Scientific Research Publishing

Journal of Information Security, 2021, 12, 177-187
https://www.scirp.org/journal/jis
ISSN Online: 2153-1242
ISSN Print: 2153-1234

A Complex Encryption System Design Implemented by AES

Zhimao Lu, Houmed Mohamed

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China
Email: houmed.med.houmed@gmail.com

How to cite this paper: Lu, Z.M. and Mohamed, H. (2021) A Complex Encryption System Design Implemented by AES. *Journal of Information Security*, **12**, 177-187.
https://doi.org/10.4236/jis.2021.122009

Received: May 1, 2020
Accepted: April 27, 2021
Published: April 30, 2021

Abstract

With the rapid development of internet technology and the increasing popularity of e-commerce, data encryption technology plays a very important role in data security. Information security has two aspects: security protocol and cryptographic algorithm and the latter is the foundation and core technology of information security. Advanced Encryption Standard (AES) encryption algorithm is one of the most commonly used algorithms in symmetric encryption algorithms. Such algorithms face issues when used in the context of key management and security functions. This paper focuses on the systematic analysis of these issues and summarizes AES algorithm implementation, comprehensive application and algorithm comparison with other existing methods. To analyze the performance of the proposed algorithm and to make full use of the advantages of AES encryption algorithm, one needs to reduce round key and improve the key schedule, as well as organically integrate with RSA algorithm. Java language is used to implement the algorithm due to its large library, then to show the efficiency of the proposed method we compare different parameters, such as encryption/decryption speed, entropies and memory consumption...) with a classic algorithm. Based on the results of the comparison between AES and the hybrid AES algorithm, the proposed algorithm shows good performance and high security. It therefore can be used for key management and security functions, particularly for sharing sensitive files through insecure channel. This analysis provides a reference useful for selecting different encryption algorithms according to different business needs.

Keywords

AES Algorithm, RSA Algorithm, Encryption, Key Management

1. Introduction

In cryptography, there are many types of encryption algorithms. Generally

speaking, they can be divided into three types: symmetric encryption algorithm, asymmetric encryption algorithm and single-entry encryption algorithm. Different encryption algorithms are used in security, encryption efficiency, implementation complexity, there are big differences in the best places to use. Among them, the symmetric encryption algorithm uses the same key for encryption and decryption. The algorithm is reversible or decipherable. Common algorithms include IDEA, DESX, RC4, RC5, RC6, DES, 3DES and AES. Symmetric encryption algorithms are often used in situations where large amounts of data are encrypted or data is frequently sent. Asymmetric encryption algorithm means that different public and private keys are used for encryption and decryption. Sometimes, asymmetric encryption is also called public key encryption. The algorithm is also reversible. Common algorithms include RSA, DSA, ECC, Diffie-Hellman, and El. Gamal *et al.* Asymmetric encryption algorithms are commonly used for public key encryption, private key decryption, private key signature, public key verification, small amount of sensitive information encryption, digital signature, and so on. The single-entry encryption algorithm and the hash encryption algorithm are unidirectional irreversible algorithms. The encrypted data cannot be decrypted. Common algorithms include MD5 and SHA. The algorithm is commonly used in non-restorable password storage, information integrity checks, etc., such as file verification, digital signature, authentication protocol, and the like How to encrypt critical information has become the focus of attention in the area of IT and networking [1]. Nevertheless, AES cryptanalysis has not ceased and several researchers are exploring new methods to allow us to achieve competitive efficiency. Literatures [2] [3] [4] [5] [6] explain how AES is planned and implemented for change. The work suggested by Reena Mehla and Harleen Kaur [2] focuses on the modification of the key extension and shift row transformation of AES to make the corresponding algorithm highly resistant against attacks. Their proposed method also reduced the time-taker for images to be encrypted, and provides better output than AES. 2016, Smaliukas and Gytis Vaicekauskas [3] reviewed and improved the AES algorithm to lower the algorithm computation and enhance the data transmission. Their suggested technique used parity bit generator to represent a high level of protection and create higher transmission of data without using Mixcolumns. Dimas Natanael, Faisal, Dewi Suryani [4], Implement ECC algorithm to secure text message in mobile messaging. They follow the approach suggested by Singh to build an Android chat application with end-to-end encryption on the device. They also offer their chat apps output experimental results such as accuracy of the received text message, average encryption time, and decryption time Lin Teng, Hang Li, Shoulin Yin, and Yang Sun [5], They incorporate random disturbance information to improve data protection, column mix process and key choreography are improved. Cheng Tan, Xiaoyan Deng, Lijun Zhang [6], they find 5 commonly used block Cipher, AES DES, 3DES, RC5 and blowfish identification. After the Cipherteext files have been tested, the identification rate is above 97%.

