Journal of Theoretical and Applied Information Technology
20th April 2014. Vol. 62 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

KEY-DEPENDENT S-BOX IN LIGHTWEIGHT BLOCK CIPHERS

[1]SUFYAN SALIM MAHMOOD ALDABBAGH, [2]IMAD FAKHRI TAHA AL SHAIKHLI, [3]MUHAMMAD REZA ZABA

[1]Department of Information Systems. IIUM, [1]Kuala Lumpur, Malaysia
[1]University of Mosul, Iraq
2Department of Computer Science, [2]IIUM [2]Kuala Lumpur, Malaysia
[3]MIMOS Berhad, [3]Kuala Lumpur, Malaysia
E-mail: [1]sufyansalim_77@yahoo.com, [2]imadf@iium.edu.my , [3]reza.zaba@mimos.my

ABSTRACT

This paper introduces new methods of how to generate a suitable key-dependent substitution boxes (S-boxes) for use in lightweight block ciphers. An S-box is commonly used in block ciphers to provide the crucial property of nonlinearity. And one of the advantages of using key-dependent S-boxes is that it is difficult for an attacker to perform attacks such as differential cryptanalysis on the cipher since the S-boxes are unknown. However, implementing such S-box incurs costs, which is one of the criteria that a lightweight block cipher has to minimize. It is therefore important to have a key-dependent S-box method which always outputs a cryptographically strong S-box and has a small footprint. In this paper, we propose novel methods on how to generate s-boxes based on the value of the secret key. Moreover, this research showed the intensive analysis for cost and security for each method.

Keywords: *S-Box, PRESENT Lightweight Block Cipher, Key Dependent S-Box, Linear Cryptanalysis And Differential Cryptanalysis*

1. INTRODUCTION

Information technology changes rapidly and the need to protect data is becoming crucial. Generally, it is difficult to suggest a cryptographic algorithm that can suit all type of target devices. Common cryptographic algorithms may not be suitable for use in extremely constrained resources [1][2]. The design of cryptographic algorithms for such low resources therefore needs to take into account the implementation aspects without sacrificing too much on security. These are some of the issues involved in the field lightweight cryptography and this paper focuses on lightweight block ciphers.

A designer of lightweight block ciphers must balance between these three design goals: security, cost (normally given in Gate Equivalents or GEs), and performance. It is generally easy to optimize any two of the three design goals, which are security and cost, security and performance, or cost and performance.

However, at the same time, it is difficult to optimize all three design goals at once as shown in figure (1).

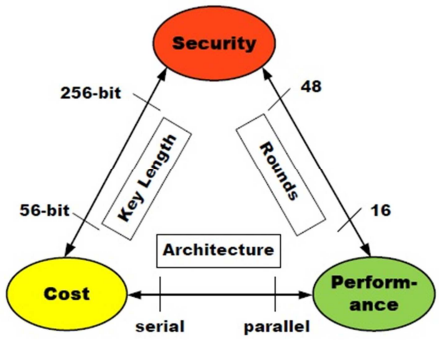

Figure.1: Three factors for designing lightweight algorithm [3].

One of the important components in lightweight block ciphers is the substitution box (s-box) since it is the only nonlinear component in the cipher. An s-box S is a nonlinear function that maps an m-bit value to another n-bit value. Such s-box is called an $m \times n$ s-box. In this paper, the focus would be on 4×4 S-boxes since it is commonly used in many existing lightweight block ciphers such as PRESENT [4], mCrypton [5], KLEIN [6], LBlock [7] and LED [8].

In this paper, we propose novel methods to generate s-boxes based on the value of the secret key. The generation is performed such that the cryptographic properties of all s-boxes produced by

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

this method are the same as the initial S-box set by the designer. This is essential to ensure that the security of the cipher is not weakened by poorly chosen s-boxes and to allow the method to be implemented in lightweight block ciphers. At the same time, the cost of implementing these methods is kept at a minimum.

2. KEY DEPENDENT S-BOX METHOD

This method is based on Theorem 2 in [9]. The theorem basically states that a new S-box can be created from an existing S-box by applying a series of linear functions to the existing S-box. The resulting S-box has the same cryptographic properties as the original S-box.

