Hindawi
Security and Communication Networks
Volume 2018, Article ID 3780407, 11 pages
https://doi.org/10.1155/2018/3780407

Research Article

Differential Cryptanalysis on Block Cipher Skinny with MILP Program

Pei Zhang[1,2] **and Wenying Zhang**[1,2]

[1] *School of Information Science and Engineering, Shandong Normal University, Jinan, Shandong 250358, China*
[2] *Cyberspace Security Lab, Jinan, Shandong 250358, China*

Correspondence should be addressed to Wenying Zhang; wenyingzh@sohu.com

Received 27 July 2018; Revised 11 September 2018; Accepted 16 September 2018; Published 4 October 2018

Guest Editor: Zhaoqing Pan

With the widespread use of RFID technology and the rapid development of Internet of Things, the research of lightweight block cipher has become one of the hot issues in cryptography research. In recent years, lightweight block ciphers have emerged and are widely used, and their security is also crucial. Skinny-64/192 can be used to protect data security such as the applications of wireless multimedia and wireless sensor networks. In this paper, we use the new method to verify the security of Skinny-64/192. The method is called mixed-integer linear programming (MILP) which can characterize precisely the linear operation and nonlinear operation in a round function. By applying MILP program, we can automatically find a 11-round differential characteristic for Skinny-64/192 with the minimum number of active s-boxes. The probability of differential trail is 2^{-147}, that is, far greater than 2^{-192} which is the probability of success for an exhaustive search. In addition, comparing this method with the one proposed by Sun et al., we also have a great improvement; that is, no new variables will be added in ShiftRows operation. It can reduce greatly the number of variables and improve the running speed of the computer. Besides, the experimental result proves that Skinny-64/192 is safe on 11-round differential analysis and validates the effectiveness of the MILP method.

1. Introduction

Nowadays, with the development of big data and artificial intelligence technology, data security problem becomes increasingly serious. The problem of data security exists in the whole life cycle of data, from data collection and transfer to data usage, with the focus on data confidentiality [1, 2], integrity, and availability. Due to the continuous occurrence and intensification of data leakage events on the Internet, the confidentiality of data is particularly important. The confidentiality of data, on the one hand, is the direct protection of data and, on the other hand, is providing further privacy protection on the basis of leaks. From a technical point of view, to ensure the confidentiality of data without hindering the availability of data, the general approach is to encrypt the data by encryption algorithms.

The traditional encryption algorithms are DES [3] and AES [4]; they are widely used in the field of hardware and software. The hardware applications include data security in radio frequency IC card and encryption of hard disk data. And the software applications include voice, video information encryption, and database data encryption. The most classic applications are security applications on wireless network: one is the IEEE 803.11 protocol for WLAN and the IEEE 803.16 protocol for WMAN, and the other is the ZigBee protocol. The application of these encryption algorithms ensures the security of data effectively.

At present, the traditional cryptographic algorithm for encrypting wireless multimedia data faces a lot of challenges, such as fast resource consumption, high implementation cost, and other disadvantages. In this context, lightweight block cipher emerged. Compared with traditional cryptographic algorithms, lightweight block ciphers run faster and have lower resource consumption and implementation costs while ensuring data security. They are more suitable for radio frequency identification (RFID) tags, wireless sensor network

(WSN) [5], wireless multimedia, and other micro devices. In recent years, many lightweight block ciphers have been proposed, such as PRESENT [6], PRINCE [7], Midori [8], and Skinny [9, 10], many of which have been defined as ISO standards and widely used in various fields.

Skinny is a family of tweakable lightweight block ciphers proposed by Beierle et al. at CRYPTO in 2016. It is a Substitution Permutation Network (SPN) [11, 12] structure. It supports two block lengths n=64 or 128 bits and for each of them, the tweakey t can be either n, 2n, or 3n. Skinny has been analyzed by many methods since it was proposed, such as impossible differential cryptanalysis [10] and related-key impossible differential attack [13].

Differential cryptanalysis [3, 13, 14] was firstly introduced by Biham and Shamir to analyze DES block cipher in 1990. Differential analysis is one of the most effective attack methods in block ciphers. Differential analysis is a selective plaintext attack, and its basic idea is to study the probability of differential propagation of specific plaintext differential values in the encryption process. We separate the block cipher from the permutation area and then carry out the key recovery attack on this basis. In other words, we find a high probability differential trail. Finally, by adding several rounds before and after the differential characteristic, guessing Round-keys used in these rounds, encrypting plaintexts, and decrypting ciphertexts, we can determine the right key of block cipher.

Mixed-integer linear programming (MILP) [14, 15] is a mathematical optimization or feasibility scheme, where some or all variables are limited to integers. In many cases, the term refers to an integer linear program (ILP), which is linear in terms of objective function and constraint except for the integer constraint. MILP is frequently used in business and economics to solve problems of optimization.

In [14], Mouha et al. proposed an automatic search method based on MILP. However, the drawback of this method is that the proposed constraint cannot describe the trail of the differential propagation of the linear diffusion layer. Besides, Sun et al. [15] perfected the MILP method, then combined the MILP method and differential analysis into PRESENT, and finally obtained satisfactory experimental results. In this paper, we apply the new MILP method to obtain a lower bound on the number of active s-boxes for differential cryptanalysis. Then, we use the maximum differential probability of the s-boxes to derive an upper bound for the probability of the best characteristic.

The Organizational of the Paper. The paper is organized as follows: In Section 2, we introduce some basic properties and definitions and describe how to construct a MILP program by constraints to get the minimum number of active s-boxes and the corresponding trail of differential propagation. In Section 3, the MILP program is constructed to search the differential trail of Skinny-64/192. Through specific instances, the optimal solution of the minimum number of active s-boxes is obtained for 11-round differential characteristic of Skinny-64/192. We conclude the paper and look forward to the future work in Section 4. The auxiliary materials are given in Appendix.

Our Contributions. In this paper, we apply a method proposed recently for obtaining a high probability of differential characteristic in Skinny-64/192 called MILP method, which is used to search the minimum number of active s-boxes automatically. Its minimum number of active s-boxes is 54 of 11-round differential characteristic. The number of active s-boxes is one of the commonly used methods for evaluating the security of symmetric key encryption schemes against differential attack. As far as we know, this is the first time to combine differential analysis with MILP method to be applied to Skinny-64/192.

