Differential Sieving for 2-Step Matching Meet-in-the-Middle Attack with Application to LBlock

Riham AlTawy and Amr M. Youssef(✉)

Concordia Institute for Information Systems Engineering,
Concordia University, Montréal, QC, Canada
youssef@ciise.concordia.ca

Abstract. In this paper, we propose a modified approach for the basic meet-in-the-middle attack which we call differential sieving for 2-step matching. This technique improves the scope of the basic meet in the middle attack by providing means to extend the matching point for an extra round through differential matching and hence the overall number of the attacked rounds is extended. Our approach starts by first reducing the candidate matching space through differential matching, then the remaining candidates are further filtered by examining non shared key bits for partial state matching. This 2-step matching reduces the total matching probability and accordingly the number of remaining candidate keys that need to be retested is minimized. We apply our technique to the light weight block cipher LBlock and present a two known plaintexts attack on the fifteen round reduced cipher. Moreover, we combine our technique with short restricted bicliques and present a chosen plaintext attack on Lblock reduced to eighteen rounds.

Keywords: Cryptanalysis · Meet-in-the-middle · Differential sieving · LBlock · Short bicliques

1 Introduction

Meet in the middle (MitM) attacks have drawn a lot of attention since the inception of the original attack which was first proposed in 1977 by Diffie and Hellman [13] for the analysis of the Data Encryption Standard (DES). Soon after, the attack became a generic approach to be used for the analysis of ciphers with non complicated key schedules. For this class of ciphers, one can separate the execution into two independent parts where each part can be computed without guessing all the bits of the master key. The first execution part covers encryption rounds from the plaintext to some intermediate state and the other part covers decryption rounds from the corresponding ciphertext to the same internal state. At this point, the attacker has knowledge of the same intermediate state from two independent executions where the right key guess produces matching states. A typical MitM attack can be launched with as low as one known

© Springer International Publishing Switzerland 2015
T. Eisenbarth and E. Öztürk (Eds.): LightSec 2014, LNCS 8898, pp. 126–139, 2015.
DOI: 10.1007/978-3-319-16363-5_8

plaintext-ciphertext pair. Accordingly, with the recent growing interest in low data complexity attacks [9], the MitM attack has witnessed various improvements and has been widely adopted for the analysis of various cryptographic primitives. The increasing motivation for adopting low data complexity attacks for the analysis of ciphers is backed by the fact that security bounds are better perceived in a realistic model. Particularly, in a real life scenario, security protocols impose restrictions on the amount of plaintext-ciphertext pairs that can be eavesdropped and/or the number of queries permitted under the same key.

With the current popularity of lightweight devices such as RFID chips and wireless sensor networks, the demand for efficient lightweight cryptography is increasing. These devices offer convenient services on tiny resource constrained environments with acceptable security and privacy guarantees. In particular, the employed ciphers must obey the aggressive restrictions of the application environment while maintaining acceptable security margins. As a result, the designers of lightweight ciphers are often forced to make compromising decisions to fulfil the required physical and economical constraints. Among the designs that have been proposed to address these needs are PRESENT [7], KATAN and KTANTAN [12], LED [15], Zorro [14], and LBlock [25]. All of these proposals have received their fair share of cryptanalytic attacks targeting their weak properties [4,8,18,19,21]. Such unfavourable properties usually are the result of the desire of the designers to conform to the resources constraints.

The success of the MitM attack depends of the speed of key diffusion. Complex key schedules amount for quick diffusion and hence the knowledge of the state after few rounds involves all the bits in the master key. Indeed, one can say that the witnessed renewal of interest in MitM attacks is due to the emergence of lightweight ciphers which often tend to employ simple key schedules with relatively slow diffusion to meet the resources constraints.

In this work, we present *differential sieving for 2-step matching*, a technique that improves the scope and the key retesting phase of the basic meet in the middle attack. More precisely, our technique enables the attacker to cover at least one extra round in an execution direction when all state knowledge is not available. This extension is accomplished by matching possible differential transitions through the Sbox layer instead of matching actual state values. Afterwards, the remaining candidate keys from both directions are used to evaluate the state for only one round, which is further matched by the actual bit values. The proposed 2-step differential-value matching reduces the total matching probability and accordingly the number of remaining key candidates that need to be retested is minimized. We demonstrate our technique on the light weight block cipher LBlock and present a two known plaintexts attack on the fifteen round reduced cipher. Finally, we combine our approach with restricted short bicliques and present an eighteen round attack with a data complexity of 2^{17}.

The rest of the paper is organized as follows. In the next section, we explain our proposed technique and give a brief overview on the basic meet-in-the-middle attack and the idea of short restricted bicliques. Afterwards, in Sect. 3, we give the specification of the lightweight block cipher Lblock and provide detailed

application of our attack on it. Specifically, we present a low-data complexity attack on the fifteen round reduced cipher and a restricted biclique attack on the cipher reduced to eighteen rounds. Finally, the paper is concluded in Sect. 4.

