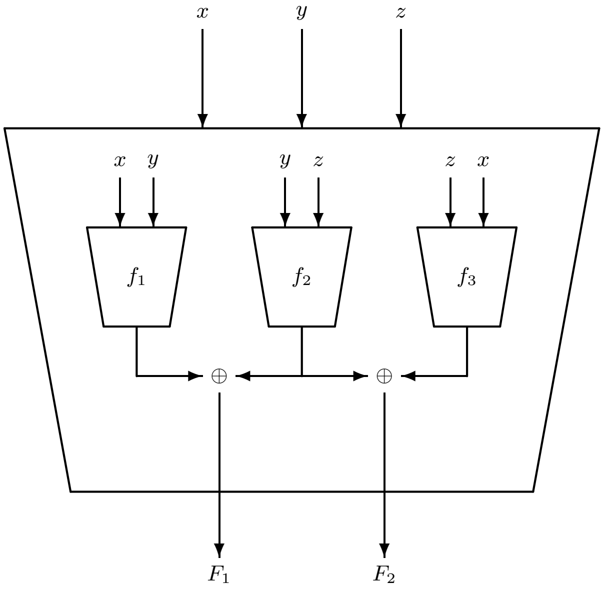

Fig. 1. A double length compression function

one of the oracles f_i and he will get a response c such that $f_i(a, b) = c$. We restrict the number of queries for each f_i by at most q. Finally he outputs a pair $M \neq N$ (for collision attack of F) such that both $F(M)$ and $F(N)$ can be computed from the set of queries he made. We say adversary wins if $F(M) = F(N)$.

Definition 1. (Computable message)
Let $\mathcal{Q}_1 = \{(x_i^1, y_i^1)\}_{1\leq i\leq q}$, $\mathcal{Q}_2 = \{(y_i^2, z_i^2)\}_{1\leq i\leq q}$ and $\mathcal{Q}_3 = \{(z_i^3, x_i^3)\}_{1\leq i\leq q}$ be the three sets of queries for the random oracles f_1,f_2 and f_3, respectively. We say a message $M = (x, y, z)$ is computable *if $(x, y) \in \mathcal{Q}_1, (y, z) \in \mathcal{Q}_2$ and $(z, x) \in \mathcal{Q}_3$.*

Thus it is easy to observe that a message M is *computable* if and only if $F(M)$ can be computed from the set of queries. Because of Theorem 1 of this section if we can bound the number of computable message by some number say Q then it is easy to check that the adversary will get a collision with probability at most $Q(Q-1)/2^{2n+1}$. In case of preimage attack, the probability is at most $Q/2^{2n}$. Thus the question reduces how to get an upper bound of the number of computable messages from any set of queries $\mathcal{Q}_1, \mathcal{Q}_2$ and $\mathcal{Q}_3$ where $|\mathcal{Q}_i| \leq q$, $1 \leq i \leq 3$. To have an upper bound we can convert our problem into a combinatorial graph theoretical problem. In the next subsection we study that problem.

3.2 A Combinatorial Graph Theoretical Problem

Tripartite Graph. A graph $G = (V, E)$ is known as a tripartite graph if $V = A \sqcup B \sqcup C$ (disjoint union) and for any edge $\{u, v\} \in E$ either $u \in A, v \in B$

or $u \in A, v \in C$ or $u \in B, v \in C$ (see Figure 2). Thus there are no edges between vertices in A or between vertices in B or between vertices in C. We use the notation $e(A, B, G)$ (or simply $e(A, B)$) for the set of edges between A and B. Similarly we can define $e(B, C)$ and $e(A, C)$. Note that for every triangle $\triangle$ in G, the vertices of $\triangle$ are from A, B and C with one vertex from each one. Now we can state the following problem.

Problem : Given an integer q, what is the maximum number of triangles of a tripartite graph G on $A \sqcup B \sqcup C$ such that $|e(A, B)|, |e(B, C)|, |e(A, C)| \leq q$.

