International Journal of Security and Its Applications
Vol.8, No.6 (2014), pp.295-302
http://dx.doi.org/10.14257/ijsia.2014.8.6.26

A Secret Sharing Scheme Based on AES

Jie Cui, Lei Chen, Yiming Zhang, Zhiqiang Xie and Hong Zhong*

School of Computer Science and Technology, Anhui University, Hefei, China
cvjxabcd@126.com

Abstract

In order to solve the key setting difficulty and the key security problem in the file encryption, key distributed storage technology may be a proper choice to help improve the safety of the key. In the paper, a novel secret sharing scheme is proposed by AES encryption algorithm for file confidentiality, dynamic key generation mechanism to generate keys, multi-secret-sharing ideas on key pre-treatment, using Shamir threshold scheme for secret dispersed storage. Finally, a few tests are carried out and the test results suggest that the efficiency of the whole scheme is good.

Keywords: *AES, secret sharing, Rijndael, file security*

1. Introduction

With the rapid development of computer and network communication technology, the information security, meanwhile, has become increasingly prominent, and events such as information leaks and other incidents happen more frequently nowadays. Cryptography provides many practical techniques to solve information security problems. In order to guarantee security and confidentiality of the private documents, people mainly resort to encryption to protect files, and make it impossible for users who do not have a key to steal information. This makes the security and confidentiality mainly hinges on the security key, thus an effective method of key management is needed. In recent years, the key dispersed storage technology has become a trend in key management. It helps to solve the problem that the key text can not be decrypted caused by losing or forgetting the key. What is more, the application in the field of computer and network security will be of significance both in theory and practice.

The first secret sharing scheme is (t, n) threshold secret sharing scheme, which is proposed independently by Blakley [1] and Shamir [2] in 1979. On that basis, Shamir's scheme, which is based on the theory of polynomials, is easy to understand and implement, and catch more attention of researchers. On the one hand, it solves on the problem of the original threshold scheme, on the other hand, it can be applied to other fields extensively. The threshold scheme provides a good way to improve other technologies. For example, Zhang, *et al.*, [3] adopted the combination technology of Shamir scheme and cloud storage technology to construct a secure network disc. Xu, *et al.*, [4] resolved copyright issues on digital media, and the threshold scheme and broadcast encryption algorithm are combined to construct the identity-based broadcast encryption scheme. In the literature [5], the threshold scheme was used to protect software copyrights, copyright information, enhanced anti-aggression and survivability and other issues. In the literature [6], an existing Watermarking Algorithms and (t, n) threshold were adopted for the copyright watermark sharing, and the improved algorithm had good security and anti-attack capability. In the literature [7], security forensics and Shamir threshold scheme were integrated to improve the security of the forensic

ISSN: 1738-9976 IJSIA
Copyright ⓒ 2014 SERSC

information. Based on analysis to the literature and research, this paper combines the AES algorithm and Shamir's scheme to ensure the security of the file.

In order to improve the effectiveness of space Shamir threshold scheme, a number of scholars put forward some corresponding solutions using the multi-secret sharing technology. First, divide the secret S into p blocks, with $S_i\,(i = 1, 2, \ldots, p)$ to represent p pieces secret, then use multi-secret sharing technology while sharing this secret p pieces. For this propose, this paper will split the key based on this idea.

This paper combines AES algorithms, dynamic key generation mechanisms, Shamir's (t, n) threshold secret sharing scheme and technology of multi-secret sharing ideas together, and sets up the file encryption key for distributed storage to ensure the security of the key and resolve the issues of key complexity, key management and file security.

2. Preliminary

2.1. AES Algorithm

Rijndael specification is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits. The number of cycles of repetition may be 10, 12 or 14 correspondingly. Each round consists of several processing steps, which are Byte Sub, Shift Row, Mix Column and Add Round Key. Each round starts from Add Round Key, then iterate 11 times. By the way, the last round doesn't include Mix Column.

The content of this paper mainly talks about the 192-bit Rijndael algorithm. The plaintext, the cipher text, the key and all the intermediate states are in form of 4×6 matrix. Here we are taking the process of one round as an example to help explain the algorithm's mathematical model [8].