In this paper we proposed to solve the problem of difficult key management of AES algorithm and low efficiency of RSA by reducing the execution round and the modification of the initial key. The complex encryption algorithm is introduced and Section 2 describes AES algorithm modification, and the RSA algorithm combination and the methodology are presented in Section 3. The simulation, experimental and the analyze of the result are presented in Section 4 and 5. By using the hybrid encryption, the encryption speed is effectively improved and the security strength is improved.

2. AES Algorithm Modification

The Advanced Encryption Standard (AES), also known as Rijndael, has not been breached at this time. Nonetheless, AES cryptanalysis has not stopped and many researchers are seeking new approaches to allow us to achieve competitive efficiency. To provide a high efficiency and a malleable algorithm for different kind of business needs (e-commerce, emails, bank card...).

The following two modifications to the original AES algorithm have been made in our research:

1) Adding new key: functions are conducted as shown in **Figure 1** in the proposed AES algorithm and additional key (modified key) is introduced. The key is generated first, and then encrypted by the public key.

Until we do the key expansion stage of the encryption process, the additional key will be XORed with plain text first. XOR's call this process InitialAddRoundKey. The new output resulting from the operation Initial AddRoundKey is used as plaintext for the following steps. Before that the conventional key is spent to produce the subkeys;

2) Reconfiguration in the SubBytes function: rather than the existing SubBytes operation, we inserted a new operation in the original SubBytes operation, called Modified Transport. And we update SubBytes to this function as ModSubBytes. Next, the data in the ModSubBytes process is transmitted before the values of S-Box have been replaced. The state array is divided into two halves (4 bits each) each part of the State array (a value of 8 bits) and is transferred or exchanged in order to achieve a new state value in the transportation process.

3. Methodology Complex Encryption System Design

Both the methods of symmetric and asymmetric encryption are used to achieve the information confidentiality [7] [8]. A single key is used in symmetric encryption and in order to communicate safely all individuals who will receive the message must have that secret key. Asymmetric encryption makes use of a pair of keys, a private key and a public key. Asymmetric encryption technique can ensure the security and non-repudiation by using the digital signature. The key management of AES is much more complicated than the RSA. If we need to encrypt large amounts of data, AES is a good choice, which can effectively improve the encryption and decryption speed. Electronic signatures require the use of

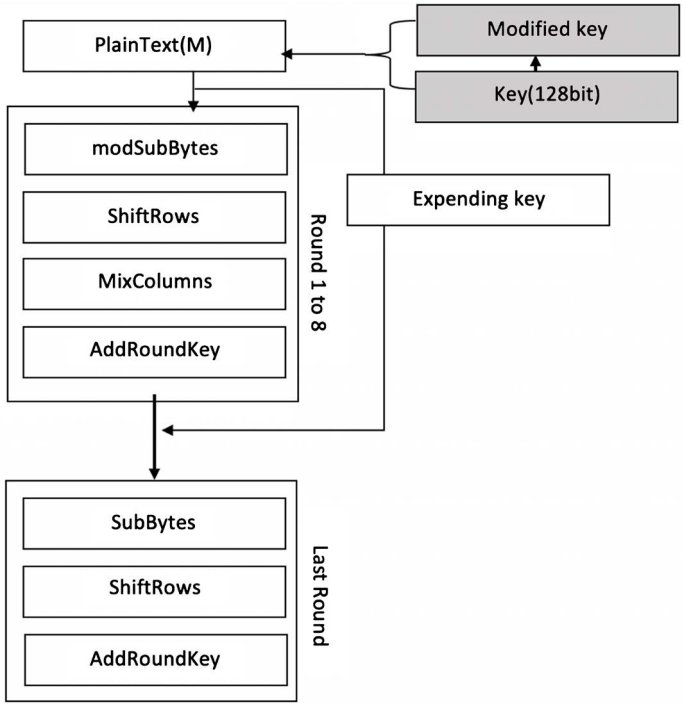


Figure 1. Flowchart of Aes encryption.

asymmetric algorithms. AES encryption algorithms cannot implement signatures. RSA can be used to implement electronic signatures. These two types of encryption algorithms have their own advantages and disadvantages [9]. In order to fully promote the advantages of each algorithm, AES and RSA encryption algorithms can be comprehensively used in the process of use. Because RSA algorithm has high security but slow encryption speed, it can be used for encryption. The AES key sends the encrypted AES key to the other party. After receiving the AES key, the other party decrypts the AES key by using the RSA, and then decrypts the received data with the AES key. In this way, since the key data length of AES is generally short, the encryption and decryption efficiency are high, and a large amount of data encryption adopts the AES encryption algorithm, and the encryption and decryption speed of the AES itself is relatively fast, and the hardware can further improve the encryption. And decryption speed [10] [11], encryption integrated application process (as shown in **Figure 2**)

The implementation process of the encryption algorithm is as follows:

1) The receiver creates the RSA public key and the private key (key pair), the receiver saves the private key, and sends the RSA public key to the sender of the data through the Internet;

2) The data sender creates the extend AES key, encrypts the AES key with the RSA public key sent by the receiver, and encrypts the plaintext data to be sent with the created AES key;

3) After receiving the ciphertext and the encrypted AES key, the receiver decrypts the AES key by using the RSA private key saved by the receiver, and then

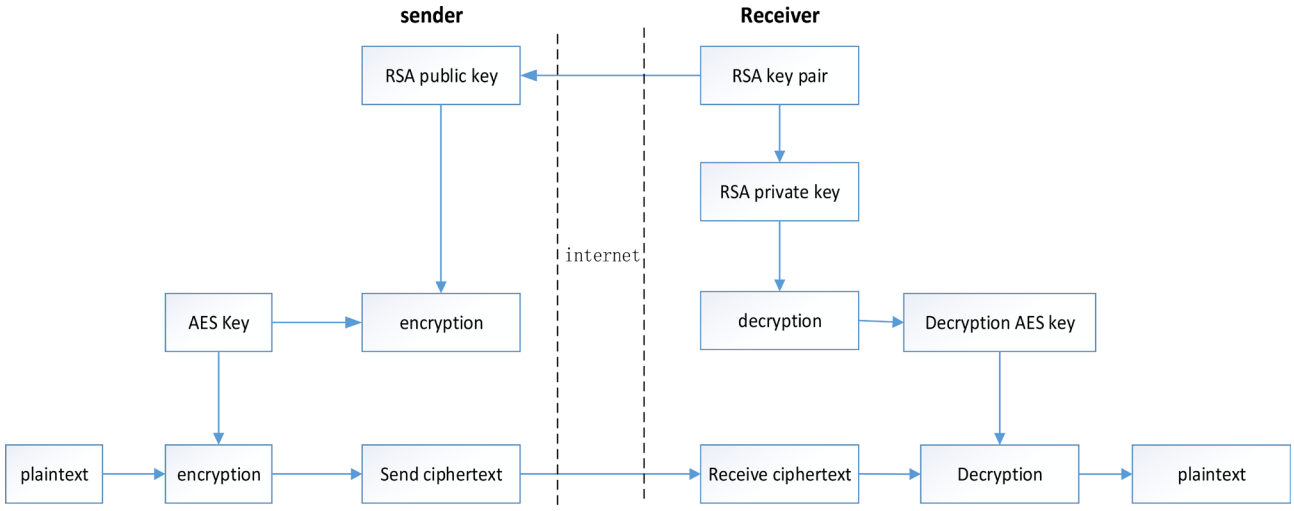


Figure 2. Encryption integrated application process.

decrypts the received ciphertext data with the key to obtain the plaintext data.