Let P_1 and P_2 denote two 4×4 permutation matrices, and a, b and x are three 4-bit values. Let S denote a 4×4 S-box. A new S-box S' is generated as

$$S'(x) = P_2(S(P_1(x) + a)) + b \quad (1)$$

whereS' has the same cryptographic properties as S. Examples of P_1 and P_2 are given below.

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, P_2 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Based on Equation (1), we can create five methods.

First method: it uses only one matrix as in equation (2).

$$S'(x) = S(P_1(x)) \quad (2)$$

Second method: it uses two matrices as in equation (3).

$$S'(x) = P_2(S(P_1(x))) \quad (3)$$

Third method: it uses one matrix and one constant as in equation (4).

$$S'(x) = S(P_1(x) + a) \quad (4)$$

Fourth method: it uses two matrices and one constant as in equation (5).

$$S'(x) = P_2(S(P_1(x) + a)) \quad (5)$$

Fifth method: it uses two matrices and two constants as in equation (6).

$$S'(x) = P_2(S(P_1(x) + a)) + b \quad (6)$$

As mentioned earlier, the resulting S-box S'generated by the above methods have the same cryptographic properties as the original S-box S.For example, if the initial S-box $S$is the S-box of PRESENT, then the characteristics of the generated S-box $S'$is the same as S.

In this paper, we propose novel key dependent S-box methods which are suitable for lightweight block ciphers. The following sections are describing the analysis of security through applying linear and differential cryptanalysis. Another section is discussing the cost for the proposed method through calculating the number of GE. Note that, the proposed methods are generic, which means it can be used with any block cipher.

Moreover, this research discusses the cost and security when it uses one bit, two bits, four bits and more than four bits of key dependent S-box with each method.

3. COST DISCUSSION

Implementation cost is one of the most important factors to consider in designing lightweight block cipher algorithms. The cost is measured in the number of Gate Equivalence (GE). And it depends on how many key bits used and also the cost is different for each method that mentioned above.

The cost to store one bit is 6GE, the cost for AND and OR operations(denoted by & and | respectively) is 1.33GE for one bit and the cost for XOR (denoted by⊕) is 2.67 GE for one bit. Also, the cost of rotation matrix by different number in each round is 2.33GE for one bit.

3.1 First Case One Bit as Key Dependent S-box

The cost in this case for each method is as follows:

First method: The cost of storing one 4×4binaray matrix is:

$$16\ bits \times 6 = 96\ GE \quad (7)$$

since there are$4 \times 4 = 16$ bits in a 4×4binaray matrix and each bit requires 6 GE.

The computation of the offset number that uses to rotate each row of the matrix P_1 is shown by equation (8):

$$RC = key\ \&\ b \quad (8)$$

Where RC is the offset to rotate each row of the matrix P_1, key is any 4-bit from master key and b is a value of any bit in key because this case uses one bit as key dependent S-box.

The cost for 4-bit AND operation is 5.32 GE.

Rotate each row of the matrixP_1 by the offset RC.

The cost of rotation one matrix 4×4 by different offset in each round is shown by equation (9):

$$16\ bits \times 2.33 = 37.28\ GE \quad (9)$$

Total cost N is

$$N = 96 + 37.28 + 5.32 = 138.6GE$$

Second method: The cost of storing one $4 \times$ 4binaray matrix is shown by equation (7`). This method has two matrices and the total cost for two matrices is shown by equation (10):

$$96 \times 2 = 192\ GE \quad (10)$$

Journal of Theoretical and Applied Information Technology
20th April 2014. Vol. 62 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The computation of the offset that uses to rotate each row of the matrixP_1 is shown by equation (8).
The cost for 4-bit AND operation is 5.32 GE.
Rotate each row of two matrices $P_1$and P_2 by the offset RC.
The cost of rotation one matrix 4×4 by different offset in each round is shown by equation (9) and the total cost for two matrices is shown by equation (11):

$$37.28 \times 2 = 74.56\ GE \quad (11)$$

Total cost N is

$$N = 192 + 74.56 + 5.32 = 271.88GE \quad (12)$$

Third method: The cost of this method is the same of first method with additional cost of XOR. The cost for XOR operation is 10.68 GE.
The total cost N is

