IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 23, 2018, accepted August 2, 2018, date of publication August 10, 2018, date of current version September 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2864663

Preimage Attacks on Some Hashing Modes Instantiating Reduced-Round LBlock

SHIWEI CHEN[1,2] AND CHENHUI JIN[1]

[1]Zhengzhou Information Science and Technology Institute, Zhengzhou 450002, China
[2]Science and Technology on Information Assurance Laboratory, Beijing 100072, China

Corresponding author: Shiwei Chen (chenshiwei1012@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 61572516 and Grant 61772547, mainly on the analysis of the block ciphers and hash functions.

ABSTRACT In this paper, we present preimage attacks on several hashing modes instantiating reduced-round LBlock. It is observed that the omission of the network twist in the last round and the diffusion of the permutation in round function are the key points for our successful attack. First, to guarantee the validity of our attack, we prove one proposition on the round function. Then, utilizing the property of LBlock and several meet-in-the-middle techniques, we present a preimage attack on Davies-Meyer hashing mode instantiating 13-round LBlock, of which the time complexity is about $O(2^{55.4})$ 13-round compression function computations, less than the ideal complexity $O(2^{64})$ and the memory complexity is about 2^{12} 32-bit memory. Furthermore, we extend our results to the Matyas-Meyer-Oseas mode and MP mode with some changes. Finally, we convert the preimage attack into preimage attack or second preimage attack on the corresponding hash functions with Merkle-Damgard structure.

INDEX TERMS Hashing modes, preimage attack, LBlock, initial structure, splice-and-cut technique, partial matching.

I. INTRODUCTION

Hash function is an important kind of cryptographic algorithms, which is to transfer messages with arbitrary lengths to hash values with fixed length. Generally, it should 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 $H(M') = H(M)$.

Specially, for an n-bit hash function, if the time complexity of finding a preimage or second preimage is less than 2^n, then this hash function cannot resist preimage attack or second preimage attack. That is to say, it is not security with respect to preimage resistance or second preimage resistance.

To design security hash function, there are two mainstream methods. One is dedicated hash functions, such as MD5, SHA-2, SHA-3, and another is hash functions based on block ciphers. Particularly, if one needs both a block cipher and a hash function in a resource-restricted environment, and there only a block cipher, then we can build a hash function based on the block cipher through the mode-of-operations. In [8], Preneel *et al.* proposed 11 mode-of-operations to build a compression function from a block cipher. Among all these PGV modes, the Davies-Meyer (DM), Matyas-Meyer-Oseas (MMO) and Miyaguchi-Preneel (MP) modes are used in practice. Recently, Stam [9] revisited the hashing modes. Due to the practical application value of these hash functions, cryptographers analyze their security assuming the underlying block ciphers having different structures. In FSE 2011, Sasaki [10] used the meet-in-the-middle techniques to propose preimage attack on DM, MMO and MP hashing modes instantiating AES, where AES has the Substitute-Permutation (SP) structure. Known to everyone, besides SP network, Feistel and generalized Feistel networks are other important structures used in block cipher. In 2012, Moon et al. [6] proposed the meet-in-the-middle preimage attacks on hash modes of generalized Feistel and Misty schemes with SP round function. And then, Sasaki [11] further analyzed the security of the hashing modes instantiating Feistel or generalized Feistel network with an SP-round function, and presented preimage attacks on 11-round Feistel network and 15-round generalized Feistel network, where he utilized the omission of the twist operation in the last

round to finish 4-round shrink if the key schedule satisfied one condition. Then, he also applied the preimage attacks on Camellia-128 and ClEFIA-128, which have the Feistel network and 4-branch type-2 generalized Feistel network respectively. During the attacks, several meet-in-the-middle techniques, such as splice-and-cut technique [1], initial structure technique [12], partial matching [2] and so on, are used to guarantee the preimage attacks succeed on more rounds. Moreover, many literatures [15-18] uses different techniques to analyze the capability of Feistel or generalize Feistel network resisting meet-in-the-middle attack.

LBlock[14], designed by Wu *et al.* in 2011, utilizes the variant Feistel network, in which there is an additional rotation operation in the right side of the LBlock's Feistel network, which leads that we could not directly applied Sasaki's method in [11] on the hashing modes instantiating LBlock. So far, though there are many attacks on the LBlock block cipher [13,3,4], no result about the hashing modes instantiating LBlock was proposed. In this paper, based on the property of LBlock's round function, we utilize the techniques of initial structure, splice-and-cut and partial matching to propose preimage attack on the DM hashing mode instantiating reduced-round LBlock, and then we analyze the complexity of our attack.

