Security Analysis of the LED Cipher Based on MILP Methods: Postprint

Authors: Liu Botao, Peng Zhanggen, Wu Ruixue, Ding Hongfa, Xie Mingming

Date: 2018-12-13T00:00:00+00:00

Abstract

Automated search algorithms for solving differential characteristics and linear approximations have become a research hotspot in differential and linear cryptanalysis of block ciphers. This paper proposes a nibble-oriented MILP model for automated search of differential characteristics and linear approximations in cryptographic algorithms, and applies it to analyze the lightweight LED cipher. By solving for the number of active S-boxes with fewer variables and constraint inequalities, it is demonstrated that four rounds of operation contain at least 25 active S-boxes. This result coincides with the theoretical value of active S-boxes provided by the algorithm designers, thereby verifying the correctness of the proposed method. Finally, the maximum differential characteristic probability and linear approximation probability of the LED algorithm are computed, demonstrating its resistance to differential and linear attacks.

Full Text

Preamble

Vol. 37 No. 2
Application Research of Computers
Accepted Paper

Security Analysis of LED Block Cipher Based on MILP Method

Liu Botao[1],[2], Peng Changgen[1],[2], Wu Ruixue[1],[3], Ding Hongfa[2], , Xie Mingming[3],
[1]College of Computer Science & Technology, [2]Guizhou Province Key Laboratory of Public Big Data, [3]Institute of Cryptography & Data Security, College of Mathematics & Statistics, Guizhou University, Guiyang 550025, China

Abstract: Automated search algorithms for solving differential characteristics and linear approximations have become a research focus in differential and linear

cryptanalysis of block ciphers. This paper proposes a method for automatically searching differential characteristics and linear approximations of cryptographic algorithms based on a half-byte MILP model. Applying this method to analyze the lightweight LED block cipher, we solve for the minimum number of active S-boxes with fewer variables and constraint inequalities. For 4 rounds of operation, at least 25 active S-boxes are obtained, which matches the theoretical value of active S-boxes given by the algorithm designers, thereby verifying the correctness of the proposed method. Finally, we calculate the maximum differential characteristic probability and linear approximation probability of the LED algorithm, demonstrating its resistance to differential and linear attacks.

Keywords: block cipher; differential attack; linear attack; MILP model; LED

0 Introduction

In recent years, lightweight block ciphers have attracted significant attention from academia and industry as a security guarantee for Internet of Things (IoT) devices. Several lightweight block cipher algorithms have been proposed both domestically and internationally, including notable designs such as PRESENT[1], LED[2], LBlock[3], KLEIN[4], RECTANGLE[5], SKINNY[6], and SFN[7]. The lightweight LED block cipher was introduced by Guo Jian et al. from Nanyang Technological University at CHES 2011. The algorithm design balances software and hardware implementation performance, offering fast software execution efficiency. For hardware implementation, it employs an ultra-lightweight (or even non-existent) key schedule where the key is directly implemented through single-input flip-flops, saving substantial hardware area resources. This ultra-lightweight key scheduling approach has been adopted and learned from by multiple algorithms including SKINNY. Compared with PRESENT, LBlock, KLEIN, and RECTANGLE, LED achieves a smaller hardware footprint. To ensure confidentiality, integrity, and authentication for information transmission in micro wireless sensor devices, wireless sensor networks (WSN)[8], vehicular ad-hoc networks (VANET)[9], and to protect user privacy in mobile smart devices[10], the LED algorithm stands out among lightweight block ciphers for its strong adaptability and high execution efficiency in both software and hardware, making it highly suitable for intelligent micro-devices in IoT applications.

While lightweight block cipher algorithms offer advantages in hardware implementation such as low resource consumption and power usage, their security remains a primary concern for cryptographers. For block ciphers, differential[11] and linear[12] attacks represent two powerful cryptanalytic methods, and resistance against these attacks constitutes a crucial security evaluation metric. Consequently, designers must first assess differential and linear attacks to validate the security of their cryptographic algorithms. The key to these attacks lies in finding high-probability differential characteristics and linear approximations. The maximum differential characteristic probability and maximum linear

approximation probability are calculated by analyzing the minimum number of active S-boxes in the algorithm. Currently, automated search algorithms for obtaining the minimum number of active S-boxes have become a research hotspot in differential and linear cryptanalysis.

