MULTI-COLLISIONS ATTACK IN RING HASH STRUCTURE

Nasour Bagheri[1], Babak Sadeghiyan[2]
[1] *Electrical Engineering Department, Iran University of Science and Technology (IUST)*
[2] *Computer Engineering Department, Amirkabir University of Technology Tehran, Iran*

Majid Naderi
Electrical Engineering Department, Iran University of Science and Technology (IUST), Iran

Keywords: Ring Hash Structure, Single Feedback Ring Hash, Multi Feedback Ring Hash, Hash function, Multi-collision Attack, Joux Attack, Preimage Attack, r-way collision.

Abstract: Ring hash structure is a new hash structure which has been introduced by Joux to strengthen the current hash structures against multi-collision attacks. In this paper, we present a cryptanalysis on Ring hash structure. We show that finding multi-collisions, i.e. 2^k-way collision, for a Ring hash structure is not much harder than finding such multi-collisions for ordinary MD hash structure. The complexity of our attack is approximately $\log(n)$ times harder than the complexity of attacks against MD structures. We employ these multi-collisions to find a D-way pre-image for this structure. We show the complexity of finding 2^K-way multi-collision and 2^k-way preimage are $O((k+1)\times(n/2)\times 2^{n/2})$ and $O(k\times n/2\times 2^{n/2}+2\times 2^n)$ respectively. We also show that Ring structure should not be used to create a hash function of 2n-bit length, by concatenating this structure to any other hash structure of n-bit output length. We show that the time complexity of finding a collision for this concatenated structure is $O((k+1)\times(n/2)\times 2^{n/2})$ that is much smaller than $\Omega(2^n)$, which is expected for a generic-birthday attack.

1 INTRODUCTION

Hash functions are used widely as a cryptographic primitive for generating digital signatures and message authentication codes. Each cryptographic hash functions should satisfy some security criteria. The main security requirement for a hash function is its collision freeness. Informally, it means that no attacker should be able to find a pair of different messages M and M' leading to equal hash values. Moreover, hash functions with output block length smaller than 160 bits are nowadays considered as insecure, due to general birthday attack.

In practice, building a cryptographic function with an input of variable size is not a simple task. Most hash functions are based on an iterated construction that makes use of a compression function, whose inputs have fixed sizes. A well-known family of such a construction are MDx hash function family, including MD4 (Rivest, 1992), MD5 (Rivest, 1995), and SHA (FIPS 180–1, 1995). The principle behind this structure is that if there is a computationally collision-free function f from m bits to n bits, where n < m (Damgard, 1990) then there is a collision-free hash function h mapping a message of arbitrary polynomial length to a k-bit string. Due to Merkle-Damg°ard theorem, it is claimed that if IV is fixed and if the padding procedure includes the length of the input into the padding bits, then h is collision-resistant if f is collision-resistant (Damgard, 1990, Merkle, 1990). Hence, it has been generally thought that the problem of designing a collision-resistant hash function has been reduced to the problem of designing a collision-resistant compression function. However, the multi-collision attack (Joux, 2004) and the multi-block differential collision attack on MD5, SHA-0 and SHA-1 (Biham, 2005, Wang, Yin, 2005, Wang, Yu, 2005) indicates that a collision-resistant compression function is not a sufficient condition of a collision- resistant hash function, but only a necessary condition. It means that a secure and collision- resistant hash function will be based not only on a collision-resistant compression function, but also on a collision-resistant structure.

Bagheri N., Sadeghiyan B. and Naderi M. (2008).
MULTI-COLLISIONS ATTACK IN RING HASH STRUCTURE.
In *Proceedings of the International Conference on Security and Cryptography*, pages 278-284
DOI: 10.5220/0001917102780284

Attacks to Hash function can divide in two groups. First group, e.g. Wang Attack (Wang, Yin, 2005, Wang, Yu, 200), includes attacks that use weaknesses of compression function. Second group of attacks, e.g. Joux (Joux, 2004) attack, includes attacks that use weaknesses of iterative structures, which are generic attacks to iterative structures such as MD structure. Actually, in this attack, increasing the security of compression function does not lead to strengthen the structure against multi-collision attack. This attack was shown that there is a 2^k -way collision attack for the classical iterated hash function based on a compression function, $f:\{0,1\}^{m+n}\to\{0,1\}^n$ where the time complexity of the attack is $O\left(k\times 2^{n/2}\right)$. This complexity is much less than the complexity for the generalized birthday attacks which is $\Omega\left(2^{\frac{n(2^k-1)}{2^k}}\right)$. This is the basic idea of Joux's attack. The main strategy of Joux's attack is to the first find k successive collisions by performing k successive birthday attacks. 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 $\left(m_1,m_2,...,m_k,Padding\right)$ (where m_i is one of the two blocks 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.

