SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2016; **9**:27–33
Published online 6 October 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1375

RESEARCH ARTICLE

Biclique cryptanalysis of MIBS-80 and PRESENT-80 block ciphers

Mohammad Hossein Faghihi Sereshgi[1] *, Mohammad Dakhilalian[1] and Mohsen Shakiba[2]

[1] Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran
[2] Department of Electrical and Computer Engineering, Jundi-Shapur University of Technology, Dezful, Iran

ABSTRACT

PRESENT and MIBS are two lightweight block ciphers that are suitable for low resource devices such as radio-frequency identification tags. In this paper, we present the first biclique cryptanalysis of MIBS block cipher and a new biclique cryptanalysis of PRESENT block cipher. These attacks are performed on full-round MIBS-80 and full-round PRESENT-80. Using matching without matrix technique in the attack on MIBS and choosing a sub-key space of an internal round for key division eventuate to reduce the security of this cipher by 1 bit, while the data complexity of attack is 2^{52} chosen plaintext. The attack on PRESENT-80 has a data complexity of at most 2^{22} chosen plaintext and computational complexity of $2^{79.34}$ encryption that both complexities are lower than of other cryptanalyses of full-round PRESENT-80 so far. Also, in this paper, we use early abort technique to efficiently filter out wrong keys in matching phase of biclique attack of PRESENT-80. Copyright © 2015 John Wiley & Sons, Ltd.

KEYWORDS
cryptography; block cipher; biclique cryptanalysis; MIBS cipher; PRESENT cipher

***Correspondence**
Mohammad Hossein Faghihi Sereshgi, Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran.
E-mail: mh.faghihi@ec.iut.ac.ir

1. INTRODUCTION

Along advances in low resource applications such as radio-frequency identification tags and Internet of things, lightweight cryptography became a popular field of study to find appropriate solutions to different purposes of security in low-resource devices. So far, many block ciphers like MIBS [1], LBlock [2] and PRESENT [3] are introduced to satisfy conditions of constrained applications.

In ASIACRYPT 2011, biclique cryptanalysis of block ciphers, which is a kind of meet in the middle cryptanalysis, was introduced by Bogdanov *et al.* [4]. After that, many studies have been performed on security of different lightweight ciphers against biclique attack [5], [6], [7], and [8].

Structures of MIBS and PRESENT are standard Feistel network and substitution–permutation network, respectively. The key schedule of MIBS, with some minor differences, is adopted from key schedule of PRESENT. In this paper, we consider 80-bit key length versions of MIBS (MIBS-80) and PRESENT (PRESENT-80).

In this paper, which is a revised version of [9], we introduce two biclique cryptanalyses on these block ciphers. These attacks are conducted on full-round versions of MIBS-80 and PRESENT-80 ciphers and are in single key mode.

The biclique cryptanalysis of MIBS, which is compounded to the *matching without matrix* technique for the first time, is the first full-round cryptanalysis of this cipher. In the initial phase of this attack and attack on PRESENT-80, which is partitioning the key space by choosing appropriate bits, we tried to choose those bits that, first, differences caused by these bits activate less S-boxes in the matching phase, second, do not lead the attack to need the entire code book for data complexity. This features alongside *early abort* technique in addition to considering the bit-wise feature of the PRESENT's permutation layer result in better data complexity and computational complexity for the attack on PRESENT-80 comparing with other biclique attacks that have been introduced on full-round version of this cipher so far. Results of published attacks on MIBS and published attacks on PRESENT, those which cover more than 25 rounds of PRESENT, are represented in Table I.

The rest of this paper is organized as follows. Section 2 provides a description of MIBS-80 cipher. Section 3 shows

Table I. Results of cryptanalysis of MIBS and PRESENT ciphers.

Number of rounds	Data	Time	Cipher	Attack type	PoS	References
12	2^{59} CP	2^{58}	MIBS-80	IDC	100%	[10]
13	2^{61} CP	2^{56}	MIBS-80	DC	99.9%	[10]
14	2^{40} CP	2^{40}	MIBS-80	DC	50.15%	[10]
17	2^{58} KP	2^{69}	MIBS-80	LC	99.9%	[10]
18	$2^{62.47}$ KP	$2^{78.62}$	MIBS-80	LC	99.9%	[10]
19	$2^{57.87}$ CP	$2^{78.22}$	MIBS-80	MLC	99.9%	[11]
19	$2^{57.87}$ CP	$2^{74.23}$	MIBS-80	MLC	99.9%	[11]
Full (32)	2^{52} CP	$2^{78.98}$	MIBS-80	BC	100%	This
25	$2^{62.4}$ CP	2^{65}	PRESENT-80	MLC	95%	[12]
26	2^{66} CP	2^{72}	PRESENT-80	MLC	95%	[12]
26	$2^{63.16}$ CP	2^{76}	PRESENT-80	TDC	50%	[13]
26	$2^{62.08}$ KP	2^{76}	PRESENT-80	TDC	50%	[13]
27	2^{62} CP	2^{74}	PRESENT-80	FFT-MLC	95%	[14]
Full (31)	2^{23} CP	$2^{79.76}$	PRESENT-80	BC	100%	[15]
Full (31)	2^{22} CP	$2^{79.34}$	PRESENT-80	BC	100%	This