In 2011, Mouha et al.[13] applied Mixed Integer Linear Programming (MILP) to automatically solve for the minimum number of active S-boxes, using bytes as the unit for differential variables and combining SPN (Substitution-Permutation Network) and Feistel structures to constrain and programmatically solve byte-level confusion and diffusion components. In 2014, Sun et al.[14] extended the MILP automation method at ASIACRYPT, expanding the differential variable unit from byte-level to bit-level. In 2016, Fu et al.[15] proposed a MILP-based automated search method for ARX-structure block ciphers at FSE, analyzing the differential characteristics and linear approximations of the lightweight SIMON algorithm[16]. In 2017, Yin et al.[17,18] built upon the method in[14] to automatically search for differential characteristics and linear approximations of the lightweight ESF algorithm[19], demonstrating that full-round ESF can resist differential attacks.

Currently, security analysis of the LED block cipher primarily relies on traditional theoretical methods for solving differential characteristics and linear approximations[20], and the algorithm designers did not provide specific inference or calculation processes to prove their claims. No publicly published results have been found domestically or internationally regarding the application of MILP automated search models to solve for differential characteristics and linear approximations of the LED algorithm. Therefore, for protecting sensitive information in IoT devices, using MILP models for automated half-byte-based security analysis of the LED algorithm and other ciphers provides significant value for strengthening the design and analysis of proprietary cryptographic algorithms.

This paper proposes a half-byte-oriented MILP model for automatically searching differential characteristics and linear approximations of cryptographic algorithms. Through programming implementation, this model solves for the number of active S-boxes with fewer variables and constraint inequalities, improving the efficiency of MILP-based automated analysis. We accurately verify the number of active S-boxes in LED and calculate its maximum differential characteristic probability and maximum linear approximation probability, proving that LED can resist differential and linear attacks.

---

1 LED Algorithm Description

1.1 Common Symbols and Terminology

- P: 64-bit plaintext
- C: 64-bit ciphertext

- K: Original key
- SK: 64-bit round key
- $\oplus$: XOR operation
- $||$: Concatenation operator
- *state*: Intermediate state value
- *rc*: Round constant value

1.2 LED Algorithm

LED is a typical SPN-structure lightweight block cipher with a 64-bit block size and two key sizes: 64-bit and 128-bit, denoted as LED-64 and LED-128, respectively, with 32 and 48 rounds of iteration.

The encryption process consists of five module components in the round function: AddRoundKey, AddConstants, SubCells, ShiftRows, and MixColumnsSerial. The LED encryption flow is illustrated in [Figure 1: see original paper]. LED operates on half-byte (*nibble*) units, with plaintext input denoted as P and ciphertext output as C.

[Figure 1: see original paper] The encryption process of LED

The following describes and analyzes the five module components of LED' s round function and its key schedule:

1) AddRoundKey Transformation
AddRoundKey performs bitwise XOR between the intermediate state value (*state*) and the round key (SK). LED applies AddRoundKey once every 4 rounds. The main difference between LED-64 and LED-128 lies in their AddRoundKey operations:

For LED-64:

$$state[0:63] \leftarrow state[0:63] \oplus SK_1$$

For LED-128, odd-numbered 4-round operations:

$$state[0:63] \leftarrow state[0:63] \oplus SK_1$$

Even-numbered 4-round operations:

$$state[0:63] \leftarrow state[0:63] \oplus SK_2$$

2) AddConstants Transformation
AddConstants performs XOR between the 32-bit intermediate state value and fixed constants (0, 1, 2, 3) along with round constants *rc* (detailed in[2]):

$$state[0:7]||state[4i+4:4i+7] \leftarrow state[0:7]||state[4i+4:4i+7] \oplus rc_j$$

where $i = 0, 1, 4, 5, 8, 9, 12, 13$ and $j = 0, 1, 2, 3$.

3) SubCells Transformation
SubCells is the non-linear layer of LED, using the 4×4 S-box from PRESENT. The S-box is shown in , and the substitution operation is:

$$state[0:3]||state[4i:4i+3] \leftarrow Sbox(state[0:3]||state[4i:4i+3]), \quad 0 \leq i \leq 15$$

** S-box in hexadecimal form**

4) ShiftRows Transformation
ShiftRows performs cyclic shifts on nibble blocks. In the 4×4 nibble matrix, Row 1 remains unchanged, Row 2 is cyclically left-shifted by 1 nibble, Row 3 by 2 nibbles, and Row 4 by 3 nibbles, as shown in [Figure 2: see original paper].