This paper is outlined as follows. Section 2 introduces the LBlock and the workflow of some hashing modes; Section 3 proposes our preimage attack on the DM mode instantiation LBlock; Section 4 extends our attack to MMO and MP hashing modes; Section 5 concludes the whole paper and explains the future work.

II. DESCRIPTION OF BASIC KNOWLEDGE ON RELATED WORKS

In this section, we outline three hashing modes used in practice popularly, the specification of LBlock and the previous works about preimage attacks on hash functions.

A. HASH FUNCTIONS BASED ON BLOCK CIPHER

Generally, a hash function mainly includes two parts, domain extension and compression function, where the domain extension is to iterate the compression function. The most popular domain extension may be the Merkle-Damgård (MD) structure [5], which is used in MD5, SHA-1, SHA-2 and so on. In this paper, we assume the MD domain extension is used.

Then the following question is how to build a compression function based on a block cipher. For a block cipher E, the inputs are plaintext M and key K, while for a compression function, the inputs are message M and initial chaining value $h_0(IV)$. So, we could regard the message, or the initial chaining value, or the combination of message and initial chaining value, as plaintext or key, which produces 64 mode-of-operations described in [8]. Among all of them, DM, MMO and MP modes are used in practice. For example, MD5 uses DM mode, and MMO mode used in Lesamnt, *et al.*.

Let E be one block cipher, $M_i(i \geq 0)$ be the i-th input message, and H_i be the i-th chaining value, that is , the result after hashing the i-th input message M_i and $H_0 = IV$. Then the DM, MMO and MP modes are described as follows:

(1) DM mode : $CF(H_i, M_i) = E_{M_i}(H_i) \oplus H_i$
(2) MMO mode : $CF(H_i, M_i) = E_{H_i}(M_i) \oplus M_i$
(3) MP mode : $CF(H_i, M_i) = E_{H_i}(M_i) \oplus M_i \oplus H_i$

For the DM mode, the preimage attack is equal to the key recovery attack on block cipher and for the MMO mode and MP mode, the preimage attack is to recover the internal state of the block cipher. Moreover, in the analysis of hash functions, the messages could be chosen.

B. BRIEF DESCRIPTION OF LBLOCK

LBlock, a lightweight block cipher, was designed by Wenling Wu and Lei Zhang, which employs a variant Feistel structure (See Figure 1.) and consists of 32 rounds. For a 64-bit plaintext $M = X_0^L||X_0^R$, 32-bit round subkey $K_i(i = 1, 2, \cdots, 32)$ and the round function F, the encryption procedure is described as follows:

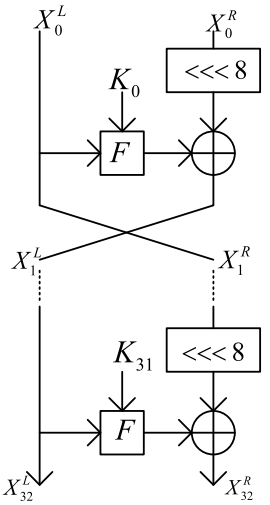


FIGURE 1. Encrytion procedure of LBlock.

Step1: For $i = 1$ to 31, do

$$X_i^L = (X_{i-1}^R <<< 8) \oplus F(X_{i-1}^L, K_i), X_i^R = X_{i-1}^L$$

Step 2: Compute

$$X_{32}^L = X_{31}^L, X_{32}^R = (X_{31}^R <<< 8) \oplus F(X_{31}^L, K_{32})$$

and output $X_{32}^L||X_{32}^R$ as the ciphertext.

Specifically, the round function F includes three layers (see Figure 2.), that is, Xor subkey, confusion function S and diffusion function P. For a 32-bit input X and 32-bit subkey K, the round function F is defined as follows:

$$F(X, K) = P(S(X \oplus K))$$

Confusion function S denotes the non-linear layer of round function F, and it consists of eight 4-bit S-boxes s_j in parallel.

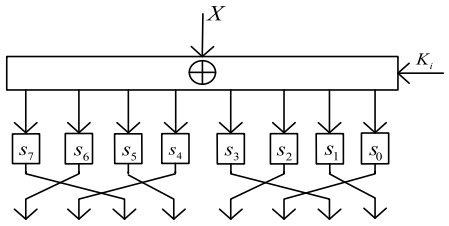


FIGURE 2. Round function F of LBlock.

