A Student Information Management System Based on Fingerprint Identification and Data Security Transmission

Pengtao Yang, Guiling Sun, Jingfei He, Peiyao Zhou, Jiangjiang Liu
College of Electronic Information and Optical Engineering, Nankai University, Tianjin 300350, China
Mailing Address: Guiling Sun, Nankai University, No.38 Tongyan Road, Jinnan District, Tianjin, China 300350
Correspondence should be addressed to Guiling Sun: sungl@nankai.edu.cn

Abstract: In this paper, a new type of student information management system is designed to implement student information identification and management based on fingerprint identification. In order to ensure the security of data transmission, this paper proposes a data encryption method based on an improved AES algorithm. A new S-box is cleverly designed, which can significantly reduce the encryption time by improving ByteSub, ShiftRow and MixColumn in the round transformation of the traditional AES algorithm with the process of look-up table. Experimental results show that the proposed algorithm can significantly improve the encryption time compared with the traditional AES algorithm.

Key Words: Information Management System, Data Encryption, Advanced Encryption Standard Algorithm, Optimization Algorithm

1 Introduction

At present, there are a large number of college students, so the identification and verification of student identity information occurs at all times in the campus, as well as the corresponding services given by the students' identification. Therefore, safe and efficient student information management, convenient identification to obtain the required service, safe and reliable information transmission have become an important task for the student information management[1-3]. Three main features of the proposed system are the following:

1) This system uses the fingerprint identification terminal to collect the fingerprint information. By means of replacing the campus card with the physiological characteristics of lifelong invariance, uniqueness and convenience, it has become the basis of student identity authentication. The maturity of the fingerprint identification technology ensures the safety and speed of the process, and also eliminates the disadvantages of the campus card which is easy to be stolen, forged and easily lost.

2) In order to ensure the safety of the students' information, the fingerprint characteristic value is encrypted and transmitted, using the improved AES encryption algorithm[17], which has the same security guarantee with traditional AES algorithm but, reduces the required time for encryption. Therefore, this student management system not only is convenient for students in the college, but also protects the privacy of students.

3) After the system has been built, because of its easy to maintain and popularize, the modular system design is easier to improve, and it can be widely used in other fields.

2 Description of the Student Information Management System

The system is mainly composed of two parts: terminal and host computer. The terminal is composed of fingerprint identification module and micro controller. The host computer can use personal computers or large servers according to the number of users, and the management of student information database uses SQL Server. The terminal fingerprint

sensor uses optical fingerprint recognition module, while the microcontroller uses STM32F4, with 192KB of SRAM[4]. Each terminal processes and encrypts the collected fingerprint data, and then transmits it to the host computer. In the host computer database for storage in order to manage, and only in the terminal data collection and not stored, so that the security of the data is guaranteed. The system structure is shown in Figure 1.