MILP can characterize accurately the linear operation and nonlinear operation in the round function. Then a high probability of 11-round differential characteristic is automatically searched. The probability of differential trail is 2^{-147}, that is, far greater than 2^{-192} which is the probability of success for an exhaustive search. This experimental result proves that 11-round differential analysis of Skinny-64/192 is safe, which can provide a safe reference for data encryption on wireless devices. At the same time, we also verified the effectiveness of MILP method through this experiment. In addition, we also have an improvement on MILP program; that is, no new variables can be added in ShiftRows operation, which can reduce the number of total variables greatly and improve the running speed of the computer.

2. The Minimum Number of Active S-Boxes for Differential Cryptanalysis

In this section, we will describe how to construct the MILP program to calculate the number of active s-boxes for differential analysis. This requires an accurate description of the nonlinear layers and linear layers in order to ensure the number of active s-boxes is minimum. In general, if a large number of active s-boxes exist, which indicates that the differential diffusion is fast, this suggests that the cryptographic algorithm is not vulnerable to attacks and has a high security. The core theorem for constructing the MILP program will be described in detail in the following.

2.1. Differential Cryptanalysis

Definition 1. For every input bit-level difference, a 0-1 variable x_i is introduced such that $x_i = 1$ if and only if the difference at this bit is nonzero, as

$$x_i = \begin{cases} 0, & \textit{the differences do not exist}, \\ 1, & \text{otherwise}. \end{cases} \tag{1}$$

2.2. Constraints for Nonlinear and Linear Operation. Generally, the SPN-structured encryption algorithm consists of s-box, XOR, ShiftRows, and MixColumn operations. In this subsection, we describe these four basic operations by constraints. Based on this, we can construct an r-round inequality model for a specific encryption algorithm. This model can describe the trail of differential propagation accurately. Then by selecting the appropriate objective function, we can convert this model into a MILP program, using this MILP program to search automatically for the objective function.

Constraints Describing the S-Box Operation [15]. Suppose $(x_{i_0},\ldots,x_{i_{w-1}})$ and $(y_{j_0},\ldots,y_{j_{v-1}})$ are the input and output bit-level differences of a w×v s-box marked by S_t. Firstly, to ensure that $S_t = 1$ holds if and only if $(x_{i_0},\ldots,x_{i_{w-1}})$ are not all zero, we require the following.

$$\begin{gathered} S_t - x_{i_k} \geq 0, \quad k \in \{0,\ldots,w-1\} \\ x_{i_0} + x_{i_1} + \cdots + x_{i_w} - S_t \geq 0 \end{gathered} \tag{2}$$

For bijective s-boxes, nonzero input difference must result in nonzero output difference and vice versa.

$$\begin{aligned} & wy_{j_0} + wy_{j_1} + \cdots + wy_{j_{v-1}} - \left(x_{i_0} + x_{i_1} + \cdots + x_{i_{w-1}}\right) \\ & \quad \geq 0 \\ & vx_{j_0} + vx_{j_1} + \cdots + vx_{j_{w-1}} - \left(y_{i_0} + y_{i_1} + \cdots + y_{i_{v-1}}\right) \geq 0 \end{aligned} \tag{3}$$

Constraints Describing the XOR Operation. The bit-wise input difference is (x_i, x_{i+1}) and the corresponding bit-wise output difference is y for the XOR operation. The following linear constraints describe the relation between the input and output difference.

$$\begin{aligned} x_i + x_{i+1} - y &\geq 0 \\ x_i - x_{i+1} + y &\geq 0 \\ -x_i + x_{i+1} + y &\geq 0 \\ x_i + x_{i+1} + y &\leq 2 \end{aligned} \tag{4}$$

Constraints Describing the ShiftRows or ShuffleCell Operation. For every ShuffleCell operation, its input difference $(y_0, y_1, \ldots, y_{i-1}, y_i)$ and output difference $(z_0, z_1, \ldots, z_{i-1}, z_i)$ are based on bit. If $(z_0 = y_2, z_1 = y_i, \ldots, z_{i-1} = y_0, z_i = y_1)$, the constraints include the following.

$$\begin{aligned} z_0 - y_2 &= 0 \\ z_1 - y_i &= 0 \\ &\vdots \\ z_{i-1} - y_0 &= 0 \\ z_i - y_1 &= 0 \end{aligned} \tag{5}$$

Constraints Describing the MixColumn Operation. Let $(z_0, z_1, \ldots, z_{j-1}, z_j)$ and $(x_0, x_1, \ldots, x_{j-1}, x_j)$ be the input and output bit-wise differences for the MixColumn operation. Suppose $x_i = z_{j-2} + z_{j-1} + z_j$; it is essential to set an intermediate variable u and let $u = z_{j-2} + z_{j-1}$ to get $x_i = u + z_j$, so the constraints can be described as follows.

$$\begin{aligned} z_{j-2} + z_{j-1} - u &\geq 0 \\ z_{j-2} - z_{j-1} + u &\geq 0 \\ -z_{j-2} + z_{j-1} + u &\geq 0 \\ z_{j-2} + z_{j-1} + u &\leq 2 \\ u + z_j - x_0 &\geq 0 \\ u - z_j + x_0 &\geq 0 \\ -u + z_j + x_0 &\geq 0 \\ u + z_j + x_0 &\leq 2 \end{aligned} \tag{6}$$

Definition 2 (the objective function [16]). Some notations for differential are used in the model; e.g., S_j denotes the activity of an s-box and the objective function is as follows.

$$\begin{aligned} & \min \quad \sum_j S_j \\ & \text{s.t.} \quad S_j = \begin{cases} 0, & \textit{the sbox is not active}, \\ 1, & \textit{othersise}. \end{cases} \end{aligned} \tag{7}$$

The smaller the number of active s-boxes is, the slower the differential diffusion is. This illustrates that the encryption algorithm will be attacked in more rounds, and this will threaten its security.

3. Constructing the MILP Program to Calculate the Minimum Number of Active S-Boxes of Skinny-64/192

It is well known that the security of an encryption algorithm must be evaluated before being put into use. In this section, we use the newly proposed MILP method to evaluate the security of Skinny-64/192 on differential analysis. This is the first time to combine differential analysis with MILP method to be applied to Skinny-64/192; the method is called MILP program. The MILP program can automatically obtain the minimum number of active s-boxes on the 11-round differential analysis. And MILP program consists of inequalities which can describe precisely the linear and nonlinear operation.