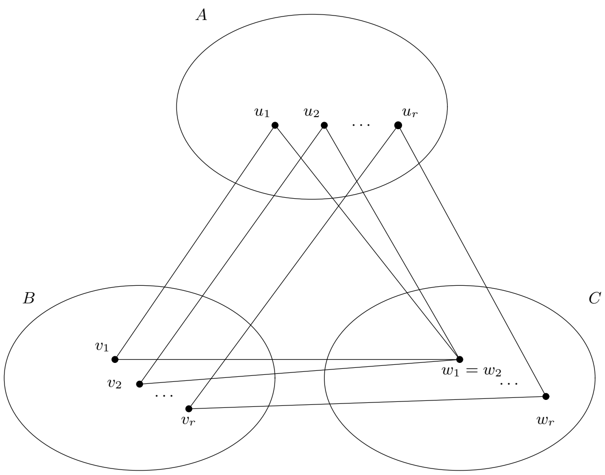

Fig. 2. A tripartite graph

We first prove a Proposition which will be useful for finding the upper bound of the problem stated above.

Proposition 2. *Let G be a tripartite graph on $A \sqcup B \sqcup C$ such that $|e(A, B)| \leq q$. For a set of edges $E_{BC} = \{v_1 w_1, \cdots, v_r w_r\} \subseteq e(B, C)$ such that v_i's are distinct vertices from B, the number of triangles in G whose one of the sides is from E_{BC} is at most q.*

Proof. Let T be the set of triangles in G one of whose side is from E_{BC}. Now we can define an injective map ρ from the set T to the set $e(A, B)$. Given a triangle $uvw \in T$ with $vw \in E_{BC}$ and $v \in B$, define $\rho(T) = uv$. Obviously the map $\rho : T \to e(A, B)$ is well defined. To see it is an injective map we just note that all v_i's are distinct (see Figure 2). So, $\rho(uvw) = \rho(u'v'w')$ with $v, v' \in B$

and $u, u' \in A$ implies that $u = u'$ and $v = v'$. Since $v = v'$ and $vw, v'w' \in E_{BC}$ implies that $w = w'$. So the two triangle uvw and $u'v'w'$ are identical. $\square$

Thus if we can divide the set $e(B, C)$ into r sets E^i_{BC}, $1 \leq i \leq r$ such that each E^i_{BC} has the property stated in the Proposition 2 for B or C then the number of triangles in G will be at most $r \times q$. Assume $q = n^2$. We will show now that we can always divide $e(B, C)$ into $2n$ many such sets. Thus upper bound of triangles is $2n^3$. Let $G = (V, E)$ be a bipartite graph on $B \sqcup C$ with $|E| \leq n^2$. We say a set of edges $E' = \{u_1v_1, \cdots, u_rv_r\}$ in G is *good* if all $u_i \in B$ or C and u_i's are distinct.

Proposition 3. *Given a bipartite graph $G = (V, E)$ with $V = A \sqcup B$ and $|E| \leq n^2$ we can divide E into at most $2n$ good sets of edges.*

Proof. The proof is by induction on n. Assume $|E| > (n - 1)^2$. Thus we can find a set B or C where number of vertices with positive degree is at least n. Without loss of generality we assume that the set B has n vertices $u_1, \cdots, u_n$ with degree at least one. Let $u_iv_i \in E$, where $v_i \in C$, $1 \leq i \leq n$. Note that v_i's are not necessarily distinct. So $E_1 = \{u_1v_1, \cdots, u_nv_n\}$ is a good set. Now consider $E - E_1$. Again, if $|E - E_1| \leq (n - 1)^2$ then we can apply induction hypothesis and we will get $2(n-1)$ good sets for $E-E_1$. So the result is true. If $|E - E_1| > (n - 1)^2$. Again we can find a good set E_2 of size at least n by using similar argument. Now $|E| - |E_1| - |E_2| \leq n^2 - 2n \leq (n - 1)^2$. So by induction hypothesis we can get $2(n - 1)$ good sets in $E - (E_1 \cup E_2)$. Thus we have $2n$ good sets whose union is the whole set E. For $n = 1$ the result is trivial. $\square$

Theorem 2. *Given a positive integer n, the number of triangles of any tripartite graph G on $A \sqcup B \sqcup C$ such that, $|e(A, B)|, |e(B, C)|, |e(A, C)| \leq n^2$ is at most $2n^3$.*

The proof of the above theorem is immediate from Proposition 2 and 3. In fact we have better and sharp bound which is n^3. The proof is given by one of the anonymous referee. He proved a general statement as follow :

Theorem 3. *Given a positive integer n, the number of triangles of any tripartite graph G on $A \sqcup B \sqcup C$ is at most $(XYZ)^{1/2}$ such that, $|e(A, B)| \leq X, |e(A, C)| \leq Y$ and $|e(B, C)| \leq Z$. In particular, when $X = Y = Z = n^2$ the number of triangle is at most n^3.*