Let

$$S : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$$
$$Y = Y_7||Y_6||\cdots||Y_0 \rightarrow Z = Z_7||Z_6||\cdots||Z_0$$

Then for $j = 0, 1, \cdots, 7$, we have $Z_j = s_j(Y_j)$, where $Y_j (0 \leq j \leq 7)$ and $Z_j (0 \leq j \leq 7)$ are 4-bit words.

Diffusion function P is defined as a permutation of eight 4-bit words. Let

$$P : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$$
$$Z = Z_7||Z_6||\cdots||Z_0 \rightarrow U = U_7||U_6||\cdots||U_0$$

we have

$$U_7 = Z_6, \quad U_6 = Z_4, \quad U_5 = Z_7, \quad U_4 = Z_5,$$
$$U_3 = Z_2, \quad U_2 = Z_0, \quad U_1 = Z_3, \quad U_0 = Z_1$$

From the above definition, we obtain the inversion of diffusion function P described as follows:

$$P^{-1} : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$$
$$U = U_7||U_6||\cdots||U_0 \rightarrow Z = Z_7||Z_6||\cdots||Z_0$$

where

$$Z_0 = U_2, \quad Z_1 = U_0, \quad Z_2 = U_3, \quad Z_3 = U_1,$$
$$Z_4 = U_6, \quad Z_5 = U_4, \quad Z_6 = U_7, \quad Z_7 = U_5$$

C. PREVIOUS WORKS OF PREIMAGE ATTACK ON HASH FUNCTIONS

One-wayness is one important secure principle for a secure hash function. In 2008, Aoki *et al.* [1] proposed some new techniques used in the meet-in-the-middle attack, including splice-and-cut, partial matching and partial fixing, and thereby proposed a preimage attack on the reduced-round MD5. Then, a series of new techniques are presented, such as initial structure[12], biclique, indirect matching, *et al.*. In the following, we introduce some of them used in our attack.

1) SPLICE-AND-CUT TECHNIQUE

Splice-and-cut technique [1] is to consider the first and last steps of the attack target as consecutive steps. Then, the attack target is divided into two chunks of steps so that each chunk includes at least one message word that is independent from the other chunk and such message words are called "neutral words."

2) INITIAL STRUCTURE

Initial structure [12] is a few consecutive steps including at least two neutral words named m^{2nd} and m^{1st}, where steps after the initial structure (2^{nd} chunk) can be computed independently of m^{1st} and steps before the initial structure (1^{st} chunk) can be computed independently of m^{2nd}.

3) PARTIAL MATCHING

Partial matching is executing only one-word matching instead of all-word matching, and up to three consecutive steps could be skipped from the attack target.

Based on the property of the target, we could combine the above techniques to improve the efficiency of our attacks.

D. NOTATIONS

X_i : the 64-bit state after i-th round;
X_i^R, X_i^L : the right and left 32 bits of X_i respectively;
S_i : the 32-bit state after S function in round i;
P_i : the 32-bit state after P function in round i;
K_i : the 32-bit subkey used in round i;
$Y[j_1, \cdots, j_k]$: the $j_1, \cdots, j_k$ 4-bit words of the variable Y, where Y can be X_i^R, X_i^L, P_i or S_i;
$Y[j_1 - j_k]$: the j_1 to j_k 4-bit words of the variable Y, where Y can be X_i^R, X_i^L, P_i or S_i;
$P_i^{[j_1-j_k]}$: the 32-bit state after P permutation in round i with the input that only the $j_1, \cdots, j_k$ 4-bit words are nonzero.

If there is no special explanations, these notations are used in the rest paper, and the order is from right to left and from down to up.

III. PREIMAGE ATTACK ON DM HASHING MODE WITH 13-ROUND LBLOCK

In this section, we explain the basic idea of our preimage attack on the LBlock compression function. In the whole attack procedure, we fix the key to a constant. Firstly, we find out the free 4-bit words for the forward chunk and backward chunk respectively, and construct 2-round initial structure. Due to the omission of the twist operation in the last round and diffusion property of the P transformation used in LBlock, we could compute forward 7 rounds and backward 5 rounds, and then utilize partial matching technique in the meeting point.

A. 2-ROUND INITIAL STRUCTURE PLUS 2 ROUNDS

To guarantee the validity of our attack, we firstly prove the following proposition based on the definition of the variant Feistel structure and the round function F of LBlock.