3.1. Description of Skinny-64/192. Skinny is a family of tweakable lightweight block ciphers proposed by Beierle et

Table 1: S-box of Skinny-64/192.

x	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
s(x)	0xc	0x6	0x9	0x0	0x1	0xa	0x2	0xb	0x3	0x8	0x5	0xd	0x4	0xe	0x7	0xf

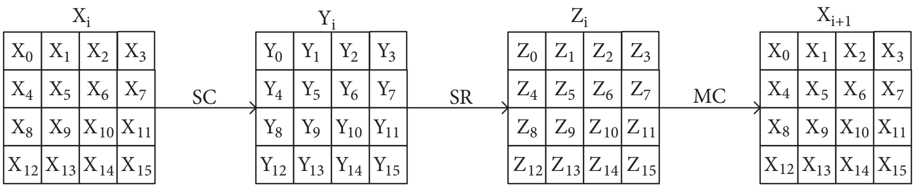

X_i
X_0 X_1 X_2 X_3
X_4 X_5 X_6 X_7
X_8 X_9 X_{10} X_{11}
X_{12} X_{13} X_{14} X_{15}
SC
Y_i
Y_0 Y_1 Y_2 Y_3
Y_4 Y_5 Y_6 Y_7
Y_8 Y_9 Y_{10} Y_{11}
Y_{12} Y_{13} Y_{14} Y_{15}
SR
Z_i
Z_0 Z_1 Z_2 Z_3
Z_4 Z_5 Z_6 Z_7
Z_8 Z_9 Z_{10} Z_{11}
Z_{12} Z_{13} Z_{14} Z_{15}
MC
X_{i+1}
X_0 X_1 X_2 X_3
X_4 X_5 X_6 X_7
X_8 X_9 X_{10} X_{11}
X_{12} X_{13} X_{14} X_{15}

Figure 1: The i-th round function of Skinny-64/192.

al. at CRYPTO in 2016. The specifications for Skinny was given in [9]. Skinny-64/192 provides 64-bit block length and 192-bit key length. We now give a short description of Skinny-64/192. Skinny-64/192 uses the SPN structure with Midori-64-like state. The state is arranged in a 4 × 4 matrix.

$$P = \begin{pmatrix} p_0 & p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 & p_7 \\ p_8 & p_9 & p_{10} & p_{11} \\ p_{12} & p_{13} & p_{14} & p_{15} \end{pmatrix} \qquad (8)$$

Every cell P_i is a nibble, $i \in 0, 1, \ldots, 15$.

The round function consists of SubCell, AddConstants, AddRoundTweakey, ShiftRows, and MixColumn. Since SubCell, ShiftRows, and MixColumn operations have an effect on differential diffusion, we only illustrate these operations in the paper. For more details, please refer to [9].

SubCells (SC). The 4×4 s-box defined in Table 1 is applied to each nibble in the state.

ShiftRows (SR). The rows of the state are rotated as in AES but to the right, i.e., the cell permutation is specified as follows.

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 & 1 & 2 & 3 \\ 7 & 4 & 5 & 6 \\ 10 & 11 & 8 & 9 \\ 13 & 14 & 15 & 12 \end{pmatrix} \qquad (9)$$

MixColumn (MC). Each column in the state is multiplied by a binary matrix MC. MC is given as follows.

$$MC = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \qquad (10)$$

Tweakey Schedule. Skinny-64/192 tweakey is updated through tweakey schedule. K = TK1 ‖ TK2 ‖ TK3 = 192 bits, and TK1 = TK2 = TK3 = 64 bits which be permuted by P_T. Then, each cell in the first and second rows of TK2, TK3 is updated using LFSR operations shown in [10].

In Figure 1, X_i, Y_i, Z_i represent the i-th round state, respectively. X_j, Y_j, Z_j represent a nibble of the state, respectively. Let $(x_0, x_1, \ldots, x_{63})$ and $(z_0, z_1, \ldots, z_{63})$ be the input and output bit-wise differences in a round for Skinny-64/192, and $X_j, Y_j, Z_j,\ j \in 0, 1, \ldots, 15$, and $x_k, y_k, z_k,\ k \in 0, 1, \ldots, 63$.

From the overall design of Skinny-64/192, its structure is compact and has the advantages of low delay, high throughput, and low number of gate circuits in hardware implementation. Therefore, Skinny-64/192 is more suitable for wireless multimedia and other micro device applications. Now, we apply the new MILP method to get the lower limit of the number of active s-boxes for the differential analysis of Skinny-64/192. Then, we use the maximum difference probability of the s-box to derive the upper bound of the best characteristic probability. Finally, the experimental results are used to determine whether Skinny-64/192 is safe on differential analysis.

3.2. Employing MILP's Method for Specific Operation

3.2.1. Compact Constraints for DDT of S-Box. In Skinny-64/192, combined with the Table 2, $y = s(x)$, $(y_0, y_1, y_2, y_3) = s1(x_0, x_1, x_2, x_3)$, it is possible to list the following vectors according to the input differential of 0001: $[0, 0, 0, 1, 1, 0, 0, 0]$, $[0, 0, 0, 1, 1, 0, 0, 1]$, $[0, 0, 0, 1, 1, 0, 1, 0]$,

Table 2: The input and output differential distribution of Skinny-64/192's s-box.

