Security Analysis of Zipper Hash Against Multicollisions Attacks

Nasour Bagheri
Electrical Engineering Department,
Shahid Rajee Teacher Training University (SRTTU)
Tehran, Iran
nbagheri@srttu.edu

Abstract— **In this paper, the existence of multicollisions in Zipper Hash structure, a new Hash structure which was introduced to strengthen the iterated Hash structures, is presented. This study shows that finding multicollisions, i.e. 2k-way collision, in this Hash structure is not much harder than finding such multicollisions in ordinary Merkle - Damgard (MD) structure. In fact, the complexity of the attacks is approximately n/2 times harder than what has been found for MD structures. Then, these large multicollisions are used as a tool to find D-way preimage for this structure. The complexity of finding 2K-way multicollisions and 2k-way preimages are $O\left(k\times(n/2+1)\times 2^{n/2}\right)$ and $O\left(k\times n/2\times 2^{n/2}+2\times 2^{n}\right)$ respectively. Similar to what has been proved by Joux for MD, it is shown in this paper that this structure could not be used to create a Hash function with 2n-bit length by concatenating this structure with any other Hash structure by Hash's output length of n-bite. It is also shown that time complexity of finding a collision for this concatenated structure is $O\left((n/2)^2\times 2^{n/2}\right)$ which is much smaller than what was expected from generic-birthday attack which would be $\Omega\left(2^n\right)$. In addition, it is shown that increasing the number of rounds of this Hash function can not improve its security against this attack significantly and the attacker can find multicollisions on this Hash function which means that this Hash function has a structural flaw.**

***Keywords-** Zipper Hash Structure; Hash function; multicollision attack; Joux attack; preimage attack; r-way collision*

I. INTRODUCTION

A. Background

In a recent paper by Joux [1], it was shown that there is a 2^k-way collision attack for the classical iterated Hash function based on a compression function, $f:\{0,1\}^{m+n}\rightarrow\{0,1\}^n$, where the attack has a complexity of $O\left(k\times 2^{n/2}\right)$. This complexity is much smaller than the complexity for the generalized birthday attack which is $O\left(2^{\frac{n\left(2^k-1\right)}{2^k}}\right)$ [2]

This is the basic idea of Joux's attack. The main strategy of Joux's attack is to first find k successive collisions by performing k successive birthday attacks using a collision finder oracle machine C. The attack works as follows:

– Let h_0 be equal to the initial value IV of H.
– For i from 1 to k do:
• Call C and find M_i and M_i' such that
$f\left(h_{i-1},M_i\right)=f\left(h_{i-1},M_i'\right)$ *and* $M_i\neq M_i'$.
• Let $h_i=f\left(h_{i-1},M_i\right)$.
– Pad and output the 2^k messages of the form
$\left(m_1,m_2,\ldots,m_k,Padding\right)$ (where m_i is
one of the two blocks of M_i or M_i'.

Clearly, the 2^k different messages built as above all reach the same final value. A schematic representation of these 2^k messages together with their common intermediate Hash values is drawn in figure 1. Since birthday attack is a probabilistic attack, there is some positive reason that Joux's attack fails. In following proposition, it is shown that an attacker needs more effort to achieve k-way collision than what has been claimed by Joux.

Proposition 1. The complexity of finding 2^k–way collision in Joux's attack is $O\left(k\times 2^{k-1}\times 2^{n/2}\right)$.

Proof: Assume that probability of each chain in Joux's attack, shown in figure 1, equals to ε. Hence, the success probability of those messages to lead to collision would be ε^k. On the other hand, in ordinary 2^k-way collision birthday attack on an ideal Hash function, for the success probability of $\frac{1}{2}$, the complexity would be $O\left(2^{\frac{n(2^k-1)}{2^k}}\right)$ [2]. Hence, if one aims to compare the Joux's attack to the ordinary birthday attack, he should improve the Joux's attack in order for the success probability to reach $\frac{1}{2}$. To improve the success probability of

Joux's attack the attack may be repeated several times. The success probability for "*N*" repetition of attack, $\Pr_N$, could be determined as follows:

$$\Pr_N = 1-\left(1-2^{-k}\right)^N$$

where for $\Pr_N = \frac{1}{2}$ the value of N should be 2^{k-1}. Recall that the complexity of original Joux's attack is $O\left(k \times 2^{n/2}\right)$, the total complexity of finding k-way collision following the Joux's attack would be $O\left(k \times 2^{k-1} \times 2^{n/2}\right)$. Proof has been completed.