Proposition 1: Let

$$P_i^{[0,1,2,3]} = P(s_0(X_{i-1}^L[0] \oplus K_i[0])||s_1(X_{i-1}^L[1] \oplus K_i[1])|| \times s_2(X_{i-1}^L[2] \oplus K_i[2])||s_3(X_{i-1}^L[3] \oplus K_i[3])||0||0||0||0)$$

$$P_i^{[4,5,6,7]} = P(||0||0||0||0||s_4(X_{i-1}^L[4] \oplus K_i[4])||s_5(X_{i-1}^L[5] \oplus K_i[5]) \times ||s_6(X_{i-1}^L[6] \oplus K_i[6])||s_7(X_{i-1}^L[7] \oplus K_i[7]))$$

and t_i be a chosen value, then we have

$X_{i-1}^R = (t_i \oplus P_i^{[4,5,6,7]}) >>> 8$ and $X_i^L = t_i \oplus P_i^{[0,1,2,3]}$

Proof: From the definition of round function F, we have

$$\begin{aligned}
P_i &= P(S(X_{i-1}^L \oplus K_i)) \\
&= P(s_0(X_{i-1}^L[0] \oplus K_i[0])||s_1(X_{i-1}^L[1] \oplus K_i[1])|| \\
&\quad \times s_2(X_{i-1}^L[2] \oplus K_i[2])||s_3(X_{i-1}^L[3] \oplus K_i[3])|| \\
&\quad \times s_4(X_{i-1}^L[4] \oplus K_i[4])||s_5(X_{i-1}^L[5] \oplus K_i[5])|| \\
&\quad \times s_6(X_{i-1}^L[6] \oplus K_i[6])||s_7(X_{i-1}^L[7] \oplus K_i[7])) \\
&= P(s_0(X_{i-1}^L[0] \oplus K_i[0])||s_1(X_{i-1}^L[1] \oplus K_i[1])|| \\
&\quad \times s_2(X_{i-1}^L[2] \oplus K_i[2])||s_3(X_{i-1}^L[3] \oplus K_i[3])||0||0||0||0 \\
&\quad \oplus 0||0||0||0||s_4(X_{i-1}^L[4] \oplus K_i[4])||s_5(X_{i-1}^L[5] \oplus K_i[5]) \\
&\quad \times ||s_6(X_{i-1}^L[6] \oplus K_i[6])||s_7(X_{i-1}^L[7] \oplus K_i[7])) \\
&= P(s_0(X_{i-1}^L[0] \oplus K_i[0])||s_1(X_{i-1}^L[1] \oplus K_i[1])|| \\
&\quad \times s_2(X_{i-1}^L[2] \oplus K_i[2])||s_3(X_{i-1}^L[3] \oplus K_i[3])||0||0||0||0) \\
&\quad \oplus P(0||0||0||0||s_4(X_{i-1}^L[4] \oplus K_i[4])||s_5(X_{i-1}^L[5] \oplus K_i[5]) \\
&\quad \times ||s_6(X_{i-1}^L[6] \oplus K_i[6])||s_7(X_{i-1}^L[7] \oplus K_i[7]))
\end{aligned}$$

That is, $P_i = P_i^{[0,1,2,3]} \oplus P_i^{[4,5,6,7]}$. Moreover, from the definition of variant Feistel structure of LBlock, we have $X_i^L = (X_{i-1}^R <<< 8) \oplus P_i$, that is,

$$X_i^L \oplus P_i^{[0,1,2,3]} = (X_{i-1}^R <<< 8) \oplus P_i^{[4,5,6,7]}.$$

Let $X_i^L \oplus P_i^{[0,1,2,3]} = (X_{i-1}^R <<< 8) \oplus P_i^{[4,5,6,7]} = t_i$, we have $X_{i-1}^R = (t_i \oplus P_i^{[4,5,6,7]}) >>> 8$ *and* $X_i^L = t_i \oplus P_i^{[0,1,2,3]}$. □

Based on the above Proposition 1, 2-round initial structure could be constructed by exploiting the small branch number of the LBlock's P-layer. 4 rounds, from round 7 to round 10, are shown in *Figure 3*. The initial structure is located in round 8 and 9. Throughout this paper, the words depending on the free 4-bit words for the forward chunk and values are shown in red, while the words depending on the free 4-bit words for the backward chunk and values are shown in blue. The fixed 4-bit words and unknown words are shown in gray and white respectively.