Input difference (x_0, x_1, x_2, x_3)	Output difference (y_0, y_1, y_2, y_3)
0000	0000
0001	1000 1001 1010 1011
0010	0001 0011 0101 0110
0011	1000 1001 1010 1011 1100 1101 1110 1111
0100	0010 0110 0111 1011 1100 1101
0101	0010 0110 0111 1010 1100 1101
0110	0001 0011 0100 0111 1000 1010 1101 1110
0111	0001 0011 0100 0111 1001 1011 1100 1111
1000	0100 0101 1100 1101 1110 1111
1001	0100 0101 1100 1101 1110 1111
1010	0101 0110 1000 1001 1010 1011
1011	0001 0011 1100 1101 1110 1111
1100	0010 0110 0111 1000 1110 1111
1101	0010 0110 0111 1001 1110 1111
1110	0001 0011 0100 0111 1001 1011 1101 1110
1111	0001 0011 0100 0111 1000 1010 1100 1111

$[0, 0, 0, 1, 1, 0, 1, 1]$. Similarly, we can get all the differential vectors, so we get the input of SageMath [17].

$$\text{points} = [[0, 0, 0, 0, 0, 0, 0, 0],$$

$$[0, 0, 0, 1, 1, 0, 0, 0], [0, 0, 0, 1, 1, 0, 0, 1],$$

$$[0, 0, 0, 1, 1, 0, 1, 0], [0, 0, 0, 1, 1, 0, 1, 1],$$

$$[0, 0, 1, 0, 0, 0, 0, 1], [0, 0, 1, 0, 0, 0, 1, 1],$$

$$[0, 0, 1, 0, 0, 1, 0, 1], [0, 0, 1, 0, 0, 1, 1, 0],$$

$$[0, 0, 1, 1, 1, 0, 0, 0], [0, 0, 1, 1, 1, 0, 0, 1],$$

$$[0, 0, 1, 1, 1, 0, 1, 0], [0, 0, 1, 1, 1, 0, 1, 1],$$

$$[0, 0, 1, 1, 1, 1, 0, 0], [0, 0, 1, 1, 1, 1, 0, 1],$$

$$[0, 0, 1, 1, 1, 1, 1, 0], [0, 0, 1, 1, 1, 1, 1, 1],$$

$$[0, 1, 0, 0, 0, 0, 1, 0], [0, 1, 0, 0, 0, 1, 1, 0],$$

$$[0, 1, 0, 0, 0, 1, 1, 1], [0, 1, 0, 0, 1, 0, 1, 1],$$

$$[0, 1, 0, 0, 1, 1, 0, 0], [0, 1, 0, 0, 1, 1, 0, 1],$$

$$[0, 1, 0, 1, 0, 0, 1, 0], [0, 1, 0, 1, 0, 1, 1, 0],$$

$$[0, 1, 0, 1, 0, 1, 1, 1], [0, 1, 0, 1, 1, 0, 1, 0],$$

$$[0, 1, 0, 1, 1, 1, 0, 0], [0, 1, 0, 1, 1, 1, 0, 1],$$

$$[0, 1, 1, 0, 0, 0, 0, 1], [0, 1, 1, 0, 0, 0, 1, 1],$$

$$[0, 1, 1, 0, 0, 1, 0, 0], [0, 1, 1, 0, 0, 1, 1, 1],$$

$$[0, 1, 1, 0, 1, 0, 0, 0], [0, 1, 1, 0, 1, 0, 1, 0],$$

$$[0, 1, 1, 0, 1, 1, 0, 1], [0, 1, 1, 0, 1, 1, 1, 0],$$

$$[0, 1, 1, 1, 0, 0, 0, 1], [0, 1, 1, 1, 0, 0, 1, 1],$$

$$[0, 1, 1, 1, 0, 1, 0, 0], [0, 1, 1, 1, 0, 1, 1, 1],$$

$$[0, 1, 1, 1, 1, 0, 0, 1], [0, 1, 1, 1, 1, 0, 1, 1],$$

$$[0, 1, 1, 1, 1, 1, 0, 0], [0, 1, 1, 1, 1, 1, 1, 1],$$

$$[1, 0, 0, 0, 0, 1, 0, 0], [1, 0, 0, 0, 0, 1, 0, 1],$$

$$[1, 0, 0, 0, 1, 1, 0, 0], [1, 0, 0, 0, 1, 1, 0, 1],$$

$$[1, 0, 0, 0, 1, 1, 1, 0], [1, 0, 0, 0, 1, 1, 1, 1],$$

$$[1, 0, 0, 1, 0, 1, 0, 0], [1, 0, 0, 1, 0, 1, 0, 1],$$

$$[1, 0, 0, 1, 1, 1, 0, 0], [1, 0, 0, 1, 1, 1, 0, 1],$$

$$[1, 0, 0, 1, 1, 1, 1, 0], [1, 0, 0, 1, 1, 1, 1, 1],$$

$$[1, 0, 1, 0, 0, 1, 0, 1], [1, 0, 1, 0, 0, 1, 1, 0],$$

$$[1, 0, 1, 0, 1, 0, 0, 0], [1, 0, 1, 0, 1, 0, 0, 1],$$

$$[1, 0, 1, 0, 1, 0, 1, 0], [1, 0, 1, 0, 1, 0, 1, 1],$$

$$[1, 0, 1, 1, 0, 0, 0, 1], [1, 0, 1, 1, 0, 0, 1, 1],$$

$$[1, 0, 1, 1, 1, 1, 0, 0], [1, 0, 1, 1, 1, 1, 0, 1,],$$

$$[1, 0, 1, 1, 1, 1, 1, 0], [1, 0, 1, 1, 1, 1, 1, 1],$$

$$[1, 1, 0, 0, 0, 0, 1, 0], [1, 1, 0, 0, 0, 1, 1, 0],$$

$$[1, 1, 0, 0, 0, 1, 1, 1], [1, 1, 0, 0, 1, 0, 0, 0],$$

$$\begin{gathered}
[1,1,0,0,1,1,1,0],[1,1,0,0,1,1,1,1],\\
[1,1,0,1,0,0,1,0],[1,1,0,1,0,1,1,0],\\
[1,1,0,1,0,1,1,1],[1,1,0,1,1,0,0,1],\\
[1,1,0,1,1,1,1,0],[1,1,0,1,1,1,1,1],\\
[1,1,1,0,0,0,0,1],[1,1,1,0,0,0,1,1],\\
[1,1,1,0,0,1,0,0],[1,1,1,0,0,1,1,1],\\
[1,1,1,0,1,0,0,1],[1,1,1,0,1,0,1,1],\\
[1,1,1,0,1,1,0,1,],[1,1,1,0,1,1,1,0],\\
[1,1,1,1,0,0,0,1],[1,1,1,1,0,0,1,1],\\
[1,1,1,1,0,1,0,0],[1,1,1,1,0,1,1,1],\\
[1,1,1,1,1,0,0,0],[1,1,1,1,1,0,1,0],\\
[1,1,1,1,1,1,0,0],[1,1,1,1,1,1,1,1]]
\end{gathered} \tag{11}$$