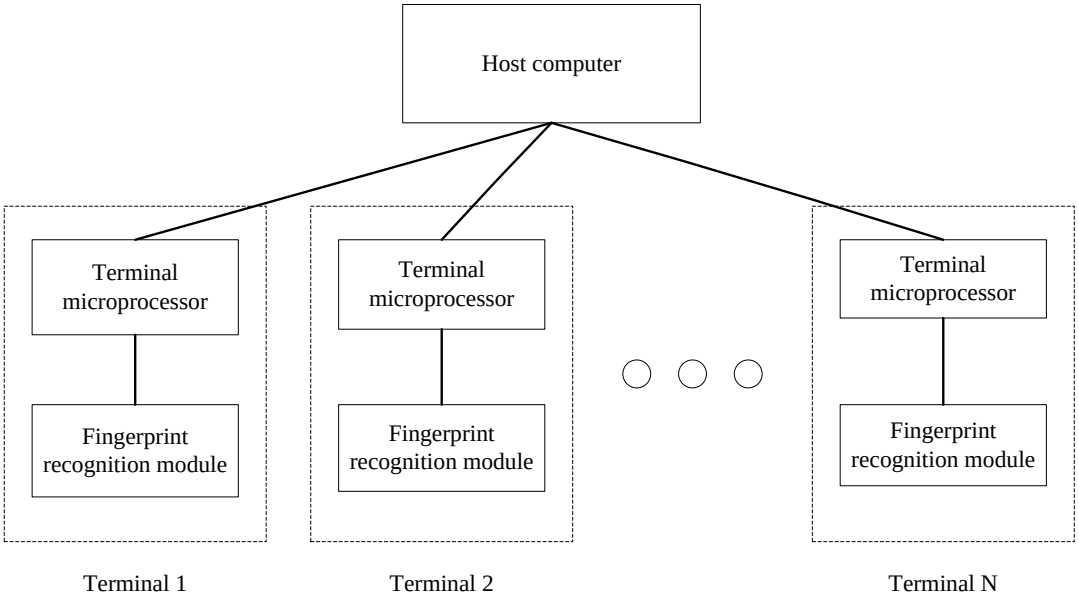

Figure 1: System structure diagram.

3 Implementation of the Student Information Management System

The system collects fingerprints through the terminal fingerprint identification sensor. And the microprocessor processes and encryptions the fingerprint information, then transmits it to the server. On the server side, it compares the fingerprint information transmitted from terminal with the fingerprint information stored in the server database. If the identity is consistent, the user is allowed to operate by verification. The overall process is shown in Figure 2.

4 Data Transmission Encryption Method

In order to achieve the campus student consumption, identity registration and other functions, the student information identification management system based on fingerprint identification and data security transmission needs transmit student fingerprint information, identity information and bank card information among the terminal. There is a risk of being intercepted during data transmission. Students' private information has a high commercial value, once intercepted by criminals, the consequences could be disastrous. If the use of plaintext transmission, security is very low, therefore, the entire data transmission using ciphertext transmission, to achieve a plaintext view, ciphertext transmission effect, greatly improving the security, so that criminals cannot take the opportunity. In order to ensure the security of encrypted transmission and user-friendly, the encryption process uses the optimized AES algorithm.