Firstly, we choose three free 4-bit words $X_7^L[0, 1, 2]$ and two free 4-bit words $X_9^R[4, 6]$ for forward chunk and backward chunk respectively. During round 8, three free 4-bit words $X_7^L[0, 1, 2]$ affect the three 4-bit words of $S_8[0, 1, 2]$ and then diffuse into $P_8[0, 2, 3]$. Similarly, two free 4-bit words $X_9^R[4, 6]$ affect $X_8^L[4, 6]$ and then $P_9[6, 7]$ through the S and P in round 9. Due to the linearity of P function, the impact from the free 4-bit words for forward chunk and backward chunk respectively can be computed independently. During round 8, the impact from the three free 4-bit words $X_7^L[0, 1, 2]$ is denoted by $P_8^{[0,1,2]}$, and we have

$$\begin{aligned}
P_8^{[0,1,2]} = P(&s_0(X_7^L[0] \oplus K_8[0])||s_1(X_7^L[1] \oplus K_8[1])|| \\
&\times s_2(X_7^L[2] \oplus K_8[2])||0||0||0||0||0)
\end{aligned}$$

Randomly choosing a value t_8 and according to proposition 1, we could compute

$$X_8^L = t_8 \oplus P_8^{[0,1,2,3]}$$

Hence, according to the definition of P, we know that $X_8^L[0, 2, 3]$ are depend on the three free words $X_7^L[0, 1, 2]$ and then affect $P_9[1, 2, 3]$. During round 9, denote the impact from the two free 4-bit words $X_9^R[4, 6]$ by $P_9^{[4,6]}$, and we have

$$\begin{aligned}
P_9^{[4,6]} = P(&0||0||0||0||s_4(X_8^L[4] \oplus K_9[4])||0|| \\
&\times s_6(X_8^L[6] \oplus K_9[6])||0)
\end{aligned}$$

Randomly choose a value t_9 and compute

$$X_8^R = (t_9 \oplus P_9^{[4,5,6,7]}) >>> 8$$

We know that $X_8^R[4, 5]$ depends on the two free words $X_9^R[4, 6]$, and then affects $P_8[4, 6]$. Hence, the free words for the forward chunk and the backward chunk do not affect $P_8^{[4,5,6,7]}$ and $P_9^{[0,1,2,3]}$ respectively, so we could compute $X_7^R = (t_8 \oplus P_8^{[4,5,6,7]}) >>> 8$ and $X_9^L = t_9 \oplus P_9^{[0,1,2,3]}$ with regardless of the free words.

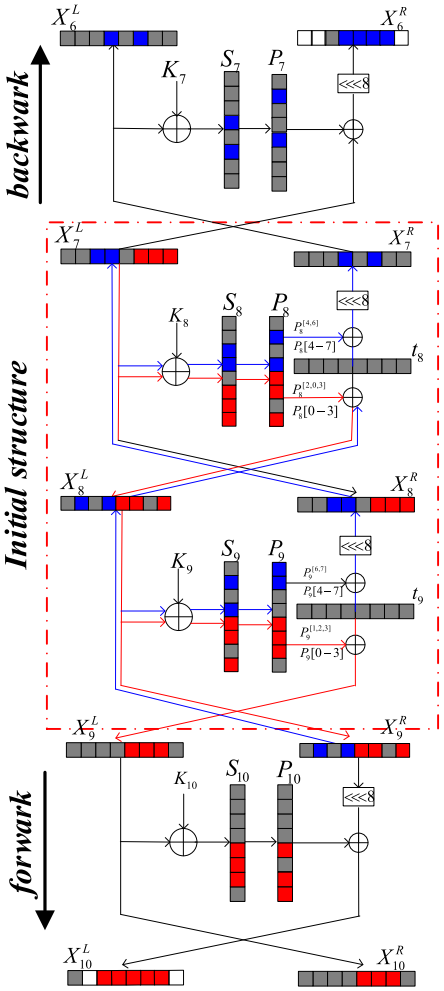


FIGURE 3. 4 rounds including the 2-round initial structure.

From Figure 3, we know that during the initial structure, the computations depending on three 4-bit words $X_7^L[0, 1, 2]$ do not affect the computations of the two 4-bit words $X_9^L[4, 6]$. Furthermore, in the computations of each chunk, free 4-bit

words of another chunk are regarded as unknown. Then, the computations of round 10 and round 7 are straight-forward.

