Cryptanalysis of Lin et al.’s Efficient Block-Cipher-Based Hash Function

Bozhong Liu*, Zheng Gong†, Xiaohong Chen‡, Weidong Qiu* and Dong Zheng*
*School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai, P.R.China.
†Distributed and Embedded Security Group, Faculty of EEMCS, University of Twente, The Netherlands.
‡Institute of Forensic Science, Ministry of Justice, P.R.China

Abstract—Hash functions are widely used in authentication. In this paper, the security of Lin et al.’s efficient block-cipher-based hash function is reviewed. By using Joux’s multicollisions and Kelsey et al.’s expandable message techniques, we find the scheme is vulnerable to collision, preimage and second preimage attacks. Some modifications are recommended to avoid those security flaws in Lin et al.’s hash construction.

Key Words: Authentication, Cryptanalysis, Block-cipher-based hash function, Multicollusion, Expandable message.

I. INTRODUCTION

Cryptographic hash functions, which operate on messages of arbitrary length and output a fixed size value, play an important role in the network’s evolution. Due to the one-way and uniform properties, hash functions are widely used for authentication such as commitment scheme, integrity identification as source code management and digital signature. The design of cryptographic hash functions often follows the Merkle-Damgard (MD) construction [10], [11], which iterates a compression function for domain extension. Under the MD construction, the final block would combine the length of the original message. Most popular hash functions are based on the MD construction, such as MD4 [12], MD5 [13], SHA-0 [14] and SHA-1 [15]. In practice, one can easily choose a well-investigated block cipher (such as DES, IDEA and AES) to construct a compression function-the block-cipher-based hash functions. They are more convenient to be constructed and can be faster in applications of authentications and digital signature than ordinary hash functions. Recent results showed that a well-designed block-cipher-based hash function not only can be used for authentication [6], but also can be benefit for message authentication codes [7] in resource-constrained environment, e.g., sensor networks, smart cards and RFID tags. According to the execution times of the underlying block cipher in the algorithm, block-cipher-based hash functions can be defined by single block length (SBL) such as the PGV hash functions [16] and double block length (DBL) such as MDC-2 [17], Parallel-DM [18], and LOKI-DBH [19]. Still, the recent advances in collision finding [8], [9] motivate renewed interest in finding good ways to turn a block cipher into a cryptographic hash function. Instructive examples can be found in [4], [5].

Recently, Lin et al.’s proposed a new block-cipher-based hash function [1], which aims at building a single-block-length hash function scheme with a higher efficiency. In Lin et al.’s scheme, the rate of the compression function is 1/2. They claimed that even the underlying compression function is insecure, the scheme can be secure after iterations. Moreover, they find that besides rate, key schedule is another very important factor. By carefully choosing an appropriate key schedule, the hash functions with small rate may be more efficient than the large ones.

In this paper, we cryptanalyze Lin et al.’s scheme by using the ideas of Joux’s multicollision attack [2] and Kelsey et al.’s expandable message technique [3]. On one hand, we construct $2^r$ multicollisions to find collisions and (second) preimages of the scheme. On the other hand, by constructing expandable messages and using fixed point technique, we indicate that the scheme is vulnerable to second preimage attack on long messages. The fixed point idea was first discussed in [20]. With this technique we can generate an message that can be extended to arbitrary length without changing the resulting hash value. More details about the attack are elaborated in Section 3.

The remainder of this paper is organized as follows. In Section 2, the definitions and properties of block-cipher-based hash functions are reviewed. Lin et al.’s scheme is described. The ideas of Joux’s multicollisions and Kelsey et al.’s expandable messages are also introduced. In Section 3, the collisions, (second) preimages attacks with multicollisions are demonstrated. Afterward, we show a complete algorithm of second preimage attack on long messages with expandable messages technique. In Section 4, some modifications are recommended to avoid those security flaws in Lin et al.’s hash construction. Section 5 concludes the paper.

