A SECOND PREIMAGE ATTACK ON THE MERKLE-DAMGARD SCHEME WITH A PERMUTATION FOR HASH FUNCTIONS

Shiwei Chen and Chenhui Jin
Institute of Information Science and Technology, Zhengzhou 450004, China

Keywords: Hash functions, MD construction, MDP, Multicollisions, Second preimage attack, Computational complexity.

Abstract: Using one kind of multicollsions of the Merkle-Damgard(MD) construction for hash functions proposed by Kelsey and Schneier, this paper presents a second preimage attack on MDP construction which is a simple variant of MD scheme with a permutation for hash functions. Then we prove that the computational complexity of our second preimage attack is $k \times 2^{n/2+1} + 2^{n-k}$ less than 2^n where n is the size of the hash value and $2^k + k + 1$ is the length of the target message.

1 INTRODUCTION

A cryptographic hash function H maps a message M with arbitrary length to a fixed-length hash value h. It has to satisfy the following three security requirements:

- Preimage resistance: For a given hash value h, it is computationally infeasible to find a message M such that $h = H(M)$;

- Second preimage resistance: For a given message M, it is computationally infeasible to find a second message $M' \neq M$ such that $H(M') = H(M)$;

- Collision resistance: It is computationally infeasible to find two different messages M' and M such that $H(M') = H(M)$.

The resistance of a hash function to collision attack or second preimage attack mainly depends on the size n of the hash value. Regardless of how a hash function is designed, an adversary will always be able to find a preimage or a second preimage after trying 2^n different messages, or find a collision pair after $2^{n/2}$ trials according to the birthday attack. Therefore, if the computational complexity of finding a collision pair or a (second) preimage for a particular hash function is less than what could be expected based on the size of the hash value, then the hash function is considered to be broken. Generally, a hash function includes two parts, that is, the compression function which maps a fixed-length value to a fixed-length value, and the domain extension transform which can transfer a message with arbitrary length to a fixed-length hash value. Aimed to these two parts, the results of analyzing on hash functions can be divided into two kinds:

- Cryptanalytic attacks: Mainly apply to the compression functions of the hash functions. Using the internal properties of the compression functions, an adversary can attack the hash functions. For example, the collision attacks on MD-family proposed in (Xiaoyun and Hongbo, 2005);

- Generic attacks: Apply to the domain extension transforms directly with some assumptions on the compression functions. Examples are long-message second preimage attack(Kelsey and Schneier, 2005), herding attack(Kelsey and Kohno, 2006) and the attack on the MD with XOR-linear/additive checksum in (Gauravaram and Kelsey, 2007).

Since Wang et al.(Xiaoyun and Hongbo, 2005) presented the collision attacks on MD-family hash functions and the recent results on the MD construction, some cryptographers have been trying to propose new domain extension transforms for hash functions, such as MD with XOR-linear/additive checksum(Gauravaram and Kelsey, 2007), ChopMD construction (Coron et al., 2005), EMD construction(Bellare and Ristenpart, 2006), MD with a permutation (MDP)(Hirose and Park, 2007), and so on. In 2007, Praveen Gauravaram and John Kelsey (Gauravaram and Kelsey, 2007) pointed out that the MD with XOR-linear/additive checksum construction gained almost no security against generic attacks. Coron et al.(Coron et al., 2005) presented that the prefix-free MD and ChopMD were indifferentiable from a random oracle and gave out the security bounds. However, Mihir Bellare and Thomas Ristenpart(Bellare and Ristenpart, 2006) proved that

Chen S. and Jin C. (2009).
A SECOND PREIMAGE ATTACK ON THE MERKLE-DAMGARD SCHEME WITH A PERMUTATION FOR HASH FUNCTIONS.
In *Proceedings of the International Conference on Security and Cryptography*, pages 245-248
DOI: 10.5220/0002302024502248

pseudorandom-oracle preserving did not imply the collision-resistance preserving and presented that the variants of MD construction presented in (Coron et al., 2005) was not collision-resistance preserving. In Asiacrypt 2007, Hirose et al.(Hirose and Park, 2007) proposed a simple variant of the Merkle-Damgard scheme with a permutation and analyzed its security by using the indifferentiability formulism. However, there is no paper discussing whether the MDP resists the second preimage attack or not.

In this paper, using the multicollsions of MD construction proposed in (Kelsey and Schneier, 2005), we will present a second preimage attack on MDP construction, the computational complexity of which is less than what could be expected based on the size of the hash value.

2 DESCRIPTION OF MDP CONSTRUCTION AND NOTATIONS

Let $f : \{0,1\}^n \times \{0,1\}^b \rightarrow \{0,1\}^n$ be a compression function and M be a l-block b-bit message. We can describe the MD^f below, which is MD construction with the compression function f:

$Function : MD^f(IV, M)$
  $let\ M = (m_1, m_2, \ldots, m_l)\ and\ h_0 = IV$
  $for\ i = 1\ to\ l\ do$
    $h_i \leftarrow f(h_{i-1}, m_i)$
  $return\ h_l$.