In case of reduced-round attacks, time complexity is the number of reduced-round encryptions. BC, Biclique cryptanalysis; CP, chosen plaintext; DC, differential cryptanalysis; FFT, fast Fourier transform; IDC, imposible differential cryptanalysis; KP, known plaintext; MLC, multidimensional linear cryptanalysis; PoS, probability of success; TDC, truncated differential cryptanalysis.

the key recovery attack on full-round MIBS-80. A brief description of PRESENT-80 is provided in Section 4. Section 5 shows the key recovery attack on full-round PRESENT-80. A conclusion of this paper is given in Section 6.

2. DESCRIPTION OF MIBS

Block cipher MIBS uses a standard Feistel structure with 64-bit block length and supports user key of lengths 64 and 80 bits. In this paper, we consider the 80-bit key version of this cipher (MIBS-80). Each round function of MIBS consists of a key addition layer, a nibble-wise S-box layer $S : GF(2^4) \rightarrow GF(2^4)$, and a mixing layer that can be represented by a simple matrix production $M : (GF(2^4)^8) \rightarrow (GF(2^4)^8)$. So, if the $state^0$ is initialized by the 80-bit user key as $state^0 = k_{79}k_{78}\ldots k_0$, then 32 round keys k^i, $0 \leq i \leq 31$ are generated as follows:

$$state^i = state^i >>> 19$$

$$state^i = Sbox(state^i_{[79:76]}) || Sbox(state^i_{[75:72]}) || state^i_{[71:0]}$$

$$state^i = state^i_{[79:19]} || \left(state^i_{[18:14]} \oplus RC\right) || state^i_{[13:0]}$$

$$k^i = state^i_{[79:48]};\ state^{i+1} = state^i$$

which RC is a round-counter. As it can be seen, round-key k^i is the 32 leftmost bits of $state^i$.

3. KEY RECOVERY FOR FULL-ROUND MIBS-80

According to the key schedule of MIBS-80, the "user key" can be computed from each $state^i$, $0 \leq i \leq 31$. For mounting the biclique attack on MIBS-80, vector space of $state^{28}$ is divided into 2^{72} subsets; each of them includes 2^8, 80-bit values of $state^{28}$, in the step that the round-key of round 28 is 32 leftmost bits of $state^{28}$. For this purpose, in each subset only bits in positions $[44, 43, 42, 41, 11, 10, 9, 8]$ are varied, and the rest remain unchanged, while we have considered bit positions of $state^{28}$ in the order [79,78,...,1,0]. An independent biclique of dimension 4 is placed in the five final rounds of the cipher. For this purpose, we consider two related-key differentials, the first one in the encryption path caused by 16 possible differences of 4 bits $[44, 43, 42, 41]$ of $state^{28}$, each of them represented by $\Delta K[i]$, $0 \leq i \leq 15$. The second related-key differential in the decryption path is also generated by 16 possible differences of 4 bits $[11, 10, 9, 8]$ of $state^{28}$, each of them represented by $\nabla K[j]$, $0 \leq j \leq 15$. As it can be seen in Figure 1, these two related-key differentials share no active S-boxes, so they can be used to make an independent biclique, which covers 2^8 possible differences $\Delta K[i,j] = \Delta K[i] \oplus \nabla K[j]$, as it was expected.

3.1. Matching without matrix

Matching without matrix technique presented in [16] could reduce the computational complexity of the matching nibbles recomputation. According to the Feistel structure of MIBS cipher, we have $L_{i+1} = M(S(L_i \oplus K_i)) \oplus R_i$, which M and S represent the operations of mix and substitution layers, respectively. Also, it could be easily derived that $L_{i+1} = M(S(R_{i+3} \oplus K_{i+2})) \oplus L_{i+3}$. So we have the equality $M(S(L_i \oplus K_i)) \oplus R_i = M(S(R_{i+3} \oplus K_{i+2})) \oplus L_{i+3}$ that leads to the equality $S(L_i \oplus K_i) \oplus M^{-1}(R_i) = S(R_{i+3} \oplus K_{i+2}) \oplus M^{-1}(L_{i+3})$.