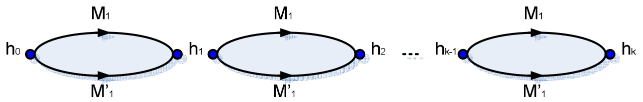

Fig. 1. Schematic representation of Joux multicollision construction.

After introducing this attack, many structure have been proposed to strength iterated structure against this type of attach such Zipper Hash [3], 3C and 3C+ [4,5], SFRH and MFRH[6], WPH and DPH [7], L-pipe[8], etc. Most of these structure have tried to strengthen against multicollision attack.

Among this new Hash structure, in this paper the strength of Zipper Hash [3] against multicollision attack is investigated.

B. *Related Work and Contribution*

In [9] Lin et. al have presented a multicollision attack for Zipper Hash. However, their attack work when the underling compression functions are weak while there is such assumption on compression function and the adversary has oracle access to compression functions. Hence, this Hash function is analyzed on random oracle model for compression functions and the results are more general in comparison with the results of [9]. In addition, it is shown that even if this structure is repaired, the attacker can find multicollision for this Hash function. It means that this Hash function is not suitable for general deployment and it has structural flaw.

C. *Paper organization*

In this paper an attack for this previously believed secure Hash structure is presented. For this purpose a brief description of Zipper Hash function is given in section 2. Section 3 contains an explanation of the attack for finding multicollision on this structure, whereas section 4 presents a preimage attack for it. In section 5 it is clarified why Zipper Hash could not be used as a primitive for 2n-bit concatenated Hash function. In section 6 it is shown that even if structure is repaired by employing more functions, the attacker can find multicollision on it, which means that it is not suitable for general deployment. Conclusion will be presented in section 7.

II. Zipper Hash Structure

The Zipper Hash structure can be considered as a general Hash function construction. To build an n-bit Hash function, two independent $(m,k)-bit$ compression functions f_0 and f_1 are needed. These functions can be seen as a $\{0,1\}^{k+m} \rightarrow \{0,1\}^k$ mapping. Same as all Hash structures, this one needs a padding function P and an initialization vector IV. For function P(x) for input x, it is guaranteed to return a padded value such that P(x) is a string that can be broken down into m-bit length blocks, and for all $M' \neq M, M' \| P(M') \neq M \| P(M)$. Moreover it uses a finalization function $g:\{0,1\}^k \rightarrow \{0,1\}^n$. Given all these pieces, the Zipper Hash function works as follows[3]:

1. Let $M_1,...,M_l$ be m-bit strings such that $M_1,...,M_l = M \| P(M)$.

2. h_1 is computed as $f_0(M_1, IV)$, and $h_2,...,h_l$ are computed iteratively as $h_i = f_0(M_i, h_{i-1})$.

3. h_1' is computed as $f_1(x_l, h_l)$, and $h_2',...,h_l'$ are computed iteratively as $h_i' = f_1(M_{l-i+1},' h_{i-1}')$.

4. Final output $H(M) = g(h_l')$.

Since existence of g function is not affecting the attack, in the rest of the paper, without losing generality, it is assumed that output length of functions f_0 and f_1 are equal to n therefore these functions can be seen as $\{0,1\}^{n+m} \rightarrow \{0,1\}^n$ mapping. Figure 2 depicts the Zipper Hash construction.

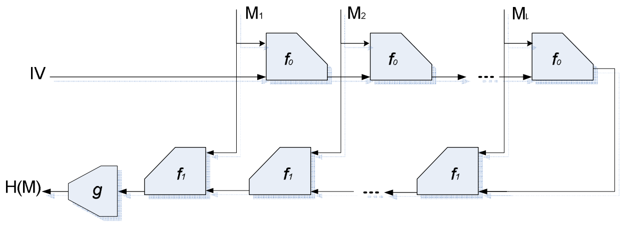

Fig. 2. The Zipper Hash structure.

III. Multicollision Attack on Zipper Hash

Zipper Hash structure was developed as a strengthen structure against multicolission attack. In this section, we show that constructing multicollisions in Zipper Hash function can be done in a efficient way. In particular, constructing 2^k -collisions costs $k \times n/2$ times as much as building ordinary 2-collisions and $n/2$ times as much as building multicollisions in ordinary MD structure. If collisions between messages of the same length are considered, the blocks of padding are identical, the padding process can be ignored and this is the case study in this paper. Moreover, if the intermediate Hash chaining values collide at some point in the Hash computation of two messages, the following values remain equal as soon as the ends of the messages are identical. Thus, on messages of the same length, collisions without the padding clearly lead to collisions with the padding.