II. PRELIMINARIES

Here we describe some necessary definitions and notions which will be used in the following cryptanalysis.
A. Hash Functions

Properties. Hash functions are one-way functions that map arbitrary length bits to fixed size bits, often denoted by $H : \{0,1\}^r \rightarrow \{0,1\}^n$. A good designed hash function should have certain computational complexity against collisions or other brute-force attacks. Usually, a secure hash function have three minimal properties. (In practice, more other properties are considered.)

1) Collision resistance: An adversary should be hard to find a pair of messages $M \neq \tilde{M}$ such that $H(M) = H(\tilde{M})$. This property is often referred to as strong collision resistance.

2) Preimage resistance: Given an hash value $Y$, an adversary should be hard to find the corresponding input message $M$ such that $Y = H(M)$. This concept is related to the properties of one-way function.

3) Second preimage resistance: Given a message $M$, an adversary should be hard to find another message $\tilde{M}$ such that $H(\tilde{M}) = H(M)$. This property is often referred to as weak collision resistance.

If any adversary can find collisions with less than $2^{n/2}$ work, or (second) preimage with less than $2^n$ work, the hash functions are not secure against collision attacks, (second) preimage attacks, respectively.

Block-cipher-based Hash Functions. The compression functions of hash functions are often built from block ciphers. A block cipher is a permutation $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ where $k$ is the key length. To turn a partial one-way block cipher into a one-way compression function, some methods are Davies-Meyer, Matyas-Meyer-Oseas, Miyaguchi-Preneel, MDC-2, MDC-4, Hirose, etc. The rate gives a measure of efficiency of a hash functions based on a certain compression function. It is defined as the number of $n$ bit message blocks operated per encryption or decryption. For example, Lin et al.’s scheme is rate 1/2.

Black-box model is a well-known security model for the analysis of block-cipher-based hash functions. In this model, a block cipher is randomly chosen from a set that containing all appropriate block ciphers. An adversary can freely encrypt and decrypt the blocks but can not access the implementation. Generally, the complexity of finding a collision or (second) preimage is based on the total number of queries of encryption and decryption by the adversary.

B. Joux’s Multicollision Attack

Joux proposed a generic multicollision attack against iterated hash functions [2]. In his paper, it shows that multicollisions in iterated hash functions are not really harder to find than ordinary collision. More precisely, it cost $r2^{n/2}$ work to find 2'-collisions instead of $2^{n(2^r-1)/2^r}$ work from an ideal hash function. On the other hand, Joux pointed out that though it is tempting to concatenate two hash function $H_1||H_2$ to gain more security without increasing the size of hash functions, in fact it is not really more secure than $H_1$ and $H_2$ itself.

Here is the basic construction of the attack. First of all, assuming that the output length of the hash function is $n$ and the size of the messages blocks is $m$. Let $h_i$ be the hash chaining values, and $h_0$ is the IV. $f : \{0,1\}^{m+n} \rightarrow \{0,1\}^n$ is the compress function. $C$ denotes a 2-collision finding machine that can output two different messages $M$ and $\tilde{M}$ such that $f(h_i, M) = f(h_i, \tilde{M})$. Finding this collision may use generic birthday attack or any other specified attacks based on the weakness of $f$. In the following, use the machine $C$ to generate $r$ pairs of messages $(M_i, \tilde{M}_i)$ satisfying $f(h_{i-1}, M_i) = f(h_{i-1}, \tilde{M}_i)$, $i$ from 1 to $r$. We can see the graphic description below in figure 1.

Fig. 1. Joux’s multicollision construction

After the construction, the set $\{m_1||m_2||...||m_r\mid m_i = M_i$ or $\tilde{M}_i, i = 1, 2, ..., r\}$ is a 2'-collision set. All the messages in the set are hash to the same value.

C. Second Preimages Attack with Expandable Messages

The Fixed Point Technique. In [20], Dean demonstrated a fixed points technique on the compression functions that could bypass the MD construction. For a compression function $f$ there exits fixed points $h, m$ such that $h = f(h, m)$. Using fixed points, we can expand the message length to arbitrary number of blocks without changing the resulting hash value. The algorithm is elaborated as follows.

Algorithm: ConstructFixedPointsMessage($IV$)

IV is the initial value and $n$ is the width of hash chaining values and output.