After introducing this attack, many structure have been proposed to strength iterated structure against this type of attach such Ring hash (Su, 2006) 3C and 3C+ (Gauravaram, 2006), Zipper hash (Liskov, 2006), WPH and DPH (Lucks, 2005), L-pipe(Speirs, 2007), etc. All this structure tries to strength against multi-collision or multi-block attack. Among this new hash structure, in our investigation, we analysis strengthen of Ring hash (Su, 2006) against multi-collision attack.

This paper is organized as following. Section 2 is a brief description of Ring hash function. Section 3 describes our attack for finding multi-collision for this structure. In section 4, we present a preimage attack for this structure. Section 5 present the security of 2n-bit Ring based hash function. Conclusions will be presented in section 6.

2 RING HASH STRUCTURE

There is two different variant of ring hash structure, called Single-feedback ring hash (SFRH) and multi-feedback ring hash (MFRH). The Ring hash structure can be considered as a general hash function construction. To build an n-bit hash function, it needs a compression functions f and an IV as initial chaining value. The function f can be seen as $\{0,1\}^{n+m}\to\{0,1\}^n$ mapping with $n\le m$ where m is single message block length and n is hash value out put length. Surprisingly, for his structure if original message length is division of m padding will be neglected (Su, 2006) , but if we consider a standard way of padding, it will not affect proposed attack. In standard padding schemas with 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 $M'\neq M, M'\|P(M')\neq M\|P(M)$. SFRH works as follow:

1. Let $M_1,...,M_l$ be m-bit strings such that $M_1,...,M_l=M\|P(M)$.
2. Fix $h_0=IV$.
3. $h_1,...,h_l$ are computed iteratively as $h_i=f_0\left(M_i,h_{i-1}\right)$.
4. Set the $M_{l+1}=M_1, M_{l+2}=M_l$, and the $h_{l+1}=f\left(M_{l+1},\leftrightarrow h_l\right)$.
5. The message digest is $H(M)=h_{l+2}=f\left(M_{l+2},h_{l+1}\right)$.

where sign $\leftrightarrow$ denotes reversal operation, that is, the bits of a variable are arranged in reverse order. For example the reverse code of '100110' is '011001'. In Figure 2, SFRH hash construction has been illustrated.

MFRH works as follow:

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

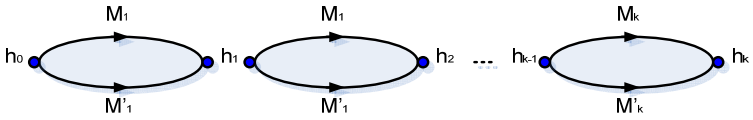

Figure 1: Schematic representation of Joux multi-collision construction.

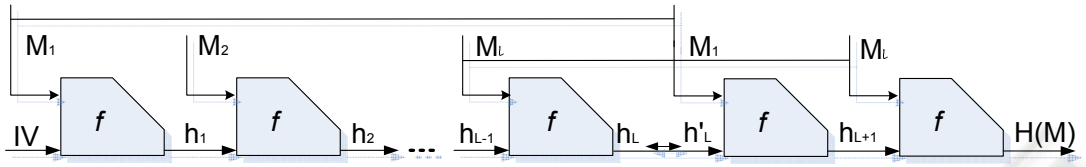

Figure 2: The Single Feedback Ring Hash (SFRH) structure, sign $\leftrightarrow$ denotes reversal operation.

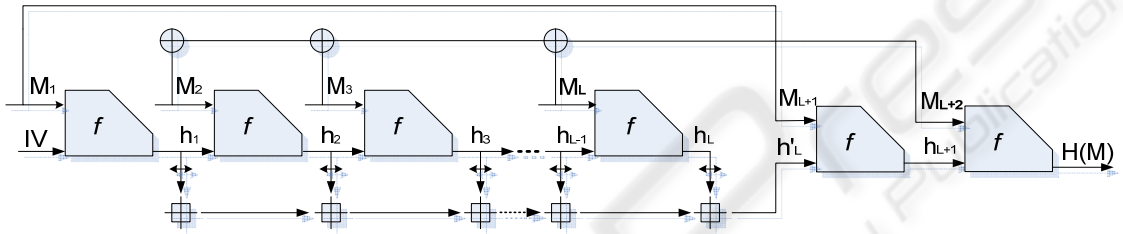

Figure 3: The Multi Feedback Ring Hash (MFRH) structure, sign $\leftrightarrow$, $\oplus$, and $\boxplus$ denote reversal operation, binary XOR and modular addition respectively.