Running SageMath will output 202 inequalities. Then, the redundant inequalities are eliminated through a specific streamlined procedure (Appendix). Finally, the s1-box can be accurately characterized with 24 inequalities. The inequality of describing s1 is shown as follows.

$$\begin{gathered}
-2x_0 + 3x_1 - 3x_2 - 2x_3 + 5y_0 + 4y_1 + y_2 + 7y_3 >= 0\\
-x_0 + x_1 + 2x_2 + x_3 + y_0 + 3y_1 - 2y_3 >= 0\\
4x_0 + 3x_1 + 2x_2 + 3x_3 - y_0 - y_1 - y_2 - y_3 >= 0\\
2x_0 - x_1 + 2x_2 + 3y_0 - y_1 + 3y_2 - y_3 >= 0\\
x_0 + 3x_1 + x_2 - 2x_3 + 2y_0 - y_1 - 2y_2 >= -2\\
-3x_0 + 2x_1 + x_2 - 2x_3 - y_0 + 3y_1 + y_3 >= -3\\
x_1 - 2x_2 + 2x_3 - y_0 - 2y_1 + y_2 + y_3 >= -3\\
-2x_0 - 3x_1 + 2x_2 + x_3 + y_0 - y_1 + 3y_2 - y_3 >= -4\\
-x_1 - 2x_2 - x_3 + y_0 - y_1 - 2y_2 + 2y_3 >= -5\\
-x_1 - x_2 - x_3 + y_0 - 2y_1 + 2y_2 - 2y_3 >= -5\\
2x_0 + x_1 + 3x_2 + 4x_3 - 3y_0 + 2y_1 - y_2 + 3y_3 >= 0\\
x_0 - 2x_1 + 2x_2 - 2x_3 - y_0 + y_1 - y_2 - 2y_3 >= -6\\
x_1 - 2x_2 + 2x_3 - y_0 - 2y_1 - y_2 - y_3 >= -5\\
x_0 - x_1 - x_3 + y_0 + 2y_1 + 2y_2 + 2y_3 >= 0\\
x_0 - x_2 + x_3 - y_0 + y_1 - y_3 >= -2\\
-x_0 - x_2 - x_3 - y_0 + y_1 - y_3 >= -4\\
-x_0 - x_1 - x_2 + x_3 + y_1 + y_3 >= -2\\
3x_0 + x_1 + x_2 - 2x_3 + 2y_0 - y_1 + 2y_2 - y_3 >= -1\\
x_1 + x_2 + y_0 - y_2 >= 0\\
x_0 - x_1 - 2x_2 - x_3 + y_0 - 2y_2 + 2y_3 >= -4\\
x_0 + x_1 + x_2 + 2x_3 - 2y_0 + y_1 + y_2 >= 0\\
-x_0 - x_1 + x_2 - y_1 + y_2 >= -2\\
x_0 + x_1 + x_2 - y_1 >= 0\\
x_0 + x_2 - y_0 - y_1 - y_2 >= -2
\end{gathered} \tag{12}$$

A round of 16 s-boxes can be characterized by 384 linear inequalities accurately. $x_0, x_1, x_2, x_3 \in X_0, y_0, y_1, y_2, y_3 \in Y_0$. X_0, Y_0 represent 4 bits. X_0 and Y_0 are input and output of s1, respectively.

In contrast, it is simple to construct constrains for the ShiftRows operation of Skinny-64/192. Referring to (5), the ShiftRows operation can be characterized precisely by the next 64 constraint equations. $y_i \in Y,\ z_i \in Z,\ i \in 0, 1, \ldots, 63$.

$$\begin{gathered}
z_0 - y_0 = 0\\
z_1 - y_1 = 0\\
z_2 - y_2 = 0\\
z_3 - y_3 = 0\\
\vdots\\
\vdots\\
z_{60} - y_{48} = 0\\
z_{61} - y_{49} = 0\\
z_{62} - y_{50} = 0\\
z_{63} - y_{51} = 0
\end{gathered} \tag{13}$$

In the ShiftRows operation, comparing with the method in [15], we also make a great improvement in which no new variables will be added. It reduces the number of variables greatly and improves the running speed of the computer. In this case, we can reduce the use of 64 variables in one round. Therefore, the variable Z can be omitted.

3.2.2. Compact Constraints for the Linear Transform. In linear layer, MixColumn operations are the most difficult to be described utilizing the novel technique, but in this work, we can introduce an intermediate variable U to solve the problems. This operation is broken down into the following steps.

Table 3: The DDT of s-box for Skinny-64/192.

	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xa	0xb	0xc	0xd	0xe	0xf
0x0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1	0	0	0	0	0	0	0	0	4	4	4	4	0	0	0	0
0x2	0	4	0	4	0	4	4	0	0	0	0	0	0	0	0	0
0x3	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
0x4	0	0	4	0	0	0	2	2	0	0	0	4	2	2	0	0
0x5	0	0	4	0	0	0	2	2	0	0	4	0	2	2	0	0
0x6	0	2	0	2	2	0	0	2	2	0	2	0	0	2	2	0
0x7	0	2	0	2	2	0	0	2	0	2	0	2	2	0	0	2
0x8	0	0	0	0	4	4	0	0	0	0	0	0	2	2	2	2
0x9	0	0	0	0	4	4	0	0	0	0	0	0	2	2	2	2
0xa	0	0	0	0	0	4	4	0	2	2	2	2	0	0	0	0
0xb	0	4	0	4	0	0	0	0	0	0	0	0	2	2	2	2
0xc	0	0	4	0	0	0	2	2	4	0	0	0	0	0	2	2
0xd	0	0	4	0	0	0	2	2	0	4	0	0	0	0	2	2
0xe	0	2	0	2	2	0	0	2	0	2	0	2	0	2	2	0
0xf	0	2	0	2	2	0	0	2	2	0	2	0	2	0	0	2

Step 1. Convert the matrix $MC_{4\times 4}$ to $MC_{16\times 16}$ of Skinny-64/192.

$$MC = \begin{pmatrix} 1&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0 \\ 0&1&0&0&0&0&0&0&0&1&0&0&0&1&0&0 \\ 0&0&1&0&0&0&0&0&0&0&1&0&0&0&1&0 \\ 0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&1 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0 \\ 1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0 \end{pmatrix} \tag{14}$$

Step 2. After the MixColumn, we can get the value x_s of X_{i+1}, in which $s \in 64, 65, \ldots, 127$. For example $x_{64} = y_0 + y_{40} + y_{52}$.