The operation based on S-box is a kind of nonlinear transformation, which independently plays a role in status byte. Here we record the S-box as S . The S-box used is constructed by combining the inverse function with an invertible affine transformation over $GF(2)$, and also derived from the multiplicative inverse over $GF(2^8)$.

1) The S-box is generated by determining the multiplicative inverse for a given number in $GF(2^8) = \dfrac{Z_2[x]}{(x^8 + x^4 + x^3 + x + 1)}$. Once given a number ω , for example, we can easily get ω 's inverse. Here we identify ω 's inverse as υ . It is clear that ω and υ are reversible with the condition below:

$$\omega \times \upsilon = 1 \bmod (x^8 + x^4 + x^3 + x + 1)$$

In that case, we can naturally get the value of:

$$\upsilon = \omega^{-1} = \begin{cases} \omega^{254} & \omega \neq 0 \\ 0 & \omega = 0 \end{cases}$$

2) The multiplicative inverse is then transformed using the following affine transformation:

$$y = La \times x + '63' = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_7 \\ x_6 \\ x_5 \\ x_4 \\ x_3 \\ x_2 \\ x_1 \\ x_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

where $[x_7 \ldots x_0]$ is the multiplicative inverse as a vector. The constant '63' makes sure that

 Copyright ⓒ 2014 SERSC

the S-box's fixed points and opposite fixed points don't appear. The trivial maximum correlation between linear transformations of input/output bits is 2^{-3}. The non-trivial maximum of the difference propagation probability is 2^{-6}. In that case, it is clear that The Rijndael S-Box can be resistant to linear and differential cryptanalysis [15].

A 4×6 matrix can be got by S-box substitution. Then the Shift Row step will transform the matrix by making row n shifted left circular by $n-1$ bytes:

$$\begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} & s_{0,4} & s_{0,5} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} & s_{1,4} & s_{1,5} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} & s_{2,4} & s_{2,5} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} & s_{3,4} & s_{3,5} \end{bmatrix} \rightarrow \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} & s_{0,4} & s_{0,5} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,4} & s_{1,5} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,4} & s_{2,5} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,4} & s_{3,5} & s_{3,0} & s_{3,1} & s_{3,2} \end{bmatrix}$$

Here $s_{i,j}$ represents the element in row i, column j. $(0 \le i \le 3, 0 \le j \le 5)$

Together with Shift Row, Mix Column provides diffusion in the key. It operates on each row independently by mapping every byte to new value. To gain this new value, 4 bytes in each row must be processed through the shifting function. Here comes the process of shifting:

$$\begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} & s'_{0,4} & s'_{0,5} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} & s'_{1,4} & s'_{1,5} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} & s'_{2,4} & s'_{2,5} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} & s'_{3,4} & s'_{3,5} \end{bmatrix} = D \cdot \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} & s_{0,4} & s_{0,5} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} & s_{1,4} & s_{1,5} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} & s_{2,4} & s_{2,5} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} & s_{3,4} & s_{3,5} \end{bmatrix}$$

Note that

$$D = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

In the Add Round Key step, the sub key is added by combining each byte of the state with the corresponding byte of the sub key using bitwise XOR for each round. The relation can be viewed below:

$$Y = X \oplus K$$

Where K refers to the round keys.

2.2. Shamir Secret Sharing Scheme

Shamir threshold scheme based on Lagrange interpolation method proposed the first (t, n) threshold secret sharing scheme. It is ideal for a comprehensive scheme, and the most extensive study and application of threshold scheme. The following system parameters, the secret distribution algorithm and secret reconstruction algorithm are three parts to a brief introduction of the program.

1) System parameters: Let $p = \{p_1, p_2, ..., p_n\}$ denote the set of N participants, P is a large prime number, and T is the threshold value. The secret space and the share space are finite field $GF(P)$. $d_1, d_2, ..., d_n$ are integers different from each other on the $GF(P)$, as identification information of the public identity of the participants. The above parameters are open.