B. BACKWARD AND FORWARD PROCEDURES

After building the initial structure as described in section A, we can compute forward and backward respectively. During the forward computation, an important observation is that the omission of the twist operation in the last round. Since our object is the reduced-round LBlock and the hashing mode is DM, we try to use the splice-and-cut technique, and thereby need to judge firstly which round is the last round to guarantee that the matching succeed and we could attack as more rounds as possible. Particularly, if the twist operation in the last round exists, all 4-bit words become unknown after two rounds. Here we choose round 13 as the last round and the forward chunk and backward chunk respectively include rounds 10-13,1-2 and rounds 3-7.

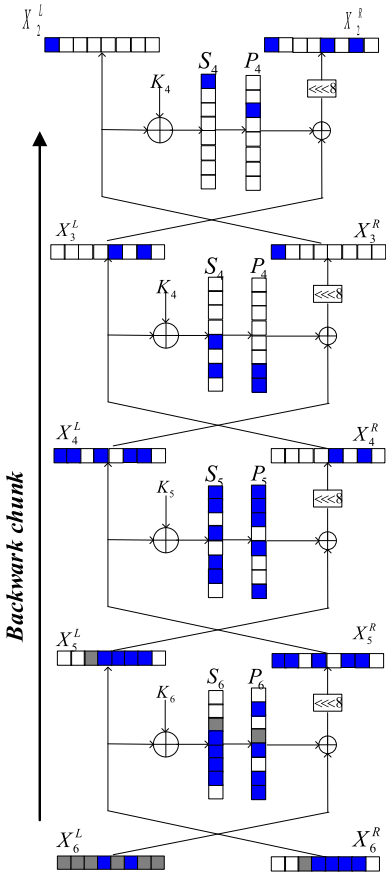


FIGURE 4. Backward including the 4 Rounds.

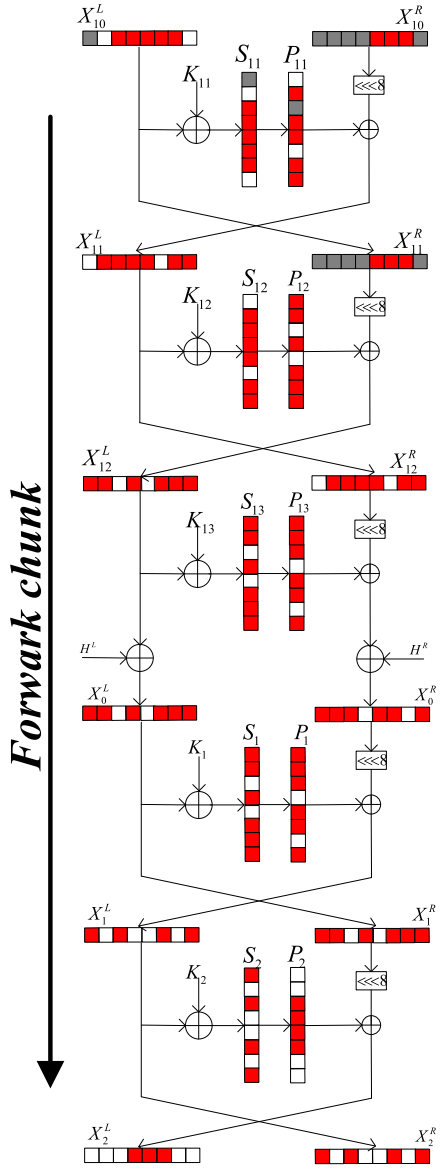


FIGURE 5. Forward phase including 6 rounds.

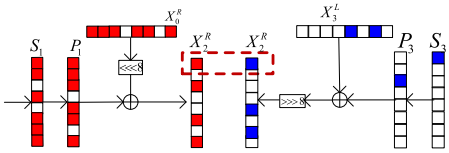


FIGURE 6. Detailed matching procedure.

From state X_6, we can compute backward straight-forward until state X_2 where only X_2^L [7] and $X_2^R[1, 3, 7]$ are determined (see Figure 4). From the state X_{10}, we compute forward until state X_2 where $X_2^L[2, 3, 4]$ and $X_2^R[0, 2, 5, 7]$ are determined (see Figure 5).

From the above backward and forward procedures, we know that the direct match can be applied due to the overlapped 4-bit word $X_2^R[7]$. See Figure 6 for the detailed matching procedure.

C. OUR ATTACK PROCEDURE AND COMPLEXITY

Based on the path of the DM hashing mode with 13-round LBlock constructed in sections 1 and 2, we could propose a preimage attack as follows.