Step 3. We introduce the intermediate variable U, $u_0 = y_{40} + y_{52}$, and then $x_{64} = y_0 + u_0$. Combining (4) and (6), the constraints between them can be expressed as follows.

$$\begin{aligned} y_{40} + y_{52} - u_0 &>= 0 \\ y_{40} - y_{52} + u_0 &>= 0 \\ -y_{40} + y_{52} + u_0 &>= 0 \\ y_{40} + y_{52} + u_0 &<= 2 \\ u_0 + y_0 - x_{64} &>= 0 \\ u_0 - y_0 + x_{64} &>= 0 \\ -u_0 + y_0 + x_{64} &>= 0 \\ u_0 + y_0 + x_{64} &<= 2 \end{aligned} \tag{15}$$

In a round, we need to use 16 intermediate variables, $u_i \in U$, $i \in 0, 1, \ldots, 15$.

3.3. *Calculate the Minimum Number of Active S-Boxes.*

In the MILP program, we must add a linear constraint to ensure that at least one s-box is active. The setting of objective function refers to (7). And all variables must be binary variable. In order to optimize the MILP model, we also need CPLEX [18] tool. Finally, we obtain the minimum number of active s-boxes which is 54 for 11-round differential analysis of Skinny-64/192. In Table 3, the

Table 4: 11-round differential trail.

Rounds	Input difference	Probability	Active s-boxes
1st	9000 0000 0000 0000	2^{-2}	1
2nd	5505 0000 0000 0000	2^{-8}	3
3rd	cc0c 00a0 0000 d000	2^{-13}	5
4th	6606 0060 0000 096f	2^{-21}	7
5th	aaff 8008 f000 a077	2^{-26}	10
6th	0505 0f40 bff0 c505	2^{-30}	10
7th	c00d 00d0 0d7d 2700	2^{-23}	8
8th	0f00 f000 0000 0196	2^{-13}	5
9th	0090 0400 7000 0000	2^{-9}	3
10th	0000 0000 0d00 0000	2^{-2}	1
11th	0000 0000 0000 0800	–	1

differential distribution table of s-box of Skinny-64/192 is presented.

First, the s1-box of Skinny-64/192 is set as an active s-box with the input difference of 1001(9). And the probability of obtaining the second round of input difference is 2^{-2}, and the number of active s-boxes is 3. By analogy, the output difference of the 11th round has an active s-box which is 1000(8). The total probability of the 11- round differential characteristic is 2^{-147}. The minimum number of the active s-boxes is 54 for 11 rounds of Skinny-64/192. The details are shown in Table 4.

According to Table 4, we can get a specific probability for each round of 11-round differential characteristic for Skinny-64/192 in Figure 2. The probability of differential trail is 2^{-147}, that is, far greater than 2^{-192} which is the probability of success for an exhaustive search.

The experimental result leads us to obtain the minimum number of active s-boxes which is 54 for the 11-round differential trail. Since the same number of rounds is attacked, the minimum active s-boxes number of the Skinny-64/192 is bigger than that of ENOCORO-128v2, PRESENR-80. This not only illustrates that Skinny-64/192 is relatively safe, but also can be implemented in hardware to protect the safety of data. The bigger the number of active s-boxes, the faster the differential diffusion; the security of cryptographic algorithm is relatively higher.

The MILP program corresponding to Skinny-64/192's 11-round differential trail consists of 7440 constraints and 1680 binary variables including 1520 continuous variables and 160 intermediate variables. Compared with Sun et al., our improvement reduced the use of 640 continuous variables in total and improved the speed of the computer. The experiments are implemented on a 64-bit operating system, Intel Core i7-7700 CPU @ 3.60GHz, with 16GB of RAM.

4. Conclusion

In this paper, a new result is obtained on the differential analysis of lightweight block cipher Skinny-64/192. We get a 11-round differential characteristic with minimum active s-boxes. The minimum number of active s-boxes is 54. The probability of 11-round differential trail is 2^{-147}, that is, far greater than 2^{-192} which is the probability of success for an exhaustive search.

The experimental result not only proves that Skinny-64/192 cannot resist 11-round differential analysis and validates the effectiveness of MILP method, but also has other important reference values. First, the lightweight block cipher Skinny-64/192 is relatively secure and can be used on wireless multimedia devices to protect data security. Second, Skinny-64/192 can resist 11-round differential analysis, so it can be used as a candidate encryption algorithm for differential privacy protection technology. Finally, by verifying the effectiveness of MILP method on differential analysis, the method can significantly reduce the workload of cryptanalysts. Besides, MILP method can be applied to more cryptanalysis, such as related-key differential analysis, impossible differential analysis, and related-key impossible differential analysis. We believe that there will be greater gains.

Appendix

The specific streamlined procedure to select certain number of inequalities (see Algorithm 1).

