International Journal of Security and Its Applications
Vol. 10, No. 3 (2016), pp.383-392
http://dx.doi.org/10.14257/ijsia.2016.10.3.33

A Random PRESENT Encryption Algorithm Based on Dynamic S-box

Zhiying Tang, Jie Cui*, Hong Zhong, Mingyong Yu

School of Computer Science and Technology, Anhui University, Hefei, 230039, China
**cuijie@mail.ustc.edu.cn*

Abstract

S-box mainly plays the role of confusion in the encryption process as an important component. For the new encryption algorithm PRESENT proposed in 2007, S-box impacts on the security of the encryption algorithm directly. This paper briefly describes the process of PRESENT algorithm and proposes an improved S-box to solve the problem that the original PRESENT S-box has anti-fixed point. Then a random PRESENT encryption algorithm based on dynamic S-box is proposed. The dynamic multiple S-boxes technology is used to implement random PRESENT algorithm, to enhance the security of the cryptographic algorithm. Finally, the security analysis is done, and it suggests that dynamic S-box has a superior ability to resist differential attack and linear attack. By comparison to the diffusion rate of original PRESENT S-box, our dynamic S-box has better avalanche effect.

***Keywords**: PRESENT; Dynamic S-box; Avalanche effect; Block cipher; diffusion rate*

1. Introduction

With the rapid development of the Internet of Things, the radio frequency identification (RFID), wireless sensor network (WSN), and other new technologies go deep into all aspects of people's work and life. However, RFID and WSN are based on wireless network to transfer information. The transmission of information is not difficult to be obtained, interfered, and even destroyed by the attackers. So information security technology is widely used in protecting networks to enhance the safety of secret information. And the traditional symmetric encryption algorithm is often restricted by hardware, which cannot achieve good effects under the condition of limited hardware and power consumption. Therefore, lightweight cipher algorithm such as PRESENT[1] and LED[2] has gradually become the realistic choice to ensure the information security of Internet of Things.

The lightweight block encryption algorithm PRESENT [1], put forward by Bogdanov in 2007, is carefully designed with area and power constraints uppermost in limited conditions. So the PRESENT algorithm still has high security in the application of limited space and power consumption, such as sensor network node. However, the permutation transformation of PRESENT algorithm is different with the column confusion of AES. It uses bit as the permutation function input unit while AES uses byte for permutation. The permutation layer of PRESENT encryption algorithm is a perfectly symmetrical structure and easy implemented. Because the structure of this kind of design has poor diffusibility, the PRESENT algorithm diffusively quality mainly depends on the diffusion properties of S-box. So S-box, as a nonlinear component in lightweight block encryption algorithm, affects the strength of the whole structure of the cipher.

To strengthen DES by using existing hardware, an improved method of DES which uses the S-box related to a key is put forward in paper [4], and the researcher has demonstrated the safety of the improved method. Meanwhile, a new key related lightweight block cipher algorithm which could resist 22 rounds differential attack is given in paper [5]. In the paper [7], it describes a design principle of dynamic S-box based on the AES and gives the main algebraic properties of dynamic S-box. In the paper [8], it introduces a kind of S-box of optimized design scheme based on genetic algorithm and finds that the S-box based on genetic algorithm in such aspects as nonlinearity and avalanche effect has a very significant improvement comparison with DES S-box.

This paper briefly describes the process of PRESENT algorithm and proposes an improved S-box to solve the problem that the original PRESENT S-box has anti-fixed point. With the dynamic encryption thoughts, this paper proposes a random PRESENT encryption algorithm based on dynamic S-box. It greatly improves dynamic, security, and the avalanche effect of the encryption algorithm, while it almost does not bring additional computation overhead. Performance analysis suggests that our encryption algorithm has good ability to resist differential attack and linear attack.

2. Present

The PRESENT algorithm is a kind of key length variable lightweight block algorithm. it uses 64-bit block and supports 80-bit and 128-bit key length. In this paper, all experimental results are given when the key length is 80 bits.

The algorithm process of PRESENT is shown in Fig. 1. The algorithm uses the SPN structure, whose operation mainly divided into two parts, key extension operation and a total of 31 rounds of iteration operation. Each iteration function F is composed of three different transformations: addRoundKey, S-box and PLayer. KeySchedue is used to generate the round key which used for round iteration operation. The 64-bit plaintext P after 31 rounds of iteration operation and the last round XOR with the round key, get 64-bit cipher C. The different transformations operate on the intermediate result, called the State.