Steps:

1) Find $O(2^{n/2})$ pairs of $(h, m)$ such that $h = f(h, m)$ with fixed points algorithm. Keep the paired results in ListA = $(h, m)$.

2) Calculate $O(2^{n/2})$ times of $h' = f(IV, m')$ as $m'$ is an unique message block. Keep the paired results in ListB = $(h', m')$.

3) Find a collision between ListA and ListB such that $h = h'$.

4) Return the message $(m'||m)$ where $m'$, $m$ are the corresponding message block in the pairs.

To expand the resulting message, it just needs adding sufficient message blocks $m$ repeatedly, e.g., $(m'||m)|...|m)$. The complexity of the algorithm is about $2^{n/2+1}$ work.

A Generic Expandable Message Technique. John Kelsey and Bruce Schneier advocated an expandable message technique to launch a second preimage attack on long messages [3]. In their paper, a sequence of collisions between messages of different lengths is found, and combined together to provide a set of messages, saying an expandable message, that covers a wide range of different lengths without changing the resulting intermediate hash value. For a given $2^n+r+1$-block message,
one can find $2^r$ distinct second preimages by expandable message technique with about $r2^{n/2+1} + 2^n - r + 1$ compression function calls.  

**ALGORITHM: ConstructExpandableMessage($IV, r$)**  
Construct a $(r, r + 2^r + 1)$ expandable message. $q$ is a fixed "dummy" message used for getting the desired length. $h_{tmp}$ is an intermediate hash value and is set to $IV$ initially. 

**Steps:**

1) Calculate $r - 1$ times of $h = f(h, q)$, starting from $h = h_{tmp}$.
2) Calculate $2^{n/2}$ times of $f(h, m)$ as $m$ is an unique message block. Keep the paired results in $ListA = (h, m)$.
3) Calculate $2^{n/2}$ pairs of $(h', m')$ such that $h' = f(h_{tmp}, m')$. Keep the paired results in $ListB = (h', m')$.
4) Find a collision between $ListA$ and $ListB$ such that $h = h'$. Then we can obtain the colliding messages $(m', q||q||...||q||m)$ where $m', m$ are the corresponding message block in the pairs and there are $r - 1$ "dummy" messages. Set $h_{tmp} = h (= h')$.
5) Repeat the four steps above with $r = r - 1$ until $r = 0$.

To describe this algorithm more vividly, we give a schematic representation in Figure 2.

**Fig. 2. A schematic representation of constructing expandable message**

The pair messages in the dash box is a resulting of the cycle from steps 1 to steps 4. By extending a pair of collisions of messages, it is easy to build a complete expandable message.

**D. Lin et al.’s block-cipher-based hash**

A new scheme of hash functions based on block cipher [1] is proposed by LIN et al. In their paper, they refer that this hash function scheme has lower rate but higher efficiency and can be built on insecure compression functions. Some proofs of the security are given under black-box model and some compress functions based on block ciphers are shown. They also emphasis the key schedule is a more important factor that affects the efficiency of a block-cipher-based hash function than rate. The scheme is shown in Fig 3.

The scheme has two branches denoted by $H_1$ and $H_2$ respectively. The input is message $M$ which are split into $l$ blocks $M_1 ... M_l$. Two values $h_{01}, h_{02}$ are set to be the different initial values for each branch. Here $K_l$ is the key of the block cipher $E_{K_l}, E : \{0, 1\}^k \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a permutation where $k$ is the key length in bits and $n$ is the block length. $E_{K_1}$ and $E_{K_2}$ denote two different and independent permutations. $h_{11}$ and $h_{12}$ are the outputs of the two branches. The final result is $H = g(h_{11}||h_{12})$ where $g$ is a transformation.

**III. CRYPTOANALYSIS ON LIN ET AL.’S SCHEME**

In this section, we will give a detailed cryptanalysis on Lin et al.’s scheme. The result shows that the scheme can not achieve the collision, preimage and second preimage resistances.