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

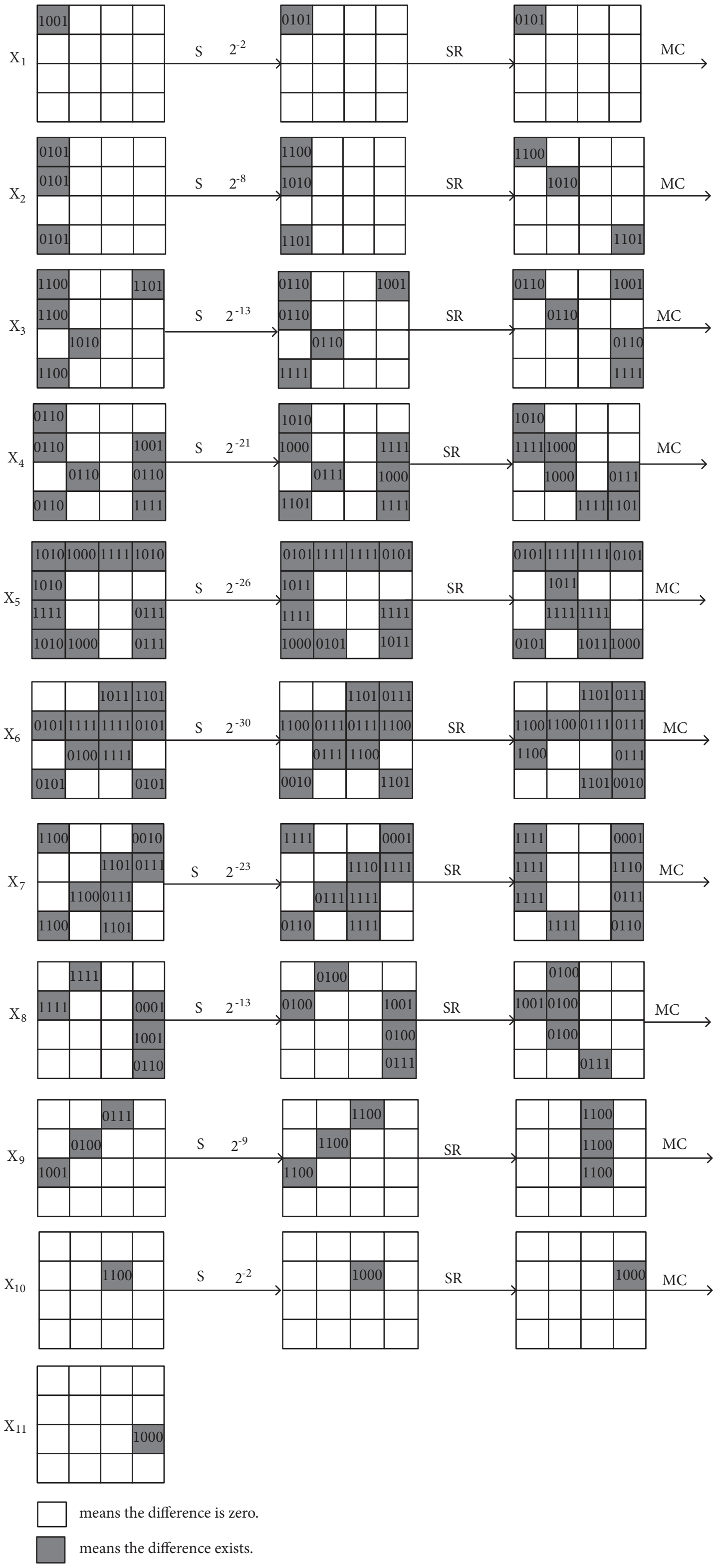


Figure 2: 11-round differential characteristic of Skinny-64/192.

```
#include <stdio.h>
# define N1 300
# define N2 200
# define M 9
int choose(int x[N1][M],int y[N2][M-1])
{
    int i, j, temp;
    int z[N1]={0};
    // How many points are not satisfied for each inequality.
    for (i=0;i<N1;i++)
     {
          for(j=0;j<N2;j++)
              if((x[i][0]*y[j][0]+x[i][1]*y[j][1]+x[i][2]*y[j][2]+x[i][3]*y[j][3]+x[i][4]*y[j][4]+x[i][5]*y[j][5]
                 +x[i][6]*y[j][6]+x[i][7]*y[j][7]+x[i][8])<0)
                z[i]++;
     }
    temp=z[0]; j=0;
    // Finding the inequality and its count is the largest.
    for(i=1;i<N1;i++)
     {
          if(z[i]>temp)
          {j=i;temp=z[j];}
     }
    if(temp!=0)
    {
          // Delete the points corresponding to the largest inequality.
          for(i=0;i<N2;i++)
            {
              if(x[j][0]*y[i][0]+x[j][1]*y[i][1]+x[j][2]*y[i][2]+x[j][3]*y[i][3]+x[j][4]*y[i][4]+x[j][5]*y[i][5]
                +x[j][6]*y[i][6]+x[j][7]*y[i][7]+x[j][8]<0)
            {
              y[i][0]=0;y[i][1]=0;y[i][2]=0;y[i][3]=0;y[i][4]=0;y[i][5]=0;y[i][6]=0;
              y[i][7]=0;
             }
            }
          // Output inequality and the number of points that are not satisfied.
          for(i=0; i<8;i++)
          {
            if(x[j][i]<0‖i==0)
              printf("%d*x%d",x[j][i],i+1);
            else
              printf("+%d*x%d",x[j][i],i+1);
           }
          printf("+%d%6d",x[j][8],temp);printf("\n");
    x[j][0]=0;x[j][1]=0;x[j][2]=0;x[j][3]=0;x[j][4]=0;x[j][5]=0;x[j][6]=0;x[j][7]=0;
    x[j][8]=0;
          return temp;
     }
    else
          return 0;
 }
void main()
{
//In SageMath, the coefficients of the inequality of the s1-box are obtained.
//Because there are so many points, I'll just list some of them here.
          int a[N1][M]={{0,-1,0,0,0,0,0,0,1},{-1,0,0,0,0,0,0,0,1}, . . . , {-1,-1,-1,-1,-1,0,1,-1,5}, {-1,-2,-1,-2,-1,-1,2,-2,8}};
   //It doesn't satisfy the s1-box.
   //Because there are so many points, I'll just list some of them here.
          int b[N2][M-1]={{0, 0, 0, 0, 0, 0, 0, 1}, {0, 0, 0, 0, 0, 0, 1, 0}, . . . , {1, 1, 1, 1, 1, 1, 0, 1}, {1, 1, 1, 1, 1, 1, 1, 0}};
    printf("     inequalities                                       counting\n");
    while(choose(a, b)!=0)
    choose(a, b);
 }
```

Algorithm 1

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (Nos. 61672330, 61602287, and 61802235), the Key Research Development Project of Shandong Province (No. 2016GGX101024), China Postdoctoral Science Foundation (2018M632712), the Key Research and Development Plan of Shandong Province (2018GGX101037), and the Major Innovation Project of Science and Technology in Shandong Province (2018CXGC0702).

References

[1] L. S. Melro and L. R. Jensen, "Influence of functionalization on the structural and mechanical properties of graphene," *Computers, Materials and Continua*, vol. 53, no. 2, pp. 111–131, 2017.

[2] Z. Pan, J. Lei, Y. Zhang, and F. L. Wang, "Adaptive fractional-Pixel motion estimation skipped algorithm for efficient HEVC motion estimation," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–19, 2018.

[3] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of Cryptology*, vol. 4, no. 1, pp. 3–72, 1991.