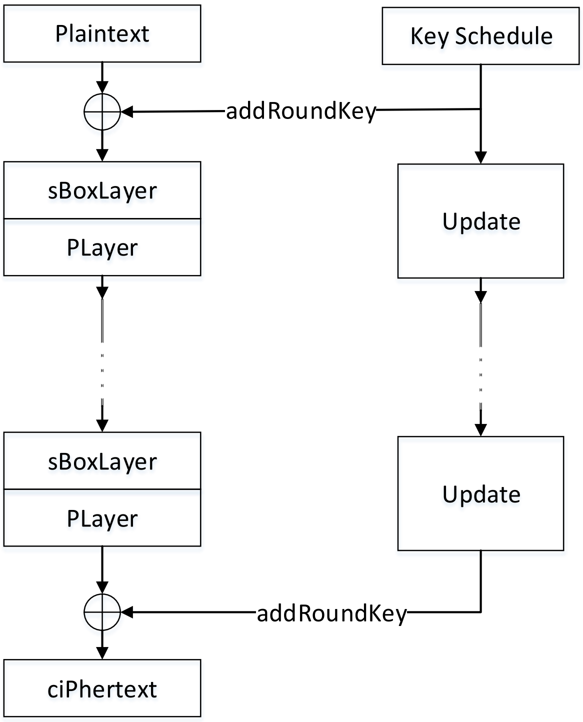


Figure 1. PRESENT Algorithmic Process

Pseudo code to describe the PRESENT algorithm is as follows:

```
Generateoundkeys()
For i=1 to 31 do
    addRoundKey(STATE,Ki)
    sBoxLayer(STATE)
    pLayer(STATE)
end for
addRoundKey(STATE,K32)
```

The encryption process is detailed as follows:

1) addRoundKey: 64-bit input XOR with the round key.

2) S-box: The S-box transformation is a non-linear 4-bit word substitution, operating on each of the State 4-bit words independently.

3) PLayer: PLayer is a permutation transformation, operating on the 64-bit State.

3. S-boxes Improvement Ideas

3.1 Improvement of the PRESENT S-box

The non-linear substitution layer uses a single 4-bit S-box in PRESENT. The implementation of such an S-box typically is much more compact than that of an 8-bit S-box. The action of this box in hexadecimal notation is given by Tab. 1.

A good performance of S-box should have good nonlinearity, differential uniformity, immune correlation and avalanche effect, and should avoid having fixed points or anti-fixed points. This article based on the thought of literature [8] using the genetic algorithm to design S-box gets the improved S-boxes (S-box S1 and S-box S2) with good diffusion rate, which solve the problem that the PRESENT S-box has anti-fixed point. The improved S-boxes are shown in Tab. 2 and Tab. 3 respectively.

Table 1. The PRESENT's S-box

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 2. Improvement S-box S1

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	6	1	4	9	0	A	D	3	E	F	8	B	7	5	2

Table 3. Improvement S-box S2

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	2	5	E	9	1	D	0	B	4	3	7	C	F	6	A	8

In this paper, the improved method mainly adopts the idea of the crossover and mutation in the genetic algorithm. The crossover operator is used to exchange genetic information between pairs, or larger groups, of individuals. The simplest recombination operator is that of single-point crossover [14]. As two parent binary strings:

P1 = 1 0 0 1 0 1 1 0,
P2 = 1 0 1 1 1 0 0 1.

If an integer position n is selected uniformly at random between 1 and the string length, L, minus zone [1, L-1], and the genetic information exchanged between the individuals about this point, then two new offspring strings are produced [14]. The two offspring below are produced when the crossover point n = 5 is selected:

O1 = 1 0 0 1 0 0 0 1,
O2 = 1 0 1 1 1 1 1 0.

In the binary string representation, mutation will cause a single bit to change its state, 0 ⇒ 1 or 1 ⇒ 0. So, for example, mutating the sixth bit of O1 leads to the new string：

O1 = 1 0 0 1 0 1 0 1.

3.2 Performance and Testing

In this article, the improved S-boxes which will replace the original S-box in PRESENT encryption algorithm directly affect the properties of dynamic S-box later designed. Then test the avalanche effect of the encryption algorithm after replacing S-box. To compare the diffusion rate of the S-box-replaced algorithm and the diffusion rate of the original encryption algorithm, the principal test content is input binary data 00000001(if the input is key, then the initial adjust by 0000000001), other inputs are randomly. Every test time the data left shift one bit, then check the diffusion rate. The test consists of four parts: plaintext to ciphertext diffusion rates(Tab.4), ciphertext to plaintext diffusion rates(Tab.5), key to ciphertext diffusion rates(Tab.6), key to plaintext diffusion rates(Tab.7). The S0 in experimental results is that the PRESET encryption algorithm own S-box.