For a target $H^L||H^R$, we aim to find a one-block preimage M such that $H^L||H^R = H(IV, M)$ as follows:

Step1: Randomly choose a one-block message M;

Step2: For all choices of $X_7^L[3, 6, 7]$, t_8 and t_9, do as follows.

Step2.1: Choose all the three free 4-bit words $X_7^L[0, 1, 2]$ for the forward chunk, and compute the values of X_2^R. Store the results in a table;

Step2.2: Choose all the two free 4-bit words $X_9^R[4, 6]$ for the backward chunk, and compute the values of X_2^R. Then check if the same value exists in the table with respect to the 7^{th} 4-bit word of X_2^R;

Step2.3: If the match with the 4-bit word X_2^R [7] is found, check the match of other 60 bits. If all bits match, output the chosen key in Step 1; Or else, return to Step2.

In the following, we analyze the complexity of our attack.

For each value of Step2, Step2.1 is iterated 2^{12} times, and requires 2^{12} 32-bit memory. Step2.2 is iterated 2^8 times. The sum of the complexities for Step2.1 and Step2.2 is about $2^{12} \times (8 - round) + 2^8 \times (7 - round)$. Since the probability that the 7^{th} 4-bit word of X_2^R is matched is about 2^{-4}, there are $2^{12+8-4} = 2^{16}$ values remaining after Step2.2, which will be checked in Step2.3. Hence, in order to match all the remaining 15 4-bit words, Step2.1-2.2 need to be iterated $2^{60-16} = 2^{44}$ times. So all the time complexity of our attack is about

$$2^{44} \times [2^{12} \times (8 - round) + 2^8 \times (7 - round)] \approx 2^{55.4}$$

13-round LBlock computations. And the memory complexity is about 2^{12} 32-bit memory.

Remark: The above attack on the DM modes with 13-round LBlock can be applied on MMO mode and MP mode with 13-round LBlock directly, where the only difference is the variable chosen in Step1. For MMO and MP, we should choose the initial chaining value IV at first to guarantee the key be known in our attack and output the initial state as the preimage.

IV. EXTENSION OF OUR ATTACK TO HASH FUNCTIONS

We propose preimage attack on the DM/MMO/MP modes with 13-round LBlock in section 3. Now we would extend those attacks to hash functions with MD structure, which depends on the mode-of-operation used.

For the DM mode $E_{M_{N-1}}(H_{N-1}) \oplus H_{N-1}$, this attack finds (M_{N-1}, H_{N-1}) where the value of M_{N-1} is chosen and the value of H_{N-1} is determined during the attack. Hence, we choose the message such that the padding string can be satisfied. However, H_{N-1} cannot be fixed to IV. So, we could only find a pseudo-preimage.

Proposition 2 (Pseudo-Preimages Yielding Preimages [7]): If the compression function f of an n-nnbit iterated hash function h does not have ideal computational security against pseudo preimage attacks and pseudo-preimages can be found in 2^s operations, then preimages for h can be found in $2^{1+(n+s)/2}$ operations.

According to the conversion method in [7] described above, the pseudo-preimages for DM mode can be translated into preimages with the time complexity is about $2^{1+(55.4+64)/2} \approx 2^{60.7}$, and For the MMO mode and MP mode, second preimage attack can be proposed with the same time complexity of $2^{60.7}$.

V. CONCLUSIONS

In this paper, we study the security of LBlock hashing modes in terms of preimage attack and second preimage attack. By combining meet-in-the-middle with initial structure and direct partial matching, we present preimage attacks on the three hashing modes instantiating reduced-round LBlock. And then based on the preimage attacks, we obtain a preimage attack on the 13-round DM-LBlock with MD structure with the time complexity of $2^{60.7}$ and a second preimage attack on the 13-round MMO-LBlock and MP-LBlock with MD structure with the complexity of $2^{60.7}$, less the ideal time complexity of 2^{64}. This paper is mainly to evaluate the capability of resisting meet-in-the-middle attack of hash functions based on the block cipher LBlock and in the future we will try other new methods to analyze the security of this kind of hash functions.

REFERENCES

[1] K. Aoki and Y. Sasaki, "Preimage attacks on one-block MD4, 63-step MD5 and more," in *Selected Areas in Cryptography—SAC* (Lecture Notes in Computer Science), vol. 5381. Springer, 2009, pp. 103–119.