[Figure 2: see original paper] Operation of shiftrows

5) MixColumnsSerial Transformation
MixColumnsSerial multiplies the 4×4 nibble intermediate state matrix with a fixed mixing matrix over the finite field $GF(2^4)$:

$$\begin{bmatrix} state[0:3] & state[16:19] & state[32:35] & state[48:51] \\ state[4:7] & state[20:23] & state[36:39] & state[52:55] \\ state[8:11] & state[24:27] & state[40:43] & state[56:59] \\ state[12:15] & state[28:31] & state[44:47] & state[60:63] \end{bmatrix} \leftarrow \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} state[0:3] & state[16:19 \\ state[4:7] & state[20:23 \\ state[8:11] & state[24:27 \\ state[12:15] & state[28:31 \end{bmatrix}$$

6) Key Schedule
LED employs a non-expanded key schedule. In LED-64, the 64-bit original key is directly assigned as the round key $SK_1 = K$. In LED-128, the 128-bit original key is split into two equal parts assigned to round keys SK_1 and SK_2, where $SK_1||SK_2 = K$.

---

3 Automated MILP Model Construction for LED

MILP is an optimization problem in operations research that seeks to maximize or minimize an objective function under linear constraints. In constructing automated differential characteristic analysis, each nibble' s differential value is either $\Delta = 0$ or $\Delta \neq 0$, described using binary variables $\{0, 1\}$ to characterize the differential pattern across r rounds.

Definition 1 In a string, n nibble differences are represented as $(\Delta_0, \Delta_1, ..., \Delta_{n-1})$, with corresponding binary variables $(x_0, x_1, ..., x_{n-1})$. When a nibble difference $\Delta_i = 0$, the variable $x_i = 0$; when $\Delta_i \neq 0$, then $x_i = 1$, where $0 \leq i \leq n-1$.

In block ciphers, round functions achieve confusion and diffusion: non-linear components provide confusion while linear components provide diffusion.

a) Non-linear Transformation Constraints
Non-linear transformations use S-box substitution. Lightweight block ciphers

employ 4×4 S-boxes operating on half-bytes. Let the 4-bit input difference be $(\Delta a_0, \Delta a_1, \Delta a_2, \Delta a_3)$ and output difference be $(\Delta b_0, \Delta b_1, \Delta b_2, \Delta b_3)$. The active status of this 4×4 S-box is denoted by variable t_A.

In non-linear constraints, at least one S-box must be active in round r' s differential characteristic. To ensure x_0, x_1, x_2, x_3 has at least one active state, $t_A = 1$ is enforced by:

$$\begin{cases} t_A - x_0 \geq 0 \\ t_A - x_1 \geq 0 \\ t_A - x_2 \geq 0 \\ t_A - x_3 \geq 0 \end{cases}$$

Conversely, when $t_A = 1$, at least one of x_0, x_1, x_2, x_3 must be active:

$$x_0 + x_1 + x_2 + x_3 - t_A \geq 0$$

For a bijective 4×4 S-box, when the input nibble difference is non-zero, the output nibble difference must also be non-zero; when input is zero, output is zero. This is constrained by:

$$\begin{cases} \sum_{j=0}^{3} b_j - \sum_{j=0}^{3} a_j \geq 0 \\ \sum_{j=0}^{3} a_j - \sum_{j=0}^{3} b_j \geq 0 \end{cases}$$

b) Linear Transformation Constraints

For linear transformation L with differential branch number B, given input nibble difference variables $(x_{in_0}, x_{in_1}, ..., x_{in_{n-1}})$ and output variables $(x_{out_0}, x_{out_1}, ..., x_{out_{n-1}})$, let d be a binary variable. When all input and output differences of L are zero, $d = 0$; otherwise, $d = 1$. The linear transformation constraints are:

$$\begin{cases} \sum_{i=0}^{n-1} x_{in_i} + \sum_{i=0}^{n-1} x_{out_i} - B \cdot d \geq 0 \\ B \cdot d - \sum_{i=0}^{n-1} x_{in_i} \geq 0 \\ B \cdot d - \sum_{i=0}^{n-1} x_{out_i} \geq 0 \end{cases}$$

LED-Specific MILP Model

LED' s round function includes five modules: AddRoundKey, AddConstants, SubCells, ShiftRows, and MixColumnsSerial, all operating on half-byte units. Among these, SubCells provides confusion while ShiftRows and MixColumnsSerial provide linear diffusion. We construct a half-byte-oriented automated MILP model for LED.

a) SubCells Transformation Constraints

Since LED' s key schedule performs no operations, we focus on the S-box substitution in the encryption process. As described in Section 2, for a bijective 4×4