The PRESENT encryption algorithm using different S-box carries on the diffusion rate test. The results show that the performances of improved S-boxes are better than the PRESENT own S-box. At the same time, the improved S-boxes do not have the problem as the PRESENT S-box with anti-fixed point.

Table 4. Plaintext to Ciphertext Diffusion Rates

Input	S0	S1	S2
00000001	0. 49926	0. 50033	0. 49859
00000010	0. 49720	0. 50094	0. 50093
00000100	0. 50009	0. 49993	0. 50024
00001000	0. 49935	0. 49873	0. 49951
00010000	0. 49802	0. 50000	0. 50095
00100000	0. 50106	0. 50057	0. 50030
01000000	0. 50036	0. 50020	0. 50079
10000000	0. 50032	0. 50080	0. 50021
Average	0. 49945	0. 50018	0. 50019

Table 5. Ciphertext to Plaintext Diffusion Rates

Input	S0	S1	S2
00000001	0. 49883	0. 49917	0. 50053
00000010	0. 49842	0. 50000	0. 49985
00000100	0. 49998	0. 49968	0. 49980
00001000	0. 49792	0. 50003	0. 49913
00010000	0. 49963	0. 49982	0. 50001
00100000	0. 50055	0. 49961	0. 50047
01000000	0. 50030	0. 50078	0. 49918
10000000	0. 50022	0. 50078	0. 50041
Average	0. 49948	0. 49998	0. 49992

Table 6. Key to Ciphertext Diffusion Rates

Input	S0	S1	S2
0000000001	0.49957	0.50013	0.49993
0000000010	0.50093	0.50143	0.49937
0000000100	0.50001	0.50036	0.49960
0000001000	0.49997	0.50100	0.50027
0000010000	0.49984	0.50007	0.50053
0000100000	0.49931	0.49946	0.50031
0001000000	0.49988	0.50060	0.50121
0010000000	0.49928	0.50066	0.50036
0100000000	0.50041	0.50035	0.50042
1000000000	0.49972	0.50105	0.49883
Average	0.49989	0.50051	0.50008

Table 7. Key to Plaintext Diffusion Rates

Input	S0	S1	S2
0000000001	0.49950	0.50075	0.5011
0000000010	0.50041	0.50055	0.5028
0000000100	0.49860	0.50096	0.5010
0000001000	0.49903	0.50124	0.5011
0000010000	0.49993	0.49974	0.5000
0000100000	0.50103	0.50017	0.4998
0001000000	0.50059	0.50077	0.5007
0010000000	0.49919	0.50064	0.4994
0100000000	0.49892	0.50101	0.4993
1000000000	0.50054	0.50002	0.4999
Average	0.49977	0.50058	0.5005

4. Dynamic S-box Design

As the PRESENT S-box is used in a static way in the process of encryption, this article proposes a random PRESENT encryption algorithm based on dynamic S-box. The design principle of dynamic S-box is shown in Fig. 2, in which an S-box dynamic selection function is designed. In each iteration, the block is split into 8 bytes (or 16 nibbles) when the 64-bit block is replaced by the S-box. Two continuous nibble replacements need the extended key to dynamically choose corresponding S-box in the process of encryption. So it is most likely to use eight different S-boxes as each round of 64-bit group during the iteration operation. Then 31 rounds of iteration operation may use 248 different S-boxes in total. Without considering the cost of storage area, with the increase of dynamic S-box size, S-box selected uncertainty will also increase, which will enhance the security of cryptographic algorithms.

Concrete implementation steps are as follows:

Step 1: Initialize the array flag[] which is used to store the S-box using state in selection function.

Step 2: 64-bit block XOR with the corresponding round key.

Step 3: 64-bit intermediate result which was gotten from the second step is divided into 8 bytes and with the corresponding key input selection function. Corresponding flags XOR with the input of 8-bit key in selection function, the results stored in the flag array.

Step 4: According to the state of the flags to choose the corresponding S-box, which is carried out on the byte substitution.

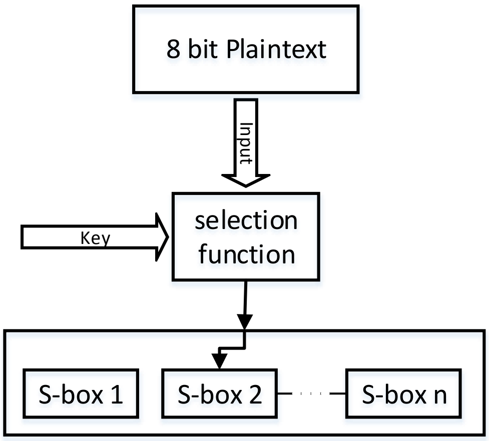


Figure 2. Dynamic S-box Design Principle

5. Dynamic S-box Analysis

5.1 Security Analysis

S-Boxes can be viewed in two distinct ways. The first is the static view of the S-box which describes the S-box when the inputs are not changing. The second is the dynamic view of the S-box which describes the S-box when the inputs are changing [10]. The use of dynamic S-box can improve the security of encryption. In the paper [4], an improved method of DES which using S-box related to the key is put forward. After the analysis of improved DES algorithm, the literature's authors think only on the premise of knowing S-box of content in advance, then implementation of the linear attack and differential attack is possible. If encryption is used with a strong password selection method S-box associated with the key, related means of attack would be extremely difficult.

With reference to the thought of literature [7], this paper designed a dynamic S-box based on the PRESENT. The dynamic S-box is comprised of several nonlinear S-boxes. In the process of S-box replacement, with each round of encryption, the selected S-box only related to the current key. It is said that selected S-box is along with the key state of dynamic change. With the expansion of the dynamic S-box, the uncertainty of the S-box using to encrypt also increases. So the attacker is difficult to predict each S-box used state. According to the basic theory of literature [4], we can know, the dynamic S-box can effectively resist differential attack and linear attack.

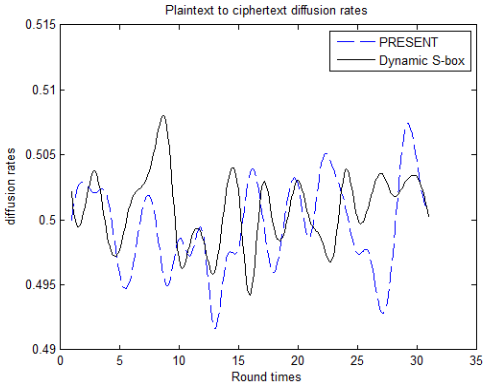
Meanwhile, in the improved algorithm, the number of S-boxes used to construct the dynamic S-box can be n (n is an integer greater than 1). In order to study conveniently only use the two S-boxes which are designed in the third part. Regarding the process of data encryption and decryption as an operation, each operation runs 1000 times. Efficiency is measured when the program running, and the experiment conducted a total of six times. Finally, this paper found using the dynamic S-box to encrypt and decrypt, the time loss compared to the original PRESENT algorithm only increases 13.37%. Further study found, the lightweight encryption algorithm PRESENT uses a 4x4 S-box that can be directly stored by software, so the cost of storage space is not big. And even more S-boxes form dynamic S-box, it has little influence on the efficiency of the program running.

Based on the above analysis, dynamic S-box in this paper can effectively resist differential attack and linear attack. And with dynamic S-box scale expansion, the security of the algorithm is significantly increased and the efficiency of the algorithm is not affected.

5.2 Test and Comparison of Cryptographic Properties

Diffusion rate is an important indicator to measure the avalanche effect of encryption algorithms. The bigger the diffusion ratio is, the better the avalanche effect of the encryption algorithm is, and the higher the security of the encryption algorithm is. For an encryption algorithm, if the diffusion rate is equal to or above 0.5 when any one input bit was reversed, then the encryption algorithm provides strict avalanche effect [12].

This Paper only studies the avalanche effect of the designed dynamic S-box when n=2. Apply the new dynamic S-box to the encryption and decryption simulation system, and test its performance from round 0 to round 31. Set the S-box uses to expand encryption key as the main S-box, while the others are only used for encryption as auxiliary S-box. The simulation results are given in Fig. 3-6. Fig. 3 shows plaintext to ciphertext diffusion rates and Fig. 4 shows key to ciphertext diffusion rates in encryption process. Fig. 5 shows ciphertext to plaintext diffusion rates and Fig. 6 shows key to plaintext diffusion rates in decryption process.



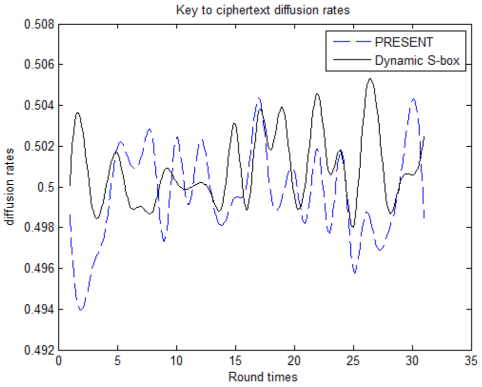
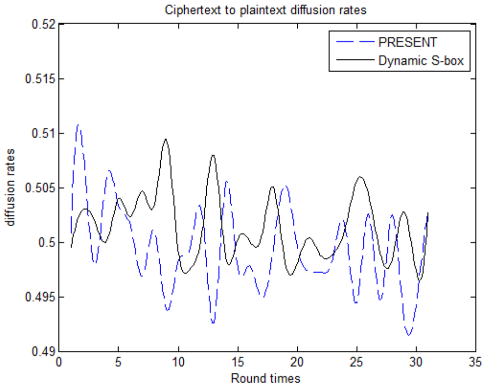


Figure 3. Plaintext to Ciphertext Diffusion Rates

Figure 4. Key to Ciphertext Diffusion Rates

As can be seen from Fig. 3 and Fig. 4, dynamic S-box diffusion rate variance is smaller than PRESENT S-box in the encryption process. In other words, the dynamic S-box of the spread of the rate fluctuation is relatively stable. And plaintext to ciphertext diffusion rates are increased by 0.13% on average and key to ciphertext diffusion rates are increased by 0.12% on average in the encryption process.



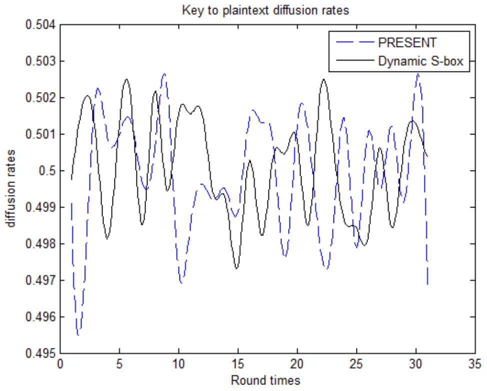


Figure 5. Ciphertext to Plaintext Diffusion Rates

Figure 6. Key to Plaintext Diffusion Rates

As from Fig. 5 and Fig. 6, can be found dynamic S-box fewer fluctuations than the original S-box, and the range is also smaller than the original S-box in the process of decryption. And ciphertext to plaintext diffusion rates are increased by 0.02% on average and key to plaintext diffusion rates are increased by 0.20% on average in the decryption process. According to the definition of strict avalanche effect, these diffusion rates have reached 0.5, so can think the dynamic S-box satisfies the requirement of strict avalanche effect.

Diffusion rates in the encryption and decryption processes increased in varying degrees and reached the requirement of strict avalanche effect. At the same time, the improved scheme resisting on differential and linear attack resistance has a better performance and less effect on the efficiency of the encryption algorithm. So the dynamic S-box compared with the original PRESENT S-box has improved.

Further study shows that the S-box selection function is designed by replacing S-box chosen in byte. An 8-bit plaintext stored in the same byte is replaced with an identical S-box. Because of the usage of a 248-bit flag array, after XOR with the corresponding round key, store the 16 kind S-box use states in each iterative operation. It can use a random function to give the flag array a uniformly initial value before each encryption. Even if the keys are the same, but the initial value of an array of marks vary randomly. So after exclusive or operation between the initial and key, its value is also unpredictable. As the dynamic S-box expanded, each iterative operation choice of S-box is unpredictable. Even if expressly with the same key but do not know the value of sign bit, the ciphertext remains uncertain. On the other hand, the ciphertext cannot also decrypt, which enhance the security of cryptographic algorithms.

6. Conclusions

This paper proposes an improved S-box to solve the problem that the original PRESENT S-box has anti-fixed point. Through combing with some improved S-boxes, this paper proposes a random PRESENT encryption algorithm with the strict avalanche effect, and the algorithm is well proved to have the ability of resisting differential and linear attack. It not only has theoretical significance, and has certain and actual application value. Of course, in this paper, there are still many needs to be improved, such as optimizing the algorithm and searching better S-box to construct dynamic S-box. Finally, this paper also advances some ideas on the further research and illustrates the diversity of designing dynamic S-box method.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (No. 61572001, No. 61502008), the Research Fund for the Doctoral Program of Higher Education (No. 20133401110004), the Educational Commission of Anhui Province, China (No. KJ2013A017), The Natural Science Foundation of Anhui Province (No. 1508085QF132), the Tender Project of the Co-Innovation Center for Information Supply & Assurance Technology of Anhui University (No. ADXXBZ2014-7), and the Doctoral Research Start-up Funds Project of Anhui University. The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

References

[1] A. Bogdanov, L. R. Knudsen and G. Leander, "PRESENT: An ultra-lightweight block cipher", Springer Berlin Heidelberg, **(2007)**.

[2] J. Guo, T. Peyrin and A. Poschmann, "The LED block cipher//Cryptographic Hardware and Embedded Systems–CHES", Springer Berlin Heidelberg, **(2011)**.

[3] T. T. K. Hue T. M. Hoang and D. Tran, "Chaos-based S-box for lightweight block cipher//Communications and Electronics (ICCE)", IEEE Fifth International Conference IEEE, **(2014)**, pp. 572-577.

[4] E. Biham and A. Biryukov, "How to strengthen DES using existing hardware", Springer Berlin Heidelberg, **(1995)**.

[5] M. Minier and M. A. Naya-Plasencia, "A related key impossible differential attack against 22 rounds of the lightweight block cipher Lblock", Information Processing Letters, vol. 112, no. 16, **(2012)**, pp. 624-629.

[6] K. Shibutani, T. Isobe and H. Hiwatari, "Piccolo: an ultra-lightweight blockcipher//Cryptographic Hardware and Embedded Systems–CHES", Springer Berlin Heidelberg, **(2011)**.

[7] X.-W. Zhang, J. Hong, C. X. Xiao, "Study on Analysis of AES's S-box and its Improvement", Micro computer information, vol. 18, **(2009)**, pp. 51-52.

[8] Y. P. Li and W. X. Ding, "An Optimal Design of S-box Based on Genetic Algorithm", Journal of chongqing university of science and technology: natural science, vol. 26, no. 2, **(2012)**, pp. 79-85.

[9] H. Chen and D. Feng, "An effective evolutionary strategy for bijective S-boxes//Evolutionary Computation", CEC, Congress IEEE, **(2004)**, vol. 2, pp. 2120-2123.

[10] M. H. Dawson and S. E. Tavares, "An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks", Advances in Cryptology—EUROCRYPT, Springer Berlin Heidelberg, **(1991)**, pp. 352-367.

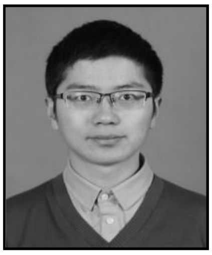
[11] L. Li, R.-F. Li, K.-L. Li, Y. Wang, G. Jiao and Y. Zou, 'Differential power analysis attacks on Present", Application Research of Computers, vol. 3, **(2014)**, pp. 843-845.

[12] D. G. Feng and W. L. Wu, "Design and analysis of block cipher", Tsinghua university press, **(2000)**.

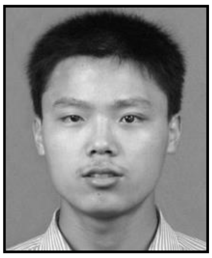
[13] X.-J. Wang, S.-Z. Guo and X.-J. Zhao, "Improved internal template attack on LED block cipher", Journal of huazhong university of science and technology (natural science edition), vol. 12, no. 12, **(2014)**.

[14] A. J. Chipperfield and P. J. Fleming, "The MATLAB genetic algorithm toolbox", Applied control techniques using MATLAB, IEE Colloquium, IET, **(1995)**.

Authors

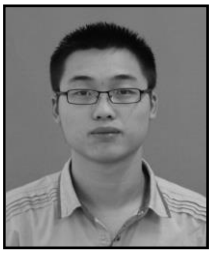


Zhiying Tang, He was born in 1992. Currently he is studying for his B.S. degree in Network Engineering from Anhui University. His research interests include network and information security.



Jie Cui, He was born in 1980, is now an associate professor in the School of Computer Science and Technology, Anhui University. He received PhD degree in University of Science and Technology of China in 2012. He has published 30 papers. His research interests include network and information security.

Hong Zhong (1965-), She is a professor (from 2009), PhD supervisor and dean of the School of Computer Science and Technology, Anhui University, China. She received her PhD degree from University of Science and Technology of China in 2005. Her research interests cover network and information security.



Mingyong Yu, He was born in 1993. Currently he is studying for his B.S. degree in Network Engineering from Anhui University. His research interests include network and information security.