In the actual application process, if the data communication parties often send a large amount of data to each other, the encryption comprehensive operation scheme can be further optimized, that is:

1) The AES key exchange is performed by RSA at regular intervals;

2) And after the AES key is exchanged, the two parties send data using the AES key of the other party key.

The specific process is that both the sender and the receiver use RSA to generate a password pair, and send the generated public key to the other party. Each party generates an AES key, and encrypts the AES generated by the other party's public key and sends it to the other party. After the AES password cipher text encrypted by the RSA public key, the AES password is decrypted by using the respective RSA private key and stored. In a certain period of time, both parties use the other party's AES to encrypt and send, and the two parties receive each other's ciphertext and then use the saved AES key to decrypt and obtain the plaintext. When adopting this scheme, the AES key must be re-reconstructed periodically and exchanged, and the probability of password leakage can be greatly reduced by the periodic replacement of the password.

4. Simulation Test and Results

The system implements the encryption scheme combining AES and RSA algorithm, and uses two fixed encryption algorithms to test the encryption algorithm. The time used for encryption and decryption by AES is basically the same. The time used for RSA public key encryption and the AES encryption time are basically the same. The difference is not big, but the time required to decrypt using RSA private key is more, the test result (Figure 3).

In order to test the performance, by transmitting an 11 M size electronic contract document for simulation transmission test, a 128-bit encryption key is first generated on the sender, and the key is encrypted by the RSA asymmetric algo-

rithm and then sent to the receiver for reception. After receiving the contract ciphertext and the symmetric key, the party successfully completed the decryption of the contract, and the combination of decryption takes about 20 seconds. The asymmetric encryption algorithm is used to encrypt the contract. After sending, it is decrypted again. The decryption process takes about 180 seconds. Therefore, the efficiency of using hybrid encryption algorithms is much better than that of asymmetric encryption algorithms. Simulation test (as shown in Table 1 and Graph 1).

It is proved by experiments that the complex encryption algorithm is close to the symmetric key encryption algorithm in terms of running speed. From the test data, the complex encryption algorithm is only 0.8 seconds longer than the

Figure 3. Test results.

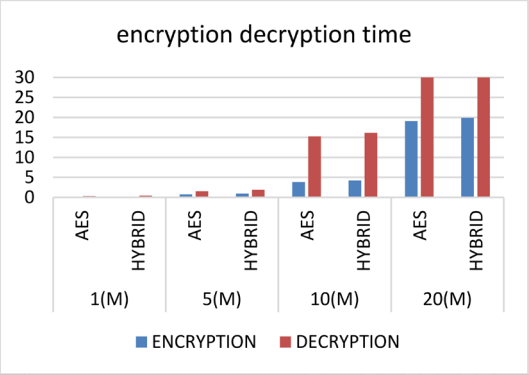
Table 1. Simulation test.

TYPE	Amount of data	Encrypted data (M)	Encryption time (s)	Decryption time (s)
AES	1	2.034	0.153	0.305
	5	10.172	0.763	1.525
	10	101.724	3.813	15.253
	20	2034.472	19.067	305.064
RSA	1	1.001	2777.778	229,166.667
	5	5.007	13,888.889	1,145,833.333
	10	25.035	69,444.444	11,458,333.333
	20	125.174	347,222.222	229,166,666.667
Hybrid encryption	1	2.134	0.183	0.405
	5	11.172	0.963	1.895
	10	109.724	4.213	16.153
	20	2087.472	19.867	315.064

symmetric encryption algorithm for 20 M data. For this kind of encryption application, the algorithm can meet the daily use of a large number of applications.