Proof. Let x_a be the number of edges from the vertex $a \in A$ between A and B. Similarly, y_a is the number of edges between A and C from the vertex a. Obviously,

$$\textstyle\sum_{a \in A} x_a = X \text{ and } \sum_{a \in A} y_a = Y.$$

Now the number or triangles containing the vertex a is bounded by $\min\{Z, x_ay_a\}$. Since a triangle containing the vertex a is determined by two edges containing a or determined by the opposite edge of a. But we have, $\min\{Z, x_ay_a\} \leq \sqrt{Zx_ay_a}$. Thus the number of triangles is bounded by

$\sum_a \sqrt{Zx_ay_a} = \sqrt{Z}\sum_a \sqrt{x_ay_a} \leq \sqrt{Z}.\sqrt{\sum_a x_a)(\sum_a y_a)} = \sqrt{XYZ}.$ □

Here, we use the Cauchy-Schwartz inequality. If we take $X = Y = Z = n^2$ then the number of triangle is bounded by n^3. We have an example where the number of triangles is exactly n^3 namely we take a complete tripartite graph. That is we have three disjoint set of vertices A, B and C each of size n. Consider all possible edges between A and B, between A and C and between B and C. Obviously the number of edges between A and B or B and C or A and C are exactly n^2. The number of triangles is n^3 since any vertex from A, from B and from C will contribute a triangle.

3.3 Security Study of The Double Length Compression Function.

We have three disjoint vertices set each of size 2^n. In particular, take $A = \{0,1\}^n \times \{1\}, B = \{0,1\}^n \times \{2\}$ and $C = \{0,1\}^n \times \{3\}$. We can correspond each query by an edge of a tripartite graph on $A \sqcup B \sqcup C$ as follow: given a query (x, y) on f_1 we add an edge $\{(x, 1), (y, 2)\}$. The number 1,2 and 3 are used to make A, B and C disjoint. Similarly we can add edges for queries on f_2 and f_3. Now it is easy to note that a computable message corresponds to a triangle in the graph G. Thus the number of computable message is equal to the number of triangles. Also the adversary can ask at most q queries to each f_i and hence the number of edges between A and B or B and C or A and C are at most q. Thus by the Theorem 2 we have at most $2q^{3/2}$ computable inputs for F. Thus the winning probability is bounded by $2q^{3/2}(2q^{3/2} - 1)/2^{2n+1}$. So the number of queries needed to get a collision is $\Omega(2^{2n/3})$. We will show an attack which makes $\mathrm{O}(2^{2n/3})$ queries to get a collision on F. So the security bound is tight. For preimage attack the winning probability is bounded by $q^{3/2}/2^{2n}$, thus the number of queries needed to get a preimage is $\Omega(2^{2n/3})$. This bound is also tight and one can find an attack very similar to the following collision attack.

A Collision Attack on F. The attack procedure is very much similar with the security proof. We first choose $2^{n/3}$ values of x_i, y_i and z_i independently, $1 \leq i \leq 2^{n/3}$. Now we will query $f_1(x_i, y_j)$ for all $1 \leq i, j \leq 2^{n/3}$. Thus we have to make $2^{2n/3}$ queries of f_1. Similarly, we query for f_2 and f_3. Now we have 2^n computable inputs and check whether there is any computable collision pair.

Remark 1. It is easy to note that, in the security proof of F we do not use the fact that $|x| = |y| = |z| = n$. In fact, if we have $f_i : \{0,1\}^{3n} \to \{0,1\}^n$, $1 \leq i \leq 3$ and define $F(x, y, z) = (f_1(x, y||0^n) \oplus f_2(y, z))||f_2(y, z) \oplus f_3(x, z)$, where $|x| = |y| = n$ and $|z| = 2n$ then we have same security level as in the previous definition. The proof for that is exactly same with the previous proof. Note that, $F : \{0,1\}^{4n} \to \{0,1\}^{2n}$. So we use two message block in each round function F and three parallel computations of f_i's are made. So rate of this compression function is 2/3.

