Improved Differential Analysis of MIBS Based on Greedy Algorithm*

Jian Liu[1], Yanjun Li[1,2,3](✉)[0000−0002−7219−0005], Runyi Liu[1], Jian Zou[4], and Zhiqiang Wang[3]

[1] Information Technology & Security Test and Evaluation Center, North China Institute of Computing Technology, Beijing 100083, China
[2] Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China
[3] Beijing Electronic Science and Technology Institute, Beijing,100070,China
[4] College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China

Abstract. MIBS is a 32-round lightweight block cipher following a Feistel structure with the block length of 64-bit and the key length of 64 or 80 bits. In this paper, the properties of the key scheduling algorithm are investigated and lots of repeated bits among the different round keys are found. Moreover, the optimal guessing order of the unknown key bits is obtained by using the greedy algorithm. At last, combined with the early abort technique, the differential cryptanalyses are improved to 15 rounds both of MIBS-64 and MIBS-80. For MIBS-64, the data complexity is 2^{59}, and the time complexity is $2^{46.2}$ encryptions. For MIBS-80, the data complexity is 2^{59}, and the time complexity is $2^{51.7}$ encryptions. The key scheduling algorithm of MIBS is similar to some other lightweight block ciphers, and we hope further similarities will help build better attacks for them as well.

Keywords: Differential Cryptanalysis · Lightweight Block Cipher · Key Scheduling Algorithm · Early Abort Technique · Greedy Algorithm.

1 Introduction

Nowadays, the Internet has been used in many areas, whereas endless security issues appear, leading to the focus on cybersecurity. As a core technique in this field, cryptography protects the security of data transmission on the Internet, preventing eavesdropping and tampering.

Symmetric cryptography is normally considered a more mature approach, compared with asymmetric cryptography, with the advantages of higher speed and more efficiency but less computation complexity. Thus, it has been widely used in many network protocols to solve related problems in this field.

* Supported by the Open Project of Henan Key Laboratory of Network Cryptography Technology (NO.LNCT2020-A09) and the Advanced Discipline Construction Project of Beijing Universities (20210101Z0401).

MIBS, proposed at the International Conference on Cryptology and Network Security 2009 [1], is a famous lightweight block cipher in the family of symmetric crypto-algorithms. The prospects of MIBS are broad due to not only the quick implementation on both software and hardware platforms but also the scope of its applications varies from advanced terminal to IoT devices.

Many cryptologists have performed the security analysis and evaluation of the MIBS cipher since it was proposed, and numerous researchers have contributed to this field. The first comprehensive analysis of MIBS was proposed by Bay et al., using sophisticated methods, such as differential analysis, linear analysis, and impossible differential analysis [2]. In the differential analysis of MIBS-64/80, they performed a 14-round key recovery attack by using the correct plain-ciphertext pairs to recommend partial key, the data complexity of this attack is 2^{40} choices of plaintext, and the computational complexity is about $2^{37.2}$ (for MIBS-64) and 2^{40} (for MIBS-80) encryptions. In 2017, Dai et al. proposed a 14-round differential analysis by guessing key bits to filter plain-ciphertext pairs and applied it to the MIBS-64 cipher, with the data complexity of 2^{59} selection plaintexts and the computational complexity of 2^{59} [3]. Yu, Pan and Li performed integral analyses on MIBS-64 respectively, for 10 and 11 rounds [4–6]. There are also impossible differential and some other cryptanalyses of reduced-round MIBS [7–10]. In 2021 Li et al. came up with quantum cryptanalysis of MIBS [11]. Among these analysis methods, differential analysis attacks more rounds of MIBS and is applied most commonly. The 15 rounds of MIBS-64 was provided with the complexity of $2^{60.7}$ times encryption [12]. Apart from these, with the development of AI technology, such as Ant Colony Optimization, genetic algorithm, etc., are utilized in the field of cryptographic design and analysis [13, 14]. In ASIACRYPT 2017, I. Nikoli' c used simulated annealing and genetic algorithms to optimize the components of SKINNY and finally got good results [15]. For example, by using a greedy algorithm, the searching technique for an active S-box can be improved, resulting in a reformed design of the crypto algorithm component [16], as well as the optimization of crypto algorithm circuit implementation, especially the circuit implementation of S-box [17]. The greedy algorithm was also applied to construct an automated searching tool to improve the cryptanalysis [18–22].

Our Contributions. By comparing various analysis methods and analyzing the key-schedule characteristics, we propose a methodology based on the early abort technique and greedy algorithm in this paper; the differential analysis of MIBS-64 and MIBS-80 can be improved to 15 rounds.

1. The 15-round differential analysis of the MIBS cipher is constructed based on the characteristic of the 12-round distinguisher.
2. Many repeated key bits among the 1st, 14th, and 15th rounds are found out. We use this key-schedule feature to figure out the optimal order of related key nibbles by using a greedy algorithm.
3. The key nibbles are guessed by performing an early abort technique to give up the wrong plain-ciphertext pairs in advance.
4. The differential cryptanalyses of MIBS are improved and the complexities are greatly reduced. For MIBS-64, the data complexity is 2^{59}, and the time

complexity is $2^{46.2}$ 15-round encryptions. For MIBS-80, the data complexity is 2^{59}, and the time complexity is $2^{51.7}$ encryptions.

The organization of this paper is as follows. The second part is preliminary knowledge, which briefly introduces the MIBS cipher and the notations to be used in this article. The third part describes MIBS's 12-round differential characteristics, key-schedule properties, and the improved early abort technique. The fourth part contains the 15-round key recovery attack and complexity analysis of MIBS-64. The fifth part gives the results of a 15-round differential analysis for MIBS-80, and the sixth part is the conclusion.

2 Preliminaries

2.1 A Brief Description of MIBS

Cybersecurity issues through the Internet have attracted attention in recent years. As a core technique in the field of information security, symmetric cryptography develops more maturely compared with asymmetric cryptography, which has advantages on the publicity as well.

The structure of MIBS is Feistel. Its block length is 64 bits, and the optional key lengths are 64 and 80 bits, which are marked as MIBS-64 and MIBS-80, respectively. The number of iteration rounds is 32. All operations in MIBS are based on nibbles, i.e., 4 bits. The round function of MIBS is SPN structure, including XOR subkey, S-box (nibble) layer, and linear permutation. The linear permutation is composed of the linear transformation and the nibble transposition, the branch number of which is 5.

The encryption structure and round function structure of the MIBS cipher are shown in Fig. 1 and Fig. 2.

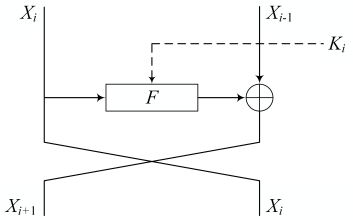

Fig. 1: One round encryption structure of MIBS.

The property of linear permutation P is used in our attack. Let $(y_8, y_7, y_6, y_5, y_4, y_3, y_2, y_1)$ be the input of the permutation P and $(y'_8, y'_7, y'_6, y'_5, y'_4, y'_3, y'_2, y'_1)$

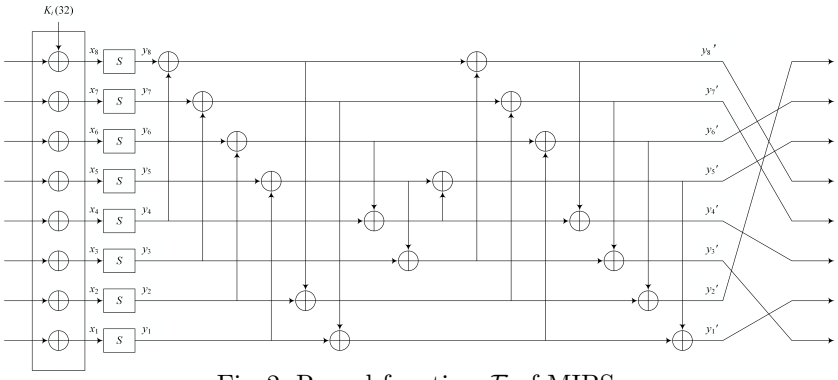

Fig. 2: Round function $\mathcal{F}$ of MIBS.

be the output which can be described as

$$
\begin{aligned}
y_1^{'} &= y_1 \oplus y_2 \oplus y_4 \oplus y_5 \oplus y_7 \oplus y_8; \quad & y_5^{'} &= y_1 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_8 \\
y_2^{'} &= y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_7; \quad & y_6^{'} &= y_1 \oplus y_2 \oplus y_4 \oplus y_5 \oplus y_6 \\
y_3^{'} &= y_1 \oplus y_2 \oplus y_3 \oplus y_5 \oplus y_6 \oplus y_8; \quad & y_7^{'} &= y_1 \oplus y_2 \oplus y_3 \oplus y_6 \oplus y_7 \\
y_4^{'} &= y_2 \oplus y_3 \oplus y_4 \oplus y_7 \oplus y_8; \quad & y_8^{'} &= y_1 \oplus y_3 \oplus y_4 \oplus y_6 \oplus y_7 \oplus y_8
\end{aligned}
$$

The inverse of permutation P described as P^{-1} is also used in our attack. Let $(y_8, y_7, y_6, y_5, y_4, y_3, y_2, y_1)$ be the output of the permutation P and $(y_8^{'}, y_7^{'}, y_6^{'}, y_5^{'}, y_4^{'}, y_3^{'}, y_2^{'}, y_1^{'})$ is the input which can be described as

$$
\begin{aligned}
y_1^{'} &= y_2 \oplus y_4 \oplus y_6 \oplus y_7 \oplus y_8; \quad & y_5^{'} &= y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_7 \oplus y_8 \\
y_2^{'} &= y_1 \oplus y_4 \oplus y_5 \oplus y_7 \oplus y_8; \quad & y_6^{'} &= y_1 \oplus y_2 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_8 \\
y_3^{'} &= y_1 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6; \quad & y_7^{'} &= y_1 \oplus y_3 \oplus y_5 \oplus y_6 \oplus y_7 \oplus y_8 \\
y_4^{'} &= y_2 \oplus y_3 \oplus y_5 \oplus y_6 \oplus y_7; \quad & y_8^{'} &= y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_6 \oplus y_7
\end{aligned}
$$

The key scheduling algorithm of MIBS adopts the same design principle as the key scheduling algorithm of the PRESENT cipher. In the key scheduling algorithm of MIBS-64, round keys K_r are generated by the 64 bits master key $K : (k_{63}, k_{62}, \cdots, k_0)$, where $0 \leq r \leq 31$. If the key state of the r-th round is expressed as $state^r$, the round function of the key scheduling algorithm can be expressed as follows.

$$
\begin{aligned}
&state^0 = user - key \\
&state^r = state^r \ggg 15 \\
&state^r = Sbox(state^r_{[63:60]}) \parallel state^r_{[59:0]} \\
&state^r = state^r_{[63:16]} \parallel state^r_{[15:11]} \oplus Round - Counter \parallel state^r_{[10:0]} \\
&K_r = state^r_{[63:32]}
\end{aligned}
$$

For MIBS-80, the master key is 80 bits, and the key scheduling algorithm can be expressed as follows.

$$
\begin{aligned}
&state^0 = user - key \\
&state^r = state^r \ggg 19 \\
&state^r = Sbox(state^r_{[79:76]}) \parallel Sbox(state^r_{[75:72]}) \parallel state^r_{[71:0]} \\
&state^r = state^r_{[79:19]} \parallel state^r_{[18:14]} \oplus Round - Counter \parallel state^r_{[13:0]} \\
&K_r = state^r_{[79:48]}
\end{aligned}
$$

where $\ggg$ means bitwise right-rotation, $[i, j]$ indicates a sequence of bit positions from the i-th bit to the j-th bit, and $\parallel$ means string concatenation. The S-box used in the round function is the same as the S-box used in function $\mathcal{F}$. Finally, the left 32 bits of the r-th round $state^r$ will be used as the r-th round key K_r.

2.2 Notations

The plaintext is denoted as (X_1, X_0), where $X_i = (x_{i,8}, x_{i,7}, \cdots, x_{i,1})$. $i = 0, 1, \cdots, r-1$. The meanings of other notations appearing in this article are as follows:

Y_r: the output of the S-boxes in the r-th round,
Z_r: the output of the linear layer P in the r-th round,
K_r: the key used in the r-th round,
$K_{r,i}$: the i-th nibble of K_r,
$y_{r,i}$: the i-th nibble of Y_r, which is $Y_r = (y_{r,8}, y_{r,7}, y_{r,6}, y_{r,5}, y_{r,4}, y_{r,3}, y_{r,2}, y_{r,1})$,
$z_{r,i}$: the i-th nibble of Z_r, which is $Z_r = (z_{r,8}, z_{r,7}, z_{r,6}, z_{r,5}, z_{r,4}, z_{r,3}, z_{r,2}, z_{r,1})$.

3 Differential Characteristics and Key Scheduling Properties

In this section, we will first introduce the 12-round differential characteristics of MIBS, then add 1 round forward and 2 rounds backward to get the position of the active S-boxes of the 1st, 14th, and 15th rounds. According to the keys used at these corresponding positions, we will introduce the relationship between the round keys of MIBS-64 and MIBS-80 in detail, respectively.

3.1 12-Round Differential Characteristics of MIBS

According to the differential distribution table of the S-box in MIBS, as shown in the Appendix, when the input differential is 5, the probability of the output differential being E is the largest, which is 2^{-2}. The input differential is E, and the probability of the output differential being 5 is also 2^{-2}. According to this feature, we can search for four 12-round differential characteristics with the least active S-boxes, and the probabilities are all 2^{-56}:

$$
\begin{aligned}
&(EE0E0EEE, 05500505) \rightarrow (EE0E0EEE, 05000000) \\
&(55050555, 0EE00E0E) \rightarrow (55050555, 0E000000) \\
&(0E000000, 55050555) \rightarrow (0EE00E0E, 55050555) \\
&(05000000, EE0E0EEE) \rightarrow (05500505, EE0E0EEE)
\end{aligned}
$$

Take the first differential characteristic as an example. The input differential value of each round is shown in Table 1.

Table 1: 12-round differential characteristic of MIBS.

Round i	ΔL_{i-1}	ΔR_{i-1}	Number of active S-boxes	Probability
1	EE0E0EEE	05500505	6	2^{-12}
2	05000000	EE0E0EEE	1	2^{-2}
3	000EEE00	05000000	3	2^{-6}
4	00050050	000EEE00	2	2^{-4}
5	0E000E00	00050050	2	2^{-4}
6	00000555	0E000E00	3	2^{-6}
7	00000000	00000555	0	1
8	00000555	00000000	3	2^{-6}
9	0E000E00	00000555	2	2^{-4}
10	00050050	0E000E00	2	2^{-4}
11	000EEE00	00050050	3	2^{-6}
12	05000000	000EEE00	1	2^{-2}
13	EE0E0EEE	05000000	-	–

The differential characteristic contains 28 active S-boxes. The input $\rightarrow$ output differential of each active S-box is 5$\rightarrow$E or E$\rightarrow$5, and the probability is 2^{-2}. It is easy to calculate that the probability of the differential characteristic is 2^{-56}.

3.2 The Key Properties of MIBS

According to the MIBS-64 key scheduling algorithm, the keys of the 1st round, 14th round and 15th round are partially duplicated or equivalent. This article mainly uses the following key scheduling properties.

Property 1 According to the MIBS-64 key scheduling algorithm, there are 17-bit repetitions between the adjacent round keys, 14-bit repetitions or equivalent (They can be obtained by querying the S-box) for the 1st round, and 15th round keys, and 29-bit repetitions for the 1st round and 14th round keys.

Property 2 For the MIBS-64 cipher, based on the 12-round differential characteristic, a 15-round differential characteristic can be obtained by adding 1 round forward and 2 rounds backward. In the first round, there are 4 active S-boxes, and 16 bits key needed to be guessed, which are reused in the 15th round with 8 bits and in the 14th round with 9 bits.

As shown in Fig. 4, for MIBS-64, K[59:52] in the 1st round are repeatedly needed in the 15th round, and K[54:52], K[59] , K[43:40], K[35] are needed repeatedly in the 14th round. K[2:60] and K[51] of the 15th round key are repeatedly needed in the 14th round. The active nibbles in the 1st, 14th, and 15th rounds

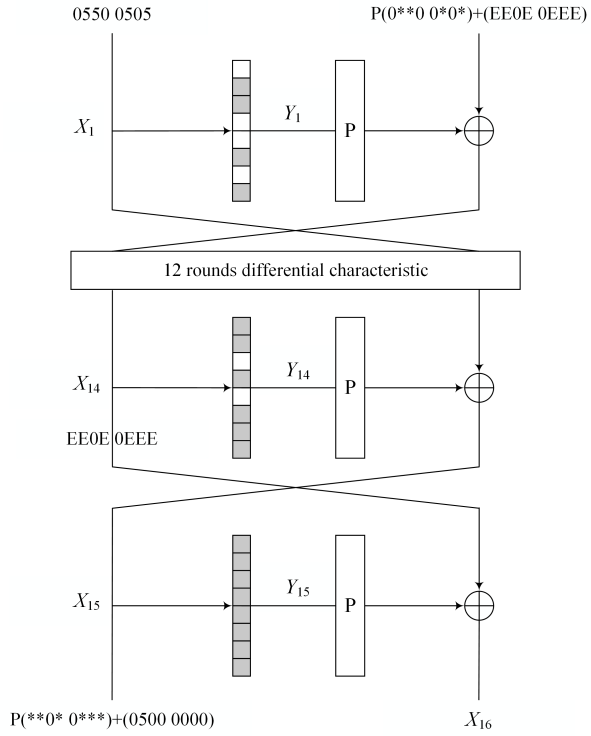

Fig. 3: 15-round Differential Attack on MIBS.

involve a 72-bit key, of which 25 bits are repeated. In fact, there are only 47 bits needed to be guessed.

MIBS-80 key scheduling has the following properties, just as MIBS-64.

Property 3 According to the MIBS-80 key scheduling algorithm, there are 13 bits of repetition between adjacent rounds of keys, 25 bits of repetition or equivalent (They can be obtained by querying the S-box.) for the 1st and 14th rounds of keys, and 6-bit repetitions between the 1st round and the 15th round of keys.

Property 4 For the MIBS-80 cipher, based on the 12-round differential characteristic, 15-round one can be obtained by adding 1 round forward and 2 rounds backward. In the first round, 4 S-boxes are active, and 16 bits key needed to be guessed, which are reused in the 15th round with 2 bits and in the 14th round with 8 bits.

As shown in Fig. 5, for MIBS-80, K[75:74] in the 1st round are repeatedly needed in the 15th round, and K[74:71] and K[59:56] are needed repeatedly in the 14th round. K[6:79] of the 15th round keys are repeatedly needed in the 14th round. The active nibbles in the 1st, 14th, and 15th rounds involve a 72-bit key, of which 18 bits are repeated. In fact, there are only 54 bits needed to be guessed.

MIBS-64	$K_{r,8}$				$K_{r,7}$				$K_{r,6}$				$K_{r,5}$				$K_{r,4}$				$K_{r,3}$				$K_{r,2}$				$K_{r,1}$			
round1	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
round2	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47
round3	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	63	62
round4	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13
round5	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28
round6	10	9	8	7	6	5	4	3	2	1	0	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43
round7	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	63	62	61	60	59	58
round8	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9
round9	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24
round10	6	5	4	3	2	1	0	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39
round11	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	63	62	61	60	59	58	57	56	55	54
round12	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5
round13	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20
round14	2	1	0	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35
round15	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	63	62	61	60	59	58	57	56	55	54	53	52	51	50

Fig. 4: The key bits guessed for 15 rounds MIBS-64.

MIBS-80	$K_{r,8}$				$K_{r,7}$				$K_{r,6}$				$K_{r,5}$				$K_{r,4}$				$K_{r,3}$				$K_{r,2}$				$K_{r,1}$			
round1	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
round2	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	79	78	77	76	75	74	73	72	71	70	69	68	67
round3	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6
round4	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25
round5	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44
round6	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63
round7	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
round8	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21
round9	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40
round10	10	9	8	7	6	5	4	3	2	1	0	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59
round11	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	79	78
round12	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
round13	67	66	65	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36
round14	6	5	4	3	2	1	0	79	78	77	76	75	74	73	72	71	70	69	68	67	66	65	64	63	62	61	60	59	58	57	56	55
round15	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	79	78	77	76	75	74

Fig. 5: The key bits guessed for 15 rounds MIBS-80.

3.3 The Early Abort Technique and The Optimized Algorithm

The main idea behind the extension of the attack is based on the early abort technique [23]. We guess smaller portions of subkeys and discard all the disqualified texts earlier than usual.

In order to use the early abort technique more effectively, we use the BP algorithm proposed by Boyar and Matthews in [13] to sort the keys that need to be guessed in the key recovery stage so that the step-by-step calculation complexity is minimized, which is known as a greedy algorithm. The idea of applying this algorithm in this article is to use the key bits guessed in the previous step as much as possible in the key needed in this step.

We give the set X that includes the nibbles in K_1 need to be guessed: $X = \{1, 3, 6, 7\}$, and the set Y that includes the nibbles in K_{14} need to be guessed: $Y = \{1, 2, 3, 5, 7, 8\}$. Then the other set P is also needed, because we can guess $K_{14,k}$ if and only if all $K_{15,j}$s of P_k(the subset of P) are known.

$$P = \{\{1,2,4,5,7,8\}, \{2,3,4,5,6,7\}, \{1,2,3,5,6,8\}, \{2,3,4,7,8\}, \\ \{1,3,4,5,8\}, \{1,2,4,5,6\}, \{1,2,3,6,7\}, \{1,3,4,6,7,8\}\}$$

For example, only when $K_{15,1}, K_{15,2}, K_{15,3}, K_{15,5}, K_{15,6}, K_{15,8}$ are all known, then $X_{14,3}$ will be obtain and we can guess the $K_{14,3}$.

Algorithm 1 Determining key guessing order with greedy algorithms

Input: $K_1, K_{14}, K_{15}, P, X, Y$
Output: D, format$(K_{i,j}, t)$

for $i \in X$ **do**
  for $j \in [1, 8]$ **do**
    $t = |K_{1,i} \cap K_{15,j}|$ //The number of elements in the intersection;
    $e = |Max(|K_{1,k} \cap K_{15,j}|)|, k \in X \setminus \{i\}$; //The maximum number of remaining intersection elements.
    $T \leftarrow (t, e, K_{1,i}, K_{15,j}, i, j)$; //Add element $(t, e, K_{1,i}, K_{15,j}, i, j)$ to list T.
  end for
end for
$T \leftarrow DeSortByt(T)$; //Descending Sort by t.
for $t \in [4, 3, 2, 1]$ **do**
  $T.t \leftarrow DeSortBye(T.t)$; //Descending Sort by e.
end for
$c = 1; k = 1;$
while $(k \leq |T|)$ **do**
  $D_c = (T_{k,3}, T_{k,1})$; //Assign element $(T_{k,3}, T_{k,1})$ to D_c.
  $D_{c+1} = (T_{k,4}, 4 - e)$; //Assign element $(T_{k,4}, 4 - e)$ to D_{c+1}.
  $R \leftarrow T_{k,6}$; //Add element $T_{k,6}$ to list R.
  $c = c + 2; k = k + 1;$
end while
for $k \in Y$ **do**
  $l^* = Min(|P_l \cap R|), l \in Y$; //The minimum number of intersection elements.
  for $j \in P_{l^*} \setminus (P_{l^*} \cap R)$ **do**
    $D_c = (K_{15,j}, 4)$; //Assign element $(K_{15,j}, 4)$ to D_c.
    $R \leftarrow j; c = c + 1;$
  end for
  $D_c = (K_{14,k}, r)$; //r is the number of remaining unguessed bits in the nibble.
  $c = c + 1;$
end for
$D_c = (K_{1,i}, r)$; //Guess the remaining bits in K_1.

First, according to the repeated bits of $K_{1,i}$ and $K_{15,j}$ that need to be guessed we optimal the order of them. Then, based on the diffusion layer of P and the $K_{15,j}$ that we know in the set of R, we determine the $K_{14,k}$ as the target key nibble, and guess the $K_{15,j}$ (For $j \in P_{l^*} \setminus (P_{l^*} \cap R)$) until we can guess $K_{14,k}$. At last, we guess all $K_{14,k}$ and $K_{1,i}$, and put out the format $(K_{i,j}, t)$.

For each guess of a key nibble, 2^{-3} ciphertext pairs can be filtered out by using the early abort technique, so as to optimize the computational complexity. For example, if there are t pairs, in the 1st step, we guess 4 bits $K_{1,7}$(=K[59:56]); in the 2nd step, we guess $K_{15,2}$ (only two bits K[55:54] need to be guessed) instead of $K_{1,6}$. This is because the two bits K[55:54] will not be guessed for $K_{1,6}$ in the next step. In the 1st-3rd steps, we guess 8 bits of the key in total to satisfy a 9-bit condition. The time complexity is $t \times 2^4 + t \times 2^{-3} \times 2^2 \times 2^4 + t \times 2^{-3\times 3} \times 2^2 \times 2^2 \times 2^4 \approx t \times 2^{4.8}$ instead of $t \times 2^8$, which is counted without the early abort technique.

4 Differential Analysis of 15-round MIBS-64

4.1 Plaintext Structure

In the key recovery attack of the 15-round MIBS-64 cipher, based on the 12-round differential characteristics in Section 3.1, we proposed the 15-round MIBS cipher as shown in Fig. 3. One round is added in front of the original rounds, and two rounds are added in the end. The intermediate rounds (round 2 to round 13) are recognized as the 12-round differential characteristic. In the following part, we will illustrate the way to recover the key bits corresponding to the active S-box positions in the 1st, 14th and 15th rounds.

In the 1st round, there are 4 active S-boxes. After the S-box substitution, 4 nibbles remain differential non-zero exactly. Thus, the differential of the input plaintext pair, only in these positions, needs to be non-zero. In other word, the input plaintext differential, which satisfies the form $[(05500505), P(0{*}{*}00{*}0{*}) + (EE0E0EEE)]$, can be considered a plaintext structure.

The plaintext structure is defined as (X, X'):

$$X = [C, P(cxyczczw) \oplus C], X' = [C \oplus T_1, P(cxyczczw) \oplus C \oplus T_2]$$

where $T_1 = 05500505$, $T_2 = EE0E0EEE$, C and c are constants, and x, y, z, w are arbitrary.

We can figure out the differential of plaintext pair

$$\Delta P = X \oplus X' = [05500505, P(0{*}{*}00{*}0{*}) \oplus EE0E0EEE]$$

where the symbol * means it can be selected arbitrarily. A total of 2^{16} plaintexts constitute 2^{31} plaintext pairs.

4.2 15-Round Differential Attack to MIBS-64

The 2^m selected structures lead to less than 2^{m+16} plaintexts in total. Thus, 2^{m+31} pairs of ciphertexts will be obtained after 15 rounds of encryption. The attack procedure is presented in this part.

Step 1. Construct plaintext pairs and obtain the corresponding ciphertext pairs.

Take 2^{m+16} plaintexts to construct 2^{m+31} pairs of (X, X') which satisfy $X \oplus X' = [05500505, P(0 * *00 * 0*) \oplus EE0E0EEE]$, and obtain their corresponding ciphertexts (Y, Y'). Next, we filter out the wrong pairs. The differential of the remained differential of ciphertext pairs should satisfy $\Delta X_{15} = P(* * 0 * 0 * **) \oplus (05000000)$. And we can know that one non-zero input differential corresponds to seven non-zero output differentials which is about one half of 16 by looking up the differential distribution table of S-box. In this way, each S-box can filter 2^{-1} pairs, 2^{-8} and 2^{-6} pairs will be filtered according to the non-zero differentials of $\Delta X_{15} \to \Delta Y_{15}$ and $\Delta X_{14} \to \Delta Y_{14}$. According to 4 active S-boxes of $\Delta X_1 \to \Delta Y_1$, another 2^{-4} pairs will be filtered. Moreover, there are 2 non-active S-boxes of $\Delta X_{14} \to \Delta Y_{14}$, so another 2^{-8} pairs will be filtered. Therefore, only $2^{m+31-8-6-4-8} = 2^{m+5}$ pairs are left.

Step 2. Guess key bits of K_1, K_{15}, and K_{14} until finding out the right pairs which satisfy the 12-round differential path.

For the remaining 2^{m+5} pairs, we use the early abort technique to guess key bits after the 1st step, filtering out unsatisfied pairs in advance at each sub-step. There are 47 key bits that need to be guessed in total: the overlap between 1st round key and 15th round key is 8 bits, the overlap between 1st round key and 14th round key is 9 bits, and the overlap between 15th round key and the 14th round key is 12 bits. These key bits involve 18 nibbles and need to be guessed in 18 substeps. According to Algorithm 1 we get the optimal order of the guessed keys just as shown in Table 2.

In the substep1, four bits $K_{1,7}$ are guessed, and three bits are filtered out, then 2^{m+2} pairs remain; in the substep2, two bits $K_{15,2}$ are guessed, and three bits are filtered out, then 2^{m-1} pairs remain; $\cdots$ By similar analysis, after guessing key bits and filtering out wrong pairs according to the order shown in Table 2, 2^{m-49} pairs remain after the substep18.

Step3. Get the correct key bits.

We count each guessed key in the remaining pairs, and the key that appears most frequently is the correct key. At last, 47 key bits will be obtained in total, and the remaining 64 – 47 = 17 bits can be obtained by brute force.

An important criterion of the differential analysis is the proportion of the probability of the right key being suggested by a right pair to the probability of a random key being suggested by a random pair with the given initial differential. This proportion is defined as the "signal-to-noise ratio". Biham and Shamir choose an appropriate value of m to make the differential analysis succeed with high probability [24].

Table 2: Key guessing order and the corresponding complexity of MIBS-64.

Substep	Key nibble	Number of key guessing bits	Complexity	
			Exponent increase	Exponent
substep1	$K_{1,7}$	4	4	4
substep2	$K_{15,2}$	2	-1	3
substep3	$K_{1,6}$	2	-1	2
substep4	$K_{15,3}$	2	-1	1
substep5	$K_{15,1}$	2	-1	0
substep6	$K_{15,6}$	4	1	1
substep7	$K_{15,7}$	4	1	2
substep8	$K_{14,7}$	1	-2	0
substep9	$K_{15,4}$	3	0	0
substep10	$K_{15,5}$	4	1	1
substep11	$K_{14,2}$	4	1	2
substep12	$K_{1,3}$	1	-2	0
substep13	$K_{15,8}$	4	1	1
substep14	$K_{14,5}$	0	-3	-
substep15	$K_{14,8}$	0	-3	-
substep16	$K_{14,3}$	3	0	-3
substep17	$K_{14,1}$	3	0	-3
substep18	$K_{1,1}$	4	1	-4

The signal-to-noise ratio can be computed according to the following formula:

$$SNR = \frac{p}{\alpha \cdot \beta / 2^k}$$

where k is the number of guessed key bits, p is the probability of the differential characteristic, α is the average number of keys suggested by a counted pair, and β is the ratio of the counted pairs to all pairs (both counted and discarded).

In the analysis above, we have guessed 47 subkey bits, and we assume the probability of the differential characteristic is 2^{-56}. For every test in Step 2, we guess 2^{47} possible keys, and a counted pair needs to satisfy the 54-bit condition, thus $\alpha = 2^{-7}$. In Step 1, a 26-bit condition is used to discard the pairs, thus $\beta = 2^{-26}$. Therefore, the signal-to-noise ratio of this attack is $2^{47} \times 2^{-56}/2^{-33} = 2^{24}$. According to the statement of Biham and Shamir, about 3∼4 right pairs are needed to perform a successful differential analysis when $S/N = 2^{24}$.

Therefore, we choose m=43, the expectation of the remaining ciphertext pairs is about $2^{43+31-16-56} = 4$ for the right key guessing, and the expectation of the remaining ciphertext pairs is about $2^{43-49} = 2^{-6}$ for a wrong key guessing.

4.3 Complexity Analysis

The data complexity of the attack is 2^{59} plaintexts; the first step needs to deal with the plaintext listed in a hash table. Otherwise, it will exceed the storage

required for the brute force attack. The second step needs to query the S-box for 2^{m+10} times, which is calculated as follows.

$$2^{m+5}\times 2^4+2^{m+5-3}\times 2^2\times 2^4+2^{m+5-3-3}\times 2^2\times 2^2\times 2^4+\cdots+2^{m+5-3\times 17}\times 2^{47} < 2^{m+10}$$

If we choose m = 43, then the search time will not exceed $2^{53}/(8\times 15)\approx 2^{46.1}$ encryption operations of 15 rounds. The remaining 17-bit brute-force searching needs less than 2^{17} encryptions, so the overall time complexity does not exceed $2^{46.2}$ encryptions of 15 rounds.

5 Differential Cryptanalysis of 15-round MIBS-80

The selected plaintext structure in the attack process is similar to MIBS-64. The main difference is the order of key bits guessed in the second step. The complexity index is affected by the order of the guessing key which can be optimized through greedy search.

Table 3: Key guessing order and the corresponding complexity of MIBS-80.

Substep	Key nibble	Number of key guessing bits	Complexity	
			Exponent increase	Exponent
substep1	K1,7	4	4	4
substep2	K15,1	2	-1	3
substep3	K15,3	4	1	4
substep4	K15,4	4	1	5
substep5	K15,5	4	1	6
substep6	K15,8	4	1	7
substep7	K14,5	1	-2	5
substep8	K1,6	3	0	0
substep9	K15,6	4	1	6
substep10	K15,2	4	1	7
substep11	K14,3	4	1	8
substep12	K15,7	4	1	9
substep13	K14,7	0	-3	-
substep14	K14,8	0	-3	-
substep15	K14,1	4	1	4
substep16	K1,3	1	-2	2
substep17	K14,2	3	0	2
substep18	K1,1	4	1	3

As shown in Table 3, in the substep1, four bits $K_{1,7}$ are guessed, and three bits are filtered out, then 2^{m+2} pairs remain; in the substep2, two bits $K_{15,1}$ are guessed, and three bits are filtered out, then 2^{m-1} pairs remain; $\cdots$ By similar analysis, after guessing key bits and filtering out wrong pairs according to the order in the table, there are 2^{m-49} pairs remaining after the substep18.

We choose m = 43, then there are $2^{43+31-16-56} = 4$ pairs remained for the right key guessed, and 2^{-6} pairs remained for the wrong key on average. The whole attack requires a data complexity of 2^{59} plaintexts, and the second step requires querying S-box for

$$2^{m+5}\times 2^4 + 2^{m+5-3}\times 2^2\times 2^4 + 2^{m+5-3-3}\times 2^2\times 2^4\times 2^4 + \cdots + 2^{m+5-3\times 17}\times 2^{54} < 2^{m+15.5}$$

times. The time complexity will not exceed $2^{58.5}/(8\times 15) \approx 2^{51.6}$ times of 15-round encryption operations. The calculation amount of the remaining 80-54=26 bits exhaustive search is 2^{26}, so the main time complexity does not exceed $2^{51.7}$ encryptions of 15 rounds.

6 Conclusions

In this paper, we propose a 15-round differential analysis of the MIBS-64/80 cipher derived from the original 12-round distinguisher with additional rounds, more specifically, 1 round forward and 2 rounds backward. Then we carry out a key recovery attack based on the feature of the key schedule algorithm. We use two methods to reduce time complexity. One is the early abort technique, and the other is the greedy algorithm, by which we optimize the guess key sequence to improve efficiency. For MIBS-64, the data complexity is 2^{59}, and the time complexity is $2^{46.2}$ encryptions. For MIBS-80, the data complexity is 2^{59}, and the time complexity is $2^{51.7}$ encryptions. The comparison of our results and the previous differential cryptanalysis results is shown in Table 4.

Table 4: The comparison of differential cryptanalysis results.

Cipher	Rounds	Data	Time	Reference
MIBS-64	14	2^{40}	$2^{37.2}$	[2]
MIBS-80	14	2^{40}	2^{40}	[2]
MIBS-64	14	2^{59}	2^{59}	[3]
MIBS-64	15	2^{58}	$2^{60.7}$	[12]
MIBS-64	15	2^{59}	$2^{46.2}$	This Paper
MIBS-80	15	2^{59}	$2^{51.7}$	This Paper

The key scheduling algorithm of MIBS is similar to other lightweight block ciphers, and we hope this feature will help us do further research about attacks for analogous encryption algorithms. Moreover, we also aim to set up the key schedule criteria, as well as better evaluation principles to design secure symmetric encryption algorithms.

Declaration of competing interest

We declare that there are no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

The differential distribution table of MIBS S-box

In\Out	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	2	0	0	2	2	2	0	4	2	0	2	0
2	0	2	0	2	0	0	0	4	0	0	2	2	2	0	0	2
3	0	0	2	0	0	2	2	2	0	0	0	2	4	2	0	0
4	0	0	0	2	0	2	2	2	2	4	0	0	0	0	0	2
5	0	0	2	2	2	0	0	2	0	0	0	0	0	2	4	2
6	0	0	2	0	0	2	0	0	4	0	2	0	2	0	2	2
7	0	2	2	2	4	2	0	0	0	2	0	0	2	0	0	0
8	0	0	0	0	2	0	2	0	0	2	2	0	2	2	0	4
9	0	4	0	0	2	2	0	0	2	0	0	2	0	2	0	2
10	0	2	0	4	0	0	2	0	2	0	0	0	2	2	2	0
11	0	0	2	2	2	0	2	0	2	0	4	2	0	0	0	0
12	0	2	2	0	0	0	4	0	0	2	0	2	0	0	2	2
13	0	2	4	0	0	0	0	2	2	2	2	0	0	2	0	0
14	0	2	0	0	2	4	2	2	0	0	2	0	0	0	2	0
15	0	0	0	2	0	2	0	0	0	2	2	2	0	4	2	0

Bibliography

[1] Izadi, M., Sadeghiyan, B., Sadeghian, S.S., Khanooki, H.A.: MIBS: a new lightweight block cipher. In: International Conference on Cryptology and Network Security. pp. 334–348. Springer (2009), https://doi.org/10.1007/978-3-642-10433-6_22

[2] Bay, A., Nakahara, J., Vaudenay, S.: Cryptanalysis of Reduced-Round MIBS Block Cipher. In: Cryptology and Network Security. pp. 1–19 (2010), https://doi.org/10.1007/978-3-642-17619-7_1

[3] Dai, Y., Tian, Y., Chen, S.: Cryptanalysis of Reduced-Round MIBS Block Cipher. Journal of Information Engineering University **18**(1), 87–92 (2017), https://doi.org/10.3969/j.issn.1671-0673.2017.01.017

[4] Yu, X., Wu, W., Li, Y.: Integral Attack of Reduced-Round MIBS Block Cipher. Journal of Computer Research and Development **50**(10), 2117–2125 (2013), https://crad.ict.ac.cn/CN/Y2013/V50/I10/2117

[5] Pan, Z., Guo, J., Cao, J., Luo, W.: Integral attack on MIBS block cipher. Journal on Communications (157-163+171) (2014), https://doi.org/10.3969/j.issn.1000-436x.2014.07.019

[6] Li, Y., Sun, Q., Ou, H., Wang, Z.: Improved Integral Attacks on MIBS-64 Block Cipher. Journal of Cryptologic Research **8**(4), 669–679 (2021), http://www.jcr.cacrnet.org.cn/CN/10.13868/j.cnki.jcr.000468

[7] Chen, P., Liao, F., Wei, H., et al.: Related-key impossible differential attack on a light-weight block cipher MIBS. Journal on Communications **2**(2), 190–193,201 (1 2014), https://doi.org/10.3969/j.issn.1000-436x.2014.02.023

[8] Cheng, L., Xu, P., Wei, Y.: New related-key impossible differential attack on MIBS-80. In: 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS). pp. 203–206. IEEE (2016), https://doi.org/10.1109/INCoS.2016.41

[9] Qiao, K., Hu, L., Sun, S., Ma, X.: Related-key rectangle cryptanalysis of reduced-round block cipher MIBS. In: 2015 9th International Conference on Application of Information and Communication Technologies (AICT). pp. 216–220 (2015), https://doi.org/10.1109/ICAICT.2015.7338549

[10] Chen, L., Wang, G., Zhang, G.: MILP-based Related-Key Rectangle Attack and Its Application to GIFT, Khudra, MIBS. The Computer Journal **62**(12), 1805–1821 (10 2019), https://doi.org/10.1093/comjnl/bxz076

[11] Li, Y., Lin, H., Yi, Z., Xie, H.: Quantum Cryptanalysis of MIBS. Journal of Cryptologic Research **8**(6), 989–998 (2021), http://www.jcr.cacrnet.org.cn/CN/Y2021/V8/I6/989

[12] Qiu, D., Mao, M., Li, Y., Li, Y.: Differential Analysis of MIBS-64 and Evaluation under Quantum Model. In: Proceedings of the 2021 5th International Conference on Electronic Information Technology and Computer Engineering. p. 1161–1165. EITCE 2021 (2022), https://doi.org/10.1145/3501409.3501614

[13] Boyar, J., Matthews, P., Peralta, R.: On the Shortest Linear Straight-Line Program for Computing Linear Forms. In: Mathematical Foundations of Computer Science 2008. pp. 168–179 (2008), https://doi.org/10.1007/978-3-540-85238-4_13
[14] Khan, S., Shahzad, W., Khan, F.A.: Cryptanalysis of Four-Rounded DES Using Ant Colony Optimization. In: 2010 International Conference on Information Science and Applications. pp. 1–7 (2010), https://doi.org/10.1109/ICISA.2010.5480260
[15] Nikolić, I.: How to Use Metaheuristics for Design of Symmetric-Key Primitives. In: Advances in Cryptology – ASIACRYPT 2017. pp. 369–391 (2017), https://doi.org/10.1007/978-3-319-70700-6_13
[16] Jin, X., Duan, Y., Zhang, Y., Huang, Y., Li, M., Mao, M., Singh, A.K., Li, Y.: Fast Search of Lightweight Block Cipher Primitives via Swarm-like Metaheuristics for Cyber Security. ACM Trans. Internet Technol. **21**(4) (jul 2021), https://doi.org/10.1145/3417296
[17] Ueno, R., Homma, N., Nogami, Y., Aoki, T.: Highly efficient $GF(2^8)$ inversion circuit based on hybrid GF representations. Journal of Cryptographic Engineering **9**, 101–113 (2019), https://doi.org/10.1007/s13389-018-0187-8
[18] Sun, S., Hu, L., Song, L., Xie, Y., Wang, P.: Automatic Security Evaluation of Block Ciphers with S-bP Structures Against Related-Key Differential Attacks. In: Information Security and Cryptology. pp. 39–51 (2014), https://doi.org/10.1007/978-3-319-12087-4_3
[19] Sun, S., Hu, L., Wang, P., Qiao, K., Ma, X., Song, L.: Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES (L) and other bit-oriented block ciphers. In: Advances in Cryptology – ASIACRYPT 2014. pp. 158–178 (2014), https://doi.org/10.1007/978-3-662-45611-8_9
[20] Sasaki, Y., Todo, Y.: New impossible differential search tool from design and cryptanalysis aspects. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 185–215. Springer (2017), https://doi.org/10.1007/978-3-319-56617-7_7
[21] Zhu, B., Dong, X., Yu, H.: MILP-based differential attack on round-reduced GIFT. In: Topics in Cryptology – CT-RSA 2019. pp. 372–390 (2019), https://doi.org/10.1007/978-3-030-12612-4_19
[22] Abdelkhalek, A., Sasaki, Y., Todo, Y., Tolba, M., Youssef, A.M.: MILP modeling for (large) s-boxes to optimize probability of differential characteristics. IACR Transactions on Symmetric Cryptology **2017**(4), 99–129 (Dec 2017), https://tosc.iacr.org/index.php/ToSC/article/view/805
[23] Lu, J., Dunkelman, O., Keller, N., Kim, J.: New Impossible Differential Attacks on AES. In: Indocrypt 2008. Lecture Notes in Computer Science, vol. 5365, pp. 279–293 (2008), https://doi.org/10.1007/978-3-540-89754-5_22
[24] Biham, E., Shamir, A.: Differential cryptanalysis of DES-like cryptosystems. Journal of CRYPTOLOGY **4**(1), 3–72 (1991), https://doi.org/10.1007/BF00630563