2 Differential Sieving for 2-Step Matching

Our approach is a modified approach of the basic meet-in-the-middle attack [13] which was first proposed in 1977. Throughout the following years, the attack has been used in the security analysis of a large number of primitives including block ciphers, stream ciphers, and hash functions. The basic attack has undergone major improvements to make it better suit the attacked primitive. In particular, the cut and splice technique [2] and the initial structure approach [20] are successfully used in MitM preimage attacks on hash function [1,24]. Moreover, partial matching [3] allows the matching point to cover more rounds through matching only known parts of the state. Other examples of these techniques include 3-subset MitM [8] and sieve-in-the-middle [10]. It is worth noting that most MitM attacks are low data complexity attacks except when used with bicliques [6], which represent a more formal description of the initial structure and can be constructed from related key differentials.

2.1 Basic Meet-in-the-middle Attack

The basic MitM attack recovers the master key of a given cipher more efficiently than by brute forcing it. As depicted in Fig. 1, the attack idea can be explained as follows: let the attacked primitive be an r-round block cipher operating under a fixed master key K. Let $E_{i,j}(p)$ denote the partial encryption of the plaintext p and $D_{i,j}(c)$ denote the partial decryption of the ciphertext c, where i and j are the starting and ending rounds for the partial encryption/decryption, respectively. If one can compute $E_{1,j}(p)$ using K_1 and $D_{r,j}(c)$ using K_2 such that K_1 and K_2 do not share some key bits and each key guess does not involve the whole master key, then the same state of the cipher can be computed independently from the encryption and decryption sides. More precisely, each guess of K_1 allows us to compute a candidate $E_{1,j}(p)$ and each guess of K_2 gives a candidate $D_{r,j}(c)$. Since the output of both executions is state j, then all the guessed (K_1, K_2) pairs that result in $E_{1,j}(p) = D_{r,j}(c)$ are considered potentially right keys and one of them must be the right key.

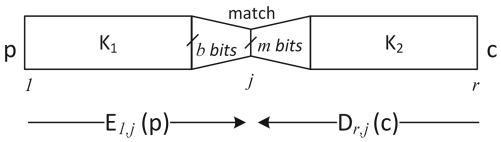

Fig. 1. Basic MitM attack

Generally, given one plaintext-ciphertext pair, the number of potentially right keys depends on the number of bits that are matched at state j. More formally, given a b-bit state, a k-bit master key, and the knowledge of m bits at the matching state, the number of potentially right keys is 2^{k-m}, where 2^{-m} is the probability that the two states match at the available m bits. Partial matching takes place if $m < b$ and hence the 2^{k-m} candidate keys need to be retested for full state matching. In the case of $m = b$, only the relation between the key size k and the state size b determines the number of potentially right keys. If $b = k$, then only the right key remains and no further testing is required. However, in some cipher, designers tend to use master keys that are larger than the state size in order to provide a higher security margin in more constrained environments. Accordingly, even if we are performing full state matching, we end up with 2^{k-b} potentially right keys. Consequently, more plaintext-ciphertext pairs are needed to find the right key. Indeed, whether we are partially or fully matching the states, if $k > b$, we get a set of 2^{k-b} potentially right keys after retesting with one plaintext-ciphertext pair. In this case, to recover the right key, the data complexity of the MitM attack is $n = \lceil \frac{k}{b} \rceil$ plaintext-ciphertext pairs.

2.2 Differential Sieving Approach

The time complexity of the MitM attack is divided into two main components: the MitM part, where both forward and backward computations take place, and the key retesting part where the remaining potentially right keys need to be rechecked. More formally, let $K_s = K_1 \cap K_2$ be the set of bits shared between K_1 and K_2, $K_f = K_1 \backslash K_1 \cap K_2$ be the set of bits in K_1 only, and $K_b = K_2 \backslash K_1 \cap K_2$ be the set of bits in K_2 only. We also assume that the master key $K = K_1 \cup K_2$. The time complexity of the MitM attack is given by:

$$\underbrace{2^{|K_s|}(2^{|K_f|} \cdot c_f + 2^{|K_b|} \cdot c_b)}_{\text{MitM}} + \underbrace{2^{|K|} \times 2^{-\rho_s} \cdot c_t}_{\text{Key testing}},$$

where $2^{-\rho_s}$ is the total matching probability, c_f and c_b denotes the costs of partial computations in the forward and backward directions, and $c_t = c_f + c_b$ is the total cost of computation. According to this equation, the MitM attack recovers the right key more efficiently than exhaustively searching the master key space if both K_f and K_b are non empty sets and the number of remaining keys that need retesting is not too large. In other words, besides efficient separation of executions, the key retesting part of the MitM attack should not be the component dominating the attack time complexity. Evidently, the lower the total matching probability $2^{-\rho_s}$ is, the less keys remain for rechecking.