2) The secret distribution algorithm: The secret dealer D constructs a random (t-1) times polynomial of secret $f(x) = a_0 + a_1 x + ... + a_{i-1} x^{i-1}$, lets $a_0 = f(0) = s$. s is a secret shared between

the participants. Then, D distributes a secret share $x_i = f(d_i) \bmod p$ to every $p_i (1 <= i <= n)$.

3) Secret reconstruction algorithm: Any T participants $\{p_{i_1}, p_{i_2}, ..., p_{i_t}\}$ can get T points $(d_{i_1}, x_{i_1}), (d_{i_2}, x_{i_2}), ..., (d_{i_t}, x_{i_t})$ by combining their share, and the shared secret S: $s = f(0) = \sum_{r=1}^{t} x_{i_r} \prod_{h=1, h \neq r}^{t} \frac{0 - d_{i_h}}{d_{i_r} - d_{i_h}} \bmod p$ can be recovered by the Lagrange interpolation method.

3. The New Secret Sharing Scheme based on AES

Dynamic key generation mechanisms are used to ensure that sufficient complexity and length of key as K. The secret sharing scheme is as follow.

1) Break the K into $\{s_1, s_2, ..., s_m\}$, and then destroy K. The process is shown in Figure 1.

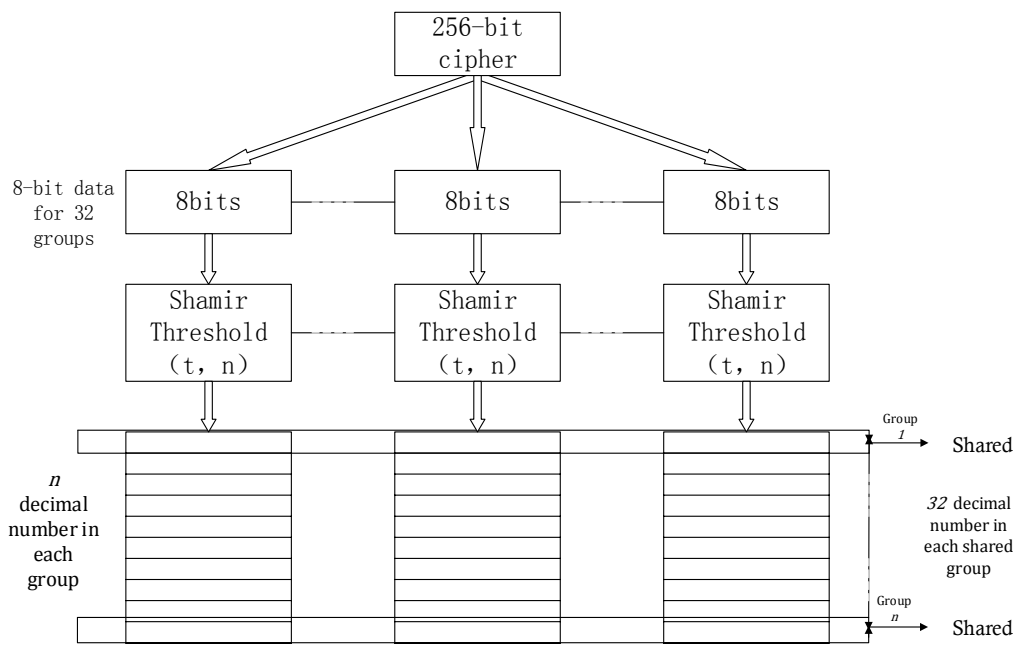

Figure 1. Key Splitting and Secret Sharing

2) Let $p = \{p_1, p_2, ..., p_n\}$ denote the set of N participants, P is a large prime number, and T is the threshold value. The secret space and the share space are finite field $GF(P)$. $d_1, d_2, ..., d_n$ Are integers different from each other on the $GF(P)$, as identification information of the public identity of the participants. The above parameters are open.

3) The secret distribution algorithm: The secret dealer D construct a random (t-1) times polynomial of secret $f(x) = a_0 + a_1 x + ... + a_{i-1} x^{i-1}$, let $a_0 = f(0) = s_j (1 \leq j \leq m)$ from $\{s_1, s_2, ..., s_m\}$, s is a secret shared between the participants. Then, D distribute a secret share $x_{ji} = f(d_i) \bmod p$ to every $p_i (1 \leq i \leq n)$.