[4] J. Daemen and V. Rijmen, *The Design of Rijndael: AES-The Advanced Encryption Standard*, Springer, Berlin, Germany, 2002.

[5] G. Cheng, C. Yang, X. Yao et al., "When Deep Learning Meets Metric Learning: Remote Sensing Image Scene Classification via Learning Discriminative CNNs," *IEEE Transactions on Geoscience & Remote Sensing*, vol. 99, pp. 1–11, 2018.

[6] M. H. Faghihi Sereshgi, M. Dakhilalian, and M. Shakiba, "Biclique cryptanalysis of MIBS-80 and PRESENT-80 block ciphers," *Security and Communication Networks*, vol. 9, no. 1, pp. 27–33, 2016.

[7] J. Borghoff, A. Canteaut, T. Güneysu et al., "PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications," in *Advances in Cryptology – ASIACRYPT 2012*, vol. 7658 of *Lecture Notes in Computer Science*, pp. 208–225, Springer Berlin Heidelberg, Heidelberg, Germany, 2012.

[8] S. Banik, A. Bogdanov, T. Isobe et al., "Midori: a block cipher for low energy," in *Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security*, Lecture Notes in Comput. Sci., pp. 411–436, Springer Berlin, 2014.

[9] C. Beierle, J. Jean, Moradi A. et al., "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS," in *Proceedings of the Part II, of the 36th Annual International Cryptology Conference on Advances in Cryptology-CRYPTO*, vol. 9815, pp. 123–153, Springer-Verlag, 2016.

[10] M. Tolba, A. Abdelkhalek, and A. M. Youssef, "Impossible Differential Cryptanalysis of Reduced-Round SKINNY," 2017.

[11] G. Han and W. Zhang, "Improved biclique cryptanalysis of the lightweight block cipher piccolo," *Security and Communication Networks*, vol. 2017, 2017.

[12] Y. Zheng, B. Jeon, and L. Sun, "Student's t-Hidden Markov Model for Unsupervised Learning Using Localized Feature Selection," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 99, no. 1-1, 2017.

[13] R. Ankele, S. Banik, A. Chakraborti et al., "Related-key impossible-differential attack on reduced-round Skinny," in *Applied Cryptography and Network Security*, vol. 10355 of *Lecture Notes in Comput. Sci.*, pp. 208–228, Springer, Cham, 2017.

[14] N. Mouha, Q. Wang, D. Gu, and B. Preneel, "Differential and linear cryptanalysis using mixed-integer linear programming," in *Information Security and Cryptology*, pp. 57–76, Springer, 2012.

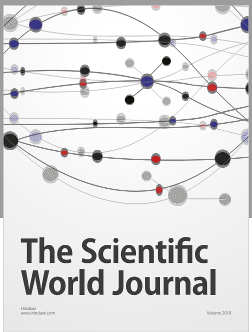
[15] S. Sun, L. Hu, P. Wang et al., "Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES (L) and Other Bit-Oriented Block Ciphers, ASIACRYPT," 2014.

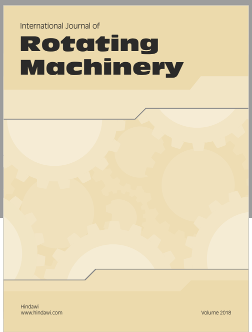
[16] L. Sun, W. Wang, R. Liu, and M. Wang, "MILP-aided bit-based division property for ARX ciphers," *Science China Information Sciences*, vol. 61, no. 11, Article ID 118102, 2018.

[17] R. A. Mezei, *An Introduction to SAGE Programming: With Applications to SAGE Interacts for Numerical Methods*, Inc, 2015.

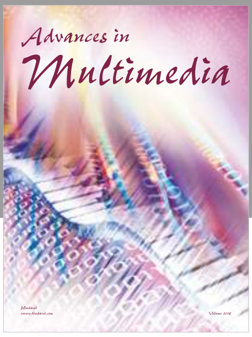
[18] "Division C.: Using the CPLEX Callable Library," 1997.

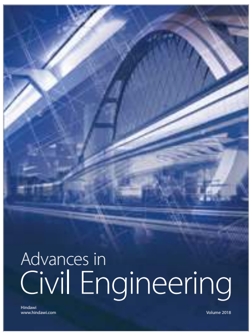
Journal of
Engineering

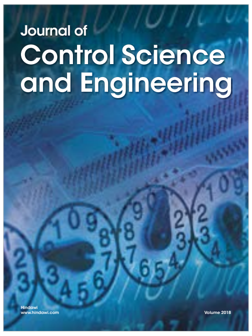

The Scientific World Journal


International Journal of
Rotating Machinery

Journal of
Sensors

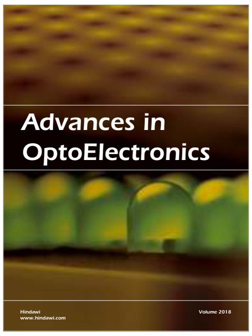

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

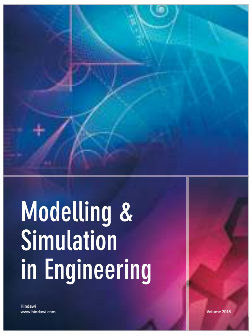
Journal of
Robotics

Journal of
Electrical and Computer Engineering


Advances in
OptoElectronics

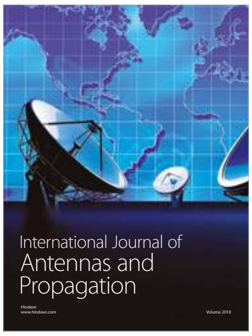
VLSI Design

International Journal of
Navigation and Observation

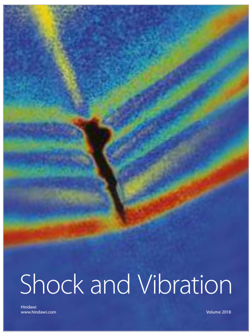

Modelling & Simulation in Engineering

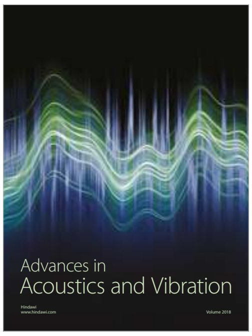
International Journal of
Aerospace Engineering

International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration