Security Analysis and Enhanced Design of a Dynamic Block Cipher

ZHAO Guosheng[1], WANG Jian[2]

[1] College of Computer Science and Information Engineering, Harbin Normal University, Harbin 150025, China
[2] School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150001, China

Abstract: There are a lot of security issues in block cipher algorithm. Security analysis and enhanced design of a dynamic block cipher was proposed. Firstly, the safety of ciphertext was enhanced based on confusion substitution of S-box, thus disordering the internal structure of data blocks by four steps of matrix transformation. Then, the diffusivity of ciphertext was obtained by cyclic displacement of bytes using column ambiguity function. The dynamic key was finally generated by using LFSR, which improved the stochastic characters of secret key in each of round of iteration. The safety performance of proposed algorithm was analyzed by simulation test. The results showed the proposed algorithm has a little effect on the speed of encryption and decryption while enhancing the security. Meanwhile, the proposed algorithm has highly scalability, the dimension of S-box and the number of register can be dynamically extended according to the security requirement.

Keywords: cryptography; block cipher; encryption algorithm; safety analysis

I. INTRODUCTION

For now, it has been proved that unsafe factors exist in both DES and AES which are advanced encryption standards of cryptography. Ref. [1-2] proposed separately that rectangle attack, boomerang attack and bypass cache attack can be used to crack the DES encryption. During the 1999 RSA conference, scientists successfully found a DES key with exhaustive-key-search-attack method. Advanced encryption standard, that is AES, also known as Rijndael encryption, but now the algorithm has been attacked [3]. Chen gives a new related-key method of AES-192 for square attack by using of the intrinsic relationships of AES-192's sub keys [4]. Dong proposed a differential attack for AES to achieve high performance [5] and then design the 5-round AES-192 attack with key dispatcher and low time complexity[6]. In summary, DES and AES algorithm that are used widely nowadays both exist a variety of security risks.

The following sections is scheduled as listed below. In the second section, the flow chart of algorithm is briefly introduced. The third section introduces the four steps of matrix transformation. The fourth section gives the enhanced design of dynamic S-box. The LFSR is introduced in the fifth section and the safety performance of proposed algorithm was analyzed in sixth section. The last section introduces the summary and future prospects.

In order to accommodate the space-time domain dynamics of mobile data traffics, a new architecture based on SuBS has been proposed for the future WAN.

II. Algorithm Description

Figure 1 shows the work flow of algorithm. Firstly, S-box is initialized by using a key, then read the data blocks (16 bytes is equal to 1 block) circularly, if data blocks less than 16 bytes will be filled with PKCS#7.

The second step is to transform the data block matrix, so that the initial internal data will be shuffled. And then put the resulting data into S-boxes and execute the S-boxes' replacement operation. This operation would greatly enhance the nonlinear strength of final cipher text. The next step, column bytes circulated displacement by confused function, to enhance the diffusion of cipher text. Finally, using n-level interaction linear feedback shift register to generate dynamic keys for enhancing the randomness of key and encrypted data block iteratively. The specific iteration number can be changed according to the demand of encryption. In this paper, we will take the 16 round of iteration as an example.

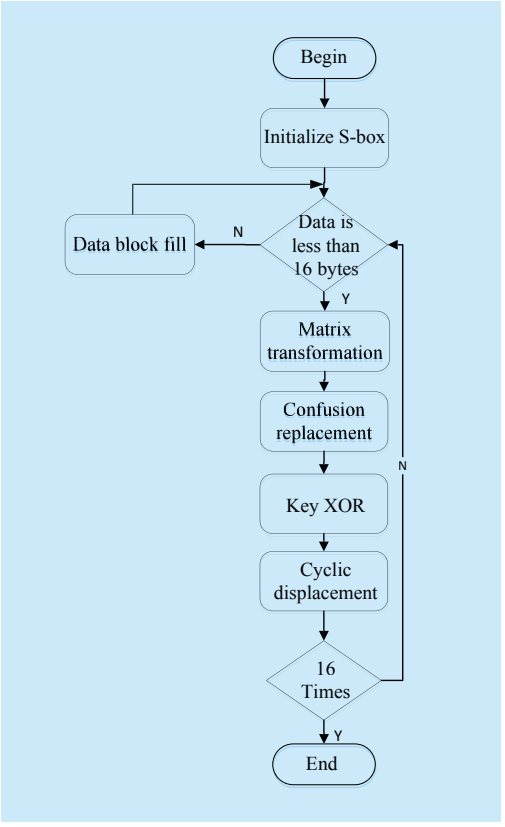

Fig.1 *Work flow of the algorithm*

III. Matrix Transformation

Matrix transformation is the first step of the algorithm. The function of the matrix transformation is used to disrupt the internal structure of the data block, which has high efficiency of encryption and decryption and is also useful to the follow-up encryption structure which can make the encryption work much more well-distributed. In this algorithm, the matrix transformation is a matrix transformation's method that make matrix decompose into a small matrix. The matrix transformation totally includes 4 steps.

First, we will enter the 16-byte data block as the matrix of the 16 elements, the matrix A generate a matrix of 4*4:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (1)$$

Decomposition matrix: the matrix A is decomposed into four sub-matrix of 2*2. They are A [1, 2;1,2], A [3, 4;1,2], A [1, 2;3,4], and A [3, 4;3,4], respectively. It contains these elements:

$$A[1,2;1,2] = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (2)$$

$$A[3,4;1,2] = \begin{bmatrix} a_{31} & a_{32} \\ a_{41} & a_{42} \end{bmatrix} \quad (3)$$

$$A[1,2;3,4] = \begin{bmatrix} a_{13} & a_{14} \\ a_{23} & a_{24} \end{bmatrix} \quad (4)$$

$$A[3,4;3,4] = \begin{bmatrix} a_{33} & a_{34} \\ a_{43} & a_{44} \end{bmatrix} \quad (5)$$

Position transformation of the sub-matrix: the position of the 4 sub matrix in the parent matrix A is used for interchange along the lateral center axis. That is, carrying the first line and third line out the transformation, the second line and fourth line interchange. Corresponding to A[1,2;1,2] and A[3,4;1,2] swap positions, A[1,2; 3,4] and A[3,4; 3,4] switch positions. Transform position of those sub-matrix in A are shown in figure 2.

Diagonal transformation of the sub-matrix: diagonal transformation of elements within the 4 sub matrix, that is, the two elements in the same diagonal swap position. After transform,

the four sub-matrixes are:

$$A[3,4;1,2] = \begin{bmatrix} a_{22} & a_{21} \\ a_{12} & a_{11} \end{bmatrix} \quad (6)$$

$$A[1,2;1,2] = \begin{bmatrix} a_{42} & a_{41} \\ a_{32} & a_{31} \end{bmatrix} \quad (7)$$

$$A[3,4;3,4] = \begin{bmatrix} a_{24} & a_{23} \\ a_{14} & a_{13} \end{bmatrix} \quad (8)$$

$$A[1,2;3,4] = \begin{bmatrix} a_{44} & a_{43} \\ a_{34} & a_{33} \end{bmatrix} \quad (9)$$

Combination of the matrix: after the two steps ahead transformation, the sub matrix is combined into the 4*4 matrix B by (5) - (8) equation. The matrix B is only the position changes of an element relative to the matrix A. The matrix B is:

$$B = \begin{bmatrix} a_{42} & a_{41} & a_{44} & a_{43} \\ a_{32} & a_{31} & a_{34} & a_{33} \\ a_{22} & a_{21} & a_{24} & a_{23} \\ a_{12} & a_{11} & a_{14} & a_{13} \end{bmatrix} \quad (10)$$

The matrix B invert the binary of all elements to final binary matrix, which can be read in sequence.

When decrypting this module, making the cipher text directly as matrix A to run so that the module can be decrypted, because the proof is simple relatively, the paper will not display the provement.

IV. Dynamic S-Box

S-box is a basic structure of the symmetric key algorithm performing replacement calculations [7], which is commonly used fuzzy relationship between the key and the cipher texts. As a non-linear structure, especially as the only certain cryptographic algorithms with nonlinear structure, its strength will directly determine the quality of cryptographic algorithms. Though some transformation S-box[8,9] has been continuously put forward, they have not fundamentally changed the structure of S-box to increase the security of S-box. However, the S-box is given in this paper which is a dynamic S-box based on the multi-dimensional space. The structure of S-box can change dynamically by using of different keys. So that the S-box nonlinear strength greatly improved compared with the previous design.

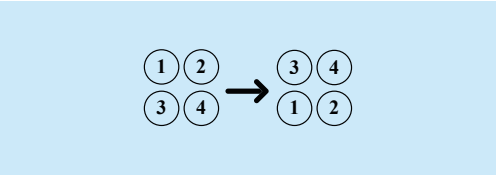

Fig.2 *The process of matrix transformation*

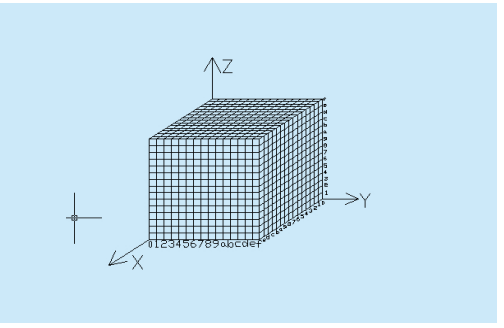

Fig.3 *Three dimensional space of S-box*

Using the present algorithm, you should reasonably choose the dimension of S-box according to the encryption requirements and computing power, because of the dimension of S-box will be influence on the security of algorithms. This paper will be discussed based on three-dimensional S-box as an example. The three-dimensional S-box, which defines a three-dimensional space that is 15 * 15 * 15, as shown in Figure 3. There are a total of 4096 points in the three-dimensional space, point coordinates from (0,0,0) to (F, F, F). The point coordinate (1.5 bytes) is filled into the relevant array as the point content directly. And then, we can transform dynamically the S-box. The three-dimensional S-box contains three two-dimensional surface.

First, we called a plane contained by x-axis, y-axis, b plane contained by x-axis, z-axis and c plane by y-axis, z-axis, which each plane contains 16 floors (each floor contain 256 points), the bottom layer is called the zeroth layer, the top layer is called the F layer. Each layer can be regarded as 8 circles in the rotation (show as Figure 4), and we agreed that all the transformation in the S box rotate in the counterclockwise direction.

Next we introduce the parsing process of key. First, the key should be divided into

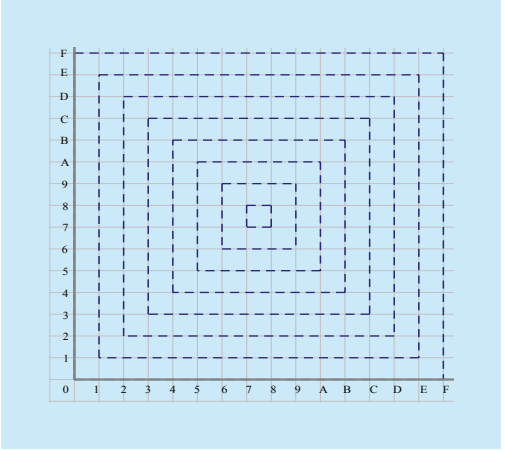

Fig.4 *Each layer can be regard as 8 circle*

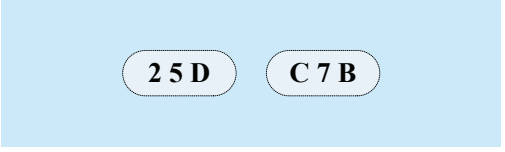

Fig.5 *Groups of the key*

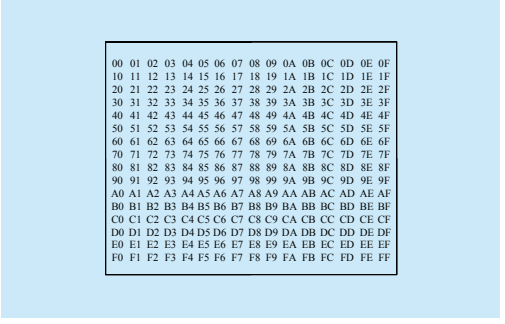

Fig.6 *Before rotated*

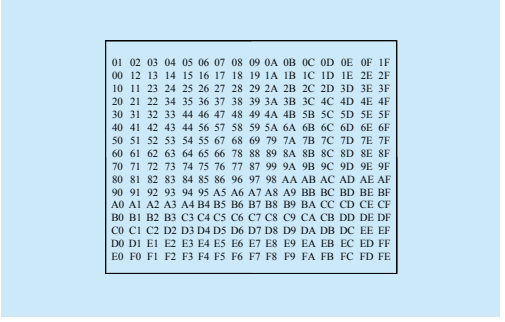

Fig.7 *After rotated*

different groups, 1.5 bytes per group, that is to say 3 characters per group. If the key is "25DC7B"(the actual length of the key should be much more than this example), so the key can be divided into two groups, "25D" and "C7B" (see figure 5).

The first character in each group keys control a plane rotation of S-box; the second character control the b plane rotation of S-box; the third character control the c plane rotation of S-box. While the value of each character is represented by a number of layers, such as the second characters is 3 in some group key, then indicates the third layer of the b plane for S box need to rotate. The position of a character in a key is used to control to rotate number of steps. The 1 step refers to the corresponding layer of relative plane rotate one point by counter-clockwise. For simplicity, the rotation process can be understood as the eight circles of one layer, in which each circle rotates around the central point of the layer by counter-clockwise. Figure 6 shows the data state of a layer in S-box before not rotated. Figure 7 shows the state after rotated one step.

Because the first character in the second group is in the fourth position of the whole key, then the corresponding layer of the corresponding plane is rotated 4 points according to the above rules.

When the rotation is changed, the data (corresponding to the array) is the same, but the location of the point in the space is changed. According to above method initializes the S-boxes, the S-boxes will be chaos, in which the entropy and the data dispersion increase with time, however data correlation is reduced. For example, using the above key "25DC7B", the initialization process of S-box is as follows: S-boxes' second layer of a plane rotates anticlockwise a point; S-boxes' fifth layer of b plane rotates anticlockwise a point; S-boxes' D layer of c plane rotates anticlockwise a point, and so on.

In the practical application, the length of key is at least 8 bytes (16 characters), so the S-box will be well confused. After initialization, the S-boxes can be used to carry out the replacement calculation. The original data received by the S-boxes is divided into a group of 1.5 bytes' data, but it can be split after 3 bytes data is accepted for easy operation. 1.5

bytes data that has accepted can be considered as 3 characters, corresponding to a coordinate in S-boxes. Remove the coordinate data(array) instead of the original data to complete the replacement, that is, the output data take out of the coordinate data to replacement to acquire data. For example: the input data is "34A", then inquire the coordinate (3, 4, A) of S-boxes and extract the corresponding data in the array data, assuming the data is [5B2], then use "5B2" to replace the original input data "34A", "5B2" is the output data.

Due to the S-boxes are a dynamic structure, the algorithm hasn't inverse S-box. When decrypting S-boxes, need to initialize S-boxes as above method, after initialization S-boxes, input 1.5 bytes to need data to decrypt , traversing all data of the S-boxes, find the corresponding coordinates, carry out the replacement operation, and get the original data, output is the plaintext data. For example: the encrypted data is "5B2", then inquire the coordinate (5, B,2) of S-boxes and extract the corresponding data in the array data, assuming the coordinates is (3,4,A), then use "34A" to replace the original input data "5B2", "34A" is the plaintext data.

V. N-Grade Lfsr

LFSR (linear feedback shift register) is the output of a given state and the linear function of the output is used as an input shift register[10,11]. XOR operation is the most common single-bit linear function: after some bits for XOR operation of the register as input, the various bits of again to register the overall shift. This register is called XOR gate by linear feedback shift register. It's principle as shown in figure 8.

The value of g_0, g_1, g_2, …,g_n is 1 or 0. Q_1,Q_2, Q_3,....,Q_n is the output of the LFSR. $M(x)$ is the input of code word polynomial, such as $M(x) = x_4+x_1+1$, represents the input order for 11001.The structure of LFSR can also be expressed as polynomial $G(x)$:

$$G(x)= g_n*x^n+ \ldots+g_1*x^1+ g_0 \qquad (11)$$

The paper uses 4 LFSR, as LFSR0, LFSR1, LFSR2 and control registers as shown in figure 9.

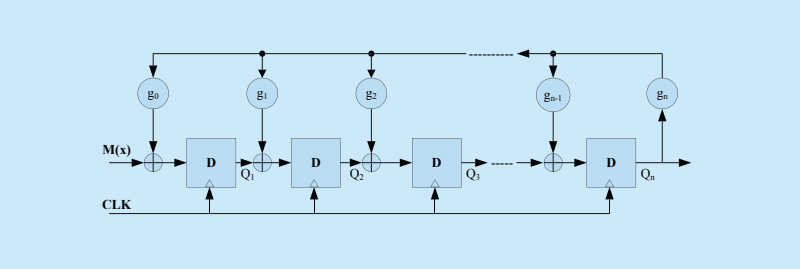

Fig.8 *XOR gate of LFSR*

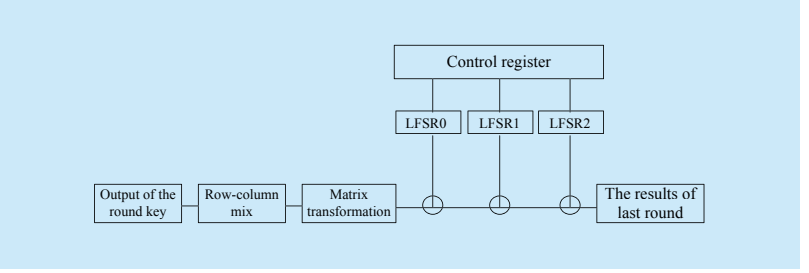

Fig.9 *Key block generator of LFSR*

In this paper, n grade linkage LFSR is a kind of strengthening design relative to the original linear feedback shift register, the role of the encrypted data block to make each round keys for use is not the same, which avoid the cryptanalysis attack for every piece of data block analysis using key effectively. At the same time, the algorithm has the cipher block chaining (CBC, Cipher-Block chaining)[12-13].

If the subscript of first block is 1, the encryption formula is as follow:

$$C_i = E_k(P_i \oplus C_{i-1}), C_0 = IV \qquad (12)$$

Then, the decryption equation is as follow:

$$P_i = D_k(C_i) \oplus C_{i-1}, C_0 = IV \qquad (13)$$

In the encryption formula, a small change in the plaintext will cause a subsequent change in the whole cipher block. However, in the decryption, the plaintext can be obtained from two adjacent cipher block. Therefore, the decryption process can be parallelized, in which, a small change in the ciphertext will only lead to the corresponding plaintext block completely changed, which does not affect to other plaintexts.

The algorithm structure is simple and the CPU instructions used is less, in the case of no

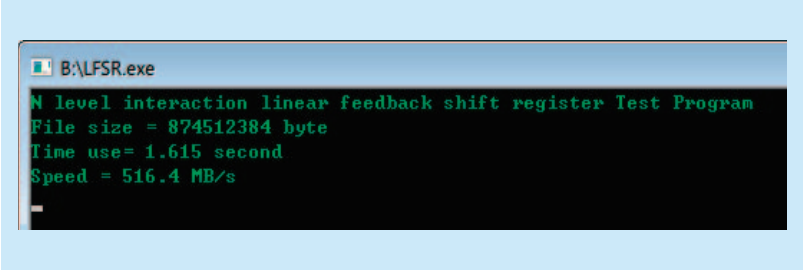

Fig.10 *Speed test of register output*

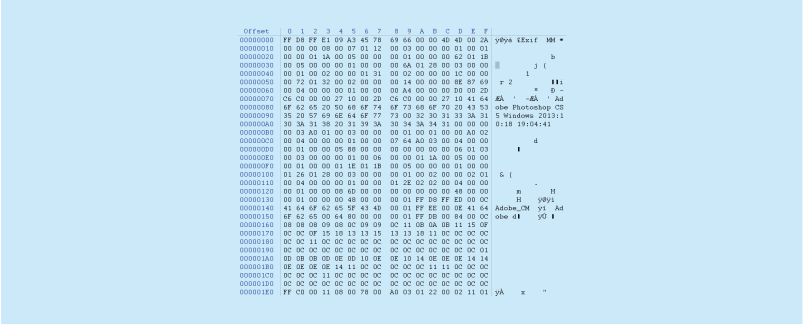

Fig.11 *Before encrypted*

Fig.12 *After encrypted*

disk I/O bottleneck, the output speed could reach 500 M/s (ram disk test, as shown in figure 10). In previous DES and AES, keys are all fixed, easy to be cracked and then obtain plaintexts.

VI. Simulation Results

6.1 Experimental environments

All of the following experiments were run on the platforms of hardware and software:

CPU: Intel® Core™ i5-4200U (up to 2.60 GHz);

RAM:4GB DDR3 1600;

GPU: NVIDIA GeForce GT 740M 2G 64 bit;

HDD: 500GB 5400rpm;

System: Windows 8.1 with Update 1.

The experiment parameters used in this paper are as follows: S-box structure using 4 dimensional dynamic confusion replaced; N grade with 128 bit - 4 level - 3 linear feedback shift register, which the iterations is 16 round.

Comparison with the AES algorithm: 128bit, 16 rounds of iterations with 2-dimension table algorithm. Comparison with DES algorithm: 56bit, 16 rounds of iterations with 2-dimension table algorithm.

6.2 Data difference degree

First, we select a test.jpg file whose size is 115KB to test the encryption effects. In this paper, the data difference degree is used to express the comparison. Figure 11 and Figure 12 shows the same part of test.jpg before and after encryption. From the comparisons of two images, we can find the data differences between plaintext and cipher text is great well degree.

6.3 Correlation coefficient

Correlation is a non-deterministic relationship, which researches the linear correlation degree between variable quantity[14,15]. A good cryptographic algorithm should make the correlation coefficient between plaintext and cipher text is very low. We use the method of Pearson correlation coefficient to calculate the correlation coefficient of two sets of data before encryption and after encryption.

Pearson is usually used to measure the linear relationship between a constant variable. When two variables are both normal continuous variables and linear relationship, the degree of correlation between two variables can be calculated. The value of correlation coefficient and meaning as shown in the Table I.

Calculate the correlation coefficient of Pearson formula is:

$$r = \frac{\Sigma XY - (\Sigma X)(\Sigma Y)/N}{\sqrt{\Sigma X^2 - (\Sigma X)^2/N} \bullet \sqrt{\Sigma Y^2 - (\Sigma Y)^2/N}} \quad (14)$$

The data were obtained before and after the experiment, and the data were used as the

sample into the above formula that we can figure out the correlation coefficient is 0.04232 (see table II).

By looking up Table I shows the correlation coefficient is 0.04232, which indicates that the data before and after encryption is no correlation. It suggests that the attacker is difficult to deduce the relationship between plaintext and ciphertext by statistical attack methods, which confirms the algorithm has a high nonlinear strength to resist linear cryptanalysis attacks.

6.4 Differential cryptanalysis

Differential cryptanalysis [16,17] is a optional plaintext attack method which was put forward by Biham and Shamir in 1991. The basic idea is that through the analysis of the specific difference influence on plaintext and ciphertext to get as far as possible key. It can be used to attack any of the password structure with fixed iteration rounds and most block ciphers.

Differential analysis involves with some comparisons about characteristics of the ciphertext and plaintext which analysts looking for the special difference of ciphertext pair. Some of these differences have a higher probability of recurrence, differential analysis using these features to calculate the probability possible key and confirm the most possible keys at last. The kind of attacks are largely dependent on the structure of S-boxes. In addition, the round of encryption is large on the effect of difference analysis relatively. If DES is only used in the 8 round, it will only take a few minutes on a personal computer to decipher.

General procedure differential analysis was attacker change a character of obtained the encrypted of an arbitrary change of a character to obtain a new data, then encrypted the data, observe transformation results. Change rate of a number of bytes (CRB) and the change of average density (CAD) are the two indexes to measure the differential attacks. The former describes the number ratio of bytes in corresponding position to two encrypted data, the second parameters describes the difference degree of bytes. The calculated results are shown in table III.

Table I *Value of correlation coefficient*

Values	Meaning
0-0.09	no correlation
0.1-0.3	low correlation
0.3-0.5	medium correlation
0.5-1.0	strong correlation

Table II *Correlation analysis results*

		Before	After
Before	Pearson correlation	1	.072
	Significant (both sides)		.363
	N	160	160
After	Pearson correlation	.072	1
	Significant (both sides)	.363	
	N	160	160

Table III *Index of differential cryptanalysis*

	Encryption 1 times	Encryption 2 times
CRB	0.632	0.873
CAD	0.492	0.573

Table 3 shows that with the increase of encryption times, the value of CRB and CAD is on the rise, which indicate that small changes in the encrypted data can cause great changes in the ciphertext. It can be seen that the proposed algorithm can effectively resist the outside differential attacks.

6.5 Resist side-channel-attacks

Side-channel-attack [18] is an attack based on bypass information, using cryptanalysis techniques and the leakaged information of password device to restore the keys being used. Bypass attacks can bypass the complex analysis of the encryption algorithm, use the leaked information of cryptographic algorithms in hardware implementation operation and then rapidly break password system based on statistical theory. Because the bypass attacks is one of the most popular key break means, so the test whether the cipher algorithm can resist side channel attack is very necessary. Through testing the power and the difference time between the right key and the wrong key to detect password algorithm whether can resist

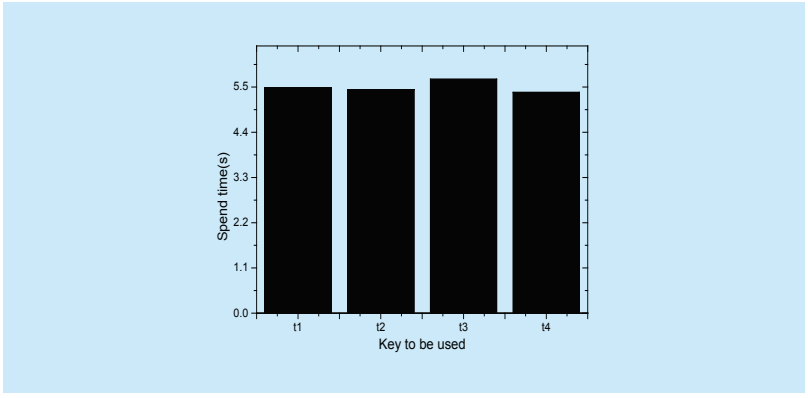

Fig.13 *Decryption keys and spend time*

Table IV *Decryption keys and time*

	Key	Time spent (s)
t1	The right key	5.48
t2	The same as the right key figures wrong key	5.44
t3	Than the right key figures more wrong key	5.69
t4	Less than the right key figures wrong key	5.37

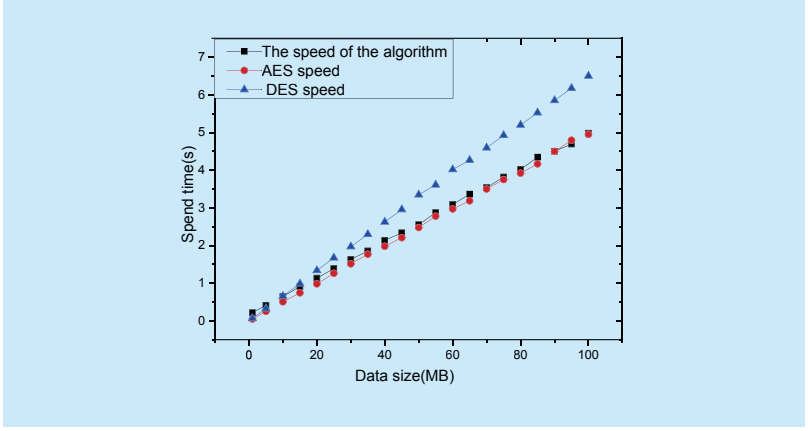

Fig.14 *Encryption time of small data*

side channel attacks, the difference degree the greater, the resistance to bypass attack cipher algorithm worse.

We use a 100MB data randomly generated to encrypt key, then we use the following keys to decrypt the encrypted data. Table Ⅳ shows the corresponding relationships between the used key and the spend time of decryption data. Figure 13 uses a column chart to compare the test results several times. We can see from table Ⅳ and figure 13 that the time spent on the encryption and decryption of data is basically the same as the time spent on the correct key and the wrong key, and the upper and lower range can be controlled in ±0.32s. Through this experiment we can prove the proposed algorithm has better ability to resist the bypass attack.

6.6 Performance analysis

A good cryptographic algorithm should not only be safe and reliable but also should have certain efficiency [19,20].Efficiency of cryptographic algorithm refers to the amount of encryption(decryption) in unit time and it also can be simply understood as the speed of encryption(decryption) data for the cryptographic algorithm.

In this experiment, we respectively use AES algorithm, DES algorithm and the proposed algorithms with different size of data to compare the time and speed of encryption(-decryption). Figure 14 shows the spend time of three kinds of cryptographic algorithm for small data. The abscissa represents the size of encrypted data; the ordinate represents the spend time of encrypted data. Then, we use v=ds/dt to work out the encryption speed of three kinds of algorithms for each sample, and connected to the line chart (see Figure 15). Figure 16 shows the encryption time of three kinds of algorithms for big data. Figure 17 shows the encryption speed of three kinds of algorithms for big data.

In the same way, we use three kinds of algorithm to decrypt the data which we have encrypted before, for comparing the decryption efficiency. Figure 18 and figure 19 respectively shows the decryption time of three kinds of algorithm spent on small sample data and corresponding decryption speed, and then figure 20 and figure 21 show the decryption time and decryption speed that three kinds of encryption algorithms spend on big data decrypted.

From figure 14 and figure 15, we can see that the proposed algorithm in the encrypted small data is slower, spend more time. The main reasons are the proposed algorithm needs to be initialized before encryption and uses a dynamic key to generate the inquired

S-boxes. For small data, the proportion of the initialization time is relatively larger, resulting in a slower encryption speed and a long time to spend. In Figure 15, we can see that with the increase of data, the encryption speed of the proposed algorithm will also speed up, when the data's size reached 10MB, the proposed algorithm's speed is faster than DES algorithm, and when the data's size reached 40MB, the proposed algorithm speed and AES algorithm's speed is equal, and the speed is still keep a growing trend.

From figure 16 and figure 17, we can see that for large data, the encryption time spent by the proposed algorithm is lower than AES algorithm, and much less than DES algorithm, and the encryption speed is slightly higher than AES algorithm, and it is higher than the DES algorithm. In addition, the curve of encryption speed in figure 17 has some fluctuates, which is mainly impacted by the contents of data blocks, however, the range of fluctuates always keep within ±2.5, the average encryption speed is 21.33MB/s, minimum speed is 18.29MB/s.

From figure 18 and figure 19, we can see that the proposed algorithm in the decrypted small data is slower, spend more time. The main reasons are the proposed algorithm needs to be initialized before decryption and uses a dynamic key to generate the inquired S-boxes. For small data, the proportion of the initialization time is relatively larger, resulting in a slower decryption speed and a long time to spend.

In Figure 19, we can see that with the increase of data, the decryption speed of the proposed algorithm will also speed up, when the data's size reached 20MB, the proposed algorithm's speed is faster than DES algorithm, and when the data's size reached 40MB, the proposed algorithm speed will be close to AES algorithm's speed.

From figure 20 and figure 21, it can be seen that the decryption speed and time of the proposed algorithm is close to AES. In addition, the curve of decryption speed in figure 19 has also some fluctuates , which is mainly impacted by the contents of data blocks, however, the range of fluctuates always keep within ±2, the average decryption speed is 20.02MB/s, minimum speed is 17.41MB/s.

From figure 15, figure 17, figure 19 and figure 21, it can be seen that the encryption speed

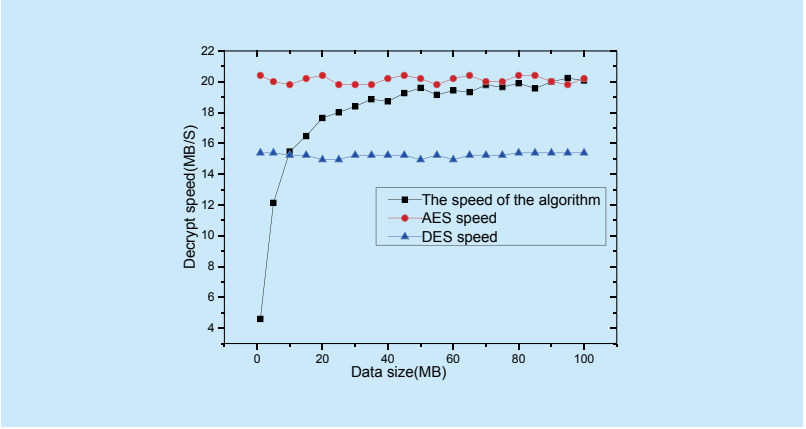

Fig.15 *Encryption speed of small data*

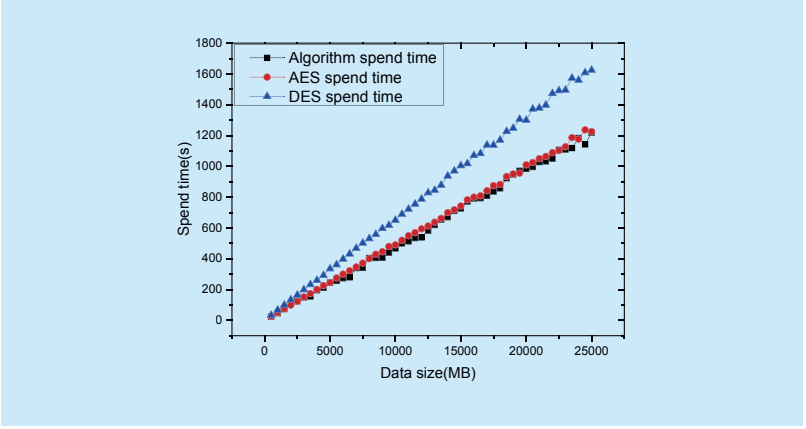

Fig.16 *Encryption time of big data*

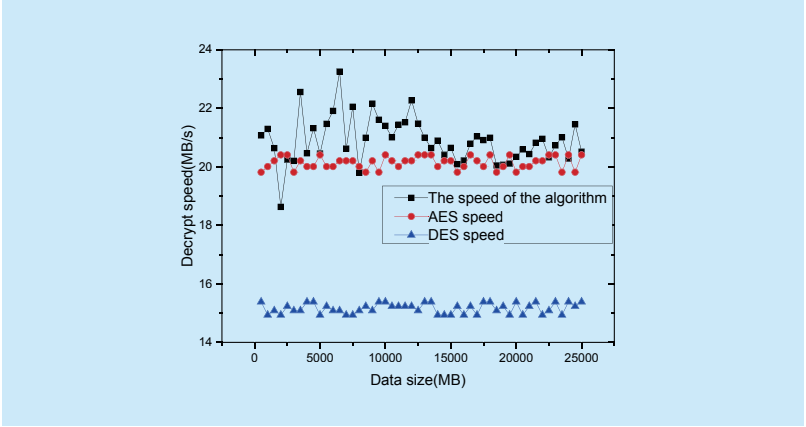

Fig.17 *Encryption speed of big data*

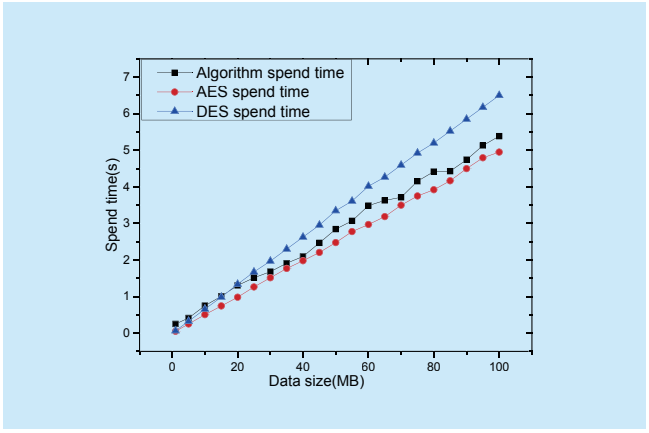

Fig.18 *Decryption time of small data*

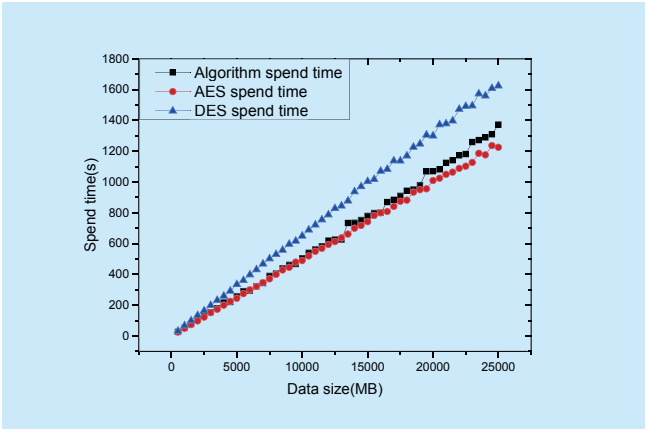

Fig.20 *Decryption time of big data*

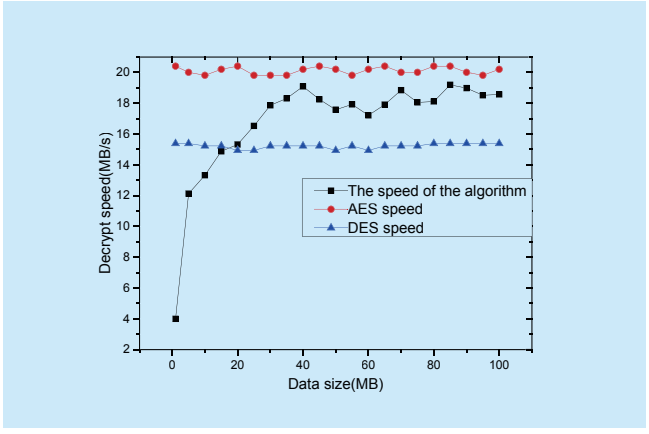

Fig.19 *Decryption speed of small data*

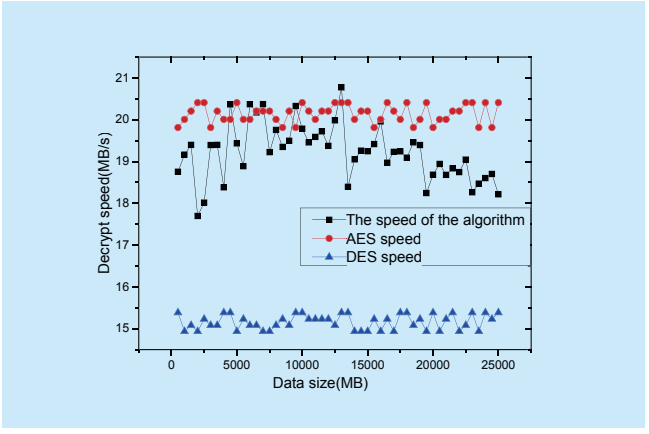

Fig.21 *Decryption speed of big data*

of the proposed algorithm is faster than the decryption speed in the same size data. This is because the proposed algorithm need traverse the S-box, so the decryption speed is slower.

VII. Conclusions

This paper presented a set of block cipher algorithm which was made of multidimensional and dynamical S-box disordered replacement and n-grade linear feedback of shifting register supplemented with the structure of matrix transformation and cyclic displacement and so on. Through experiments, the algorithm is proved to be effective. At the same time that the algorithm was a scalable algorithm, the number of register and dimensionality of S-box can be selected according to the need for encryption. Those parameters can change the safety of algorithm and then is suitable for different applications.

However, the algorithm has some deficiencies, such as odd dimensionality of S-box is difficult to achieve and small data is slow to manipulate and so on. The next step work is to further optimize the bytes split of odd dimension S-box and the encryption and decryption speed of small byte data.

Acknowledgements

I would like to express sincere gratitude to Dr. Liu Hailong for his constructive suggestions, in particular, should thank Li Guangcheng student for his work in simulation and programming, but also feel grateful to Wu Jie and Jiang Wanqiu for their good english translation. This work was supported in part by National

Natural Science Fundation of China under Grant No.61202458, 61403109.

References

[1] J Kim,S Hong,B Preneel,et al, "related-key boomerang and rectangle attacks:theory and experimental analysis", *IEEE Transactions on Information Theory*, vol.58, no.7, pp 4948-4966, July, 2012.

[2] D Bai, H.B Yu, G Wang, et al, "improved boomerang attacks on round-reduced SM3 and keyed permutation of BLAKE-256", *IET Information Security*, vol.9, no.3, pp 167-178, March, 2015.

[3] O Dunkelman,N Keller,A Shamir, "improved single-key attacks on 8-round AES-192 and AES-256", *Journal of Cryptology*, vol.28, no.3, pp 397-422, March, 2015.

[4] J Chen, Y.P Hu, Y.Y Zhang, et al, "related-key square attack on AES-192", *Journal of University of Electronic Science and Technology of China*, vol.42, no.2, pp 219-224, February, 2013.

[5] X.L Dong, Y.P Hu, J Chen, "new key-sieving algorithm in impossible differential attacks on AES", *Journal of University of Electronic Science and Technology of China*, vol.40, no.3, pp 396-400, March, 2011.

[6] X.L Dong, Y.P Hu, Y.Z Hui, et al, "a new method for meet-in-the-middle attacks on reduced AES", *China Communications*, vol.8, no.2, pp 21-25, February, 2011.

[7] M Ahmad, H Haleem, P.M Khan, "a new chaotic substitution box design for block ciphers", *Proceedings of the 1st International Conference on Signal Processing and Integrated Networks*, Noida, India, IEEE Press, pp 255-258, February 20-21, 2014.

[8] G.Q Liu, C.H Jin. "investigation on construction and differential property of a class of dynamic S-Box", *Journal Of Electronics & Information Technology*, vol.36, no.1, pp 74-81, January, 2014.

[9] S Gao, W.P Ma, N Guo, et al, "novel method for increasing the nonlinearity of S-Boxes", *Journal of Xidian University(Natural Science)*, vol.37,-no.6,pp 1017-1021,June,2010.

[10] E Ali, M Fardous, A.S Taha, et al, "dynamic linear feedback shift registers: a review", *Proceedings of the 5th International Conference on Information and Communication Technology for the Muslim World*, Kuching, Malaysia, pp 1-5, November 17-18, 2014.

[11] J.H Zhong, D.D Lin, "a new linearization method for nonlinear feedback shift registers", *Journal of Computer and System Sciences*, vol.81, no.4, pp 783-796, April, 2015.

[12] D Zheng, Q.L Zhao, Y.H Zhang, "a brief overview on cryptography", *Journal of Xi'An University Of Posts and Telecommunications*, vol.18, no.6, pp 1-10, June, 2013.

[13] W.L Wu, D.G Feng, "The State-of-The-Art of research on block cipher mode of operation", *Chinese Journal of Computers*□vol.29, no.1, pp 21-36, January, 2006.

[14] J.Y Xie, H.C Gao, "statistical correlation and k-means based distinguishable gene subset selection algorithms", *Journal of Software*, vol.25, no.9, pp 2050-2075, September, 2014.

[15] H Jiang, W.M Deng, X.Q Chen, "analysis of wine based on pearson coefficient and multiple kernel support vector classification", *Transactions of the Chinese Society for Agricultural Machinery*, vol.45, no.1, pp 203-208, January, 2014.

[16] L Ding, J Guan, "differential cryptanalysis of trivium stream cipher based on automatic deduction", *Acta Electronica Sinica*, vol.42, no.8, pp 1647-1652, August, 2014.

[17] H.R Wei, G.L Yin, "related-key impossible differential cryptanalysis on Iblock", *Journal of Computer Research and Development*, vol.51, no.7, pp 1520-1526, July, 2014.

[18] Y Zhang, K.Y Chen, X.W Li, et al, "side channel attack of cipher chips based on difference variability", *Tongxin Xuebao/Journal on Communications*, vol.36, no.3, pp 61-66, March, 2015.

[19] D.G Feng, "Status Quo and Trend of Cryptography", *Journal of China Institute of Communications*, vol.23, no.5, pp 18-26, May,2002.

[20] W Li, D.W Gu, C Zhao,et al, "security analysis of the LED lightweight cipher in the internet of things", *Chinese Journal of Computers*, vol.35, no.3, pp 424-445, March, 2012.

Biographies

***ZHAO Guosheng*,** received the M.S. and Ph.D. degrees in college of computer science and technology from Harbin Engineering University, Harbin, China, in 2005 and 2009 respectively. He has been a faculty member of College of Computer Science and Information Engineering at Harbin Normal University since 2001, where he is currently a associate professor. His major research interests include survivable system, cognitive network and autonomous computing. He was supported by the National Natural Science Foundation of China in 2013. *The corresponding author. Email: zgswj@163.com

***WANG Jian*,** received her Ph.D. degree in college of computer science and technology from Harbin Engineering University, Harbin, China, in 2009. She is currently a associate professor in the School of Computer Science and Technology, Harbin University of Science and Technology. Her research interests include survivable system, cognitive network and autonomous computing. She was supported by the National Natural Science Foundation of China in 2015.Email: wangjianlydia@163.com