 Copyright ⓒ 2014 SERSC

4) Step 3 was performed for all the data $\{s_1, s_2, ..., s_m\}$. So D distribute m pieces of secret share $\{x_{1i}, x_{2i}, ..., x_{mi}\}$ to every $p_i (1 \le i \le n)$.

5) Secret reconstruction algorithm: Any *T* participants $\{p_{i_1}, p_{i_2}, ..., p_{i_t}\}$ can get *T* points $(d_{i_1}, x_{ji_1}), (d_{i_2}, x_{ji_2}), ..., (d_{i_t}, x_{ji_t})$ by combining their share, then using Lagrange interpolation method can recover the shared secret *S*: $s_j = f(0) = \sum_{r=1}^{t} x_{ji_r} \prod_{h=1, h \ne r}^{t} \frac{0 - d_{t_h}}{d_{i_r} - d_{i_h}} \bmod p (1 \le j \le m)$. Applying such operations to $s_j (1 \le j \le m)$, we can get $\{s_1, s_2, ..., s_m\}$.

6) After the combination of $\{s_1, s_2, ..., s_m\}$ to get *K*, $\{s_1, s_2, ..., s_m\}$ is destroyed, *K* will be carried out in the corresponding decryption, and *K* will be destroyed after completing the decryption. The process is shown in Figure 2.

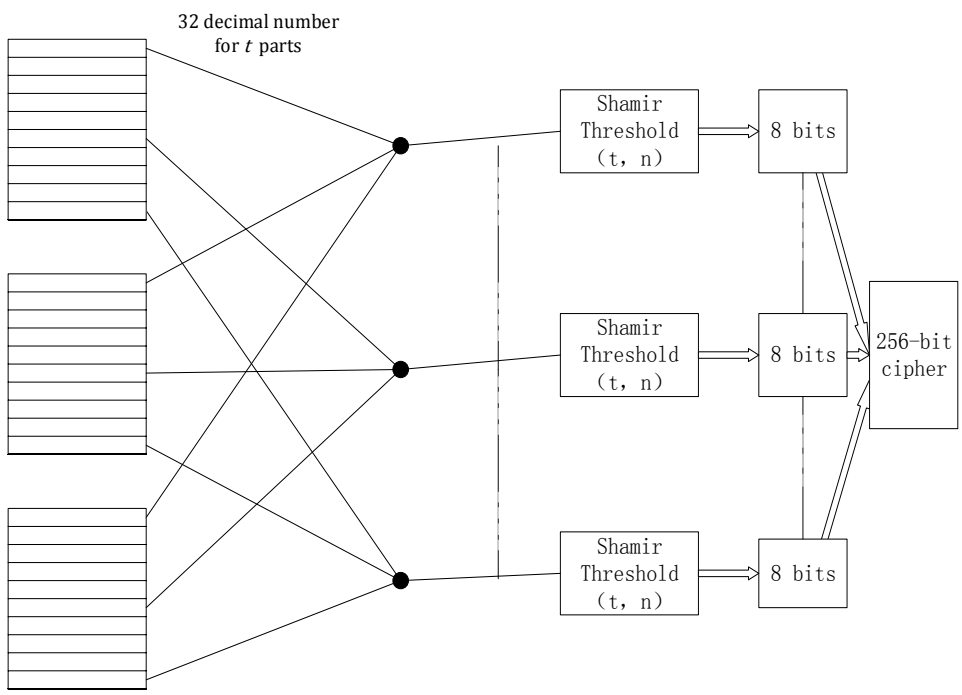

Figure 2. Secret Collecting and Key Regenerating

4. Performance Analysis of the New Scheme

4.1. Correctness Analysis

The accuracy of the step from second to fifth is based on Shamir's secret sharing scheme. The analysis of Step 1 and Step 6 is shown as follows.

Here a simple decomposition and coupling scheme will be given to help analyze the whole scheme's accuracy. If we can ensure this simple example is reliable, the correctness of the entire scheme can be guaranteed. The scheme is the following:

The decomposition: The key size used for an AES key may be 128, 192 or 256 bits. Here we take the 256-bit AES as an example. A 256-bit key K will be created based on

the stochastic dynamics key generation method. Then each 8 bits of K will be divided into a group. In that case, we will get 32 groups $\{s_1, s_2, ..., s_{32}\}$, where each group is recorded as. At last, K will be destroyed.

The composition: Corresponding to Step 1, groups $\{s_1, s_2, ..., s_{32}\}$ will be obtained in the fifth step. At this point, all these groups combine together, and the original key K will be regenerated. The decryption can be done after that. In the end, all the groups will also be destroyed.

Obviously, in the above example, the encryption and the decryption of files can be implemented. So this simple scheme is proved to be reliable. As other kinds of decomposition and coupling schemes, the bits of s_i and the number of the groups can be different. If we can combine these groups in some way, and regain the original key K according to our combination method, in other words, the decomposition and the composition are corresponding somehow, the whole scheme is still valid.

4.2. Security Analysis

The security of the step from second to fifth depends on Shamir's secret sharing scheme. Here we will analyze the first and the sixth steps' security.

In order to make it clear and easy to analyze the security of decomposition and composition, a simple example is presented below. If we can verify the safety in such condition, it is certain that the whole scheme is safe. The scheme comes as follows:

The decomposition: The key size used for an AES key may be 128, 192 or 256 bits. Here we take the 256-bit AES as an example. A 256-bit key K will be created based on the stochastic dynamics key generation method. Then each 8 bits of K will be divided into a group. In that case, we will get 32 $\{s_1, s_2, ..., s_{32}\}$ groups, each group is recorded as s_i. At last, K will be destroyed.

The composition: Corresponding to Step 1, groups $\{s_1, s_2, ..., s_{32}\}$ will be obtained in the fifth step. At this point, all these groups combine together, and the original key K will be regenerated. The decryption can be done after that. In the end, all the groups will also be destroyed.

When decomposing, the key K is destroyed after it is divided into groups $\{s_1, s_2, ..., s_{32}\}$. All these groups are shared immediately according to the Shamir's scheme afterwards. During the process, groups $\{s_1, s_2, ..., s_{32}\}$ and the key K can't be revealed and stolen because both Step 1 and 2 are carried out internally. In this way the safety of the first step can be guaranteed.

As for composition, all the groups $\{s_1, s_2, ..., s_{32}\}$ are combined into the key K, after that these groups are destroyed. Destruction of K will be in progress after the decryption is finished. Just like the decomposition part, Step 5 and Step 6 are executed internally, and there will be no related data revealed. So the safety of Step 6 is ensured.

Parenthetically, only Step 4 and Step 5 have data exchange with external environment, whose safeties are guaranteed by the Shamir's scheme. Beside this, all the other data used in the rest of the steps is only transported inside the system, and can't be revealed and stolen, which in turn ensure security of the scheme.

4.3 Performance Test

The test was made on a smart phone powered by Google's Android 4.4.2 operating system, with 2GB of RAM. Just as Table 1 shows, when encrypting some small files, the time was controlled in 10 seconds. It is still acceptable that the encryption time of larger files may be

 Copyright ⓒ 2014 SERSC

over 30 seconds. On the other hand, the key K is used in AES algorithm and the Shamir's scheme, so the secret sharing can be carried out while encrypting. In that case, when the encryption is over, the secret sharing part ends at the same time, and no waiting time will be generated. In summary, the whole process is user-friendly and satisfying.

Table 1. Test Data of the Encryption Speed

File Type	**Audio File (*.Mp3)**		**Office File (*.docx)**		**Compressed File (*.zip)**	
File Size (Unit：M)	4.38	10	3.78	46.4	6.85	34.4
Encryption Time (Unit：s)	3	8	3	44	7	31

5. Conclusion

The secret sharing scheme based on AES not only has important practical value, but also has a certain theoretical significance. Here we come up with a kind of such scheme.

Our scheme is composed of two stages: The first one is AES algorithm. The stochastic dynamic key generation method creates keys used in the algorithm, and AES algorithm carries on the encryption and the decryption to the document. The next stage is the Shamir's secret sharing scheme and the multi-secret sharing technology. When encrypting, it splits keys in several parts for distribution, and each participant will get certain part of secret. When decrypting, some of these participants (we can suppose the number of the people is t) come together. By calculating and combining *t* parts in some way, the original key can still be obtained through Shamir's scheme. Nonetheless, the key can not be got if the number of the people is less than *t*.

In a nutshell, our scheme provides a method of file encryption and key management to ensure the safety of the file. However, the key splitting method we use is simple, and more complex methods remains to be further studied.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (No. 61173188, No. 61173187, No. 61272074), the Educational Commission of Anhui Province, China (No. KJ2013A017), the Research Fund for the Doctoral Program of Higher Education (No. 20133401110004), the Science and technology project of Anhui Province (No. 1401b042015), 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] G. R. Blakley, "Safeguarding cryptographic keys, Managing Requirements Knowledge", International Workshop on, IEEE Computer Society, **(1899)**, pp. 313-313.
[2] A. Shamir, "How to share a secret", Communications of the ACM, vol. 22, no. 11, **(1979)**, pp. 612-613.
[3] Z. Lu, X. Y. Hong, Z. Hao and T. Wei, "A Safe Cloud Storage Solution based on Secret Sharing", Netinfo Security, vol. 04, **(2013),** pp. 67-69.

Copyright ⓒ 2014 SERSC

[4] X. Lei and L.Yujiang, "Improvement of threshold-based broadcast encryption algorithm", Journal of Liaoning Technical University (Natural Science), vol. 03, no. 32, **(2013),** pp. 373-376.
[5] L. Zhitao, "Research on software copyright protection scheme based on Threshold key", Mine Surveying, vol. 02, **(2012),** pp. 43-45.
[6] W. Julong and C. Jihong, "Database Watermarking Algorithm Research Based on (t,n) Threshold", Computer Science, vol. 05, no. 41, **(2014),** pp. 182-185.
[7] Y. Xiaoyuan, J. Chenli, Q. Qing and H. Yupu, "A Safe Server for Computer Forensics Based on Shamir Secret Sharing Scheme", Computer Engineering and Applications, vol. 41, no. 22, **(2005),** pp. 147-149.
[8] C. Jie, H. Liusheng, Z. Hong and Y. Wei, "Algebraic Attack on Rijndael-192 Based on Grobner Basis", Acta Electronica Sinica, vol. 05, no. 41, **(2013),** pp. 833-839.
[9] Y. Dan and L. Zhenxing, "An Overview of the Development of Secret Sharing", Natural Science Journal of Harbin Normal University, vol. 01, no. 30, **(2014),** pp. 47-49.
[10] L. Yanhong and Z. Futai, "A New Space Effcient Secret Sharing Scheme without a Secure Channel", Chinese Journal of Computers, vol. 09, no. 35, **(2012),** pp. 1816-1822.
[11] L. Jiguo, W. Fei, L. Yanqiong and Z. Yichen, "Rational Multi-secret Sharing", Journal of Chinese Computer Systems, vol. 06, no. 34, **(2013),** pp. 1392-1395.
[12] L. Quandong and Z. Yanhui, "Security decision analysis based on A. Shamir's (t,n) threshold secret sharing scheme", Network Security Technology & Application, vol. 04, **(2014),** pp. 23-26.
[13] Z. Hongfei, Y. Xiaoyuan and W. Lixian, "Audio watermarking algorithm based on Shamir secret sharing scheme and zero- watermark's technology", Computer Engineering and Applications, vol. 43, no. 26, **(2007),** pp. 100- 102.
[14] W. Haotian, C. Yue, T. Pengxu and D. Yanghuan, "Group Key Management Scheme Based on Threshold", Computer Engineering, vol. 03, no. 39, **(2013),** pp. 167-173.
[15] J. Daemen and V. Rijmen, "AES proposal: Rijndael [EB/OL]", **(1999)** October 5, http://www.east.kuleuven.ac.be/ ~rijmen/rijndael.

 Copyright ⓒ 2014 SERSC