$$N = 138.6 + 10.68 = 149.28\ GE \quad (13)$$

Fourth method: The cost of this method is the same of second method with additional cost of XOR. The cost for XOR operation is 10.68 GE.
The total cost N is

$$N = 271.88 + 10.68 = 282.56\ GE \quad (14)$$

Fifth method: The cost of this method is the same of fourth method with additional cost of XOR. The cost for XOR is 10.68 GE.
The total cost N is

$$N = 282.56 + 10.68 = 293.24\ GE \quad (15)$$

3.2 Second Case Two Bits as Key Dependent S-box

The cost in this case for each method is similar to the cost of first case one bit but the security issues is different.

3.3 Third Case Four Bits as Key Dependent S-box

The cost in this case for each method is as follows:
First method: The cost of storing one 4×4binaray matrix is shown by equation (7):
There are two offset numbers ($RC$and RR) that are use to rotate two matrices P_1 and P_2. The computations of finding two offset numbers are shown by equation (16 and 17):

$$RC = key\ \&\ '3' \quad (16)$$

$$RR = key\ \&\ 'C' \quad (17)$$

WhereRC and RR are two offset numbers to rotate the column and row of two matrices respectively, key is any 4-bit from master key.

The cost for two AND operation is

$$5.32 \times 2 = 10.64\ GE. \quad (18)$$

Rotate the column of matrix P_1 by the offset RC and rotate the row of matrix P_2 by the offset RR.
The cost of rotation one matrix 4×4 by different offset in each round is shown by equation (9):
The total cost N is

$$N = 96 + 37.28 + 10.64 = 143.92GE \quad (19)$$

Second method: The cost of storing 4×4 binary matrix is shown by equation (7`). This method has two matrices and the total cost for two matrices is shown by equation (10). There are two offset numbers ($RC$and RR) that are use to rotate two matrices P_1 and P_2. The computations of finding two offset numbers are shown by equation (16 and 17).
The cost for two AND operation is

$$5.32 \times 2 = 10.64\ GE \quad (20)$$

Rotate the column of matrix P_1 by the offset RC and rotate the row of matrix P_2 by the offset RR.
The cost of rotation one matrix 4×4 by different offset in each round is shown by equation (9) and the total cost for two matrices is shown by equation (11).
Total cost N is

$$N = 192 + 74.56 + 10.64 = 277.2GE \quad (21)$$

Third method: The cost of this method is the same of first method with additional cost of XOR. The cost for XOR is 10.68 GE.
The total cost N is

$$N = 143.92 + 10.68 = 154.6\ GE \quad (22)$$

Fourth method: The cost of this method is the same of second method with additional cost of XOR. The cost for XOR is 10.68 GE.
The total cost N is

$$N = 277.2 + 10.68 = 287.88\ GE \quad (23)$$

Fifth method: The cost of this method is the same of fourth method with additional cost of XOR. The cost for XOR operation is 10.68 GE.
The total cost N is

$$N = 287.88 + 10.68 = 298.56\ GE \quad (24)$$

3.4 Fourth Case More Than Four Bits as Key Dependent S-box

The cost in this case for each method is similar to the third case with additional cost of applying XOR to compute the value of key as follows:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

First method: The cost of this method is the same of the cost of first method of third case with additional cost of applying XOR between key bits. The cost of XOR operation computes by equation (25) , (26) and (27):

Let $k = k_0|k_1|k_2|\ldots|k_{m-1}$ (25)

Where k is the concatenating of m four bits and k is the master key.

Before computing the cost of applying XOR, you may know how many times that XOR operation applied. The equation (26) shows how many times that XOR applied in general:

$key = k_0 \oplus k_1 \oplus k_2 \oplus \ldots \oplus k_{m-1}$ (26)

Where key is a 4-bit value and this value uses to compute the offset numbers.

The cost of using XOR operation in the equation (26) is shown by equation (27):

$$Cost\ XOR = 2.67 \times 4 \times (m1) \quad (27)$$

Where m = 1, 2, 3 ...