**A. Multicollision attack**

Before getting started, $H_1$ and $H_2$ are independent hash functions on message $M$ with different IVs. If $\|H_1\|H_2$ is vulnerable to collision or (second) preimage attack, then $g$ can’t be collision resistant or (second) preimage resistant. For the scheme, the optimal security level of collision and (second) preimage resistance is $2^n$, $2^{2n}$, respectively. However, using Joux’s multicollision attack, we can find the collision and (second) preimage with less complexities.

1) **Collision Attack:** First, using Joux’s multicollision methods to construct $2^r$-collision messages with $r$ equal to $n/2$ on $H_1$. After this constructing is done, we obtain $2^r$ messages all hash to the same value on the $H_1$ side. On the other hand, since the security level of $H_2$ for collision resistance is $2^{n/2}$, we can, with a non-negligible probability, expect that among the $2^r (r = n/2)$ messages, at least two messages will also collide in the same hash value on the $H_2$ side. Thus we obtain such two messages $M_1, M_2$ that $H_1(M_1) = H_1(M_2)$ and $H_2(M_1) = H_2(M_2)$. In this case, $M_1, M_2$ are also two collide messages for $G$. Furthermore, the probability of success can be improved when $r$ is bigger. Considering the complexity of the attack, it costs $r2^{n/2} (r \geq n/2)$ operations to build multicollision on $H_1$ side and $2^{n/2}$ operations to find collision on $H_2$ side. Thus, the complexity of finding collisions on $(H_1||H_2)$ is $r2^{n/2} + 2^{n/2} (r \geq n/2)$, which is much less than $2^n$.

2) **Preimage and Second Preimage Attack:** The best general attack to find the preimage and second preimage of $H_1||H_2$ is operating exhaust search on the different messages until we hit the target. Thus the security level of (second) preimage resistance should be $2^{2n}$. However, using Joux’s multicollision construction, we can find out the preimage and second preimage with much less complexity, saying $r2^{n/2} + 2^n + 2^n (r \geq n)$ work. The attack works as follows.

First, using Joux’s multicollision methods to construct $2^r$-collision messages with $r$ equal to $n$ on $H_1$. After the
constructing is done, we obtain $2^r$ messages all hash to the same value $h_r$, on the $H_1$ side. It should be noted that the last chaining value $h_r$ is not the same as the target value on $H_1$ side, denoted as $h_{target}$. In order to maps $h_r$ to $h_{target}$, some additional search for $m_p$ should be operated such that $H_1(h_r, m_p) = h_{target}$. This search costs about $2^n$ work. Among the $2^r$ collision messages set $\{m_1 || m_2 || ... || m_r || m_p \mid m_1 = M_1$ or $M_i$, $i = 1, 2, ..., r\}$, we expect at least one of them also matches the target value on $H_2$ side, with an acceptable probability. Finally, this matched message is the preimage on $H_1 \parallel H_2$. As a consequence, it can be applied without any change when a second preimage is requested. Considering the complexity, it takes $r2^{n/2}$ work to build multicollision on $H_1$, $2^n$ work to find $m_p$, $2^n$ to find preimage on $H_2$. In total, this complexity is much less than the optimal security level $2^{2n}$.

B. Expandable message attack

In [3], a second preimage attack on long messages with expandable messages was proposed. The algorithm $LongMessageAttack(M_{target})$ was demonstrated thoroughly in the paper. Using this algorithm, not only a second preimage of the same length but also many other second preimages of different lengths are found for a given long message. However, what we expect in the attack on Lin et al.’s scheme is that all the second preimages are of the same length. Thus we make some adjustment with fixed points technique. The complete attack algorithm is elucidated as follows.

**ALGORITHM: SecondPreimageAttack$(h_{01}, r, M)$**

**Variables:**
- $h_{01}$ = the initial hash value on the $H_1$ branch;
- $r \geq n$;
- $M$ = the long target message with $2^r + r + 1$ blocks;
- $f_1$ = the compression function on $H_1$ branch;
- $m_{test}$ = an expandable message by fixed point;
- $h_{test}$ = the hash value after fixed point message;
- $M^*$ = the expandable message;
- $h^*$ = the intermediate hash value after making the expandable message;
- $M_{link}$ = a message block used to link the expandable message;
- $m_i$ = the $i$th message block of $M$;