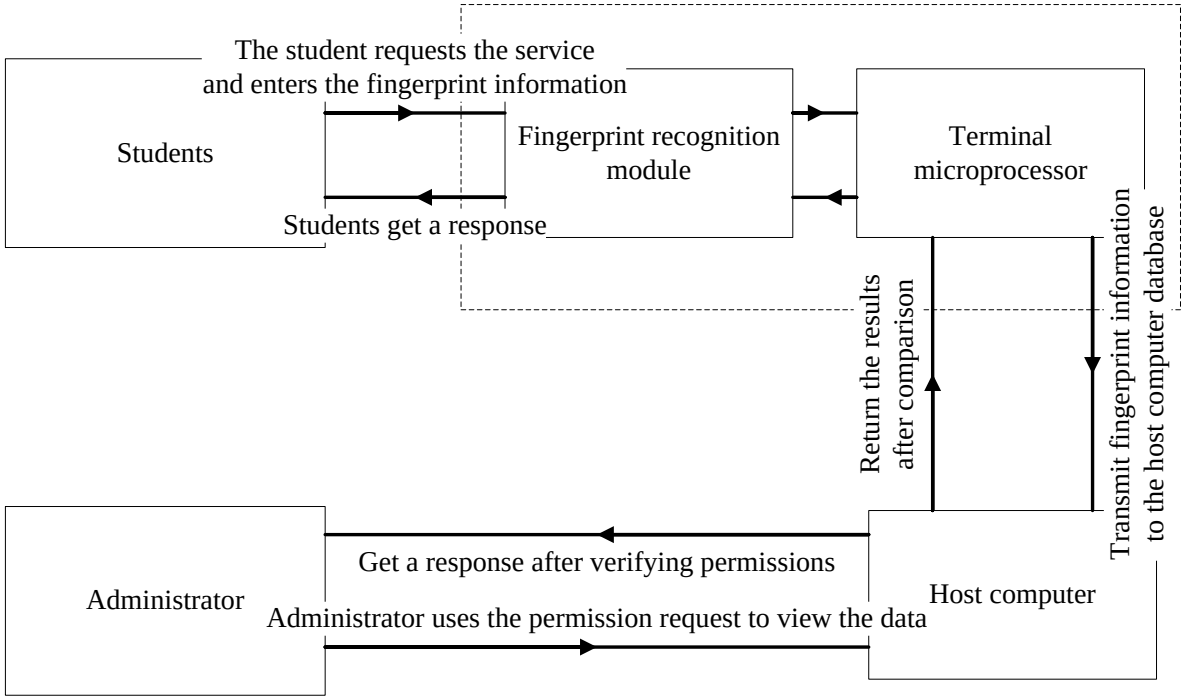

Figure 2: System flow diagram.

AES algorithm is a variable data block length and variable key length iterative block cipher algorithm, and the length of the data block and the key length can be 128, 192 or 256 bits[5]. The most important operation in the AES algorithm is the round transformation operation, where the various operations applied to the process give a high encryption strength. The round transformation operation consists of four steps: ByteSub, ShiftRow, MixColumn, and AddRoundKey, and these steps will be mathematically transformed to eventually construct a new S-box.[6-7]

4.1 Matrix Representation of AES Algorithm Round Transformation

AES algorithm mainly consists of three modules: encryption module, decryption module and key expansion module. Each round transformation of the encryption module consists of ByteSub, ShiftRow, MixColumn and AddRoundKey four operations[8]. The decryption module is also composed of four similar operations, the difference is that ByteSub, ShiftRow, MixColumn is the inverse operation of the encryption module. And the extension key used in AddRoundKey is generated by the key expansion module. The encryption module and the decryption module are the core of the AES algorithm, which are the repetition process of the round transformation, so the simplified round function can improve the operation speed of the AES algorithm. [9-10]

For convenience of description, 128 bits (16 bytes) data is used here and the key is 128 bits.

In the ByteSub transformation, it is assumed that the input is A, $A=\left[a_{i,j}\right],\left(0\le i,j\le 3\right)$, output is B, $B=\left[b_{i,j}\right],\left(0\le i,j\le 3\right)$. ByteSub transformation can be expressed as:

$$B=\left(A\right) \tag{1}$$

And it can also written as:

$$b_{i,j}=B\left(a_{i,j}\right) \tag{2}$$

In practice, this transformation can be converted to look-up table operation. The table is the AES algorithm byte conversion table, also known as S box.

In the ShiftRow transformation, the schematic diagram shown in Figure 3. It is assumed that the output is C, $C=\left[c_{i,j}\right],\left(0\le i,j\le 3\right)$.

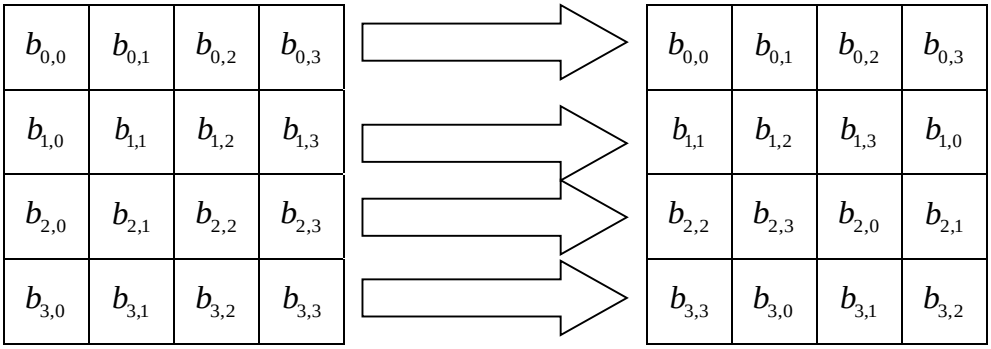

Figure 3: ShiftRow transformation schematic diagram.