S-box, a non-zero input nibble difference yields a non-zero output nibble difference, indicating an active S-box (represented as 1). The inequality constraint is:

$$x_{in_i} - x_{out_i} = 0$$

b) ShiftRows and MixColumnsSerial Constraints
In MixColumnsSerial, an n-order MDS matrix has maximum branch number $n+1$ (branch number theoretically bounds resistance to differential and linear attacks; larger branch numbers indicate better diffusion). LED' s fixed mixing matrix is a 4-order MDS matrix, achieving optimal linear diffusion. Thus, LED' s differential and linear branch numbers are both 5. Let d be a binary variable.

ShiftRows only performs cyclic shifts on nibbles without changing difference values, generating no new variables:

$$x_{out} = x_{in}$$

MixColumnsSerial involves matrix multiplication over $GF(2^4)$, where each column' s elements change, producing new difference variables.

C Implementation of MILP Constraints
The C code for MixColumnsSerial constraints:

```
void MixColumnsSerial(int a[4][4]) {
    for(i = 0; i < 4; i++) {
        int state[4];
        for(j = 1; j < 4; i++) {
            for(i = 0; i < 4; i++) {
                state[i] = a[j][(i + j) % 4];
            }
            for(i = 0; i < 4; i++) {
                a[j][i] = state[i];
            }
        }
    }
}
```

The C code for ShiftRows constraints:

```
for(j = 0; j < 4; j++) {
    printf("x%j +", a[j][i]);
    for(j = 0; j < 3; j++) {
        printf("x%j +", next + j);
    }
    printf("x%j - 5 d%j >= 0\n", next+3, dummy);
    for(j = 0; j < 4; j++) {
        printf("d%j - x%j >= 0\n", dummy, a[j][i]);
        printf("d%j - x%j >= 0\n", dummy, a[j][i] = next++);
```

```
    }
    dummy++;
}
```

LED uses 4 rounds as a major operational step, providing sufficient confusion and diffusion. We analyze one 4-round operation' s MILP constraints, with the confusion and diffusion pattern shown in equation (14).

Main MILP Module C Implementation for LED's 4-Round Operation:

```
int main() {
    int a[4][4];
    for (j = 0; j < 4; j++) {
        for (i = 0; i < 4; i++) {
            a[j][i] = next++;
        }
    }
    printf("Minimize\n");
    /* Output objective function */
    for (j = 0; j < ROUNDS*16-1; j++) {
        printf("x%j + ", j); /* ROUNDS is the round number */
    }
    printf("x%j\n\n", ROUNDS*16-1);
    printf("Subject To\n");
    /* Round function constraints */
    for (r = 0; r < ROUNDS; r++) {
        ShiftRows(a);
        MixColumnsSerial(a);
    }
    /* At least one S-box must be active */
    for (j = 0; j < ROUNDS*16-1; j++) {
        printf("x%j + ", j);
    }
    printf("x%j >= 1\n\n", ROUNDS*16-1);
    printf("Binary\n");
    /* Variable constraints */
    for (j = 0; j < 16; j++) {
        printf("x%j\n", j);
    }
    for (j = 0; j < dummy; j++) {
        printf("d%j\n", j);
    }
    printf("End\n");
    return 0;
}
```

4 MILP Model Solution for Active S-Boxes in LED

By constructing LED' s half-byte automated MILP model and implementing it in C, we analyze each round of the 4-round operation. The 4-round constraint inequalities are as follows:

Round 1 constraints:

$$\begin{cases} x_0 + x_5 + x_{10} + x_{15} + x_{16} + x_{17} + x_{18} + x_{19} - 5d_0 \geq 0 \\ d_0 - x_i \geq 0, \quad i \in \{0, 5, 10, 15, 16, 17, 18, 19\} \end{cases}$$

Round 2 constraints:

$$\begin{cases} x_{16} + x_{21} + x_{26} + x_{31} + x_{32} + x_{33} + x_{34} + x_{35} - 5d_1 \geq 0 \\ d_1 - x_i \geq 0, \quad i \in \{16, 21, 26, 31, 32, 33, 34, 35\} \end{cases}$$

Round 3 constraints:

$$\begin{cases} x_{32} + x_{37} + x_{42} + x_{47} + x_{48} + x_{49} + x_{50} + x_{51} - 5d_2 \geq 0 \\ d_2 - x_i \geq 0, \quad i \in \{32, 37, 42, 47, 48, 49, 50, 51\} \end{cases}$$

Round 4 constraints:

$$\begin{cases} x_{48} + x_{53} + x_{58} + x_{63} + x_{64} + x_{65} + x_{66} + x_{67} - 5d_3 \geq 0 \\ d_3 - x_i \geq 0, \quad i \in \{48, 53, 58, 63, 64, 65, 66, 67\} \end{cases}$$

Based on these 4-round constraints, we solve for the active S-boxes in the first 4 rounds. In the MILP model, each round' s differential characteristic must contain at least one active S-box.