**Steps:**

1) Use the $ConstructFixedPointsMessage(h_{01})$ and get $(m_0 || m')$. Set $m_{test} = (m_0 || m')$ and $h_{test} = f_1(h_{01}, m_{test})$.
2) $M^* = ConstructExpandableMessage(IV, r)$. $h^* = f_1(h_{01}, M^*)$.
3) Find $M_{link}$ such that $f_1(h^*, M_{link}) = h_i$ for some $r + 1 \leq i \leq 2^r + r + 1$ where $h_i = f_1(h_{i-1}, m_i)$ is the intermediate hash values.
4) For $(0 \leq j \leq 2^r - 1)$
   a) Output the second preimage
      $\{m_{test} || M^* || M_{link} || m_{i+1} || m_{i+2} || ... || m_{2^r+r+1}\}$

   where $M^*$ has $i - 3 - j$ blocks and $m_{test}$ has $2 + j$ blocks.

b) $m_{test} = m_{test} || m'$.

5) Among the $2^r$ second preimages of the same length produced in the last step, we expect with a non-negligible probability that one of them is also a second preimage on $H_2$ side. Finally, we find a message $M'$ such that $H_1(M') = H_1(M)$ and $H_2(M') = H_2(M)$.

Considering the complexity, it takes $2^{n/2+1}$ work for step1, $r2^{n/2} + 2^{n/2}$ work for step2, nearly negligible work for step3 with $r \geq n$, $2^n$ work respectively for step4 and step5. Thus in total, the complexity is about $2^{n/2} + 2^r$ work, less than $2^{2n}$.

**IV. Modifications**

The recent results on hash functions have motivated a lot of modifications and suggestions of fixing the flaws that were found. Here we discuss the modifications of Lin et al.’s scheme with Luck’s wide-pipe design [5] and Biham et al.’s HAIFA framework [21].

In Luck’s wide-pipe hash design, the internal hash value is widen up from $n$ bits to $\omega \geq 2n$ bits to prevent from finding the internal collisions. Thus every internal compression function $h'$ becomes $\{0, 1\}^\omega \times \{0, 1\}^m \rightarrow \{0, 1\}^\omega$ and the last compression function $h''$ becomes $\{0, 1\}^\omega \rightarrow \{0, 1\}^n$. As a result, finding the multicollision on $H_1$ branch needs $2^{\omega n/2} + 2^n$ work, much more than how much the original birthday attack needs. In other words, the multicollision attack does not take effect under the wide-pipe design. The expandable message attack fails in a similar way.

The main idea of Biham et al.’s HAsh Iterative FrAme-work (HAIFA) is adding the number of bits that were hashed so far and a salt value into the compression functions, which the chaining value is computed as $h_i = f(h_{i-1}, M_i, bits, salt)$. The bits value can prevent against the construction of the expandable message. The salt value is randomly chosen and not known in advance such that an adversary can’t pre-compute the multicollisions or the expandable message constructions before the choosing of the salt value. This indicates that the adversary have to transform the entire attack into an online attack, which is more difficult.
Figure 4 represents a modification with the solutions mentioned above. All the internal chaining values $h_i$ including the initial values $h_0$ are widen up to $\omega$ bits where $\omega \geq 2n$. The $counter_i$ keeps track of the number of bits that were hashed so far in each round. In the final round, the salt value is added into the compression function. The block cipher $E_{k_i}$ and the function $g$ should be modified accordingly.

V. Conclusion

In this paper, we have cryptanalyzed on Lin et al.’s efficient block-cipher-based hash construction. By using the multicolision and expandable messages techniques, our cryptanalysis shows that Lin et al.’s construction is not secure against the collision, preimage and second preimage attacks. The basic idea of the attacks can be concluded as follows. Firstly, try to find $2^r$ collisions or (second) preimages on one branch side (such as $H_1$ side) with $r$ is big enough. Then among the $2^r$ messages we can expect a collision or a (second) preimage with a non-negligible probability on the other branch. This attack can be applied to other similar hash constructions with two cascaded branches. As a result, a good design of a hash function should take this attack into consideration.

REFERENCES