In our attack, we choose the L_{14} (left part of data register in round 15) as matching point. So we can choose the data corresponding to the fourth and the fifth S-boxes in

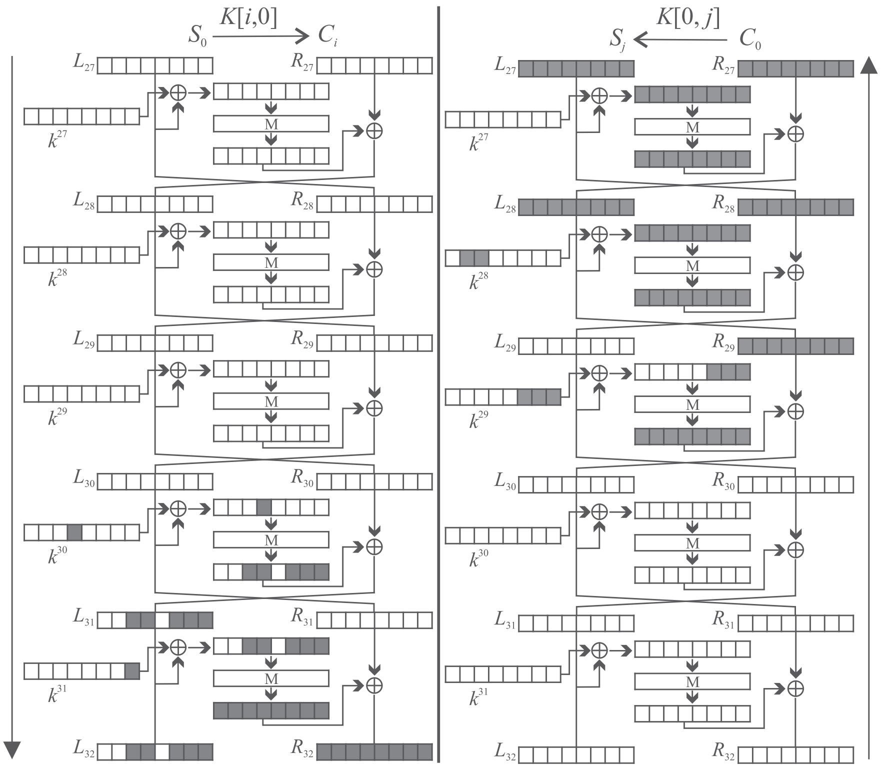

Figure 1. Four-dimension biclique for MIBS over rounds 28–32.

round 14 ($\overrightarrow{v}$) and their corresponding S-boxes in round 16, which means the fourth and fifth S-boxes in round 16 ($\overleftarrow{v}$) as matching variables.

3.2. The attack procedure

Step 1: *Constructing biclique*. Set bit numbers [44, 43, 42, 41, 11, 10, 9, 8] of $state^{28}$ to 0 and choose a new value for the other 72 bits, assume it as $K[0,0]$ and construct a biclique as follows:

- Choose $C_0 = 0_{(64)}$ as ciphertext and decrypt it to round 28 with $K[0,0]$ to obtain S_0, the data register in round 28 (i.e., L_{27} and R_{27}).
- Decrypt C_0 with keys $K[0,j] = \Delta K[0,j] \oplus K[0,0]$ and $1 \le j \le 15$, to obtain S_j (in the input of round 28).
- Encrypt S_0 with keys $K[i,0] = \Delta K[i,0] \oplus K[0,0]$ and $1 \le i \le 15$ to obtain C_i. Also, get their corresponding plaintexts P_i, $0 \le i \le 15$.