Let M^{pad} be a padded message according to the padding function given in (Hirose and Park, 2007) and ϕ is a random permutation. Then the MDP^f is defined as follows:

$Function : MDP^f(IV, M^{pad})$
  $let\ M^{pad} = (m_1, m_2, ..., m_l)\ and\ h_0 = IV$
  $h_{l-1} \leftarrow MD^f(IV, (m_1, m_2, \ldots, m_{l-1}))$
  $h_l \leftarrow f(\phi(h_{l-1}), m_l)$
  $return\ h_l$.

Since the padding function of MDP construction requires that the last block of the padded message encodes the q-bit representation of the length of the original message, the second preimage attack proposed in the following paper need to find a second preimage with the same length as the target message. Refer to (Hirose and Park, 2007) for the specifics of the padding function of MDP construction.

Note that $|M|$ represents the number of blocks of a message M, m_i is the i^{th} b-bit block of M and h_i is the i^{th} intermediate chaining value in hashing of M. If there is no special explanation, the notations represent the same means throughout this paper.

3 OUR SECOND PREIMAGE ATTACK ON MDP CONSTRUCTION

Though Hirose et al.(Hirose and Park, 2007) have analyzed the security of MDP construction using the indifferentiability formulism, up to now no paper has discuss whether the MDP resists the second preimage attack or not. In this paper, using the multicollisions of MD construction, we will present a second preimage attack on the MDP construction. Through all this paper, we assume that the compression function f is random.

3.1 Building the Multicollisions of MD^f

The k messages $M_1, M_2, \ldots, M_k$ are called k-multicollision of MD construction if

$$MD^f(M_1) = MD^f(M_2) = \cdots = MD^f(M_k)$$

The papers (Kelsey and Schneier, 2005)(Kelsey and Kohno, 2006)(Joux, 2004) have presented different methods to construct the multicollisions of MD construction. Specifically, in (Kelsey and Schneier, 2005) they introduced a way to construct (a,b)-expandable messages, which are $(b-a+1)$-multicollison naturally whose lengths can vary in a range from a to b. Using the method introduced in (Kelsey and Schneier, 2005), now we describe the algorithm 1 to construct $(t, 2^t+t-1)$-expandable messages with a starting chaining value IV and lengths varying from t to 2^t+t-1, which will be used to propose our second preimage attack on MDP^f.

Algorithm 1:

Step1. Find two messages $B_1, B_1^{'}$ such that

$$MD^f(IV, B_1) = MD^f(IV, B_1^{'}) = H_1$$

where $|B_1| = 1, |B_1^{'}| = 2^0 + 1$;

Step2. Use H_1 as the starting chaining value to construct the next collision pair $B_2, B_2^{'}$ satisfying

$$MD^f(H_1, B_2) = MD^f(H_1, B_2^{'}) = H_2$$

where $|B_2| = 1, |B_2^{'}| = 2^1 + 1$;

Step3. For the i^{th} step, we need to start with the chaining value H_{i-1} and find a collision pair $B_i, B_i^{'}$ such that

$$MD^f(H_{i-1}, B_i) = MD^f(H_{i-1}, B_i^{'}) = H_i$$

where $|B_i| = 1, |B_i^{'}| = 2^{i-1} + 1$;

Step4. Until obtaining t pairs messages $(B_i, B_i^{'})(i = 1, 2, ..., t)$, we can construct the $(t, 2^t + t - 1)$-expandable messages by choosing B_i or$B_i^{'}(i = 1, 2, ..., t)$ in every pair.

Remark:

(1) From the above algorithm 1, we know that the shortest message in the multicollisions is $B_1 \parallel B_2 \parallel \cdots \parallel B_t$ and the longest message is $B_1^{'} \parallel B_2^{'} \parallel \cdots \parallel B_t^{'}$ whose length is

$$\sum_{i=1}^{t}(2^{i-1}+1) = 2^t + t - 1$$

Moreover, by choosing B_i or $B_i^{'}(i = 1,2,\ldots,t)$ in every pair, we can obtain messages of different lengths varying from t to $2^t + t - 1$.

(2) We can use the algorithm described in (Kelsey and Schneier, 2005) to construct a collision pair $B_i, B_i^{'}$ such that

$$MD^f(H_{i-1}, B_i) = MD^f(H_{i-1}, B_i^{'}) = H_i$$

and $|B_i| = 1, |B_i^{'}| = 2^{i-1} + 1$. The specifics are as follows:

Step1. Assume m is one block chosen randomly in advance. Process 2^{i-1} given message blocks:
-$H_{temp} = H_{i-1}$;
-For $j = 0$ to $2^{i-1} - 1$ do

$$H_{temp}^{'} = f(H_{temp}, m) \text{ and } H_{temp} = H_{temp}^{'}$$

Step2. Build lists A and B as follows:
-For $j = 0$ to $2^{n/2-1} - 1$ do

$$A[j] = f(H_{i-1}, a_j) \text{ and } B[j] = f(H_{temp}, b_j)$$

where a_j and b_j are chosen randomly and $|a_j| = |b_j| = 1$;

Step3. Find j_1, j_2 such that $A[j_1] = B[j_2]$ and return the collision pairs $(a_{j_1}, m \parallel m \parallel \ldots \parallel m \parallel b_{j_2})$.