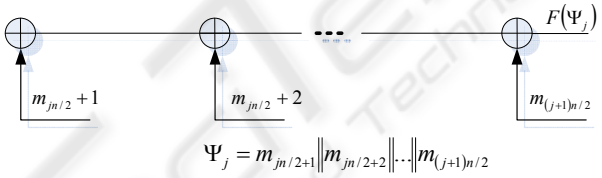

Figure 4: The $F(\Psi_j)$ function structure.

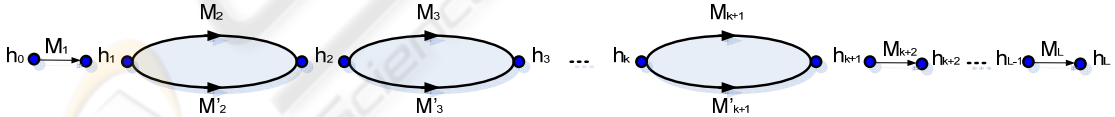

Figure 5: Schematic representation of multi-collision attack on SFRH.

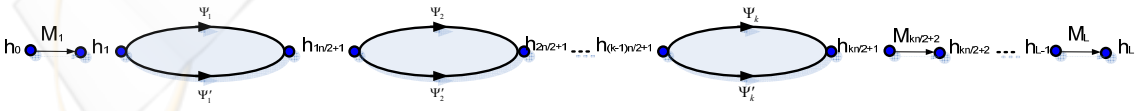

Figure 6: Schematic representation of multi-collision attack on SFRH.

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

4. Fix $h_0 = IV$.

5. $h_1,...,h_l$ are computed iteratively as $h_i = f_0(M_i, h_{i-1})$

6. Set $M_{l+1} = M_1, M_{l+2} = \bigoplus_{i=2}^{l} M_i$, $h'_l = \sum_{i=1}^{l} (\leftrightarrow h_i)$ and $h_{l+1} = f(M_{l+1}, h'_l)$.

7. Set $M_{l+1} = M_1, M_{l+2} = \bigoplus_{i=2}^{l} M_i$, $h'_l = \sum_{i=1}^{l} (\leftrightarrow h_i)$ and $h_{l+1} = f(M_{l+1}, h'_l)$.

8. The message digest is $H(M) = h_{l+2} = f(M_{l+2}, h_{l+1})$.

Where, notations $\leftrightarrow, \oplus$, and $\boxplus$ are denote reversal operation, binary XOR and modular addition respectively. Figure 2 illustrated MFRH hash construction.

3 MULTI-COLLISION ATTACK ON RING HASH

Ring hash structure has developed as a structure that could strength multi-collision attack. In this section, we show that constructing multi-collisions in Ring hash function can be done in an efficient way. If we consider collisions between messages of the same length, in this case the blocks of padding are identical and the padding process can be ignored. Moreover, if the intermediate hash chaining values collide at some point in the hash computation of two messages, the following values remain equal as while as the reminder 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 apply for any length of message block and chaining value, but for simplicity of prove, we assume that the size of the message blocks is bigger than the size of the chaining values. However, the attack can be easily generalized to the other case. We also assume that we 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 $f(h,M) = f(h,M')$. This collision finding machine may use the generic birthday attack or any specific attack based on a weakness of f. The most relevant property is that C should work properly for all chaining values. It's clear that most of these assumed conditions for our scenario are similar to what considered in (Joux, 2004), for generating multi-collision attack on ordinary MD. On the other hand, C' is a collision finding machine that given $2^{n/2}$ different single block message pairs m_i and m'_i and finds to multi-blocks Ψ and Ψ' so that $F(\Psi) = F(\Psi')$ where $F(\Psi) = \bigoplus_{i=1}^{n/2} m_i$ is ordinary XOR of all $n/2$ blocks of multi-block Ψ where illustrated in Figure 4. C' is a machine that can find collision with solving following equations:

$$
\begin{cases}
a_0 \times M_{0,0} \oplus a'_0 \times M'_{0,0} \oplus a_1 \times M_{1,0} \oplus \\
a'_1 \times M'_{1,0} \oplus ... \oplus a'_k \times M_{k,0} \oplus a'_k \times M'_{k,0} \\
= \\
b_0 \times M_{0,0} \oplus b' \times M'_{0,0} \oplus b_1 \times M_{1,0} \oplus \\
b'_1 \times M'_{1,0} \oplus ... \oplus b_k \times M_{k,0} \oplus b'_k \times M'_{k,0} \\
\\
a_0 \times M_{0,1} \oplus a' \times_0 M'_{0,1} \oplus a_1 \times M_{1,1} \oplus \\
a' \times_1 M'_{1,1} \oplus ... \oplus a'_k \times M_{k,1} \oplus a'_k \times M'_{k,1} \\
= \\
b_0 \times M_{0,1} \oplus b' \times M'_{0,1} \oplus b_1 \times M_{1,1} \oplus \\
b'_1 \times M'_{1,1} \oplus ... \oplus b_k \times M_{k,1} \oplus b'_k \times M'_{k,1} \\
\\
a_0 \times M_{0,n} \oplus a' \times_0 M'_{0,n} \oplus a_1 \times M_{1,n} \oplus \\
a' \times_1 M'_{1,n} \oplus ... \oplus a'_k \times M_{k,n} \oplus a'_k \times M'_{k,n} \\
= \\
b_0 \times M_{0,n} \oplus b' \times_0 M'_{0,n} \oplus b_1 \times M_{1,n} \oplus \\
b'_1 \times M'_{1,n} \oplus ... \oplus b_k \times M_{k,n} \oplus b'_k \times M'_{k,n}
\end{cases}
\quad (1)
$$

$$
\begin{cases}
a'_0 = a_0 \oplus 1 \text{ , } a'_1 = a_1 \oplus 1 \text{ , ... , } a'_k = a_k \oplus 1 \\
\\
b'_0 == b_0 \oplus 1 \text{ , } b'_1 = b_1 \oplus 1 \text{ , ... , } b'_k = b_k \oplus 1
\end{cases}
$$

Here, $M_{i,j}$ is the j[th] bit of the i[th] message block. But, how many messages do we need to solve this equation with non negligible probability? Actually, this linear equation would find some value for a_i, a'_i, b_i, a'_i, *for* $0 \le i \le k$. If adversary wants to insure that there is any solution for these equations, he must increase the number of messages pair. If he selects two distinct messages, the probability that they satisfy the condition is equal to 2^{-n}. If he wants to find any solution for given linear equations, he need to determine k in such a way:

$$\frac{\binom{2^k}{2}}{2^n} \geq 1/2 \qquad (2)$$

Therefore, if he selects $k = n/2$, this probability will be equal to $1/2$. Because of the second part of the equation, he can reduce the number of variables from $2 \times n$ *to* n variables, so the time complexity of solving this linear equation is $O(n^3)$.

3.1 Multi-Collision Attack on SFRH

In this section, we prove that constructing 2^k-collisions on SFRH approximately cost k times as much as building multi-collisions in the ordinary MD structure. In particular, we claim that we can generate 2^k equal collision on SFRH by only $O(k)$ calls to the C oracle machine which is much less than what we expected from birthday paradox $O\left(2^{n\times\left(\left(2^{k/2}-1\right)/2^{k/2}\right)}\right)$. Assume that l, number of message blocks, satisfies $l \geq k+2$. The attack works as follows:

1. Let h_0 be equal to the initial value IV of Ring Hash.
2. Set $h_1 = f(h_0, M_1)$
3. For i from 2 to $k+1$ do:

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

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

4. Output the 2^k messages of form $(M_1, m_2, ..., m_k, M_{k+1}, ..., M_l)$ where m_j is on of the two multi blocks M_j and M_j'.

Obviously, since M_1, M_l have been fixed and $2 \leq i \leq k+1 \rightarrow f(h_{i-1}, M_i) = f(h_{i-1}, M_i')$, for all 2^k different messages h_l, h_l', M_{l+1}, and M_{l+2} are equal and 2^k different messages generated in this way are result the same value of hash.

Time complexity of attack is equal to all attempt that adversary do. It is equal to $k \times 2^{n/2}$ for finding 2-way collision. Figure 5 is a schematic illustration of these 2^k multi-blocks messages together with their common intermediate hash values.