The total cost N is

$$N = Cost\ of\ first\ method\ of\ third\ case + Cost\ XOR \quad (28)$$

For example, if m=16 then it means that there are 64 bits used as key dependent S-box and the total cost N is

$N = 143.92 + (10.68 \times 15) = 304.1GE$ (29)

Second method: The total cost N of second method is $N = Cost\ of\ second\ method\ of\ third\ case + Cost\ XOR$ (30)

For example, if m=16 then it means that there are 64 bits used as key dependent S-box and the total cost N is

$N = 277.2 + (10.68 \times 15) = 437.4GE$ (31)

Third method: The total cost N of third method is

$$N = Cost\ of\ third\ method\ of\ third\ case + Cost\ XOR \quad (32)$$

For example, if m=16 then it means that there are 64 bits used as key dependent S-box and the total cost N is

$N = 154.6 + (10.68 \times 15) = 314.8GE$ (33)

Fourth method: The total cost N of fourth method is $N = Cost\ of\ fourth\ method\ of\ third\ case + Cost\ XOR$ (34)

For example, if m=16 then it means that there are 64 bits used as key dependent S-box and the total cost N is

$N = 287.88 + (10.68 \times 15) = 448.08GE$ (35)

Fifth method: The total cost N of fourth method is

$$N = Cost\ of\ fifth\ method\ of\ third\ case + Cost\ XOR \quad (36)$$

For example, if m=16 then it means that there are 64 bits used as key dependent S-box and the total cost N is

$N = 298.56 + (10.68 \times 15) = 458.76GE$ (37)

All the costs of each case and each method are presented in table (1). The last case used 64 bits as key dependent S-box.

4. SECURITY DISCUSSION

The cryptanalysis is the important factor to test the security of the algorithm. As it has been mentioned before, the S-box is an important part and the heart of the algorithm. To measure the security of any algorithm, this is done by using the cryptanalysis. The most important attacks are linear and differential cryptanalysis. These attacks are the basic for all other attacks [10][11]. This research will focus on differential cryptanalysis because the linear attack is closer to differential attack.

For differential cryptanalysis, the significant step is to construct the difference distribution table for each S-box. That step is very important to start the attack. The next important step is to find or determine the differential characteristic for whole algorithm. For fixed S-box like PRESENT S-box, it can calculate the probability for whole algorithm and find the path of differential characteristic for whole algorithm. For key dependent S-box like proposed algorithm, it is difficult to calculate the probability and to find the path of differential characteristic for whole algorithm because the S-boxes are unknown.

In the next subsections, this research discusses the total number to generate different S-box and different differential distribution table (DDT) for each case and each method. In the following subsections, the number of different S-boxes is higher than or equals the number of different DDT because the XOR ($\oplus$) is a linear operation and it effect on the values of S-box only not on DDT. For example, one DDT has many different S-boxes by applying XOR ($\oplus$) on the values of S-box. The following subsections show the number of different S-boxes and the number of different DDT for each case and each method:

4.1 First Case One Bit as Key Dependent S-box

The total number to generate different S-box and different DDT in this case for each method as follows:

First method: There are two different S-boxes and also the same number of different DDT.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Second method: There are four different S-boxes and also the same number of different DDT.
Third method: There is (2 *16 = 32) different S-boxes while there are two different DDT.
Fourth method: There is (2 * 2 * 16 = 64) different S-boxes while there are four different DDT.
Fifth method: There is (2 * 2 * 16 * 16 = 1024) different S-boxes while there are four different DDT.

4.2 Second Case Two Bits as Key Dependent S-box

The total number to generate different S-box and different DDT in this case for each method is as follows:
First method: There are four different S-boxes and also the same number of different DDT.
Second method: There are sixteen different S-boxes and also the same number of different DDT.
Third method: There is (4 *16 = 64) different S-boxes while there are four different DDT.
Fourth method: There is (4 * 4 * 16 = 256) different S-boxes while there are sixteen different DDT.
Fifth method: There is (4 * 4 * 16 * 16 = 4096) different S-boxes while there are sixteen different DDT.

4.3 Third Case Four Bits as Key Dependent S-box

The total number to generate different S-box and different DDT in this case for each method as follows:
First method: There are sixteen different S-boxes and also the same number of different DDT.
Second method: There are 256 different S-boxes and also the same number of different DDT.
Third method: There is (16 *16 = 256) different S-boxes while there are sixteen different DDT.
Fourth method: There is (16 * 16 * 16 = 4096) different S-boxes while there are 256 different DDT.
Fifth method: There is (16 * 16 * 16 * 16 = 65536) different S-boxes while there are 256 different DDT.