Our proposed *differential sieving 2-step matching* approach provides the means to decrease the total matching probability by matching both the possible differential transitions through the Sbox layer and actual partial state bit values. The technique also allows us to extend the basic MitM by at least one more round when the key knowledge is not available anymore. Indeed, when we have knowledge of parts of the state before the subkey mixing, we lose this

state knowledge if the bits of this subkey are not in the guessed key set. However, if we are using two plaintext-ciphertext pairs, we know the difference after the key mixing with certainty. To this end, we can extend our state difference knowledge with one extra round and use it for matching possible input/output differences through the following substitution layer. Our technique requires at least two plaintext-ciphertext pairs which is fortunate for two reasons: first, we match two different states variables thus we get lower total matching probability. Second, when we analyze a light weight cipher whose key size is larger than the block size (the case for LBlock), the MitM attack requires more than a single plaintext-ciphertext pair to recover the right key anyway.

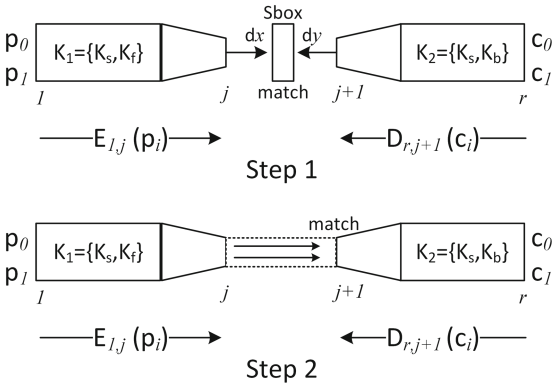

Fig. 2. Difference-value 2-step matching MitM attack

Figure 2 depicts the high level idea of the proposed 2-step matching approach and shows how it differs from the basic approach shown in Fig. 1. The core idea of the attack is to efficiently separate the execution such that we get the same partial state knowledge (value or difference) from both the backward and forward directions. Let Xf_j^i and Xb_j^i be the states at round j resulting from the forward and backward executions using the i^{th} plaintext and ciphertext, respectively. As depicted in Fig. 2, the forward execution ends at round j where we are forced to terminate the execution because the available knowledge about the m-bit partial state Xf_j^i will be lost after the key mixing in round $j+1$. More precisely, let K_m be the set of bits in the master key that are Xored with the m-bit partial state Xf_j^i at round $j+1$. If $K_m \notin K_f$ then we lose the matching knowledge in state Xf_{j+1}^i. However, since the master key $K = K_1 \cup K_2$, then with certainty $K_m \in K_b$. The procedure of our attacks is described as follows:

- For each guess k_s of the $2^{|K_s|}$ values of K_s
  - For each guess k_f of the $2^{|K_f|}$ values, partially encrypt the two plaintexts p_0 and p_1, and store the resulting m-bit partial states values (Xf_j^0, Xf_j^1) and difference $dx = (Xf_j^0 \oplus Xf_j^1)$ in a table T

- For each guess k_b of the $2^{|K_b|}$ values, partially decrypt the two corresponding ciphertexts c_0 and c_1 to get the two m-bit partial states values (Xb^0_{j+1}, Xb^1_{j+1}).
  1. Step 1: check if the resulting difference $dy = (Xb^0_{j+1} \oplus Xb^1_{j+1})$ and any of the dx entries in T is a possible differential transition through the Sbox layer. If yes then go to Step 2, else guess another k_b.
  2. Step 2: get K_m from k_b and using the forward table T entry which matched in step 1, compute $Xf^0_{j+1} = Xf^0_j \oplus K_m$ and $Xf^1_{j+1} = Xf^1_j \oplus K_m$. Check if these two states are equal to Xb^0_{j+1}, Xb^1_{j+1}. If yes then add the current master key candidate $(k_s \cup k_f \cup k_b)$ to a list L_p of potentially right keys, else guess another k_b.
- Exhaustively test if any of the candidate keys in L_p encrypts both plaintexts p_0 and p_1 to their corresponding ciphertexts c_0 and c_1. If yes output it as the correct master key, else guess another k_s

Other than extending the matching point by one more round, the main gain of the 2-step matching is reducing the total matching probability and hence the false positive keys whose retesting may dominate the total attack complexity. More formally, given an m-bit matching knowledge, let the probability of a possible differential for a given Sbox be $2^{-\alpha}$, then the total matching probability in our approach is $2^{-\rho_s} = 2^{-\alpha} \times 2^{-m} \times 2^{-m}$. Assuming $c_f \approx c_b \approx c_t/2$, the total time complexity of the attack is given by:

$$2^{|K_s|}(2^{|K_f|} + 2^{|K_b|}) \cdot c_t + 2^{|K|-\alpha-2m} \cdot c_t$$

The memory complexity is given by $2^{|K_f|} + 2^{|K_f|+|K_b|-\alpha-2m}$. This complexity may be negligible as the sizes of both lists T and L_p are usually small. The data complexity is equal to the unicity distance of the analyzed cipher.

2.3 Short Restricted Bicliques

Biclique cryptanalysis [6] was first used to present an accelerated exhaustive search on the full round AES. The basic idea of bicliques is to increase the number of rounds of the basic MitM attack. As depicted in Fig. 3, a biclique is a structure of the 3-tuple $\{p^u_j, s^u_i, K_q\}$ where K_q denotes the key bits used to encrypt the plaintext p to an intermediate state s. K_q is partitioned into three disjoint sets of key bits $K_q = \{K_5, K_3, K_4\}$ such that for a given u of the $2^{|K_5|}$ values of K_5, the states variables between p and s that are affected by a change in the value of K_3 are different than those affected by a change in the value of K_4. More formally, let $Enc_{[u,i,j]}(p)$ and $Dec_{[u,i,j]}(s)$ denote the encryption and decryption of the plaintext p and intermediate state s using the u, i, and j values of K_5, K_3, and K_4, respectively. A biclique can be constructed if for all u, i, and j of the $2^{|K_5|}, 2^{|K_3|}$, and $2^{|K_4|}$ values, respectively, the computation of $s^u_i = Enc_{[u,i,0]}(p)$ does not share any active state variables with the computation $p^u_j = Dec_{[u,0,j]}(s)$. Since both the forward and backward computations generate two independent differential paths, one can deduce that $s^u_i = Enc_{[u,i,j]}(p^u_j)$.

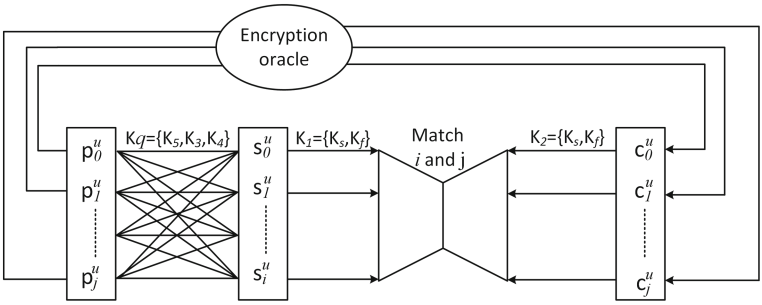

Fig. 3. Combining short restricted bicliques with basic MitM attack

In our attack we employ short restricted bicliques [10]. They are restricted in the sense that we do not have the freedom in partitioning K_q as in the case of conventional biclique cryptanalysis [6]. More precisely, we first separate the forward and backward execution of the MitM attack to get K_1 and K_2. In the sequel, the choice of K_3, K_4, and K_5 must conform to some restrictions. Particularly, $K_3 \in K_f$, $K_4 \in K_b$, and $K_5 \in K_s$ so that each choice of K_s, K_f, and K_b of the 2-step matching MitM attack gives a unique $K_q = \{K_5, K_3, K_4\}$ candidate that can be used to construct the biclique. For a given u of the $2^{|K_5|}$ values, a biclique is constructed as follows:

- Get the base states (p_0^u, s_0^u): choose the base plaintext $p_0^u = 0$ and set $K_q = \{K_5, K_3, K_4\} = \{u, 0, 0\}$. The base intermediate state $s_0^u = Enc_{[u,0,0]}(p_0^u)$. Note that $p_0^u = 0$ for all values of K_5 which is not the case for s_0^u as it depends on the value of K_5 used in K_q to partially encrypt p_0^u.
- Compute (p_j^u, s_i^u): for each i of the $2^{|K_3|}$ values of K_3, set $s_i^u = Enc_{[u,i,0]}(p_0^u)$ and for each j of the $2^{|K_4|}$ values of K_4, set $p_j^u = Dec_{[u,0,j]}(s_0^u)$.

To this end, we get $s_i^u = Enc_{[u,i,j]}(p_j^u)$ and an exhaustive search on K_5 is performed to construct the bicliques with time complexity of $2^{|K_5|}(2^{|K_3|} + 2^{|K_4|})$. Following [10], we include the biclique construction in the 2-step MitM and hence its time complexity is incorporated with the overall complexity of the MitM attack. Since we are using differential sieving, we need to build two base bicliques with two different base plaintext ${p_0}_0^u = 0$, and ${p_1}_0^u = 1$. In what follows we explain how we combine short restricted bicliques with differential sieving 2-step MitM attack as depicted in Fig. 4:

- For each guess k_s of the $2^{|K_s|}$ values, we get a guess u of the $2^{|K_5|}$ values of K_5
  - For each guess k_f of the $2^{|K_f|}$ values of K_f, we get a guess i of the $2^{|K_3|}$ values of K_3:
    1. Partially encrypt the two base plaintexts to get the corresponding intermediate states of the biclique ${s_0}_i^u = Enc_{[u,i,0]}({p_0}_0^u)$ and ${s_1}_i^u = Enc_{[u,i,0]}({p_1}_0^u)$. If $i = 0$, then save the base intermediate states ${s_0}_0^u$ and ${s_1}_0^u$

2. Partially encrypt the two intermediate states ${s_0}_i^u$ and ${s_1}_i^u$, and store the resulting m-bit partial states values (Xf_z^0, Xf_z^1) and difference $dx = (Xf_z^0 \oplus Xf_z^1)$ in a table T

– For each guess k_b of the $2^{|K_b|}$ values of K_b, we get a guess j of the $2^{|K_4|}$ values of K_4:
1. Partially decrypt the two base intermediate states to get the corresponding plaintexts of the biclique ${p_0}_j^u = Dec_{[u,0,j]}({s_0}_0^u)$ and ${p_1}_j^u = Dec_{[u,0,j]}({s_1}_0^u)$.
2. Ask the encryption oracle for the ciphertexts ${c_0}_j^u$ and ${c_1}_j^u$ corresponding to the plaintexts acquired in the previous step.
3. Partially decrypt the two ciphertexts ${c_0}_j^u$ and ${c_1}_j^u$ to get the two m-bit partial states values (Xb_{z+1}^0, Xb_{z+1}^1).
4. Perform the 2-step matching procedure described in the previous section.

– Exhaustively test if any of the candidate keys in L_p encrypts any two plaintexts ${p_0}_j^u$ and ${p_1}_j^u$ to their corresponding ciphertexts ${c_0}_j^u$ and ${c_1}_j^u$. If yes output it as the correct master key, else guess another k_s

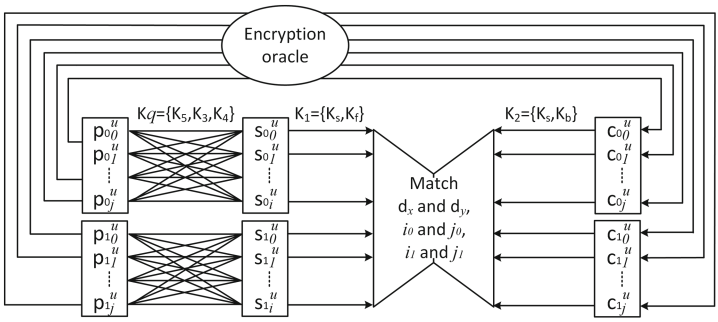

Fig. 4. Combining short restricted bicliques with differential sieving 2-step matching MitM attack

The total time complexity is composed of three parts: biclique construction, MitM, and key retesting. More formally, let c_q be the cost of biclique computation, the total time complexity of the attack is given by:

$$2 \times 2^{|K_5|}(2^{|K_3|} + 2^{|K_4|}) \cdot c_q + 2^{|K_s|}(2^{|K_f|} + 2^{|K_b|}) \cdot c_t + 2^{|K|-\alpha-2m} \cdot c_t.$$

The memory complexity is given by $2^{|K_f|} + 2^{|K_f|+|K_b|-\alpha-2m} + 2^{(\#\text{ of active bits in } p)+1}$ where the third term denotes the memory needed to store the chosen plaintext-ciphertext pairs for all the j and u values of K_3 and K_5 for two bicliques. Regardless of the value of K_3 and K_5, we only get a small number of plaintexts since they always differ in only few places. The data complexity is $2^{(\#\text{ of active bits in } p)+1}$ chosen plaintexts to get the corresponding ciphertexts of the plaintexts produced by the two bicliques.

3 Application to LBlock

In this section we demonstrate our technique on the lightweight block cipher LBlock [25]. LBlock requires about 1320 GE on 0.18 μm technology, which satisfies the required limitation of 2000 GE in RFID applications. LBlock has been analyzed with respect to various types of attacks [5,22] including: impossible differential [16,17], integral [21], boomerang [11], and biclique [23] cryptanalysis. Note that the attack presented in [23] is a typical high data complexity biclique cryptanalysis where the whole key space is exhaustively searched. More precisely, similar to the biclique attack on AES [6], in [23], the authors presented an attack with a time complexity $\approx 2^{79}$, almost equal to that of the average exhaustive search. The factor of 2 gain is due to counting the number of Sboxes which are affected by the difference only and not evaluating the whole state. In addition, the attack has a data complexity of 2^{52}.

3.1 Notation

Besides the notations used in describing our technique in Sect. 2, the notation used in applying our attack on LBlock is as follows:

- K: The master key.
- Sk_i: i^{th} round sub key.
- X_i: The eight 4-bit nibble state at round i.
- $X_i[j]$: j^{th} nibble of the i^{th} round state.
- $K_{[i,j]}$: i^{th} and j^{th} bits of master key K.

3.2 Specifications of LBlock

LBlock [25] is a 64-bit lightweight cipher with an 80-bit master key. It employs a 32-round Feistel structure variant. Its internal state is composed of eight 4-bit nibbles. As depicted in Fig. 5, the round function adopts three nibble oriented transformations: subkey mixing, 4-bit Sboxes, and nibble permutation. The 80-bit master key K is stored in a key register denoted by $k = k_{79}k_{78}k_{77}.....k_1k_0$. The leftmost 32 bits of the register k are used as i^{th} round subkey Sk_i. The key register is updated after the extraction of each Sk_i as follows:

1. $k \lll 29$.
2. $[k_{79}k_{78}k_{77}k_{76}] = S_9[k_{79}k_{78}k_{77}k_{76}]$.
3. $[k_{75}k_{74}k_{73}k_{72}] = S_8[k_{75}k_{74}k_{73}k_{72}]$.
4. $[k_{50}k_{49}k_{48}k_{47}k_{46}] \oplus [i]_2$,

where S_8 and S_9 are two 4-bit Sboxes. For further details, the reader is referred to [25].

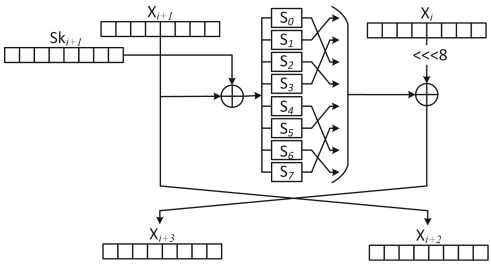

Fig. 5. LBock round function

3.3 Low Data-Complexity Attack on LBlock

In this section, we present the first low data complexity attack on reduced round LBlock. The attack exploits the weak diffusion of the key schedule. This fact enables us to find some nibbles in states at higher rounds whose values do not involve all the bits from the master key. Using symbolic evaluation of the cipher, we have found an execution separation for the first fifteen rounds of LBlock. The knowledge of the first nibble of the 8^{th} round state requires guessing 73 and 72 bits of the master key when evaluating it from the forward and backward executions, respectively. More precisely, given a known plaintext-cipher text pair, the forward execution requires guessing $K - K_{[0,1,14,15,16,17,35]}$ to compute $Xf_8[0]$ and the backward execution requires guessing $K - K_{[55,62,63,64,65,66,67,68]}$ to compute $Xb_8[0]$. For this separation, we can use a basic MitM attack since we have the knowledge of one nibble from both directions at the same round. To recover the right key, we have to try $\lceil \frac{80}{64} \rceil = 2$ plaintext-ciphertext pairs. Accordingly, the total matching probability $2^{-4\times 2}$. Following the notation used in Sect. 2, $|K_s| = 65$, $|K_f| = 8$, and $|K_b| = 7$. Hence, the time complexity of this 2 known plaintext attack is given by $2^{65}(2^8 + 2^7) + 2^{80-8} = 2^{73}$ and a memory complexity of 2^8.

While the previous attack was an application of the basic MitM attack, Fig. 6 shows an execution separation for fifteen rounds of Lblock starting from round four. We opted for demonstrating the 2-step matching approach on these specific rounds because in the following section, we will add a short restricted biclique in the first three rounds. Our attack requires the knowledge of nibbles $Xf_{11}[1]$ and $Xf_{10}[6]$ in the forward computation and nibble $Xb_{12}[0]$ in the backward computation. The adopted execution separation is depicted in Fig. 6, where the red and blue colours denote known state nibbles in consecutive rounds in the forward and backward executions, respectively. The yellow colour denotes key nibbles that are guessed in each direction and the white colour represents unknown nibbles. We have found that the knowledge of $Xf_{11}[1]$ and $Xf_{10}[6]$ involves 76 bits from the master key, namely $K_1 = K - K_{[2,1,0,79]}$. In the backward direction, $Xb_{12}[0]$ requires guessing 73 bits, specifically $K_2 = K - K_{[26,27,27,29,30,31,32]}$. Accordingly, $|K_s| = 69$, $|K_f| = 7$, and $|K_b| = 4$.

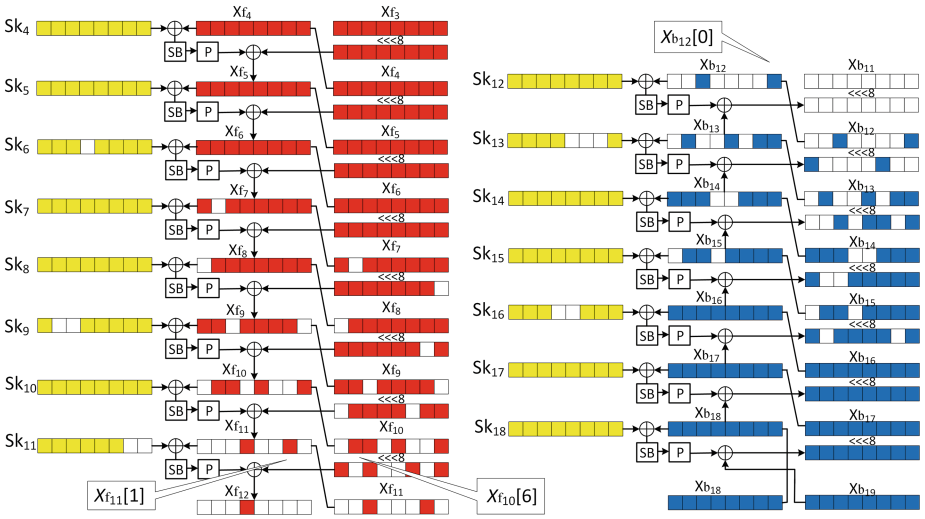

Fig. 6. Fifteen rounds execution separation for our 2-step matching MitM attack (Color figure online)

We assume that states Xf_3 and Xf_4 are loaded with the known plaintexts, while states Xb_{18} and Xb_{19} are loaded with their corresponding ciphertexts. The attack follows exactly the procedure defined in Sect. 2.2. However, due to the fact that LBlock employs a Feistel structure, the matching process involves two different states from the forward direction. Particularly, for the two known plaintexts p_0 and p_1, we store in table T nibbles $(Xf^0_{11}[1], Xf^1_{11}[1])$ and $(Xf^0_{10}[6], Xf^1_{10}[6])$ along with the guessed K_1. For each guess of K_2 in the backward execution using the two corresponding ciphertexts c_0 and c_1, we compute $Xb^0_{12}[0]$ and $Xb^1_{12}[0]$, and we apply the 2-step matching as follows:

1. Set $dx = Xf^0_{11}[1] \oplus Xf^1_{11}[1]$ and $dy = P^{-1}((Xb^0_{12}[0] \oplus Xb^1_{12}[0]) \oplus ((Xf^0_{10}[6] \oplus Xf^1_{10}[6])))$, where P^{-1} is the inverse permutation of the round function. Check if (dx, dy) is a possible differential pair from the differential distribution table (DDT) of Sbox S_1.
2. From the current K_b knowledge, one can get a candidate value for $Sk_{11}[1]$ which is unknown in the forward direction and consequently a candidate $(Xf^0_{12}[0], Xf^1_{12}[0])$. Match these two candidate nibbles with $Xb^0_{12}[0]$ and $Xb^1_{12}[0]$.

If a (K_1, K_2) pair passed the 2-step matching then it is considered for retesting. The probability of possible differentials of the Sbox S_1 is $\approx 2^{-1.4}$, and we match two different nibbles, hence the total matching probability is $2^{-1.4-4-4}$. This two known plaintext-ciphertext attack has a time complexity of $2^{69}(2^7 + 2^4) + 2^{80-9.4} \approx 2^{76}$, and a memory complexity of 2^7.

3.4 Three More Rounds with Restricted Bicliques

The previous fifteen round attack is extended to eighteen rounds by appending three round restricted biclique. As depicted in Fig. 7, if $K_3 = K_{[27,28,29,30]}$ and $K_4 = K_{[0,1,79]}$, the differential paths in the first three rounds are independent. More precisely, a change in K_3 affects $Sk_2[2]$ which consecutively propagates the effect to the red nibbles in the forward direction. In the backward direction, changing K_4 involves $Sk_1[7]$ and $Sk_3[2]$, their effect is denoted by the blue nibbles in the states. Since $K_3 \neq K_f$ and $K_4 \neq K_b$, we have to consider bits $K_{[2,26,31,32]}$ as shared bits, i.e., $K_s = K_s + K_{[2,26,31,32]}$ and thus $|K_5| = 73$. The attack follows the procedure presented in Sect. 2.3, and the 2-step matching is performed exactly as in the fifteen round attack. Since $c_q \approx c_t/4$, following the time complexity equation in Sect. 2.3, this chosen plaintext attack has a time complexity of $2^{73-1}(2^4+2^3)+2^{73}(2^4+2^3)+2^{80-9.4} \approx 2^{78}$. The memory and data complexities are determined by the number of active bits in the plaintexts of the bicliques. Since we have four active nibbles in the plaintexts of both bicliques, this attack has a memory and data complexities of 2^{17}.

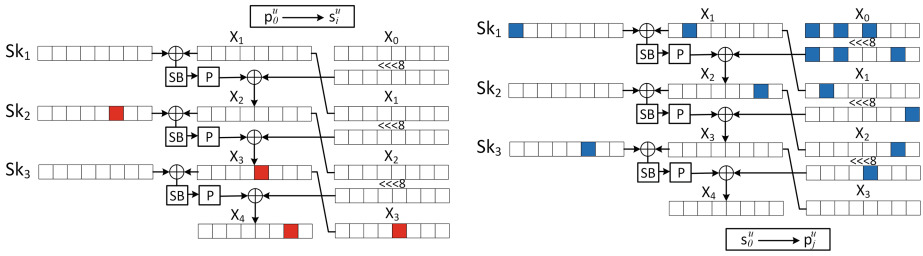

Fig. 7. First three rounds restricted biclique (Color figure online)

4 Conclusion

We have presented a modified approach to decrease the total matching probability of the MitM attack. This approach extends the attack by one more round and incorporates differential and value matching to reduce the number of false positive keys. We have shown how to combine our approach with short restricted bicliques and demonstrated the approach on the light weight block cipher LBlock. Particularly, we have presented a two known plaintexts 2-step matching MitM attack on fifteen rounds with time complexity of 2^{76} and memory complexity of 2^7. Finally, we have combined the fifteen rounds execution with 3-round restricted bicliques and an eighteen round chosen plaintext attack is presented with time, memory, and data complexities of 2^{78}, 2^{17}, and 2^{17}, respectively.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped improve the quality of the paper.

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Le Fonds de Recherche du Québec - Nature et Technologies (FRQNT).

References

1. AlTawy, R., Youssef, A.M.: Preimage attacks on reduced-round stribog. In: Pointcheval, D., Vergnaud, D. (eds.) AFRICACRYPT. LNCS, vol. 8469, pp. 109–125. Springer, Heidelberg (2014)
2. Aoki, K., Sasaki, Y.: Meet-in-the-middle preimage attacks against reduced SHA-0 and SHA-1. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 70–89. Springer, Heidelberg (2009)
3. Aoki, K., Sasaki, Y.: Preimage attacks on one-block MD4, 63-step MD5 and more. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 103–119. Springer, Heidelberg (2009)
4. Bar-On, A., Dinur, I., Dunkelman, O., Lallemand, V., Tsaban, B.: Improved analysis of zorro-like ciphers. Cryptology ePrint Archive, Report 2014/228 (2014)
5. Bogdanov, A., Boura, C., Rijmen, V., Wang, M., Wen, L., Zhao, J.: Key difference invariant bias in block ciphers. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 357–376. Springer, Heidelberg (2013)
6. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique cryptanalysis of the full AES. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 344–371. Springer, Heidelberg (2011)
7. Bogdanov, A.A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
8. Bogdanov, A., Rechberger, C.: A 3-subset meet-in-the-middle attack: cryptanalysis of the lightweight block cipher KTANTAN. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 229–240. Springer, Heidelberg (2011)
9. Bouillaguet, C., Derbez, P., Dunkelman, O., Fouque, P.-A., Keller, N., Rijmen, V.: Low-data complexity attacks on AES. IEEE Trans. Inf. Theory **58**(11), 7002–7017 (2012)
10. Canteaut, A., Naya-Plasencia, M., Vayssière, B.: Sieve-in-the-middle: improved MITM attacks. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 222–240. Springer, Heidelberg (2013)
11. Chen, J., Miyaji, A.: Differential cryptanalysis and boomerang cryptanalysis of LBlock. In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds.) CD-ARES Workshops 2013. LNCS, vol. 8128, pp. 1–15. Springer, Heidelberg (2013)
12. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
13. Diffie, W., Hellman, M.: Exhaustive cryptanalysis of the NBS data encryption standard. Computer **10**(6), 74–84 (1977)
14. Gérard, B., Grosso, V., Naya-Plasencia, M., Standaert, F.-X.: Block ciphers that are easier to mask: how far can we go? In: Bertoni, G., Coron, J.-S. (eds.) CHES 2013. LNCS, vol. 8086, pp. 383–399. Springer, Heidelberg (2013)
15. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED block cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)

16. Karakoç, F., Demirci, H., Harmancı, A.E.: Impossible differential cryptanalysis of reduced-round LBlock. In: Askoxylakis, I., Pöhls, H.C., Posegga, J. (eds.) WISTP 2012. LNCS, vol. 7322, pp. 179–188. Springer, Heidelberg (2012)
17. Liu, Y., Gu, D., Liu, Z., Li, W.: Impossible differential attacks on reduced-round LBlock. In: Ryan, M.D., Smyth, B., Wang, G. (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 97–108. Springer, Heidelberg (2012)
18. Rijmen, V., Toz, D., Mendel, F., Varici, K.: Differential analysis of the LED block cipher. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 190–207. Springer, Heidelberg (2012)
19. Nakahara Jr., J., Sepehrdad, P., Zhang, B., Wang, M.: Linear (Hull) and algebraic cryptanalysis of the block cipher PRESENT. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 58–75. Springer, Heidelberg (2009)
20. Sasaki, Y.: Meet-in-the-middle preimage attacks on AES hashing modes and an application to Whirlpool. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 378–396. Springer, Heidelberg (2011)
21. Sasaki, Y., Wang, L.: Meet-in-the-middle technique for integral attacks against feistel ciphers. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 234–251. Springer, Heidelberg (2013)
22. Wang, Y., Wu, W.: Improved multidimensional zero-correlation linear cryptanalysis and applications to LBlock and TWINE. In: Susilo, W., Mu, Y. (eds.) ACISP 2014. LNCS, vol. 8544, pp. 1–16. Springer, Heidelberg (2014)
23. Wang, Y., Wu, W., Yu, X., Zhang, L.: Security on LBlock against biclique cryptanalysis. In: Lee, D.H., Yung, M. (eds.) WISA 2012. LNCS, vol. 7690, pp. 1–14. Springer, Heidelberg (2012)
24. Wu, S., Feng, D., Wu, W., Guo, J., Dong, L., Zou, J.: (Pseudo) preimage attack on round-reduced Grøstl hash function and others. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 127–145. Springer, Heidelberg (2012)
25. Wu, W., Zhang, L.: LBlock: a lightweight block cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)