3.2 Multi-Collision Attack on MFRH

In this section we show that how the adversary can generate 2^k equal collision by only $O(k \times n/2)$ calls to the C oracle and $O(k)$ calls to C' oracle machine, which is much less than what we expected from birthday paradox $O\left(2^{(n/2)\times\left(\left(2^k-1\right)/2^k\right)}\right)$. Assume that l, number of message blocks, satisfies $l \geq k \times n/2 + 2$. The attack works as follows:

1. Let h_0 be equal to the initial value IV of Ring Hash.
2. Set $h_1 = f(h_0, M_1)$
3. For i from 2 to $k \times n/2 + 1$ do:

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

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

4. For j from 1 to k do:

a. Give the $n/2$ different single block message pairs of (M_j, M'_j) through $(M_{(j-1)n/2+2}, M'_{j\times n/2+1})$ to C' machine to find Ψ_j and Ψ_j' so that $F(\Psi_j) = F(\Psi_j')$.

5. Output the 2^k messages of form $(M_1, \psi_1, ..., \psi_k, M_{k\times n/2+2}, ..., M_l)$ where ψ_j is on of the two multi blocks Ψ_j and Ψ_j'.

Obviously, since M_1, M_l have been fixed and for $2 \leq i \leq k+1 \rightarrow f(h_{i-1}, M_i) = f(h_{i-1}, M_i')$, for all these 2^k different messages h_{l+2} and M_{l+2} are equal and all the 2^k different messages generated in this way are result the same value of hash.

Third step of this algorithm finds $2^{k\times(n/2)}$ different multi-collision for forward part of MFRH which uses f as compress function and fourth step find 2^k different multi-collision for feed-forward part of MFRH hash which produces $\bigoplus_{i=2}^{l} M_i$. Time complexity of attack is equal to all attempt that adversary doing. It is equal to $k \times (n/2) \times 2^{n/2}$ for finding 2-way collision in forward path of attack and $k \times (n/2)^3$ for finding 2-way collision in feed-forward path, related to time complexity of C' machine. Figure 6 is a schematic illustration of

these 2^k multi-blocks messages together with their common intermediate hash values.

4 FINDING K-WAY SECOND-PREIMAGE ATTACK ON RING HASH

In reference (Joux, 2004) Joux puts forward an attack method called k-way second-preimage 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 $Y = H(M) \in \{0,1\}^k$, the attackers first find 2^r collisions on r-block messages $M_1, M_2, ..., M_{2^r}$ where $H_r = H(M_1) = H(M_2) = ... = H(M_{2^r})$. Then, he finds 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 to the message M. Obviously, the time complexity of this attack is $O(r2^{n/2} + 2^n)$.

4.1 Finding k-Way Second-Preimage Attack on SFRH

For a hash function based on the SFRH structure, we show that adversary can find 2^r-way preimage and second preimage in cost of $O(r2^{n/2} + 2^n)$. The attack works as follows:

1. Fixed M_1 with some random value.
2. Set $h_1 = f(IV, M_1)$
3. For i from 2 to $r+1$ do:

a. Call C and find M_i and M_i' so that $f(h_{i-1}, M_i) = f(h_{i-1}, M_i')$.

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

4. Find M_{r+2} in such a way $f\left(f\left(\leftrightarrow f(h_{r+1}, M_{r+2}), M_1\right), M_{r+2}\right) = Y$.

5. Output the 2^r messages of form $(M_1, m_2, ..., m_r, M_{r+1})$ where m_j is on of the two multi blocks M_j and M_j'.

Obviously, all 2^k different messages generated in this way are led to value Y as hash result. This procedure can divide in two parts. Fist part of attack is finding 2^r-way collision which cost $r \times 2^{n/2}$ and second part which contain in step 4, related to finding a preimage to guaranteed the successes of attack which cost 2^n. Therefore, all time complexity of the attack is equal to what has been clamed.

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

4.2 Finding k-way Second-Preimage Attack on SFRH

For a hash function based on the SFRH structure, we show that adversary can find 2^r-way preimage and second preimage in cost of $O(r \times (n/2) 2^{n/2} + 2^n)$. The attack works as follows:

1. Fixed M_1 with some random value.
2. Set $h_1 = f(IV, M_1)$
3. For i from 2 to $r+1$ do:

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

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

4. for j from 1 to r do:

a. Call C' to find Ψ_j and Ψ_j' among $2^{n/2}$ different messages of $n/2$ blocks length in such a way that $F(\Psi_j) = F(\Psi_j')$.

5. Find $M_{r \times n/2+2}$ so that $f\left\{f\left[\left(\left(\leftrightarrow f(h_{r+1}, M_{r+2})\right) + \sum_{i=1}^{r+1}(\leftrightarrow h_i)\right), M_1\right], \bigoplus_{i=1}^{r+2} Mi\right\} = Y$.