Remark 2. One can define a function $F : \{0,1\}^{4n} \rightarrow \{0,1\}^{2n}$ by $F(x, y, z_1, z_2) = (f_1(x, y, z_1) \oplus f_2(y, z_1, z_2))||(f_2(y, z_1, z_2) \oplus f_3(x, z_1, z_2))$ hoping for more security. But an attack can be shown with complexity $\mathrm{O}(2^{2n/3})$. First, fix some z_1 and then choose $2^{n/3}$ values of x, y and z_2 independently. By the same argument like previous attack, it still has 2^n computable messages and hence we will expect to have a collision on F.

3.4 Block-Cipher based Double length Compression function

Let $E : \{0,1\}^{2n} \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher with $2n$-bit keys. Define a function $f : \{0,1\}^{3n} \rightarrow \{0,1\}^n$, as follow :

$$f(x, y, z) = E_{x||y}(z) \oplus z,$$

$|x| = |y| = |z| = n$, Here, we will assume $E(\cdot)$ as a family of random permutations. More precisely, given any integer $s > 0$, and s distinct keys $k_1, \cdots, k_s \in \{0,1\}^{2n}$, the functions $E_{k_1}, \cdots, E_{k_s}$ are independent random permutations. It is easy to check that if we sacrifice two bits then we can get three instances of f which will be independent to each other. That is we can define, $f_i(x, y, z) = E_{<i>||x||y}(z) \oplus z$, where $< i >$ is the two bit binary representation of i and $|x| = n - 2, |y| = |z| = n$. Then we can define similarly the double length compression function $F : \{0,1\}^{4n-2} \rightarrow \{0,1\}^{2n}$ i.e. $F(x, y, z, t) = (f_1(x, y, z) \oplus f_2(x, z, t)) \; || \; (f_2(x, z, t) \oplus f_3(x, y, t))$, where $|x| = n - 2, |y| = |z| = |t| = n$ (see Figure 3).

Here an adversary can ask both E and E^{-1} query. Let $\{(k, a, b)\}$ be a query response triple (in short q-r triple), where $E_k(a) = b$. We can assume that, the first two bits of k not equal to 00 otherwise the query is useless to get a collision attack. Now, if the first two bits of k is $< i >$ with $i \neq 0$ and say k' is the remaining $2n - 2$ bits then,

$$f_i(k', a) = a \oplus b \text{ if and only if } (k, a, b) \text{ is a q-r triple.}$$

Thus given a set of q q-r triples we can have at most q computation of f_i for each i and hence we can have at most $2q^{3/2}$ computable messages. Now it is enough to find a bound of $\Pr[F(M) = F(N)]$, where $M \neq N$.

Now consider $M = (x_1, y_1, z_1, t_1) \neq (x_2, y_2, z_2, t_2) = N$. We assume that $x_1 = x_2 = x$, $y_1 = y_2 = y$, $z_1 \neq z_2$ and $t_1 \neq t_2$. For the other cases one can study similarly. Now, the event $F(M) = F(N)$ is equivalent to

$$f_1(x, y, z_1) \oplus f_2(x, z_1, t_1) = f_1(x, y, z_2) \oplus f_2(x, z_2, t_2),$$
$$f_3(x, y, t_1) \oplus f_2(x, z_1, t_1) = f_3(x, y, t_2) \oplus f_2(x, z_2, t_2).$$

To compute the probability of happening above we can first condition on each term except $f_1(x, y, z_1)$ and $f_3(x, y, t_1)$. Thus the conditional event would be $f_1(x, y, z_1) = a$ and $f_3(x, y, t_1) = b$ for some string a and b. We now have,

$$\Pr[f_1(x, y, z_1) = a, f_3(x, y, t_1) = b | f_2(x, z_1, t_1) = a_1, \cdots, f_3(x, y, t_2) = a_4]$$
$$\leq 1/2^{n-1} \times 1/2^{n-1}$$

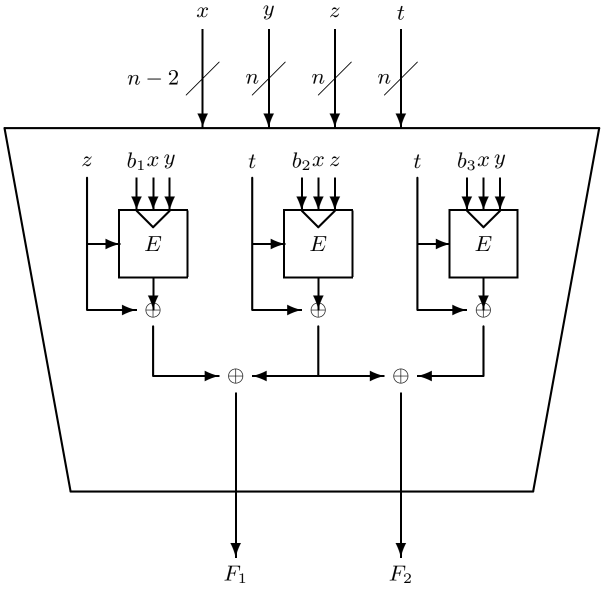

Fig. 3. A double length compression function based on a double-key length block cipher ($b_i :=< i >$)