Although this attack could be applied for any length of message block and chaining value, for simplicity of proof, it is assumed that the size of the message blocks is bigger than the size of the chaining values. However, the attack can be easily generalized. It is also assumed that one can access two collision finding machines C and C'. C is a machine that, given as input a chaining value h , outputs two different blocks M and M' such that $f_0(h,M)=f_0(h,M')$ and C' is such a one that given $2^{n/2}$ different messages of length $n/2$ blocks and h as chining value, finds the multiblocks Ψ and Ψ' among them such that $F_1(h',\Psi)=F_1(h',\Psi')$ where $F_1(h',\Psi)$ is ordinary compressed value of Ψ by applying f_1 as compress function and *h'* as initial value where illustrated in figure 3.

These collision finding machines may use the generic birthday attack or any specific attack based on a weakness of f_0 and f_1 multicollision attack on ordinary MD. The most relevant property is that C and C' should work properly for all chaining values. It's clear that most of these assumed conditions for the scenario investigated in this paper is similar to what considered in [1].

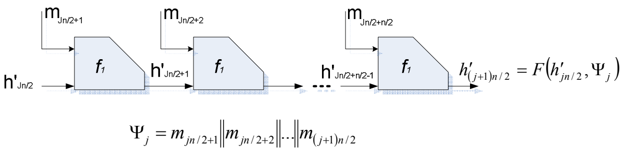

Fig. 3. The $F_1(h'_j,\Psi_j)$ function structure.

We now claim that we can generate 2^k equal collisions by only $O(k\times n/2)$ calls to the C oracle machine and $O(k)$ calls to C' oracle machine which is much less than what we expected from birthday paradox $O\left(2^{(n/2)\times\left((2^k-1)/2^k\right)}\right)$. Assuming that, l is the number of message blocks and is equal to $k\times n/2+1$, the attack works as follows:

1 Let h_0 be equal to the initial value IV of Zipper Hash.

2 For i from 1 to $k\times n/2$ do:

   a. Call C and find M_i and M'_i in such a way that $f_0(h_{i-1},M_i)=f_0(h_{i-1},M'_i)$.

   b. Set $h_i=f_0(h_{i-1},M_i)$.

3 since $M_{k\times n/2+1}$ is the padded message block determine $h_{k\times n/2+1}=f_0(h_{k\times n/2},M_{k\times n/2+1})$ and $h'_1=f_1(h_{k\times n/2+1},M_{k\times n/2+1})$

4 For j from 1 to k do:

   a. Call C' to find Ψ_j and Ψ_j' among $2^{n/2}$ different messages of $n/2$ blocks length and $h'_{(n/2)\times(j-1)+1}$ in such a way that $F_1(h'_{(n/2)\times(j-1)+1},\Psi_j)=F_1(h'_{(n/2)\times(j-1)+1},\Psi'_j)$.

5 Output the 2^k messages of form $(\psi_1,\ldots,\psi_k,M_{kn/2+1})$ where ψ_j will be one of the two multi blocks Ψ_j or Ψ_j'.

Obviously, all 2^k different messages generated in this way result to the same value of Hash. The third step of this algorithm finds $2^{k\times(n/2)}$ different multicollisions for the forward part of the Zipper Hash which uses f_0 as the compress function and the fourth step finds 2^k different multicollisions for the reverse part of the Zipper Hash which uses f_1 as the compress function. The time complexity of the attack is equal to all attempts done by the adversary. It is equal to $k\times(n/2)\times 2^{n/2}$ for finding a 2-way collision in the forward path of the attack and $k\times 2^{n/2}$ for finding a 2-way collision in the reverse path. Figure 4 is a schematic illustration of these 2^k multiblocks messages together with their common intermediate Hash values.

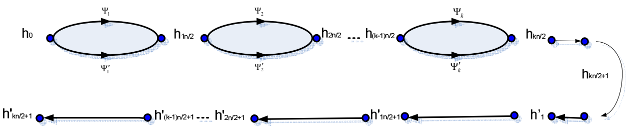

Fig. 4. Schematic representation of multicollision attack on Zipper Hash structure.