Analysis Process:
In Round 1, with one nibble difference variable $x_0 \neq 0$, after SubCells transformation (LED' s S-box is bijective), a non-zero input difference yields a non-zero output difference, resulting in 1 active 4×4 S-box. After ShiftRows and MixColumnsSerial, the nibble difference variables propagate from x_0 to $\{x_{16}, x_{17}, x_{18}, x_{19}\}$. According to the constraints, when $x_{16}, x_{17}, x_{18}, x_{19}$ are non-zero, variable $d_0 \neq 0$. To satisfy the constraints, we set $x_{16}, x_{17}, x_{18}, x_{19} \neq 0$ while other nibble differences remain zero.

In Round 2, with four difference variables $\{x_{16}, x_{17}, x_{18}, x_{19}\}$ non-zero, SubCells yields 4 active S-boxes. After linear layers, variables propagate to $\{x_{32}, x_{33}, x_{34}, x_{35}\}$. The constraints require $d_1 \neq 0$ when these four variables are non-zero.

In Round 3, with 16 difference variables non-zero from Round 2, SubCells produces 16 active S-boxes. Propagation through linear layers spreads to $\{x_{48}, x_{49}, x_{50}, x_{51}\}$, requiring $d_2 \neq 0$.

In Round 4, with four variables $\{x_{48}, x_{49}, x_{50}, x_{51}\}$ non-zero, SubCells yields 4 active S-boxes. After linear transformations, variables become $\{x_{64}, x_{65}, x_{66}, x_{67}\}$. To satisfy Round 4 constraints, we can set $x_{64} \neq 0$ while others are zero.

After Round 4, the pattern returns to a single non-zero nibble difference, making Round 5 identical to Round 1. Thus, LED exhibits periodic behavior every 4 rounds. By calculating active S-boxes for the first 4 rounds (1+4+16+4 = 25), we can precisely extrapolate the full-round count.

** MILP Model for Differential Cryptanalysis of LED**

Rounds	Nibble Variables	Constraints	Active S-boxes
1	20	5	1
2	36	10	5
3	52	15	21
4	68	20	25
...	...	...	...
14	228	70	125

** The Number of Active S-Boxes in LED-64**

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Count	1	5	21	25	26	30	46	50	51	55	71	75	76	80	96	100

Round	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Count	101	105	121	125	126	130	146	150	151	155	171	175	176	180	196	200

** The Number of Active S-Boxes in LED-128**

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Count	1	5	21	25	26	30	46	50	51	55	71	75	76	80	96	100

Round	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Count	101	105	121	125	126	130	146	150	151	155	171	175	176	180	196	200

Round	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Count	201	205	221	225	226	230	246	250	251	255	271	275	276	280	296	300

The analysis shows that 4-round encryption contains at least 25 active S-boxes, LED-64 has at least 200 active S-boxes in full rounds, and LED-128 has at least 300. These results match the theoretical values provided by LED' s designers, validating our half-byte MILP model.

LED' s S-box has differential probability 2^{-2} and linear probability 2^{-2}. Using the Piling-Up Lemma[10], LED-64' s maximum differential characteristic probability is $2^{-2\times200} = 2^{-400}$, and maximum linear approximation probability is $2^{-2\times200} = 2^{-400}$. For LED-128, the maximum differential characteristic probability is $2^{-2\times300} = 2^{-600}$, and maximum linear approximation probability is $2^{-2\times300} = 2^{-600}$. These calculations prove LED' s strong resistance to differential and linear attacks.

---

5 Conclusion

This paper proposes a half-byte-oriented MILP model for automatically searching differential characteristics and linear approximations of cryptographic algorithms, implemented in C based on LED's structure. The model efficiently solves for active S-boxes using fewer variables and constraints, precisely determining that LED' s 4-round operation contains at least 25 active S-boxes, LED-64 has at least 200, and LED-128 has at least 300. These results match the designers' theoretical values, confirming the model' s correctness. LED-64' s maximum differential characteristic probability is 2^{-400} and maximum linear approximation probability is 2^{-400}; LED-128' s are 2^{-600} and 2^{-600}, respectively. Our analysis proves LED can resist differential and linear attacks.