Step 2: *Matching*. For a fixed i, Encrypt P_i with $K[i,0]$ to round 14 to obtain the matching variable $\overrightarrow{v_{i,0}}$ and store the process $P_i \overset{K[i,0]}{\longrightarrow} \overrightarrow{v_{i,0}}$. Then, encrypt P_i with $K[i,j] = \Delta K[i,j] \oplus K[0,0]$, $1 \le j \le 15$ to obtain $\overrightarrow{v_{i,j}}$. Only recompute those parts of the processes that differ from the stored process. For a fixed j, Decrypt S_j with $K[0,j]$ to round 16 to obtain the matching variable $\overleftarrow{v_{0,j}}$ and store the process $\overleftarrow{v_{0,j}} \overset{K[0,j]}{\longleftarrow} S_j$. Then, decrypt S_j with $K[i,j]$, $1 \le i \le 15$ to obtain $\overleftarrow{v_{i,j}}$ and only recompute those parts that differ from $\overleftarrow{v_{0,j}} \overset{K[0,j]}{\longleftarrow} S_j$. The procedure of matching is shown in Figure 2. A candidate key $K[i,j]$ leads to $\overrightarrow{v_{i,j}} = \overleftarrow{v_{i,j}}$. Because matching variables are 8 bits, we anticipate $2^{8-8} = 1$ candidate key for each key set. Exhaustive search is needed to filter out wrong candidate keys.

In the left (or right) side of Figure 2, light gray S-boxes are those that are required to be computed once per P_i (or S_j), and dark gray S-boxes are those that are required to be computed 2^4 times per P_i (or S_j). There is no need to compute white S-boxes.

Data complexity. According to Figures 1, 3 nibbles of ciphertext are not affected by key differences. So the data complexity will not exceed $2^{64-12} = 2^{52}$ chosen plaintexts.

Computational complexity. In step 1, $2^4 \times (19+6)+15 = 415$ S-boxes; in step 2, $2^4 \times (34 + 2^4 \times (71)) = 18720$ S-boxes (in encryption path) plus $2^4 \times (9 + 2^4 \times (80)) = 20624$ (in decryption path) S-boxes are computed. In key schedule, $2^4 \times (3 + 9) + 52 = 244$ S-boxes are computed. Also, there is one candidate key per key set in average. The MIBS-80 uses 320 S-boxes in a full-round encryption, so the computational complexity is as follows:

$$C_{full} = 2^{72}\left(\frac{39344 + 415 + 224}{320} + 1\right) = 2^{78.98}$$

Because we used all keys in this attack, its success probability is 100%.

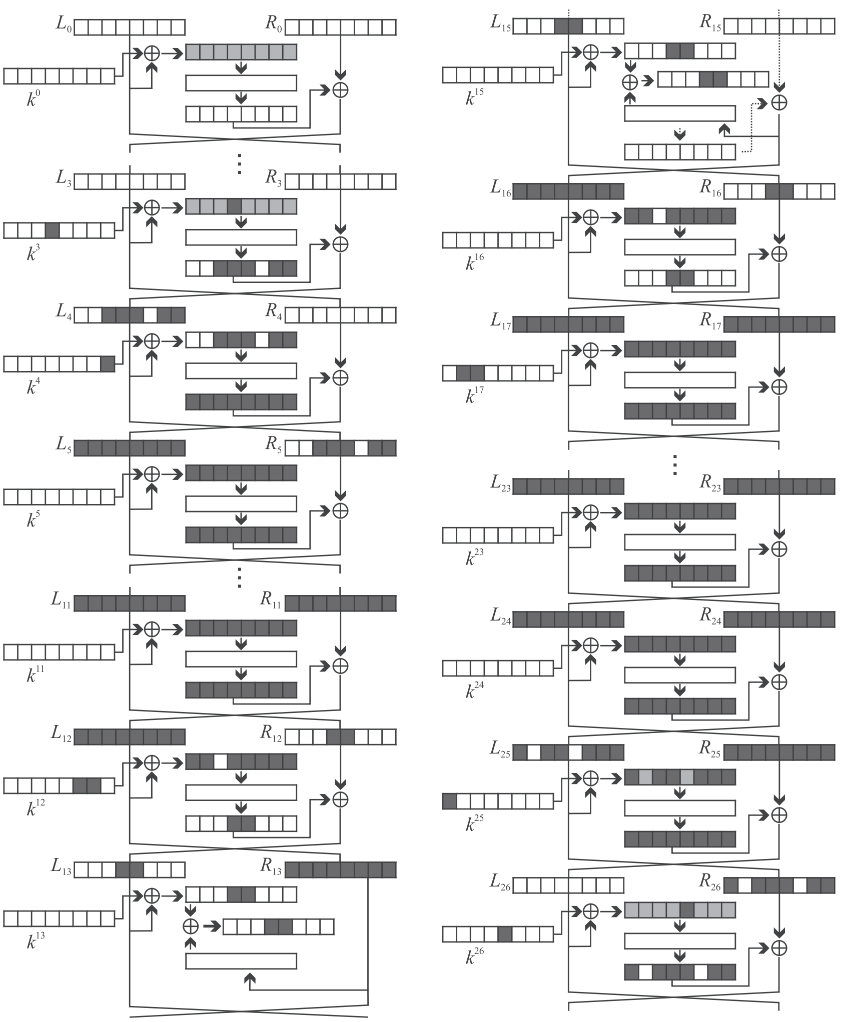

Figure 2. Matching over 27 rounds. Left: forward computation and right: backward computation.