4.4 Fourth Case More Than Four Bits as Key Dependent S-box

The total number to generate different S-box and different DDT in this case is the same with third case but this case more secure than the previous cases because it used more bits.

The table (1) shows the cost and number of S-box and DDT of each case and each method.

Table (1): Cost, number of S-boxes and number of DDT

Case	Method	Cost GE	Number of S-box	Number of DDT
First case	First method	138.6	2	2
	Second method	271.88	4	4
	Third method	149.28	32	2
	Fourth method	282.56	64	4
	Fifth method	293.24	1024	4
Second case	First method	138.6	4	4
	Second method	271.88	16	16
	Third method	149.28	64	4
	Fourth method	282.56	256	16
	Fifth method	293.24	4096	16
Third case	First method	143.92	16	16
	Second method	277.2	256	256
	Third method	154.6	256	16
	Fourth method	287.88	4096	256
	Fifth method	298.56	65536	256
Fourth case	First method	304.12	16	16
	Second method	437.4	256	256
	Third method	314.8	256	16
	Fourth method	448.08	4096	256
	Fifth method	458.76	65536	256

5. CONCLUSION

In this research, novel methods to generate S-boxes based on the value of the secret key are proposed. This research showed the intensive

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

analysis for cost and security for 1bit, 2bits, 4bits and more than 4 bits.

For the cost and security perspective, the least cost was with the first and second method of first case 138.6GE and 271.88GE respectively, but the least number of different S-box with the first method of first case. The highest cost was with fifth method of fourth case when it used whole key bits. While the highest number of different S-box was with fifth method of third and fourth case. The fourth case is more secure because it used more than 4 bits as key dependent S-box.

Depend on the application, we can choose which method is suitable

REFERENCE

[1] Panasenko, S., & Smagin, S., "Lightweight Cryptography: Underlying Principles and Approaches", International Journal of Computer Theory and Engineering, Vol 3 No.4, (2011).

[2] Sufyan Salim Mahmood AlDabbagh and Imad Al Shaikhli," Lightweight Block Ciphers: a Comparative Study", in Journal of Advanced Computer Science and Technology Research Vol.2 No.4, November 2012, 159-165.

[3] Eisenbarth, T., & Kumar, S., "A Survey of Lightweight-Cryptography implementations", Design & Test of Computers, IEEE, 24(6), 522-533, (2007).

[4] A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher Cryptographic Hardware and Embedded Systems - CHES 2007." Vol. 4727, P. Paillier and I. Verbauwhede, Eds., ed: Springer Berlin / Heidelberg, 2007, pp. 450-466.

[5] C. Lim and T. Korkishko, "mCrypton – A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors Information Security Applications." Vol. 3786, J.-S. Song, et al., Eds., ed: Springer Berlin / Heidelberg, 2006, pp. 243-258.

[6] Z. Gong, S. Nikova, and Y. Law, "KLEIN: A New Family of Lightweight Block Ciphers RFID. Security and Privacy." Vol. 7055, A. Juels and C. Paar, Eds., ed: Springer Berlin / Heidelberg, 2012, pp. 1-18.

[7] W. Wu and L. Zhang, "LBlock: A Lightweight Block Cipher Applied Cryptography and Network Security." Vol. 6715, J. Lopez and G. Tsudik, Eds., ed: Springer Berlin / Heidelberg, 2011, pp. 327-344.

[8] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED Block Cipher Cryptographic Hardware and Embedded Systems – CHES 2011." Vol. 6917, B. Preneel and T. Takagi, Eds., ed: Springer Berlin / Heidelberg, 2011, pp. 326-341.

[9] G. Leander and A. Poschmann, "On the Classification of 4 Bit S-Boxes Arithmetic of Finite Fields." vol. 4547, C. Carlet and B. Sunar, Eds., ed: Springer Berlin / Heidelberg, 2007, pp. 159-176.

[10] Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1990).

[11] Matsui, M.: Linear cryptoanalysis method for des cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1993).