Future work will apply this half-byte MILP model to more block ciphers to develop a generalized, efficient automated MILP framework.

References

[1] Bogdanov A, Knudsen L R, Leander G, et al. PRESENT: an ultra-lightweight block cipher [C]//Proc of International Workshop on Cryptographic Hardware and Embedded Systems. Berlin: Springer, 2007: 450-466.

[2] Guo Jian, Peyrin T, Poschmann A, et al. The LED block cipher [C]//Proc of International Conference on Cryptographic Hardware and Embedded Systems. Berlin: Springer-Verlag, 2011: 326-341.

[3] Wu Wenling, Zhang Lei. LBlock: a lightweight block cipher [C]//Applied Cryptography and Network Security. Berlin: Springer, 2011: 327-344.

[4] Gong Zheng, Nikova S, Law Y W. KLEIN: a new family of lightweight block ciphers [C]//Proc of International Conference on RFID Security and Privacy. Springer-Verlag, 2011: 1-18.

[5] Zhang Wentao, Bao Zhenzhen, Lin Dongdai, et al. RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms [J]. Science China, 2015, 58(12): 122103-122103.

[6] Beierle C, Jean J, Kölbl S, et al. The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS [C]//Advances in Cryptology. Berlin: Springer, 2016: 123-153.

[7] Li Lang, Liu Botao, Zhou Yimeng, et al. SFN: a new lightweight block cipher [J]. Microprocessors & Microsystems, 2018, 60: 138-150.

[8] Wang Ding, Li Wenting, Wang Ping. Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks [J]. IEEE Trans on Industrial Informatics, 2018, 99: 1-10.

[9] Li Wei, Zhang Wenwen, Gu Dawu, et al. Impossible differential fault analysis on the LED lightweight cryptosystem in the vehicular Ad-hoc networks [J]. IEEE Trans on Dependable & Secure Computing, 2016, 13(1): 84-92.

[10] Wang Ding, Cheng Haibo, He Debiao, et al. On the challenges in designing identity-based privacy-preserving authentication schemes for mobile devices [J]. IEEE Systems Journal, 2018, 21(1): 916-925.

[11] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems [J]. Journal of Cryptology, 1991, 4(1): 3-72.

[12] Matsui M. Linear Cryptanalysis Method for DES Cipher [C]//Proc of Eurocrypt-Advances in Cryptology. 1993, 765: 386-397.

[13] Mouha N, Wang Qingju, Gu Dawu, et al. Differential and linear cryptanalysis using mixed-integer linear programming [C]//Proc of International Conference on Information Security and Cryptology. Springer-Verlag, 2011: 57-76.

[14] Sun Siwei, Hu Lei, Wang Peng, et al. Automatic security evaluation and (related-key) differential characteristic search: application to SIMON, PRESENT, LBlock, DES(L) and other bit-oriented block ciphers [C]//Advances in Cryptology–ASIACRYPT. Berlin: Springer, 2014: 158-178.

[15] Fu Kai, Wang Meiqin, Guo Yinghua, et al. MILP-based automatic search algorithms for differential and linear trails for speck [C]//Fast Software Encryption. Berlin: Springer, 2016: 268-288.

[16] Beaulieu R, Treatman-Clark S, Shors D, et al. The SIMON and SPECK lightweight block ciphers [C]//Proc of Design Automation Conference. IEEE, 2015: 1-6.

[17] Yin Jun, Ma Chuyan, Song Jian, et al. Security Analysis of lightweight block cipher ESF [J]. Journal of Computer Research and Development, 2017, 54(10): 2224-2231.

[18] Yin Jun, Song Jian, Zeng Guang, et al. Related-key differential attack on lightweight block cipher ESF [J]. Journal of Cryptologic Research, 2017, 4(4):

333-344.

[19] Liu Xuan, Zhang Wenying, Liu Xiangzhong, et al. Eight-sided fortress: a lightweight block cipher [J]. Journal of China Universities of Posts and Telecommunications, 2014, 21(1): 104-108.

[20] Li Wei, Gu Dawu, Zhao Chen, et al. Security analysis of the LED lightweight cipher in the Internet of things [J]. Chinese Journal of Computers, 2012, 35(3): 434-445.

Note: Figure translations are in progress. See original paper for figures.