IV. FINDING K-WAY SECOND-PREIMAGE ATTACK ON ZIPPER HASH

In reference [1] Joux puts forward the attack method called k-way which is applicable for finding the second preimage of an output of a Hash function based on the MD structure. For a given Hash target value $Y=H(M)\in\{0,1\}^k$, at first the attackers find 2^r collisions on r-block messages $M_1,M_2,\ldots,M_{2^r}$ making $H_r=H(M_1)=H(M_2)=\ldots=H(M_{2^r})$. Then, to find the block M_{r+1} such that $f(H_r,M_{r+1})=Y$. In this way, the attackers succeed in finding 2^r second preimages with the message M. Obviously, the time complexity of this attack is $O(r2^{n/2}+2^n)$.

For the Hash function based on the Zipper Hash structure, we show that theadversary can find 2^r-way preimage and second preimage with cost of $O\left((r+1)\times(n/2)2^{n/2}+2\times 2^n\right)$. For this purpose, as mentioned in [3], it is assumed that g function is a identical function. The attack works as follows:

1. Fixed h_1 with some random value.
2. For i from 2 to $l = r\times n/2+1$ do:
   a. Call C and find M_i and M'_i in such a way that $f_0(h_{i-1}, M_i) = f_0(h_{i-1}, M'_i)$.
   b. Set $h_i = f_0(h_{i-1}, M_i)$
3. since $M_{k\times n/2+1}$ is the padded message block determine $h_{r\times n/2+2} = f_0(h_{r\times n/2+1}, M_{r\times n/2+2})$ and $h'_1 = f_1(h_{k\times n/2+2}, M_{k\times n/2+2})$
4. For j from 1 to k do:
   a. Call C' to find Ψ_j and $\Psi_j{}'$ among $2^{n/2}$ different messages of length $n/2$ blocks and $h'_{(n/2)\times(j-1)+1}$ in such a way that $F_1(h'_{(n/2)\times(j-1)+1}, \Psi_j) = F_1(h'_{(n/2)\times(j-1)+1}, \Psi'_j)$.
5. Find M_1 in such a way that $Y = g(f_1(h'_{r\times n/2+1}, M_1))$.
6. Find IV in such a way that $h_1 = f_0(IV, M_1)$.

Obviously, all 2^k different messages generated in this way lead to value Y as a Hash result. This procedure can be divided in two parts. Fist part of the attack is finding the 2^r-way collision whose cost is $r\times(n/2)\times 2^{n/2}$ and the second part, described in steps 5 and 6, is related to finding two preimages to guarantee the successes of the attack with cost 2×2^n. Hence, the total complexity of the attack equals to the claimed value.

By applying a similar procedure, the adversary can find a 2^r-way second preimage with identical cost of time complexity which is far from ideal value $2^r\times 2^n$.

Clearly this kind of attack can not be used for the original version of Zipper Hash structure which use prefixed IV value, but it shows that this structure is far from ideal structure.

V. ZIPPER HASH IN 2N-BIT CONCATENATED STRUCTURE

A natural construction to build large Hash values is to concatenate several smaller Hashes. For example, given two Hash functions F and G, it seems reasonable given a message M to form the large Hash value $\left(F(M)\|G(M)\right)$. In this construction, F and G can either be two completely different Hash functions or two slightly different instances of the same Hash function. In [1] Joux has shown that if at least one of these Hash function is a MD iterated Hash function, the complexity of finding a collision for this structure is slightly more than finding collision for one branch and it is equal to $O(n/2\times 2^{n/2})$.

The basic idea in this attack is to find a $2^{n/2}$-way collision for MD structure and find a collision among this $2^{n/2}$ different messages for the second Hash function. Clearly this collision is applicable to booth branches. A similar attack can be applied to find a collision on the "2n-bit" Hash construction $\left(F(M)\|G(M)\right)$ when one replaces either F or G, namely F, by the Zipper Hash structure. The attack complexity includes the complexity of finding $2^{n/2}$-way collision on Zipper Hash which is $O\left((n/2)\times(n/2)\times 2^{n/2}\right)$ plus the complexity of finding a collision on G which is $O(2^{n/2})$. Hence, the total complexity would be $O\left((n/2+1)\times(n/2)\times 2^{n/2}\right)$."

VI. INCREASING THE ROUNDS OF THE ZIPPER HASH

Assume that someone tries to protect the Zipper Hash by adding another layer to this structure. Figure 5 has illustrated this modified version of Zipper Hash. In general it is assumed that $f_2 \neq f_0$ and $f_2 \neq f_1$. The following theorem shows that this structure is vulnerable to multicollision attack.