Through experimental tests, the complex encryption algorithm is based on security. The complex encryption algorithm (hybrid encryption) uses AES for the main data. From the perspective of data security, it is safer, and the AES encryption key is through RSA. Encrypted, so the encrypted data is more secure than AES.

Above, we compare our method approach with performance of other methods for that we use a text.txt (Figure 4) file of 409 chars which is 4 kb size. The results of the analysis are set out in Table 2. Compared to Vigila and Muneeswaran [12], our approach turns out to be surpassing and it is also outperforming compared to Dimas Natanael, Faisal, Dewi Suryani [5] when it comes to encryption,



Graph 1. Encryption and decryption time comparison AES standard vs hybrid algorithm.

The implementation process of the encryption algorithm is as follows.txt — Edited

```
The implementation process of the encryption algorithm is as follows: the receiver creates
the RSA public key and the private key (key pair), the receiver saves the private key, and
sends the RSA public key to the sender of the data through the Internet.The data sender
creates the extend AES key, encrypts the AES key with the RSA public key sent by the
receiver, and encrypts the plaintext data to be sent with the created AES key. After
receiving the cipher text and the encrypted AES key, the receiver decrypts the AES key by
using the RSA private key saved by the receiver, and then decrypts the received cipher
text data with the key to obtain the plaintext data. In the actual application process, if
the data communication parties often send a large amount of data to each other, the
encryption comprehensive operation scheme can be further optimized, that is: the AES key
exchange is performed by RSA at regular intervals, and after the AES key is exchanged, the
two parties send data using the AES key of the other party key. The specific process is
that both the sender and the receiver use RSA to generate a password pair, and send the
generated public key to the other party. Each party generates an AES key, and encrypts the
AES generated by the other party's public key and sends it to the other party. After the
AES password cipher text encrypted by the RSA public key, the AES password is decrypted by
using the respective RSA private key and stored. In a certain period of time, both parties
use the other party's AES to encrypt and send, and the two parties receive each other's
cipher text and then use the saved AES key to decrypt and obtain the plaintext. When
adopting this scheme, the AES key must be re-reconstructed periodically and exchanged, and
the probability of password leakage can be greatly reduced by the periodic replacement of
the password. increases implementation requirements.
```

Figure 4. TEXT.txt file used for the comparison of several approaches.

Table 2. Comparison result of several approaches.

Methods	#chars	Ciphertext size(kb)	Encryption time(s)	Decryption time(s)
Vigila and Munees [12]	409	459.118	1.95	0.83
Dimas, faisal, dewi [5]	409	15.587	0.263	0.206
Proposed	409	17.966	0.258	0.312

but we still need to improve our performance by optimizing the decryption 0.312.

5. Algorithm Analysis

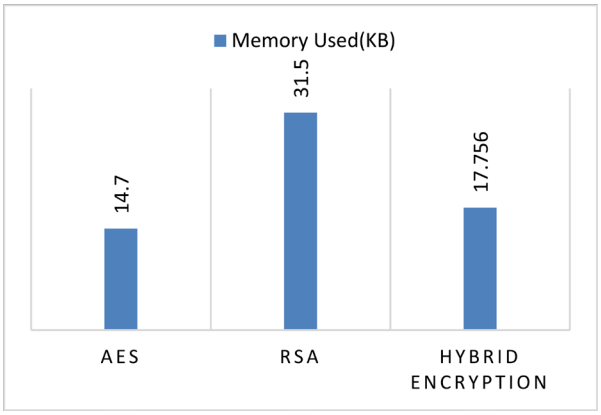
5.1. Memory Used

For implementation, various encryption techniques require different memory sizes. The needed memory depends on the number of operations the algorithm must perform, the key size used, the vectors used for initialization and the types of operations. Memory used Program Affect Costs [13] [14]. It is important that the necessary memory be as tiny as possible.