6. Output the 2^r messages of form $(M_1, \psi_1, ..., \psi_{r+1}, M_{r+2})$ where ψ_j is on of the two multi blocks Ψ_j and Ψ_j'.

Obviously, all 2^k different messages generated in this way are led to the value Y, as hash result. This procedure can divide in two parts. The first part of the attack, steps 3 and 4, is finding 2^r-way collision in MFRH which cost $O(r \times (n/2) \times 2^{n/2})$ and second part which contain in step 5, related to finding a preimage to guaranteed the successes of attack which cost $O(2^n)$. Therefore, all time complexity of attack is equal what has been clamed.

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

5 RING HASH IN 2N-BITE 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 (Joux, 2004) Joux has shown that if at least one of these hash function be a MD iterated hash function, complexity of finding a collision for this structure is slightly more than finding collision for one branch and equal to $O\left(n/2\times 2^{n/2}\right)$.

The basic idea in this attack is finding $2^{n/2}$-way collision for MD structure and find a collision among this $2^{n/2}$ different message for the second hash function. Clearly this collision is applicable to booth branches. Whit similar task, adversary could seek a collision for Ring hash structure. These attacks are difference in the first part complexity and complexity of the first par for SFRH and MFRH are $O\left((n/2)\times 2^{n/2}\right)$ and $O\left((n/2)\times(n/2+1)\times 2^{n/2}\right)$ respectively.

6 CONCLUSIONS

In this paper, we showed that finding multi-collisions in Ring hash structure are not much harder to find than finding multi-collisions in MD hash structure. Actually, we proved that finding multi-collisions in SFRH is as difficult as finding it on ordinary MD, and for MFRH, it is a little harder than what for MD is. Also, we have shown that finding 2^r-way preimages and second preimages on these structures are not really harder to find than ordinary preimages and second preimages. Moreover, we shown that ring hash structures can not be used as a building block for creating 2n-bite concatenated hash structure because of its strength against collision attack, which is much less than ideal one. Our study have shown that although this structure is slightly more secure than MD iterated hash structure, but is really far from perfect hash function.

REFERENCES

Biham, E., Chen, R, and Joux, A., etc, 2005. Collisions of SHA-0 and Reduced SHA-1, Advances in Cryptology-EUROCRYPT'05, pp.36–57, Springer-Verlag.

Damgard, I., 1990. A design principle for hash functions, in Advances in Cryptology – Crypto'89 (G. Brassard, ed.), no. 435 in Lecture Notes in Computer Science, pp. 416–427, Springer-Verlag.

FIPS, 180–1, 1995. Secure hash standard. FIPS publication.

Gauravaram, P., Millan, W., Dawson, E. and Viswanathan, K., 2006. Constructing Secure Hash Functions by Enhancing Merkle-Damgard Construction., Information Security and Privacy, (Batten, L., Safavi-Naini, R., ed.) volume 4058 of Lecture Notes in Computer Science, pp. 407–420, Springer.

Joux, A., 2004. Multi-collisions in Iterated Hash Functions. Application to Cascaded Constructions Advances in Cryptology-CRYPTO'04, pp. 306–316, Springer-Verlag.

Lucks,S. , 2005. A failure-friendly design principle for hash functions. In Bimal Roy, editor, Advances in Cryptology-ASIACRYPT'05, volume 3788 of Lecture Notes in Computer Science, pp. 474-494, Springer-Verlag.

Merkle, R., C., 1990. One-way hash functions and DES in Advances in Cryptology – Crypto'89 (G. Brassard, ed.), no. 435 in Lecture Notes in Computer Science, pp. 428–446, Springer-Verlag.

Rivest, R., L., 1992. The MD4 Message – Digest Algorithm. Network MIT laboratory for Computer Science and RSA Data Security , Inc RFC 1320.

Rivest, R., L., 1992. The MD5 message-digest algorithm, Request for Comments (RFC1320), Internet Activities Board, Internet Privacy Task Force.

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

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

Wang, X., Yin, Y., L., and Yu, H., 2005. Finding collisions in the full SHA-1, Advances in Cryptology-CRYPTO'05, pp. 17–36, Springer-Verlag.

Wang, X. and Yu, H., 2005. How to Break MD5 and Other Hash Functions, Advances in Cryptology - EUROCRYPT'05, pp. 19–35, Springer-Verlag.