To finding multicollision in this new structure we introduce an new oracle machine C'' which is such a one that given $2^{n/2}$ different messages of length $\left(\frac{n}{2}\right)^2$ blocks, i.e. $2^{n/2}$ messages of form $\left(\psi_1,\ldots,\psi_{\frac{n}{2}}\right)$ where ψ_j will be one of the two multi blocks Ψ_j or Ψ'_j, and h' as the chining value, to find the multiblock Υ and Υ' among them such that $F_2(h',\Upsilon)=F_2(h',\Upsilon')$ where $F_2(h',\Upsilon)$ is ordinary compressed value of Υ by applying f_2 as compress function and h' as initial value. These collision finding machines may use the generic birthday attack or any specific attack based on a weakness of f_2. The most relevant property is that C'' should work properly for nay chaining values.

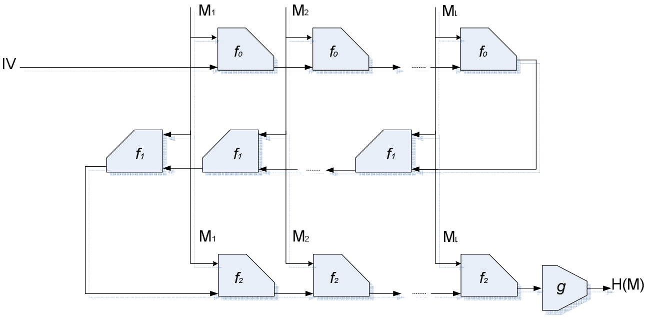

Fig. 5. Schematic representation of multicollision attack on Zipper Hash structure.

We now claim that we can generate 2^k equal collision by only $O\left(k\times(n/2)^2\right)$ calls to the C oracle machine, and $O\left(k\times {}^{n}\!/_{2}\right)$ calls to C' oracle machine and $O(k)$ calls to C' oracle machine which is much less than what we expected from birthday paradox $O\left(2^{(n/2)\times\left(\left(2^k-1\right)/2^k\right)}\right)$. Assuming that, l is the number of message blocks and is equal to $k\times n/2+1$, the attack works as follows:

1 Let h_0 be equal to the initial value IV of Zipper Hash.

2 For *i* from 1 to $k\times(n/2)^2$ do:

a. Call C and find M_i and M_i' in such a way $f_0\left(h_{i-1},M_i\right)=f_0\left(h_{i-1},M_i'\right)$.

b. Set $h_i=f_0\left(h_{i-1},M_i\right)$.

3 Since $M_{k\times n/2+1}$ is padded block fined $h_{k\times n/2+1}=f_0\left(h_{k\times n/2},M_{k\times n/2+1}\right)$ and $h_1'=f_1\left(h_{k\times n/2+1},M_{k\times n/2+1}\right)$

4 For *j* from 1 to $k\times {}^{n}\!/_{2}$ do:

a. Call C' to find Ψ_j and $\Psi_j{}'$ among $2^{n/2}$ different messages of $n/2$ blocks length and $h'_{(n/2)\times(j-1)+1}$ in such a way $F_1\left(h'_{(n/2)\times(j-1)+1},\Psi_j\right)=F_1\left(h'_{(n/2)\times(j-1)+1},\Psi'_j\right)$.

5 For *l* from 1 to $k\times {}^{n}\!/_{2}$ do:

a. Call C'' to find Υ_j and Υ'_j among $2^{n/2}$ different messages of $(n/2)^2$ blocks length and $h'_{(n/2)\times(j-1)+1}$ in such a way $F_1\left(h'_{(n/2)\times(j-1)+1},\Upsilon_j\right)=F_1\left(h'_{(n/2)\times(j-1)+1},\Upsilon'_j\right)$

6 Output the 2^k messages of form $\left(\gamma_1,\ldots,\gamma_k\right)$ where γ_j will be one of the two multiblocks γ_j or γ'_j.

Obviously, all 2^k different messages generated in this way result to the same value of Hash. The third step of this algorithm finds $2^{k\times(n/2)}\times 2^{(n/2)}$ different multicollisions for the first round of the Zipper Hash which uses f_0 as a compress function, the fourth step finds $2^{k\times(n/2)}$ different multicollisions for the second round of the Zipper Hash which uses f_1 as a compress function and the fifth step finds 2^k different multicollisions for the third round of the Zipper Hash which uses f_2 as a compress function. The time complexity of the attack is equal to all attempts done by the adversary. It is equal to $k\times(n/2)^2\times 2^{n/2}$ for finding 2-way collision in the first round of attack, $k\times(n/2)\times 2^{n/2}$ for finding 2-way collision in the second round and $k\times 2^{n/2}$ for finding 2-way collision in the third round.