4. DESCRIPTION OF PRESENT-80

PRESENT uses a substitution–permutation network structure with 64-bit block length and supports user keys of lengths 80 and 128 bits. Each round function of PRESENT consists of a key addition layer, a nibble-wise S-box layer $S : GF(2^4) \to GF(2^4)$, and a simple bit-wise permutation layer. In this paper, we consider 80-bit key length version of PRESENT (PRESENT-80). So if $state^0$ is initialized by the 80-bit user key as $state^0 = k_{79}k_{78}\dots k_0$, then 31 round keys and the post whitening key k^i, $0 \leq i \leq 31$ are generated as follows:

$$k^i = state^i_{[79:16]}$$
$$state^i = state^i >>> 19$$
$$state^i = Sbox(state^i_{[79:76]})||state^i_{[75:0]}$$
$$state^i = state^i_{[79:20]}|| \left(state^i_{[19:15]} \oplus RC\right) ||state^i_{[14:0]}$$
$$state^{i+1} = state^i$$

The round-key k^i is the 64 leftmost bits of $state^i$.

5. KEY RECOVERY FOR PRESENT-80

In the case of PRESENT, the vector space of $state^{28}$, in the step that the round-key of round 29 is 64 leftmost bits of $state^{28}$, is divided into 2^{72} subsets; each of them includes 2^8 value of $state^{28}$. To construct an independent biclique of dimension 4, we considered the 4 bits $[8, 7, 6, 4]$ of $state^{28}$ to cause 16 differences $\Delta K[i], 0 \leq i \leq 15$, and 4 bits $[40, 39, 38, 31]$ of $state^{28}$ to cause 16 differences $\nabla K[j], 0 \leq j \leq 15$, which according to Figure 3 share no active S-boxes in differential trails in the last three rounds.

5.1. The attack procedure

Step 1: Similar to step 1 of part 3.1 and by assuming $C_0 = 0_{(64)}$ and internal state S in round 29, we construct S_j, $0 \leq$

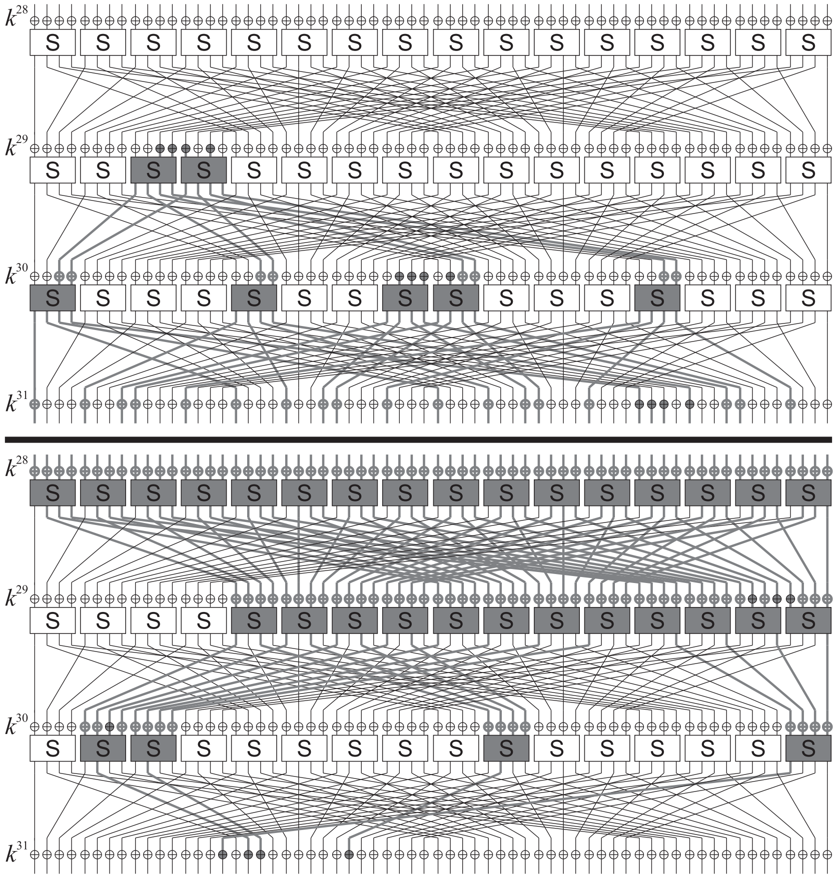
k^{28}
S S S S S S S S S S S S S S S S
k^{29}
S S S S S S S S S S S S S S S S
k^{30}
S S S S S S S S S S S S S S S S
k^{31}
k^{28}
S S S S S S S S S S S S S S S S
k^{29}
S S S S S S S S S S S S S S S S
k^{30}
S S S S S S S S S S S S S S S S
k^{31}

Figure 3. Four-dimension biclique for PRESENT-80 over rounds 29–31.