Then C can be expressed as a matrix:

$$\begin{bmatrix} c_{0,j} \\ c_{1,j} \\ c_{2,j} \\ c_{3,j} \end{bmatrix} = \begin{bmatrix} b_{0,(j+0)\%4} \\ b_{1,(j+1)\%4} \\ b_{2,(j+2)\%4} \\ b_{3,(j+3)\%4} \end{bmatrix} \quad (3)$$

In the MixColumn transformation, each column of the state array obtained in ShiftRow is treated as a polynomial on $GF(2^8)$, and modulo x^4+1 multiplication with a fixed polynomial $03x^3+01x^2+01x+02$.

It is assumed that the output is D, $D=\left[d_{i,j}\right],(0\le i,j\le 3)$, then MixColumn can also be written as matrix multiplication [11-13]:

$$\begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} c_{0,j} \\ c_{1,j} \\ c_{2,j} \\ c_{3,j} \end{bmatrix} \quad (4)$$

In the AddRoundKey transformation, the expansion round key generated by the key expansion module begins to function. Set the round key to K, $K=\left[k_{i,j}\right],(0\le i,j\le 3)$. Set the output to E, $E=\left[e_{i,j}\right],(0\le i,j\le 3)$. Then AddRoundKey can be expressed as a matrix:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix} \quad (5)$$

(2), (3), (4) into (5) can get:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} S\left[a_{0,(j+0)\%4}\right] \oplus \begin{bmatrix} 03 \\ 02 \\ 01 \\ 01 \end{bmatrix} S\left[a_{1,(j+1)\%4}\right] \oplus \begin{bmatrix} 01 \\ 03 \\ 02 \\ 01 \end{bmatrix} S\left[a_{2,(j+2)\%4}\right] \oplus \begin{bmatrix} 01 \\ 01 \\ 03 \\ 02 \end{bmatrix} S\left[a_{3,(j+3)\%4}\right] \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix} \quad (6)$$

Above we have come to a matrix representation between input A and output E of each round transformation of AES algorithm. [14-16]

4.2 Optimized AES algorithm

In equation (6), in order to calculate $\begin{bmatrix} 02 \\ 01 \\ 01 \\ 03 \end{bmatrix} S\left[a_{0,(j+0)\%4}\right]$ requires one xtime[17] operation and one exclusive-OR operation. Thus, getting each

column vector of a round transformation result E requires four xtime operations and eight exclusive-OR operations (regardless of round key generation). According to the observation we can see that in the column vector multiplied by $S\left[a_{0,(j+0)\%4}\right]$, $S\left[a_{1,(j+1)\%4}\right], S\left[a_{2,(j+2)\%4}\right], S\left[a_{3,(j+3)\%4}\right]$, only the three elements: 01, 02, 03. So we can create a new S box to get directly each element in the $\begin{bmatrix}02\\01\\01\\03\end{bmatrix} S\left[a_{0,(j+0)\%4}\right]$、$\begin{bmatrix}03\\02\\01\\01\end{bmatrix} S\left[a_{1,(j+1)\%4}\right]$、$\begin{bmatrix}01\\03\\02\\01\end{bmatrix} S\left[a_{2,(j+2)\%4}\right]$、$\begin{bmatrix}01\\01\\03\\02\end{bmatrix} S\left[a_{3,(j+3)\%4}\right]$ four column vectors by look-up table method, so that we can save four xtime operations and four exclusive-OR operations and get each column vector of a round transformation result E requires only four exclusive-OR operations (regardless of round key generation). Let data in the original S box operate respectively with 01,02,03, and get a new byte conversion table, as shown in Table 1:

Table 1: The byte conversion table of optimized AES algorithm.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	63	7c	77	7b	f2	6b	6f	5c	30	01	67	2b	fe	d7	ab	76
0	c6	f8	ee	f6	ff	d6	de	91	60	02	ce	56	e7	b5	4d	ec
	a5	84	99	8d	0d	bd	b1	54	90	03	a9	7d	19	62	e6	9a
	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
1	8f	1f	89	fa	ef	b2	8e	fb	41	b3	5f	45	23	53	e4	9b
	45	9d	40	87	15	eb	c9	0b	ec	67	fd	ea	bf	f7	96	5b
	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
2	75	e1	3d	4c	6c	7e	f5	83	68	51	d1	f9	e2	ab	62	2a
	c2	1c	ae	6a	5a	41	02	4f	5c	f4	34	08	93	73	53	3f
	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
3	08	95	46	9d	30	37	0a	2f	0e	24	1b	df	cd	4e	7f	ea
	0c	52	65	5e	28	a1	0f	b5	09	36	9b	3d	26	69	cd	9f
	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
4	12	1d	58	34	36	dc	b4	5b	a4	76	b7	7d	52	dd	5e	13
	1b	9e	74	2e	2d	b2	ee	fb	f6	d4	61	ce	7b	3e	71	97
	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
5	a6	b9	02	c1	40	e3	79	b6	d4	8d	67	72	94	98	b0	85
	f5	68	02	2c	60	1f	c8	ed	be	46	d9	4b	de	d4	e8	4a
	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
6	bb	c5	4f	ed	86	9a	66	11	8a	e9	04	fe	a0	78	25	4b
	6b	2a	e5	16	c5	d7	55	94	cf	10	06	81	f0	44	ba	e3
	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
7	a2	5d	80	05	3f	21	70	f1	63	77	af	24	20	e5	fd	bf
	f3	fe	c0	8a	ad	bc	48	04	df	c1	75	63	30	1a	0e	6d
	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
8	81	18	26	c3	be	35	88	2e	93	55	fc	7a	c8	ba	32	e6
	4c	14	35	2f	e1	a2	cc	39	57	f2	82	47	ac	e7	2b	95

	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
9	c0	19	9e	a3	44	54	3b	0b	8c	c7	6b	28	a7	bc	16	ad
	a0	98	d1	7f	66	7e	ab	83	ca	29	d3	3c	79	e2	1d	76
	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
a	db	64	74	14	92	0c	48	b8	9f	bd	43	c4	39	31	d3	f2
	3b	56	4e	1e	db	0a	6c	e4	5d	6e	ef	a6	a8	a4	37	8b
	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
b	d5	8b	6e	da	01	b1	9c	49	d8	ac	f3	cf	ca	f4	47	10
	32	43	59	b7	8c	64	d2	e0	b4	fa	07	25	af	8e	e9	18
	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
c	6f	f0	4a	5c	38	57	73	97	cb	a1	e8	3e	96	61	0d	0f
	d5	88	6f	72	24	f1	c7	51	23	7c	9c	21	dd	dc	86	85
	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
d	e0	7c	71	cc	90	06	f7	1c	c2	6a	ae	69	17	99	3a	27
	90	42	c4	aa	d8	05	01	12	a3	5f	f9	d0	91	58	27	b9
	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
e	d9	eb	2b	22	d2	a9	07	33	2d	3c	15	c9	87	aa	50	a5
	38	13	b3	33	bb	70	89	a7	b6	22	92	20	49	ff	78	7a
	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16
f	03	59	09	1a	65	d7	84	d0	42	29	5a	1e	7b	a8	6d	2c
	8f	f8	80	17	da	31	c6	b8	c3	b0	77	11	cb	fc	d6	3a

In the use of C language to implement, the table will be set to a two-dimensional array $S_{new}[256][3]$, so that we can get each element of

$\begin{bmatrix}02\\01\\01\\03\end{bmatrix} S\left[a_{0,(j+0)\%4}\right]$、$\begin{bmatrix}03\\02\\01\\01\end{bmatrix} S\left[a_{1,(j+1)\%4}\right]$、

$\begin{bmatrix}01\\03\\02\\01\end{bmatrix} S\left[a_{2,(j+2)\%4}\right]$、$\begin{bmatrix}01\\01\\03\\02\end{bmatrix} S\left[a_{3,(j+3)\%4}\right]$ four

column vectors by look-up table method. For example, in $\begin{bmatrix}02\\01\\01\\03\end{bmatrix} S\left[a_{0,(j+0)\%4}\right]$, the lower four bits and higher four bits of $a_{0,(j+0)\%4}$ correspond separately to the abscissas and ordinates of the table, so that we get the row coordinates of the two-dimensional array, which is equivalent to determining which grid in above table. The 2,1,1,3 of the column vector correspond separately to the 1,0,0,2 in two-dimensional array column coordinates, which is equivalent to determining which element of the grid in above table. The optimized AES encryption algorithm flow chart is shown in Figure 4.

Likewise, a similar new byte conversion table can be created at the time of decryption to achieve decryption optimization.

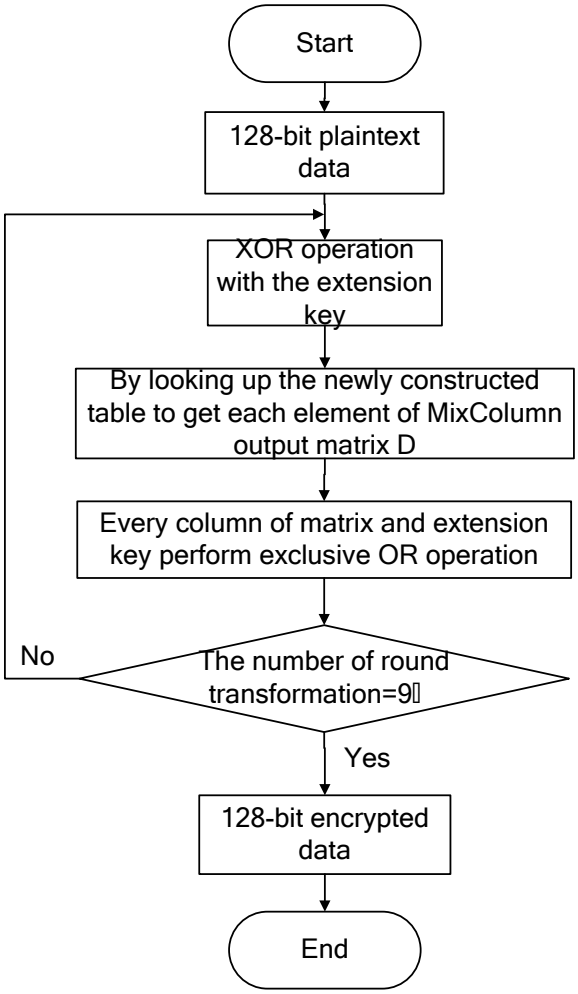

Figure 4: optimized AES encryption algorithm flow chart

4.3 Experimental results and analysis

In order to test the encryption speed between classical AES algorithm and optimized AES algorithm in this paper, we use C ++ language to implement the two algorithm encryption process respectively the encryption process in Windows7 operating system, Core i5-3230M 2.60GHz CPU and 8G memory environment. Each experiment we take 100,000 times the encryption time, and we get respectively a total of 10 sets of data in five experiments. The data got in the experiments are shown in Table 2.

Through the test results in Table 2 we can see that the encryption speed of optimized AES algorithm has a great improvement compared to the classic AES algorithm. In terms of memory footprint, this optimized AES encryption algorithm require $256\times3\times2=1536\mathrm{B}=1.5\mathrm{KB}$ to store two new byte conversion tables(encryption and decryption). The traditional AES algorithm requires $256\times2=512\mathrm{B}=0.5\mathrm{KB}$ to store two bytes conversion tables, so the optimized AES algorithm does not significantly increase the memory resource occupancy.

Table 2: Experimental test results.