This approach can be easily extended to Zipper Hash with extra rounds. In similar way, it can be shown that the complexity of finding 2^k -way collision in Zipper Hash with *I* round is $O\left(k(n/2)^i 2^{n/2}\right)$. This approach can be easily extended to Zipper Hash with extra rounds. Following the given algorithms to find 2^k -way collision on the Zipper Hash with two and three rounds that have the complexities of $O\left(k(n/2)2^{n/2}\right)$ and $O\left(k(n/2)^i 2^{n/2}\right)$ respectively, it can be shown that the complexity of finding 2^k -way collision in Zipper Hash extended to *"i"* rounds would be $O\left(k(n/2)^{i-1}2^{n/2}\right)$. Whenever this complexity reaches the birthday bound to find a 2^k -way collision on an ideal Hash function, $O\left(2^{(n)\times\left(\left(2^k-1\right)/2^k\right)}\right)$, the extended Zipper Hash would be strengthen to the multicollision attack. To determine a bound for "*i*" we do as follows:

$$k(n/2)^{i-1}2^{n/2}\geq 2^{n\times\left(\left(2^k-1\right)/2^k\right)}\Rightarrow k(n/2)^{i-1}2^{n/2}\geq 2^n$$

$$\Rightarrow (n/2)^{i-1}\geq 2^{n/2}\Rightarrow i\geq n/2\times\log_{n/2}^{2}+1$$

This means that, in oracle model of compression function, the extended Zipper Hash to "*i*" rounds, for $i\geq n/2\times\log_{n/2}^{2}+1$, is not vulnerable to the multicollision attack.

VII. Conclusion

In this paper, it is shown that multicollisions in Zipper Hash structure are not much harder to find than finding multicollisions in the MD one. It is also shown that finding 2^r -way preimages and second preimages on this structure are not really harder to find than ordinary preimages and second preimages. Another important result is the fact that Zipper Hash structure can not be used as a building block for creating 2n-bit concatenated Hash structure because of its strength against collision which is much less than the ideal one. The study shows that although this structure is slightly more secure than iterated Hash function, it is really far from an ideal Hash function. It is shown that modifying the Zipper Hash by an extra round can not make it resistant to this attack. Finally, it is shown that in oracle model of compression function, the extended Zipper Hash to "*i*" rounds, for $i\geq n/2\times\log_{n/2}^{2}+1$, is not vulnerable to the multicollision attack

Reference

[1] A. Joux, "Multicollisions in Iterated Hash Functions. Application to Cascaded Constructions", Advances in Cryptology-CRYPTO '04, Springer-Verlag, pp. 306–316, 2004

[2] M. Nandi, D. R. Stinson, "Multicollision Attacks on Some Generalized Sequential Hash Functions", IEEE Transactions on Information Theory, Vol. 53, No. 2, pp. 759-767, 2007

[3] M. Liskov, "Constructing an Ideal Hash Function from Weak Ideal Compression Functions", 13th International Conference on Selected Areas in Cryptography, pp. 358-375, 2007

[4] P. Gauravaram, W. Millan, E. Dawson, K. Viswanathan, "Constructing Secure Hash Functions by Enhancing Merkle-Damgard Construction", Lecture Notes in Computer Science, Vol. 4058, pp. 407–420, 2006.

[5] P. Gauravaram, W. Millan, E. Dawson, K. Viswanathan, "Constructing Secure Hash Functions by Enhancing Merkle-Damga°rd Construction

(Extended Version). " Information Security Institute (ISI), Queensland University of Technology (QUT), number QUT-ISI-TR-2006-013, http://www.isi.qut.edu.au/ research/ publications/technical/qut-isi-tr-2006-013.pdf, July 2006.

[6] S. Su, Y. Yang, B. Yang, S. Zhang ,"The Design and Analysis of a Hash Ring-iterative Structure", available: http://eprint.iacr.org/2006/384.pdf

[7] S. Lucks, "A failure-friendly design principle for Hash functions", Advances in Cryptology - ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, Chennai, India, 2005

[8] W. R. Speirs, J. Molly, "Making large Hash Functions from small compression function", available:http://eprint.iacr.org/2007/239.ps.

[9] P. Lin, W. Wu, C. Wu1, T. Qiu, "Analysis of Zipper as a Hash Function", Lecture Notes in Computer Science, Vol. 4991, pp. 392-403, 2008