$j \le 15$ and C_i, $1 \le i \le 15$ and their corresponding P_i, $0 \le i \le 15$.

Step 2: *Matching*. In this step, for a fixed i, compute the value of $\overrightarrow{v_{i,j}}$s that are 12 bits in positions [63,62, 61,59,58,57,55,54,53,51,50,49] of input of round 16 (R_{16}) by encrypting P_i with $K[i,j]$, $0 \le j \le 15$. Also, for a fixed j, compute the values of $\overleftarrow{v_{i,j}}$s, which are the same bits of input of round 16, by decrypting S_j using $K[i,j]$, $0 \le i \le 15$. Figure 4 shows the procedure of matching.

In the upper (or bottom) part of Figure 4, light gray S-boxes are those that are required to be computed once per P_i (or S_j), and dark gray S-boxes are those that are required to be computed 2^4 times per P_i (or S_j). There is no need to compute white S-boxes.

The recomputations that are needed to filter out wrong keys can be decreased by using the *early abort* technique [7,17,18]. In this technique, we fist recompute those parts of cipher that are needed to obtain only a part of matching variable. If these computations lead to different values for partial $\overrightarrow{v_{i,j}}$ and corresponding part of $\overleftarrow{v_{i,j}}$, then the key is wrong. But if these values were equal, we compute another part of $\overrightarrow{v_{i,j}}$ and $\overleftarrow{v_{i,j}}$ and so on. We organize this procedure (selecting the cumulative parts of matching variable) to minimize recomputations' time complexity. Those keys that lead computations to equal $\overrightarrow{v_{i,j}}$ and $\overleftarrow{v_{i,j}}$ will be considered as candidate keys. Exhaustive search is needed to eliminate wrong candidate keys.

In using *early abort* technique for this attack on PRESENT-80, all S-box recomputations for all parts of matching variables will be the same as before except for some S-boxes in rounds 14, 15, and 16. In case of this attack, we consider the matching variable consisting of 12 parts that are its bits. Then, we recompute the value of the bit in position 63 of input of R_{16}. It is anticipated that this bit filters out half of the keys in average in every key set. For other half of the keys, we go for the next part of matching variable, which in this case, we consider it the bit in position 59. By this choice, only recomputations of two S-boxes will be added to the previous computations. It is anticipated that this bit filters out half of the remaining keys. To filter out the remaining keys, we go for the next bits in this order of positions 62,58,61,57,..., and it is anticipated that in each step, half of the keys will be eliminated. Because the recomputations of additional S-boxes will be negligible after some steps, the order of bit positions after these steps will not be important. Therefore, only the first six steps of recomputations of matching variable will be considered in computational complexity.