for some $a_1, \cdots, a_4$. Thus, probability of collision for a given pair is bounded by $1/2^{2n-2}$ and hence success probability after q many queries is bounded by $2q^3/2^{2n-2}$. Note that $2q^{3/2}$ is the maximum number of computable messages and hence the number of pairs of computable messages is at most $2q^3/2^{2n-2}$. Thus we need $\Omega(2^{2n/3})$ many queries to have non-negligible success probability.

4 Future Work and Conclusion

This paper deals with a new double length compression function which can uses three parallel computations of a compression function or a double key block cipher. Although the security of this compression function is not maximum possible (i.e. there is a better attack than birthday attack) the lower bound of the number of queries is $\Omega(2^{2n/3})$. So it has better security than a most secure single length compression function. Also the security is proved for compression function. So the hash function based on the compression function has same security level for free-start collision attack. So it would be interesting to study the security level for collision attack. Also one can try to design an efficient (if possible, rate-1) double block length hash function which is maximally secure against collision attack even if the underlying compression function is not secure.

Acknowledgement. The authors would like to thank the anonymous referees of FSE-05 who gave helpful comments and provide a solution of the graph theoretical problem. This research was supported by the MIC(Ministry of Information and Communication), Korea, supervised by the IITA(Institute of Information Technology Assessment). The second author was supported by the 21st Century COE Program 'Reconstruction of Social Infrastructure Related to Information Science and Electrical Engineering' of Kyushu University.

References

1. 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.
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. Dobbertin.*Cryptanalysis of MD4.* Fast Software Encryption, Cambridge Workshop. Lecture Notes in Computer Science, vol 1039, D. Gollman ed. Springer-Verlag 1996.
4. H. Dobbertin.*Cryptanalysis of MD5* Rump Session of Eurocrypt 96, May. http//www.iacr.org/conferences/ec96/rump/index.html.
5. 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.
6. H. Finney. *More problems with hash functions.* The cryptographic mailing list. 24 Aug 2004. http://lists.virus.org/cryptography-0408/msg00124.html.
7. M. Hattori, S. Hirose and S. Yoshida. *Analysis of Double Block Lengh Hash Functions.* Cryptographi and Coding 2003, LNCS 2898.
8. S. Hirose. *Provably Secure Double-Block-Length Hash Functions in a Black-Box Model*, to appear in ICISC-04.
9. A. Joux. *Multicollision on Iterated Hash Function.* Advances in Cryptology, CRYPTO 2004, Lecture Notes in Computer Science 3152.
10. J. Kelsey. *A long-message attack on SHAx, MDx, Tiger, N-Hash, Whirlpool and Snefru.* Draft. Unpublished Manuscritpt.
11. L. Knudsen, X. Lai and B. Preneel. Attacks on fast double block length hash functions. *J.Cryptology, vol 11 no 1, winter 1998.*
12. 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.*
13. 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.
14. S. Lucks. *Design principles for Iterated Hash Functions*, e-print server : http://eprint.iacr.org/2004/253.
15. 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.
16. M. Nandi. *A Class of Secure Double Length Hash Functions..* e-print server : http://eprint.iacr.org/2004/296.

17. NIST/NSA. *FIPS 180-2 Secure Hash Standard*, August, 2002. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf
18. 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.
19. R. Rivest *The MD5 message digest algorithm.* http://www.ietf.org/rfc/rfc1321.txt
20. T. Satoh, M. Haga and K. Kurosawa. *Towards Secure and Fast Hash Functions.* IEICE Trans. VOL. E82-A, NO. 1 January, 1999.
21. B. Schneier. *Cryptanalysis of MD5 and SHA.* Crypto-Gram Newsletter, Sept-2004. http://www.schneier.com/crypto-gram-0409.htm#3.