[2] K. Aoki and Y. Sasaki, "Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 5677. Berlin, Germany: Springer, 2009, pp. 70–89.

[3] C. Boura *et al.*, "Scrutinizing and improving impossible differential attacks: Applications to CLEFIA, Camellia, LBlock and Simon," Cryptol. ePrint Arch., Tech. Rep. 2014/699, 2014.

[4] J. Chen *et al.*, "Impossible differential cryptanalysis of LBlock with concrete investigation of key scheduling algorithm," Cryptol. ePrint Arch., Tech. Rep. 2014/272, 2014.

[5] I. B. Damgård, "A design principle for hash functions," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 435. Berlin, Germany: Springer, 1990, pp. 416–427.

[6] D. Moon, D. Hong, D. Kwon, and S. Hong, "Meet-in-the-middle preimage attacks on hash modes of generalized feistel and misty schemes with SP round function," *IEICE Trans.*, vol. E95.A, no. 8, pp. 1379–1389, 2012.

[7] A. J. Menezes *et al.*, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1997.

[8] B. Preneel, R. Govaerts, and J. Vandewalle, "Hash functions based on block ciphers: A synthetic approach," in *Advances in Cryptology—CRYPTO* (Lecture Notes in Computer Science), vol. 773. Berlin, Germany: Springer, 1994, pp. 368–378.

[9] M. Stam, "Blockcipher-based hashing revisited," in *Fast Software Encryption—FSE* (Lecture Notes in Computer Science), vol. 5665. Berlin, Germany: Springer, 2009, pp. 67–83.

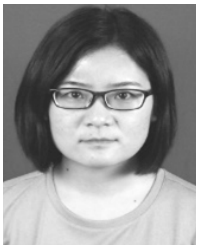
[10] Y. Sasaki, "Meet-in-the-middle preimage attacks on AES hashing modes and an application to whirlpool," in *Fast Software Encryption—FSE* (Lecture Notes in Computer Science), vol. 6733. Berlin, Germany: Springer, 2011, pp. 378–396.

[11] Y. Sasaki, "Preimage attacks on feistel-SP functions: Impact of omitting the last network twist," in *Applied Cryptography and Network Security—ACNS* (Lecture Notes in Computer Science), vol. 7954. Berlin, Germany: Springer, 2013, pp. 170–185.

[12] Y. Sasaki and K. Aoki, "Finding preimages in full MD5 faster than exhaustive search," in *Advances in Cryptology—EUROCRYPT* (Lecture Notes in Computer Science), vol. 5479. Berlin, Germany: Springer, 2009, pp. 134–152.

[13] Y. Sasaki and L. Wang, "Meet-in-the-middle technique for integral attacks against feistel ciphers," *Selected Areas in Cryptography—SAC* (Lecture Notes in Computer Science), vol. 7707. Berlin, Germany: Springer, 2013, pp. 234–251.

[14] W. Wu and L. Zhang, "LBlock: A lightweight block cipher," in *Applied Cryptography and Network Security—ACNS* (Lecture Notes in Computer Science), vol. 6715. Berlin, Germany: Springer, 2011, pp. 327–344.
[15] J. Guo, J. Jean, I. Nikolic, and Y. Sasaki, "Meet-in-the-middle attacks on classes of contracting and expanding feistel constructions," *IACR Trans. Symmetric Cryptol.*, vol. 2016, no. 2, pp. 307–337, 2016.
[16] L. Lin, W. Wu, and Y. Zheng, "Improved meet-in-the-middle distinguisher on feistel schemes," in *Proc. Int. Conf. Sel. Areas Cryptogr.* Berlin, Germany: Springer, 2015, pp. 122–142.
[17] J. Guo, J. Jean, I. Nikolić, and Y. Sasaki, "Meet-in-the-middle attacks on generic feistel constructions," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2014, pp. 458–477.
[18] L. Dong and Y. Mao, "Meet-in-the-middle attacks on 3-line generalized feistel networks," Cryptol. ePrint Arch., Tech. Rep. 2017/1071, 2017.



SHIWEI CHEN was born in Nanyang, Henan, China, in 1983. She received the Ph.D. degree in cryptography from the Zhengzhou Information Science and Technology Institute, Zhengzhou, Henan, China, in 2012. Her current research interests include the analysis and designs of cryptographic algorithms.

CHENHUI JIN was born in Zhoukou, Henan, China, in 1965. His current research interests include the analysis and designs of cryptographic algorithms.