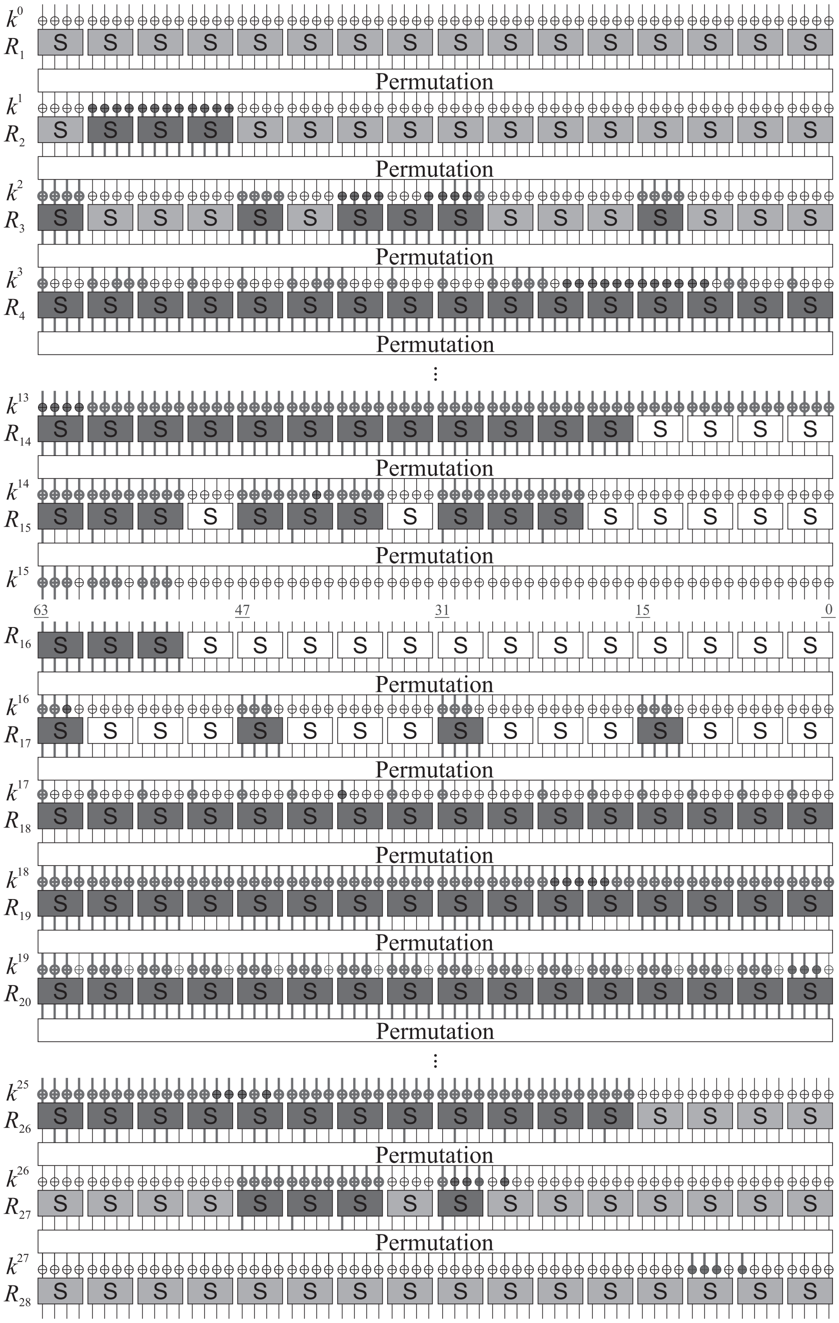

Figure 4. Matching over 28 rounds. Up: forward computation and down: backward computation.

Because matching variable consists of 12 bits, we anticipate $2^{8-12} = 0.0625$ candidate key for each key set. Exhaustive search is needed to filter out wrong candidate keys.

Data complexity. According to Figure 3, 42 bits of ciphertext are not affected by differences between keys in each key set. So the data complexity will not exceed $2^{64-42} = 2^{22}$ chosen plaintexts.

Computational complexity. In step 1 , $2^4 \times (32 + 7) + 9 = 633$ S-boxes; in step 2, using *early abort* technique, $2^4 \times \left(39 + 2^4 \times \left(175 + \frac{2}{2} + \frac{5}{4} + \frac{1}{8} + \frac{5}{16} + \frac{1}{32}\right)\right) = 46104$ S-boxes (in encryption path) plus $2^4 \times (32 + 2^4 \times (148)) = 38400$ S-boxes (in decryption path) are computed. $2^4 \times (3 + 4) + 25 = 137$ S-boxes are computed in key schedule. Also, there is 0.0625 candidate key per key set in average. The PRESENT-80 uses 527 S-boxes in a full-round encryption, so the computational complexity is as follows:

$$C_{full} = 2^{72}\left(\frac{84504 + 633 + 137}{527} + 0.0625\right) = 2^{79.34}$$

6. CONCLUSION

In this paper, we presented the first independent biclique attack on full-round MIBS-80 and a new independent biclique attack on PRESENT-80. The attack on MIBS-80 uses *matching without matrix* technique with biclique cryptanalysis for the first time and results in reduction of the cipher security about 1 bit. The attack on PRESENT-80 results in data and computational complexities better than other introduced attacks on full-round PRESENT-80 so far mainly due to considering the bit-wise feature of the cipher's permutation layer, and choosing bits that activate less S-boxes in beginning rounds of matching phase of the attack. Also, we showed how we can use *early abort* technique efficiently to decrease the computational complexity of biclique attack on PRESENT-80.