Graph 2 and Table 3 display the memory used for specified algorithms for the unit operations. Current AES consumes the least amount of memory while RSA consumes the maximum amount of memory per operating device. The hybrid algorithm requires medium memory size and shows a slight difference compared to AES but a better outcome than RSA. Therefore, if the requirement for any application is the smallest memory size, the AES is the best choice, but in terms of security the hybrid encryption is more effective, as shown in the previous entropy comparison.

Table 3. Memory consumption comparison.

Algorithm	Memory Used(KB)
AES	14.7
RSA	31.5
Hybrid encryption	17.756



Graph 2. Memory consumption comparison.

5.2. Entropy Comparison

A popular and classical measure of uncertainty in the theory of knowledge was described in 1948 by (Shannon, 1948) [15].

Shannon suggested that entropy H(X) could be determined by the average amount of information of a discrete random variable X.

$$H(X) = \sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$$

on the following terms:

- X consists of a finite of a sample space $x_1, x_2, x_3, \cdots, x_n$;
- P(x_i) probability distribution, $x_i \geq X$;
- And $\sum_i^n 1p(x_i)$.

Figure 5 shows that hybrid encoding averages the maximum mean entropy per byte of encoding. Entropy is a random measure of information. Cryptographic algorithms make randomness an integral and desirable property. Hybrid encryption results in high randomness of the output data, making the data less susceptible to attack.

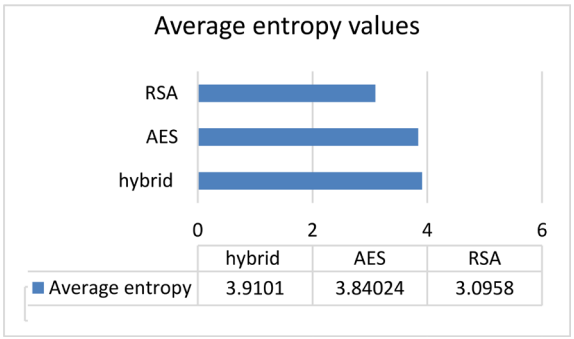


Figure 5. Entropy per byte comparison over existing result.

5.3. Key Management Problem

We present security analyses of the proposed here. Theoretical analysis indicates that the proposed algorithm overcomes security problems, key management problem the key management is easier by bringing in a trustworthy third party [16]. Trend exchanged only one key for each person. Any two computers do not need to share a key. Therefore, only n keys would be available in the entire network with n computers. The principle of the public key cryptosystem is that the encryption key and the decryption key are separated. Everyone can make their own designed encryption keys and algorithms public, and only secret decrypt keys. Anyone who uses this encryption key and algorithm to send encrypted information to the user can restore it. The advantage of public key cryptography is that the key transmission path does not need to be too high, which greatly simplifies key management.

5.4. Brute Force

AES already represents a safe algorithm that goes beyond cryptanalysis. Hackers often aim to find the cryptanalysis cipher key, which can be used to decode cipher text [17] [18]. The most that cryptanalysis in theory is brute force that can be used against all the cryptographic algorithms. In brute force attack, hackers hunt for the cipher key for all possible key combinations. They test every possible

key combination and perform a decryption of the trail to verify if it is the correct key. The question now is how long does it take for brute force to locate the actual key? The time for brute force attack depends on key size. This can be found very easily if the size of the key is small. But if the key size is longer than it can take quite a long time to locate the actual key.

6. Conclusion

In this paper, the encryption system combining AES and RSA algorithm makes full use of the advantages of symmetric key and asymmetric key. The session key used in the file is encrypted by RSA, and the encryption of data file is encrypted by AES. The system's encryption processing efficiency is high. The encryption algorithms commonly used in cryptography are analyzed and summarized. The AES encryption algorithm is implemented based on JAVA language. The algorithm is packaged and designed for the mixed use of AES and RSA encryption algorithms, which can reduce the understanding of encryption algorithms. Finally, the system analysis of the commonly used encryption algorithm is carried out. Through the system design of the encryption algorithm, it can help the relevant user.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Odeh, A., Masadeh, S.R. and Azzazi, A. (2015) A Performance Evaluation of Common Encryption Techniques with Secure Watermark System (SWS). *International Journal of Network Security & Its Applications*, **7**, 31-38. https://doi.org/10.5121/ijnsa.2015.7303