The algorithm used	Experiment number					the average time of 100,000 times encryption(s)
	1	2	3	4	5	
Traditional AES algorithm	8.472	8.443	8.382	8.427	8.430	8.43
Optimized AES algorithm	1.550	1.471	1.471	1.469	1.533	1.50

5 Conclusion

The system implements the verification of the student identity through the fingerprint, which can make the campus life more convenient. The data encryption transmission and the terminals only processing without storing the data, which makes the convenience greatly improved on the basis of ensuring security. Each terminal connected with the host computer constitutes an integral system to achieve the information sharing among each terminal, and the host computer stores the terminal data and manages the students' information efficiently with less time. The encryption method based on the improved AES optimize the implementation method of algorithm in the process of simplifying the operation step, and the mathematical structure of the original algorithm is not changed, so that the encryption speed increases rapidly under the condition that the security is not reduced， while the memory occupation doesn't increase significantly, so it is easy to be achieved in the embedded system. Taking an example of AES with 128-bit plaintext length and key length, this paper proposes an optimization scheme based on actual requirement .The scheme can also be extended to the AES with other data length, which is suitable for various situations of data encryption, so it has a wide range of applications and strong practicability.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

Special thanks National University Student Innovation Program and Nankai University here for the assistance provided to this project.

References

[1] Kai, Zhao. (2016). Design and implementation of college students' entrepreneurship management system based on B/S structure. RISTI (Revista Iberica de Sistemas e Tecnologias de Informacao). March 30, 2016, Issue 17B:102-114.

[2] S.R.Bharamagoudar, Geeta R.B., and S.G.Totad3.(2013)Web Based Student Information Management System. International Journal of Advanced Research in Computer and Communication Engineering ,Vol. 2, Issue 6

[3] Ahmad, R.; and Ismail, W. (2016). Performance comparison of advanced encryption standard-128 algorithms for wimax application with improved power-throughput. Journal of Engineering Science and Technology, December 2016, 11(12):1678-1694.

[4] STMicroelectronics,2009.STM32 Reference Manual . 10th ed.

[5] US Department of Commerce, NIST. (2006). Advanced Encryption Standard. National Computer Conference (Vol.3373, pp.83-87).

[6] Ahmad, R.; and Ismail, W. (2013). A survey of high performance cryptography algorithms for WiMAX applications using SDR. Selforganization and green applications in cognitive radio networks (1st ed.). USA: IGI-Global, 231-246.

[7]Monteiro, C.; Takahashi, Y.; and Sekine, T. (2015). Low-power secure S-box circuit using charge-sharing symmetric adiabatic logic for advanced encryption standard hardware design. IET Circuits, Devices and Systems, 1 September 2015, 9(5):362-369.
[8]Youssef, A.M.; and Tavares, S.E. (2005). Affine equivalence in the AES round function. In Discrete Applied Mathematics 2005 148(2):161-170
[9]Bertoni, G., Breveglieri, L., Koren, I., Maistri, P., & Piuri, V. (2003). Error analysis and detection procedures for a hardware implementation of the advanced encryption standard. Computers IEEE Transactions on, 52(4), 492-505.
[10] Blömer, J., & Seifert, J. P. (2003). Fault Based Cryptanalysis of the Advanced Encryption Standard (AES). Financial Cryptography, International Conference, FC 2003, Guadeloupe, French West Indies, January 27-30, 2003, Revised Papers (Vol.2742, pp.162-181). DBLP.
[11]Daemen, J., Rijmen, V.(2002) The Design of Rijndael: AES - The Advanced Encryption Standard.
[12]Schneier, B.(1996) Applied Cryptography: Protocols, Algorithms, and Source Code in C.
[13]Stallings, W.(1999) Cryptography and Network Security: Principles and Practice.
[14] Mcloone, M., & Mccanny, J. V. (2001). Rijndael fpga implementation utilizing look-up tables. Signal Processing Systems IEEE Workshop on, 34, 349-360.
[15]Gong, J., Liu, W., & Zhang, H. (2012). Multiple lookup table-based aes encryption algorithm implementation. Physics Procedia, 25, 842-847.
[16]Wang, J. F., Chang, S. W., & Lin, P. C. (2003). A novel round function architecture for AES encryption/decryption utilizing look-up table. IEEE, 2003 International Carnahan Conference on Security Technology, 2003. Proceedings (pp.132-136). IEEE.
[17] Daor, J., Daemen, J., & Rijmen, V. (1999). Aes proposal: rijndael. Vazirani: Efficient and Secure Pseudo-Random Number Generation. Proceedings, 25th IEEE FOCS.