REFERENCES

1. Izadi M, Sadeghiyan B, Sadeghian S, Khanooki H. MIBS: a new lightweight block cipher. In *Cryptology and Network Security, Lecture Notes in Computer Science*, Vol. 5888, Garay J, Miyaji A, Otsuka A (eds). Springer: Berlin Heidelberg, 2009; 334–348.
2. Wu W, Zhang L. Lblock: a lightweight block cipher. In *Applied Cryptography and Network Security, Lecture Notes in Computer Science*, Vol. 6715, Lopez J, Tsudik G (eds). Springer: Berlin Heidelberg, 2011; 327–344.
3. Bogdanov A, Knudsen L, Leander G, Paar C, Poschmann A, Robshaw M, Seurin Y, Vikkelsoe C. PRESENT: an ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems—CHES 2007, Lecture Notes in Computer Science*, Vol. 4727, Paillier P, Verbauwhede I (eds). Springer: Berlin Heidelberg, 2007; 450–466.
4. Bogdanov A, Khovratovich D, Rechberger C. Biclique cryptanalysis of the full AES. In *Advances in Cryptology ASIACRYPT 2011, Lecture Notes in Computer Science*, Vol. 7073, Lee D, Wang X (eds). Springer: Berlin Heidelberg, 2011; 344–371.
5. Hong D, Koo B, Kwon D. Biclique attack on the full hight. In *Information Security and Cryptology—ICISC 2011, Lecture Notes in Computer Science*, Vol. 7259, Kim H (ed). Springer: Berlin Heidelberg, 2012; 365–374.
6. Ahmadi S, Ahmadian Z, Mohajeri J, Aref M. Low-data complexity biclique cryptanalysis of block ciphers with application to piccolo and hight. *IEEE Transactions on Information Forensics and Security* 2014; **9**(10): 1641–1652.
7. Shakiba M, Dakhilalian M, Mala H. Cryptanalysis of mcrypton-64. *International Journal of Communication Systems* 2015; **28**(8): 1401–1418.
8. Wang Y, Wu W, Yu X, Zhang L. Security on Lblock against biclique cryptanalysis. In *Information Security Applications, Lecture Notes in Computer Science*, Vol. 7690, Lee D, Yung M (eds). Springer: Berlin Heidelberg, 2012; 1–14.
9. Faghihi Sereshgi MH, Dakhilalian M, Shakiba M. Biclique cryptanalysis of MIBS-80 and PRESENT-80. Cryptology ePrint Archive, Report 2015/393 2015.
10. Bay A, Nakahara JJ, Vaudenay S. Cryptanalysis of reduced-round MIBS block cipher. In *Cryptology and Network Security, Lecture Notes in Computer Science*, Vol. 6467, Heng SH, Wright R, Goi BM (eds). Springer: Berlin Heidelberg, 2010; 1–19.
11. Bay A, Huang J, Vaudenay S. Improved linear cryptanalysis of reduced-round MIBS. In *Advances in Information and Computer Security, Lecture Notes in Computer Science*, Vol. 8639, Yoshida M, Mouri K (eds). Springer: International Publishing, 2014; 204–220.
12. Cho J. Linear cryptanalysis of reduced-round present. In *Topics in Cryptology—CT-RSA 2010, Lecture Notes in Computer Science*, Vol. 5985, Pieprzyk J (ed). Springer: Berlin Heidelberg, 2010; 302–317.
13. Blondeau C, Nyberg K. Links between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities. In *Advances in Cryptology—EUROCRYPT 2014, Lecture Notes in Computer Science*, Vol. 8441, Nguyen P, Oswald E (eds). Springer: Berlin Heidelberg, 2014; 165–182.
14. Zheng L, Zhang Sw. FFT-based multidimensional linear attack on PRESENT using the 2-bit-fixed characteristic. *Security and Communication Networks* 2015.
15. Lee C. Biclique cryptanalysis of PRESENT-80 and PRESENT-128. *The Journal of Supercomputing* 2014; **70**(1): 95–103.
16. Isobe T, Shibutani K. Generic key recovery attack on Feistel scheme. In *Advances in Cryptology—ASIACRYPT 2013, Lecture Notes in Computer Science*, Vol. 8269, Sako K, Sarkar P (eds). Springer: Berlin Heidelberg, 2013; 464–485.
17. Shakiba M, Dakhilalian M, Mala H. Non-isomorphic biclique cryptanalysis of full-round crypton. *Computer Standards & Interfaces* 2015; **41**: 72–78,.
18. Shakiba M, Dakhilalian M, Mala H. Non-isomorphic biclique cryptanalysis and its application to full-round mcrypton. Cryptology ePrint Archive, Report 2013/141 2013.