[2] Kazlauskas, K., Smaliukas, R. and Vaicekauskas, G. (2016) A Novel Method to Design S-Boxes Based on Key-Dependent Permutation Schemes and Its Quality Analysis. *International Journal of Advanced Computer Science and Applications*, **7**, 93-99. https://doi.org/10.14569/IJACSA.2016.070412

[3] Priyadarshini, P., Narayankar, P., Narayan, D.G. and Meena, S.M. (2016) A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish. *Procedia Computer Science*, **78**, 617-624. https://doi.org/10.1016/j.procs.2016.02.108

[4] Ali, A.H. and Sagheer, A.M. (2017) Design of an Android Application for Secure Chatting. *International Journal Computer Network and Information Security*, **9**, 29-35. https://doi.org/10.5815/ijcnis.2017.02.04

[5] Dimas, N., Faisal and Suryani, D. (2018) Text Encryption in Android Chat Applications using Elliptical Curve Cryptography (ECC). *Procedia Computer Science*, **135**, 283-291. https://doi.org/10.1016/j.procs.2018.08.176

[6] Teng, L., Li, H., Yin, S. and Sun, Y. (2019) A Modified Advanced Encryption Standard for Data Security. *International Journal of Network Security*, **22**, 112-117.

[7] Tan, C., Deng, X. and Zhang, L. (2018) Identification of Block Ciphers under CBC

Mode. *Procedia Computer Science*, **131**, 65-71. https://doi.org/10.1016/j.procs.2018.04.186

[8] Abdel-hafeez, S., Sawalmeh, A. and Bataineh, S. (2017) High Performance AES Design using Pipelining Structure over GF($(2^4)^2$). 2007 *IEEE International Conference on Signal Processing and Communications*, Dubai, 24-27 November 2007, 716-719. https://doi.org/10.1109/ICSPC.2007.4728419

[9] Cao, T. (2016) Design and Implementation of Encryption System Based on AES. *Software Development and Application*, **21**, 141-142.

[10] Indra Sena, R.M. and Siva Kumar, A.P. (2016) Secured Data Transmission Using Wavelet Based Steganography and Cryptography by Using AES Algorithm. *Procedia Computer Science*, **85**, 62-69. https://doi.org/10.1016/j.procs.2016.05.177

[11] Yang, B. and Bo, L. (2018) Complex Encryption Computer System Realized by AES World Smart Home. *Information System Engineering*, No. 12, 31-35.

[12] Vigila, S.M.C. and Muneeswaran, K. (2009) Implementation of Text Based Cryptosystem Using Elliptic Curve Cryptography. 2009 1*st International Conference on Advanced Computing*, Chennai, 13-15 December 2009, 82-85. https://doi.org/10.1109/ICADVC.2009.5378025

[13] Xia, G. (2019) Technical Browsing of Complex Encryption Computer System Realized by AES. *Automotive Application*, No. 6, 26-28.

[14] Zhang, M. (2019) Alarm System of Complex Encryption Computer System Implemented by AES. Inner Mongolia University of Science and Technology, Baotou, 21-22.

[15] Neenu, S. and Bonifus, P.L. (2016) Design of AES Architecture with Area and Speed Tradeoff. *Procedia Technology*, **24**, 1135-1140. https://doi.org/10.1016/j.protcy.2016.05.066

[16] Jie, K. and Liu, Y. (2015) Analysis of Data Encryption Algorithms. *China Science and Technology*, **18**, 33-34.

[17] Rakesh, K. and Geetu, M. (2015) A Novel Framework for Secure File Transmission Using Modified Aes and md5 Algorithms. *International Journal of Information and Computer Security*, **7**, Article No. 91. https://doi.org/10.1504/IJICS.2015.073012

[18] Berry, R., Berry, K. and Kumar, A. (2016) Review on Network Security and Cryptography. *International Journal of Innovative Research in Technology*, **3**, 44-45.