Therefore, the computational complexity of finding $B_i, B_i^{'}$ such that

$$MD^f(h_{i-1}, B_i) = MD^f(h_{i-1}, B_i^{'}) = h_i$$

and $|B_i| = 1, |B_i^{'}| = 2^{i-1} + 1$ is about $2^{i-1} + 2^{n/2+1}$ compression function operations. Hence, the computational complexity of algorithm 1 is about

$$\sum_{i=1}^{t}(2^{i-1} + 2^{n/2+1}) = t \times 2^{n/2+1} + 2^t \approx t \times 2^{n/2+1}.$$

3.2 Our Second Preimage Attack on MDP^f Hash Function

Let $M = (m_1, m_2, \ldots, m_{2^k+k+1})$ be the target message of $2^k + k + 1$ blocks. Our attack is to find another message $M^{'}$ of $2^k + k + 1$ blocks different from M such that $MDP^f(M^{'}) = MDP^f(M)$. The specific algorithm is described below:

Preprocessing step: Construct $(k, 2^k + k - 1)$-expandable messages with a starting value IV and an arbitrary target value H_k according to algorithm 1;

Algorithm 2:

Step1. Randomly choose a one-block message B such that the value of $f(H_k, B)$ equals to one of the chaining values $h_1, h_2, \ldots, h_{2^k+k}$ produced in the hashing of M, that is, $f(H_k, B) = h_{i_0}$ where $k + 1 \le i_0 \le 2^k + k$;

Step2. Choose a message M_0 of $i_0 - 1$ blocks from the $(k, 2^k + k - 1)$-expandable messages constructed in the preprocessing step;

Step3. Form a message

$$M^{'} = M_0 \parallel B \parallel m_{i_0+1} \parallel \cdots \parallel m_{2^k+k+1}$$

satisfying $MDP^f(M^{'}) = MDP^f(M)$ (If $i_0 = 2^k + k$, then only the last block of original message is included in the second preimage).

3.3 Analysis of the Computational Complexity of the Above Algorithm

In the above algorithm, since the one-block message B is chosen randomly and $k + 1 \le i_0 \le 2^k + k$, the probability of guaranteeing that $f(H_k, B) = h_{i_0}$ is $2^k/2^n$. So the computational complexity of step1 is about 2^{n-k}. And the computational complexity of the step2 and step3 can be ignored. Additionally, the computational complexity of the preprocessing step is about $k \times 2^{n/2+1}$. Hence, the computational complexity of the above algorithm is about $k \times 2^{n/2+1} + 2^{n-k}$ which is less than 2^n.

4 CONCLUSIONS

In this paper, using the $(k, 2^k + k - 1)$-expandable messages with a starting chaining value IV, we present a second preimage attack on hash functions with MDP construction and analyze the computational complexity of our second preimage attack which is $k \times 2^{n/2+1} + 2^{n-k}$ less than 2^n.

REFERENCES

Xiaoyun W. and Hongbo Y. (2005), How to break MD5 and other hash functions. In *Eurocrypt 2005, LNCS 3494, pp. 474-490*. Berlin: Springer-Verlag, 2005.

Kelsey J.and Schneier B.(2005), Second preimages on *n*-bit hash functions for much less than 2^n word. In *Eurocrypt 2005, LNCS 3494, pp. 19-35*. Berlin: Springer-Verlag, 2005.

Kelsey J. and Kohno T.(2006), Herding hash functions and the Nostradamus attack. In *Eurocrypt 2006, LNCS 4004, pp. 183-200*. Berlin: Springer-Verlag, 2006.

Gauravaram P. and Kelsey J.(2007), Cryptanalysis of a Class of Cryptographic Hash Functions. In *CT-RSA 2008, http://eprint.iacr.org/2007/277*.

Coron J.S., Dodis Y., Malinaud C. and Puniya P.(2005), Merkle-Damgard Revisited: How to Construct a Hash Function. In*CRYPTO 2005. LNCS 3621, pp. 430-448*. Berlin: Springer-Verlag, 2005.

Bellare M. and Ristenpart T.(2006), Multi-Property-Preserving Hash Domain Extension and the EMD Transform. In *ASIACRYPT 2006. LNCS, vol. 4284, pp. 299-314*. Berlin: Springer-Verlag, 2006.

Hirose S., Park J.H. and Yun A.(2007), A Simple Variant of the Merkle-Damgard Scheme with a Permutation. In *ASIACRYPT 2007. LNCS, vol. 4833, pp. 113-129*. Berlin: Springer-Verlag, 2007.

Joux A.(2004), Multicollisions in Iterated Hash Functions. In *CRYPTO 2004, LNCS 3152, pp. 306-316*. Berlin: Springer-Verlag, 2004.